

INGENIERÍA DE TELECOMUNICACIÓN

Departamentos:

Arquitectura y Tecnología de Computadores
Teoría de la Señal, Telemática y Comunicaciones

Universidad de Granada



PROYECTO FIN DE CARRERA

Desarrollo de una aplicación de visualización de datos y configuración para sistemas de monitorización y análisis del movimiento del cuerpo humano.

Realizado por:

Javier López García

Dirigido por:

D. Alberto Olivares Vicente

D. Gonzalo Olivares Ruiz

Desarrollo de una aplicación de visualización de datos y configuración para sistemas de monitorización y análisis del movimiento del cuerpo humano

PALABRAS CLAVE:

Sensores inerciales, IMU, MEMS, acelerómetros, giróscopos, magnetómetros, calibración, cálculo de la orientación, GaitWatch, ECnsole, Matlab, GUI.

RESUMEN:

Este proyecto se centra en desarrollar una aplicación gráfica en Matlab que permita la configuración y visualización de los datos recogidos por los sistemas de medida de parámetros de la marcha GaitWatch y ECnsole. Los objetivos principales son: elaborar un análisis de los sistemas de monitorización de la marcha existentes en la actualidad que presentan características similares; desarrollar un software específico que reuna los requisitos necesarios establecidos inicialmente (visualizar y configurar los datos extraídos de los sistemas citados anteriormente); e integrar los algoritmos de calibración y cálculo de la orientación calculando las señales requeridas de forma imperceptible para el usuario de la aplicación.

D. Gonzalo Olivares Ruiz profesor del departamento de Arquitectura y Tecnología de los Computadores de la Universidad de Granada, y D. Alberto Olivares Vicente, profesor del dpto. de Teoría de la Señal, Telemática y Comunicaciones, como directores del Proyecto Fin de Carrera de D. Javier López García,

Informan:

Que el presente trabajo, titulado:

Desarrollo de una aplicación de visualización de datos y configuración para sistemas de monitorización y análisis del movimiento del cuerpo humano.

Ha sido realizado y redactado por el mencionado alumno bajo nuestra dirección, y con esta fecha autorizamos a su presentación.

Granada, a 15 de Septiembre de 2014 Fdo:

D. Alberto Olivares Vicente

D. Gonzalo Olivares Ruiz

Agradecimientos

En primer lugar me gustaría darle las gracias al departamento de Arquitectura y Tecnología de computadores y al departamento Teoría de la Señal, Telemática y Comunicaciones por darme la posibilidad de realizar este proyecto.

Además quisera agradecer a mis tutores Gonzalo Olivares y Alberto Olivares su inestimable ayuda y dedicación. Sin su ayuda no hubiera sido posible.

Darle las gracias también a mi familia que durante estos años me ha dado la oportunidad de formarme adecuadamente y llegar a conseguir los objetivos que me había fijado.

Índice

Agradecimientos	II
1. Introducción	3
1.1. Contexto	3
1.2. Motivación	4
1.3. Objetivos	5
1.4. Estructura del documento	6
2. Monitorización del cuerpo humano	9
2.1. Definición y ámbito de uso	9
2.1.1. Definición: Análisis de la marcha	10
2.1.2. Historia	10
2.1.3. Fáctores y parámetros	11
2.1.4. Aplicaciones de los sistemas de captura del movimiento humano . . .	12
2.1.5. Clasificación de los sistemas de captura del movimiento humano . . .	16
2.2. Estado del arte	24
2.2.1. Sistemas basados en sensores inerciales	24
2.2.2. Sistemas basados en cámaras	28
3. Hardware de monitorización	33

3.1.	GaitWatch	33
3.1.1.	Descripción general	33
3.1.2.	Descripción del Hardware	33
3.1.3.	Configuración inicial	35
3.2.	ECnsole	42
3.2.1.	Descripción general	42
3.2.2.	Descripción del Hardware	43
4.	Fundamentos	47
4.1.	Fundamentos de sensores	47
4.1.1.	Acelerómetros	47
4.1.2.	Giróscopo	51
4.1.3.	Magnetómetro	54
4.2.	Fundamentos de la monitorización	58
4.2.1.	Calibración	58
4.2.2.	Cálculo de orientación	70
5.	Software de monitorización	91
5.1.	GUI en Matlab	91
5.1.1.	GUI en Matlab utilizando GUIDE	92
5.1.2.	GUI en Matlab sin hacer uso de GUIDE. Programando	103
5.2.	Descripción de la aplicación	105
5.2.1.	Pantalla principal (GaitWatch Manager)	105
5.2.2.	File (GaitWatch Manager)	105
5.2.3.	Calibration (GaitWatch Manager)	128
5.2.4.	Preferences (GaitWatch Manager)	130
5.2.5.	Pantalla principal (ECnsole)	131
5.2.6.	File (ECnsole)	132

5.2.7. Calibration (ECnsole)	138
5.2.8. Preferences (ECnsole)	141
6. Conclusiones y trabajo futuro	143
6.1. Conclusiones	143
6.2. Trabajo futuro	144

Índice de figuras

1.1. Estructura del proyecto del movimiento de la UGR.	4
2.1. Longitud de paso y de zancada [1].	12
2.2. Anchura del paso [1].	12
2.3. Ángulo del pie [1].	12
2.4. Imágenes de la producción <i>El origen del planeta de los simios</i> [2].	13
2.5. Animación de personajes virtuales en videojuegos [2].	14
2.6. Aplicaciones médicas [2].	14
2.7. Interfaces humano-máquina avanzados [2].	15
2.8. Sistema de vigilancia [2].	15
2.9. Sistema de interacción y aprendizaje para para un robot [2].	16
2.10. Clasificación de sistemas de captura del movimiento	16
2.11. Exoesqueleto con monitorización de movimientos <i>Lokomat</i> [3].	18
2.12. Sistema inercial. Ángulos de orientación [4].	18
2.13. Sistema magnético [3].	20
2.14. Sistema acústico [28].	20
2.15. Sistema CODAMotion [3].	21
2.16. Cámara rodeada de un anillo de LEDs infrarrojos [3].	23
2.17. Sistema de marcadores pasivos [3].	23

2.18. 3DN-GX4-45 [5].	25
2.19. Xsens MVN [6].	26
2.20. Traje Xsens MVN [6].	27
2.21. T9 DOF-Sensor Stick [7].	27
2.22. VICON 621 [8].	28
2.23. VICON 621: Marcadores [8].	29
2.24. Oqus Camera series [9].	29
2.25. Kinect Xbox 360 [10].	30
3.1. Diagrama general del GaitWatch siendo usado por un sujeto.	34
3.2. Definición del sistema de coordenadas deseado (a). Convención para la orientación de los ejes de rotación (b).	35
3.3. Orientación y ejes de coordenadas del GaitWatch.	36
3.4. Identificación del eje Z del acelerómetro en la unidad de caja.	37
3.5. Identificación de la señal de aceleración recogida en el eje Z.	38
3.6. Identificación de los ejes X(arriba) e Y(abajo) de la aceleración de la caja. .	38
3.7. Identificación delos eje Z (izquierda) y X (derecha) del acelerómetro en la unidad de la espinilla derecha.	39
3.8. Identificación del sentido de la rotación del eje Y del giróscopo en la unidad de la mano izquierda.	40
3.9. Señal de la tasa angular recogida recogida durante las maniobras de identificación de la rotación.	41
3.10. Señal de campo magnético registrada durante las maniobras de identificación del eje Y.	42
3.11. Diagrama general de ECnsole usado por un sujeto.	43
3.12. Sistema ECnsole.	44
3.13. Plantillas instrumentadas.	45
4.1. Acelerómetro mecánico [11].	49
4.2. Acelerómetro piezoeléctrico [12].	49

4.3. Sensor de efecto Hall con salida lineal de proporción [13].	50
4.4. Acelerómetro de condensador [14].	50
4.5. Fundamentos del giróscopo [15].	51
4.6. Fuerzas aplicadas sobre el sólido de rotación [15].	52
4.7. Giróscopo [16].	53
4.8. Giróscopo MEMS [17].	54
4.9. Funcionamiento interno de un sensor giroscópico MEMS [17].	54
4.10. Magnetómetro de precesión de protones [18].	55
4.11. Magnetómetro cuántico [19].	56
4.12. Magnetómetro vectorial [20].	57
4.13. Magnetómetro MEMS [21].	57
4.14. Aceleración recogida en las posiciones casi-estáticas y salida del algoritmo de detección	60
4.15. Representación 3D de la aceleración recogida durante las posiciones casi-estáticas	61
4.16. Representación 3D de las posiciones casi-estáticas calibradas	62
4.17. Aceleración original recogida cuando ambos ejes X y Z son colocados paralelo y anti-paralelo al vector gravedad	63
4.18. Campo magnético original recogido por el movimiento aleatorio de la unidad del tronco que contiene el magnetómetro	64
4.19. Campo magnético calibrado recogido moviendo aleatoriamente la unidad del tronco que contiene el magnetómetro.	65
4.20. Campo magnético original (izquierda) vs. Campo magnético calibrado (derecha) recogidos con movimientos aleatorios de la unidad del tronco que contiene el magnetómetro	66
4.21. Dispositivo de calibración del giróscopo	67
4.22. Rutina de calibración del giróscopo: selección del período para calcular el offset.	68
4.23. Rutina de calibración del giróscopo: selección de las rotaciones positivas para el cálculo del factor de escala	69
4.24. Esquema de coordenadas de navegación ENU [22].	71

4.25. Esquema de coordenadas de Body en un vehículo [23]	72
4.26. Representación bidimensional de los giros en roll, pitch y yaw [23].	73
4.27. Representación en el espacio de un cuaternion unitario [24].	76
4.28. Efectos del offset dinámico en la salida integrada de la señal del giróscopo . .	79
4.29. Diagrama de flujo de un filtro de Kalman de propósito general	84
4.30. Diagrama de flujo del enfoque estándar del filtro de Kalman con fusión de sensores	89
5.1. Ventana de comandos en <i>MATLAB</i>	92
5.2. Plantillas para MATLAB GUI.	93
5.3. Ventana de desarrollo de GUI.	94
5.4. Elemento Push Button. Sumador.	96
5.5. Opciones para un componente gráfico.	97
5.6. <i>Property Inspector</i>	97
5.7. Propiedades. <i>PushButton</i>	98
5.8. Opción. <i>Show functions</i>	98
5.9. Primer <i>Edit Text</i> . Sumador.	99
5.10. Segundo <i>Edit Text</i> . Sumador.	100
5.11. Static Text. Sumador.	101
5.12. Componentes GUI. Sumador.	102
5.13. Ventana gráfica para el programa <i>Sumador</i>	102
5.14. GUI sin hacer uso de GUIDE. Programa ejemplo.	104
5.15. Apariencia de la pantalla principal de la aplicación del GaitWatch.	105
5.16. Apariencia de la pantalla principal de la aplicación del GaitWatch. Opción: <i>File</i>	106
5.17. Selección del origen de los datos.	106
5.18. Selección del archivo de datos.	107
5.19. Selección de los errores del magnetómetro para su eliminación. Ejes: X,Y y Z.	107

5.20. Pantalla de selección de los componentes corporales.	108
5.21. Pantalla de selección de los componentes corporales. Select all/Unselect all. .	109
5.22. Pantalla de selección de los componentes corporales. Error.	109
5.23. Pantalla de selección de los componentes corporales. Aviso.	110
5.24. Pantalla de selección de señales.	111
5.25. Pantalla de selección de señales. Todas las señales seleccionadas.	112
5.26. Selección de parámetros: Acelerómetro, Giróscopo y Magnetómetro.	113
5.27. Selección de parámetros: Acelerómetro, Giróscopo y Magnetómetro.	113
5.28. Modificación de archivos de parámetros de calibración: Acelerómetro, Giróscopo y Magnetómetro.	114
5.29. Configuración de los algoritmos de estimación de la orientación.	114
5.30. Configuración de los algoritmos de estimación de la orientación. Piernas. . .	115
5.31. Configuración de los algoritmos de estimación de la orientación. Brazos. . .	115
5.32. Configuración de los algoritmos de estimación de la orientación. Tronco. . .	116
5.33. Barra de progreso.	116
5.34. Componentes <i>Set structure figures</i>	117
5.35. Figure Options.	119
5.36. Mensajes de error: Figure Options.	119
5.37. Set structure of figures: 1x1,1x2 y 1x3.	120
5.38. Set structure of figures: 2x1,2x2 y 2x3.	121
5.39. Set structure of figures: 3x1,3x2 y 3x3.	121
5.40. GaitWatch Manager: 1x1.	121
5.41. GaitWatch Manager: 1x2.	122
5.42. GaitWatch Manager: 1x3.	122
5.43. GaitWatch Manager: 2x1.	122
5.44. GaitWatch Manager: 2x2.	123
5.45. GaitWatch Manager: 2x3.	123

5.46. GaitWatch Manager: 3x1.	123
5.47. GaitWatch Manager: 3x2.	124
5.48. GaitWatch Manager: 3x3.	124
5.49. Figura eliminada.	125
5.50. Figura eliminada: Representación gráfica.	125
5.51. Save figure as.	127
5.52. Formatos de guardado.	128
5.53. Apariencia de la pantalla principal de la aplicación del GaitWatch. Opción: <i>Calibration</i>	128
5.54. Calibración de parámetros del acelerómetro.	129
5.55. Calibración de parámetros del giróscopo.	130
5.56. Calibración de parámetros del magnetómetro.	130
5.57. Selección de dispositivo.	130
5.58. Selección de dispositivo.	131
5.59. Pantalla principal: ECnsole.	132
5.60. Pantalla: Select the data file.	132
5.61. Pantalla: Select the data you wish to be displayed.	133
5.62. Pantalla: Select calibration parameters.	134
5.63. Selección de parámetros: Acelerómetro,Giroscopio y Magnetómetro.	135
5.64. Ventana <i>Select orientation estimation algorithm</i> para el pie izquierdo.	136
5.65. Ventana <i>Select orientation estimation algorithm</i> para el pie derecho.	136
5.66. Aviso: Pérdida de datos.	137
5.67. Pantalla principal: Set structure of figures.	137
5.68. Aviso: Pantalla principal: Representación gráfica.	138
5.69. Pantalla principal: Representación gráfica.	138
5.70. Acelerómetro: Cálculo de los parámetros de calibración. Aceleración original. .	139
5.71. Acelerómetro: Cálculo de los parámetros de calibración. Aceleración calibrada. .	139

5.72. Giroscopio: Cálculo de los parámetros de calibración. Aceleración calibrada.. .	140
5.73. Magnetómetro: Cálculo de los parámetros de calibración. Campo magnético original.	140
5.74. Magnetómetro: Cálculo de los parámetros de calibración. Campo magnético calibrado.	141
5.75. Select device.	142

Índice de tablas

3.1.	Lista de acelerómetros del GaitWatch y sus canales asociados	39
3.2.	Lista de giróscopos del GaitWatch y sus canales asociados.	41
3.3.	Desplazamiento del magnetómetro y ejes	42
4.1.	Lista de sensores que componen los sistemas GaitWatch y ECnsole	58
4.2.	Parámetros de calibración de los acelerómetros biaxiales	64
4.3.	Parámetros de calibración de los giróscopos	69
5.1.	Tabla de componentes en GUIDE	95
5.2.	Señales del sistema GaitWatch Manager	111
5.3.	Componentes <i>Set structure of figures.</i>	118
5.4.	Barra de herramientas <i>Matlab.</i>	126
5.5.	Señales del sistema ECnsole	134

Introducción

1.1. Contexto

Actualmente, las enfermedades neurodegenerativas son una de las causas de muerte más importantes de los países desarrollados. El incremento de la esperanza de vida y el envejecimiento de la población constituyen un factor determinante que implica que cada vez existan más casos de enfermedades de este tipo.

En el últimos años el grupo de investigación SiPBA (TIC 218) del Departamento de Teoría de la Señal, Telemática y Comunicaciones, adscrito al CITIC-UGR, ha venido aplicando los métodos desarrollados en el pasado, en la detección y diagnóstico de Alzheimer, en la iniciativa PPMI de la Fundación Michael J. Fox para el estudio de la enfermedad de Parkinson. Además se han empleado otras bases de datos como las de los hospitales Virgen de la Victoria de Málaga y Virgen de las Nieves de Granada, con los que el grupo mantiene una estrecha colaboración.

Por otro lado, a finales del año 2013 de la mano de D. Alberto Olivares (integrante del grupo de investigación SipBA) se trabajó en colaboración con médicos para desarrollar una caja de herramientas Matlab que fuese completa. El sistema determina la posición humana utilizando los datos iniciales y se está utilizando actualmente en el Laboratorio de Análisis de Movimiento del Servicio de Neurología (Klinikum Großhadern) para obtener informes objetivos de la marcha y el movimiento de los pacientes de Parkinson. Mi proyecto supone el desarrollo de una aplicación para que permita visualizar los datos y se pueda configurar dicho sistema.

El trabajo se engloba dentro del proyecto de análisis del movimiento del cuerpo humano, llevado a cabo entre los departamentos de ATC (Arquitectura y Tecnología de los Computadores) y TSTC (Teoría de la Señal, Telemática y Comunicaciones), de la Universidad de

Granada.

Para ser más concretos, el proyecto se incluiría en la capa de *aplicaciones* puesto que consiste en el desarrollo de una aplicación de visualización de datos y configuración para sistemas de monitorización y análisis del movimiento del cuerpo humano.



Figura 1.1: Estructura del proyecto del movimiento de la UGR.

1.2. Motivación

Como ya habíamos hablado anteriormente, a la hora de desarrollar nuestra aplicación partíamos de un sistema capaz de determinar la posición humana utilizando los datos iniciales. La búsqueda de un resultado óptimo, requiere el procesamiento y transformación de los datos iniciales. Aquí es cuando entran en juego los algoritmos de monitorización de la marcha.

En los estudios aplicados en su tesis [25] por el Dr. Alberto Olivares Vicente se analizan de forma detallada los algoritmos de calibración existentes en la actualidad y se observa que existen dos grandes grupos de métodos, aquellos basados en la minimización del error o coste para estimar los parámetros de calibración, y aquellos que calculan los parámetros de forma directa aplicando una serie de trasformaciones algebraicas.

En cuanto a los métodos de estimación de la orientación comenzaron a ser desarrollados en los años setenta para las misiones espaciales. Se desarrollaron una serie de algoritmos con el objeto de determinar la orientación de la nave u objeto espacial respecto a distintos sistemas

de referencia. Los métodos TRIAD Y QUEST (por citar algunos), pueden ser considerados ejemplos de estos primeros algoritmos para la estimación de la orientación.

Hoy en día y sirviéndonos del análisis de los algoritmos de estimación de la orientación efectuado en [25], se llega a la conclusión de que prácticamente el 100 % de los algoritmos existentes estarían basados en la fusión de varios sensores utilizando el filtro de Kalman.

Ahora que ya podemos decir que estamos mejor situados, cabe destacar que la motivación principal de nuestro proyecto surge de la idea de integrar en una aplicación visual los distintos algoritmos de monitorización desarrollados hasta la fecha para el sistema sobre el que partíamos inicialmente. De hecho, lo que se pretende es facilitar el uso de los algoritmos por parte de personal no especializado.

1.3. Objetivos

El **objetivo principal** del proyecto es desarrollar una aplicación gráfica en Matlab que permita la configuración y visualización de los datos recogidos por los sistemas de medida de parámetros de la marcha GaitWatch y ECnsole. Los sistemas están compuestos por una serie de módulos que contienen sensores magnéticos e iniciales del tipo MEMS que son colocados en diferentes segmentos corporales. Este sistema permite la monitorización del movimiento del cuerpo humano y está orientado al cálculo de parámetros de la marcha (longitud de zancada, cadencia, etc.). Además, este sistema está siendo utilizado en la actualidad en la unidad de análisis del movimiento dentro del Departamento de Neurología del Hospital Clínico Großhadern de Múnich.

A continuación procedemos a detallar el objetivo principal de una forma más específica:

- **Análisis de los sistemas de monitorización de la marcha.** Si queremos cubrir los objetivos globales que se imponen inicialmente, debemos hacer un estudio general de las características del hardware utilizado. A partir de las conclusiones extraídas, aplicamos un análisis de los requisitos que debe cumplir la aplicación.
- **Desarrollo del software.** Una vez finalizada la etapa anterior construimos la interfaz gráfica teniendo en cuenta los requerimientos fijados. Es la fase más larga y de mayor esfuerzo.
- **Integración de algoritmos.**

En esta fase se incluyen los algoritmos ya desarrollados para la monitorización de la marcha en la aplicación. La ejecución debe resultar imperceptible para el usuario mientras hace uso del programa.

- ◊ **Algoritmos de calibración.** El usuario tendrá la posibilidad de cargar los parámetros de calibración calculados anteriormente.

- ◊ **Algoritmos de orientación.** El usuario podrá modificar los parámetros de los algoritmos en función de los requerimientos del movimiento que ha sido monitorizado.

Además del objetivo principal, el proyecto intentará cumplir otra serie de objetivos:

- **Revisión del estado del arte:** Esta parte consiste en hacer un recorrido por todos los sistemas desarrollos recientemente relacionados con nuestro dispositivo, que han sido probados en la industria y han sido acogidos y aceptados por diferentes fabricantes.
- **Estudio teórico de la materia.** Este objetivo tiene como finalidad comprender todos los conceptos teóricos relacionados con los sistemas de monitorización de la marcha.

1.4. Estructura del documento

Este proyecto comienza con el Capítulo 1, el cual es una introducción, y cuyo fin por un lado es poner el contexto en que se enmarca el proyecto y por otro lado guiar al lector en la motivación y objetivos de la presente obra.

El Capítulo 2 está estructurado en dos bloques principalmente, el primero define el concepto de *análisis de la marcha* y hace un repaso breve de su historia. Posteriormente se describen todos los factores y parámetros que se necesitan tener en cuenta para el análisis de la marcha. Después, se ven las aplicaciones que podemos encontrar de los sistemas de captura del movimiento humano (es necesario aclarar que este tipo de sistemas engloban a los sistemas orientados al análisis de la marcha). Finalmente, se establece una clasificación de los sistemas de captura del movimiento humano. Una vez que pasamos al segundo bloque, se pretende establecer una visión general de los dispositivos disponibles en el mercado con características similares, los cuales han servido de referencia y motivación. Para ello se hace una revisión del estado del arte.

En el Capítulo 3 se realiza una descripción general de los sistemas GaitWatch y ECnsole. Además, se especifican las características técnicas de cada uno de los dispositivos. En el caso del GaitWatch, se le añade adicionalmente una explicación exhaustiva de como se identifican los ejes de los sensores iniciales que contiene.

En el Capítulo 4 se realizará un extenso análisis a nivel teórico de los fundamentos de los sensores iniciales (acelerómetro, giróscopo y magnetómetro) utilizados en los sistemas GaitWatch y ECnsole. La segunda parte del capítulo se centrará primero en explicar el proceso de calibración, cuyo objetivo es transformar los datos originales en unidades con sentido físico. Después se describirán los algoritmos para el cálculo de la orientación. Primero el método de *Integración de la Velocidad Angular* para el giróscopo. Posteriormente, el método de *Descomposición de la Gravedad* donde el acelerómetro y el magnetómetro trabajan

de manera conjunta. Después, el *Filtro de Kalman* que fusiona los datos del acelerómetro, giróscopo y manetómetro obteniendo unos resultados más precisos. Y por último, el *Filtro de Kalman Extendido* (una mejora del *Filtro de Kalman* gracias al uso de un elemento nuevo, los cuaterniones, que minimizan el cálculo computacional).

El Capítulo 5 comienza con una introducción teórica acerca de Matlab GUI para que el lector se pueda familiarizar con la creación de interfaces gráficas a través de Matlab. Inicialmente se ofrece la posibilidad de uso del entorno de desarrollo GUIDE para el diseño de aplicaciones relativamente sencillas, GUIDE es una herramienta que permite facilitar el diseño de mismas. Posteriormente se dan las pautas a seguir si decidimos prescindir de dicha herramienta, es decir, si se opta por la creación de las GUIs de forma programática (el proyecto ha sido desarrollado de esta manera). Una vez finalizado este bloque, se continúa con la descripción de la aplicación (la interfaz gráfica desarrollada que constituye el proyecto). Aquí se van metiendo en distintas secciones cada una de las partes de la aplicación con sus capturas y explicaciones correspondientes.

Para terminar, en el Capítulo 6 se extraen las conclusiones finales del proyecto y se incorpora una visión sobre posibles mejoras y líneas futuras de trabajo.

Capítulo **2**

Monitorización del cuerpo humano

Este capítulo está estructurado en dos bloques principalmente, el primero define el concepto de *análisis de la marcha* y hace un repaso breve de su historia. Posteriormente se describen todos los factores y parámetros que se necesitan tener en cuenta para el análisis de la marcha. Después, se ven las aplicaciones que podemos encontrar de los sistemas de captura del movimiento humano (es necesario aclarar que este tipo de sistemas engloban a los sistemas orientados al análisis de la marcha). Finalmente, se establece una clasificación de los sistemas de captura del movimiento humano. Una vez que pasamos al segundo bloque, se pretende establecer una visión general de los dispositivos disponibles en el mercado con características similares, los cuales han servido de referencia y motivación. Para ello se hace una revisión del estado del arte.

2.1. Definición y ámbito de uso

GaitWatch y ECnsole son sistemas para la monitorización y análisis de los parámetros de la marcha. Inicialmente vamos a definir el concepto, la historia y los parámetros que caracterizan la marcha humana. Para situarnos mejor y poner en contexto los sistemas utilizados, debemos incluir estos dispositivos dentro de los sistemas de captura del movimiento. El concepto es más amplio y engloba un mayor número de aplicaciones. La captura de movimiento es el proceso de registrar el movimiento de objetos o personas, es decir, no sólo bajo condiciones de desplazamiento. En los apartados siguiente se describirán varias de las aplicaciones y la forma de clasificar los sistemas de captura del movimiento presentes en la actualidad.

2.1.1. Definición: Análisis de la marcha

La marcha humana [1] es un proceso de locomoción en el que el cuerpo humano, en posición erguida, se mueve hacia delante, siendo su peso soportado alternativamente por ambas piernas. Mientras el cuerpo se desplaza sobre la pierna de soporte, la otra pierna se balancea hacia delante como preparación para el siguiente apoyo.

La ciencia encargada del análisis de la marcha es la biomecánica [26]. En la actualidad continúa en desarrollo debido a la dificultad para conseguir establecer una normalización. Una misma persona puede realizar el mismo movimiento de formas muy distintas, ya que influyen múltiples factores como la carga exterior añadida, la velocidad y, en el caso del caminar, también el calzado. Además, cada movimiento humano, realizado por las extremidades superiores o inferiores, depende en gran medida de las dimensiones de las mismas y de la distribución de la masa, con lo que el mismo movimiento puede diferir mucho de una persona a otra. El hecho de que no exista el *hombre medio* complica este tipo de estudios. Las dimensiones del cuerpo humano varían según edad, sexo, raza, e incluso grupo laboral.

2.1.2. Historia

Los pioneros de la investigación científica sobre el análisis de la marcha fueron Aristóteles en *De Motu Animalium* (En la marcha de los Animales) y más tarde en 1680, Giovanni Alfonso Borelli, también llamado *De Motu Animalium* (I et II). En la década de 1890, el anatomista alemán Christian Wilhelm Braune y Otto Fischer publicó una serie de documentos sobre la biomecánica de la marcha humana en condiciones de carga y descarga.

Con el desarrollo de la fotografía, se hizo posible la captura de secuencias de imágenes que revelan detalles de la locomoción humana y animal que pueden ser percibidos con el ojo humano. Eadweard Muybridge y Marey Étienne-Jules fueron los pioneros de esta práctica a principios de 1900. Por ejemplo, fue la fotografía la que reveló por primera vez la secuencia detallada de la marcha del caballo "a galope", que solía estar mal representada en las pinturas realizadas con anterioridad a este descubrimiento.

Aunque las investigaciones anteriores fueron hechas en el campo de las cámaras de película, la aplicación generalizada de análisis de la marcha se realiza en seres humanos que presentan patologías como parálisis cerebral, enfermedad de Parkinson y trastornos neuromusculares, que se inició en la década de los 70 con la ayuda de sistemas de cámara de vídeo que pueden realizar los estudios detallados de cada uno de los pacientes dentro de los límites realistas de costes y tiempo. El desarrollo de regímenes de tratamiento, a menudo en cirugía ortopédica, basados en los resultados de análisis de la marcha han avanzado significativamente en la década de los 80. Muchos de los principales hospitales ortopédicos en todo el mundo tienen ahora los laboratorios para realizar estos estudios de la marcha que se utilizan habitualmente en un gran número de casos médicos, tanto para el diseño de planes de tratamiento como para el control de seguimiento.

La implementación de ordenadores modernos con la tecnología para realizar estos estudios se produjo de forma independiente a finales de la década de los 70 y principios de los 80 en laboratorios de investigación de varios hospitales, algunos a través de colaboraciones con la industria aeroespacial. Pronto siguió con el desarrollo comercial, con la aparición de sistemas de movimiento Vicon y BTS, y con el marketing de sistemas de análisis de la marcha a mediados de la década de 1980.

Para más información sobre la historia del estudio de la marcha consulte [27].

2.1.3. Fáctores y parámetros

El proceso de deambulación está modulado o modificado por muchos factores [27], y los cambios que imprimen en el patrón de marcha habitual pueden ser transitorios o permanentes. Los factores pueden ser de diversos tipos:

- **Extrínsecos:** Hay varios factores extrínsecos, como por ejemplo el terreno, el calzado, la vestimenta, transporte de carga.
- **Intrínsecos:** Los factores intrínsecos son sexo (hombre y mujer), peso, altura, edad,etc.
- **Físicos:** Los factores físicos incluyen el peso, la talla y la constitución física.
- **Psicológicos:** Los factores psicológicos incluyen la personalidad y las emociones.
- **Fisiológicos:** Cuando se habla de factores fisiológicos se trata de características antropométricas.
- **Patológicos:** Los factores patológicos pueden ser por ejemplo traumatismos, patologías neurológicas, músculo esquelética y trastornos psiquiátricos.

Los parámetros [1] que se tiene en cuenta para el análisis de la marcha son:

- **Longitud del paso completo:** Es la distancia lineal entre los sucesivos puntos de contacto del talón de mismo pie. Aproximadamente 156 cm por paso completo.
- **Longitud del paso:** Es la distancia lineal en el plano de progresión entre los puntos de contacto de un pie y el otro pie. La mitad en cm de la longitud de paso completo. Estos 2 parámetros se pueden ver en la figura 2.1:

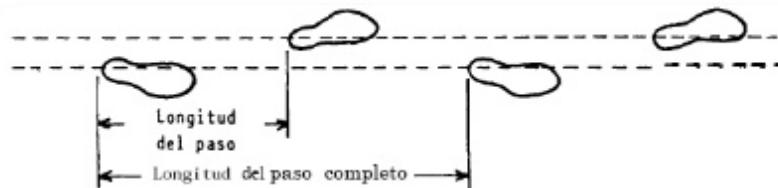


Figura 2.1: Longitud de paso y de zancada [1].

- **Cadencia:** El número de pasos por unidad de tiempo (pasos por minuto). Aproximadamente entre 60 y 117 pasos por minuto.
- **Anchura del paso:** La distancia entre la línea media de un pie y línea media del otro pie. De 5 a 10 cm. Se muestra en la figura 2.2:

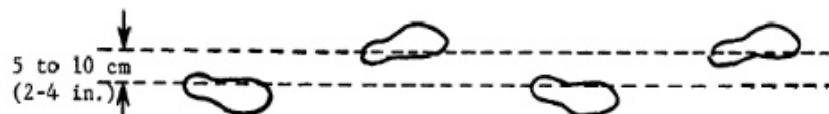


Figura 2.2: Anchura del paso [1].

- **Ángulo del pie:** El ángulo en el cual normalmente se desvía la punta del pie hacia fuera de la línea de progresión. De 6.7 a 6.8 grados. Se muestra en la figura 2.3:

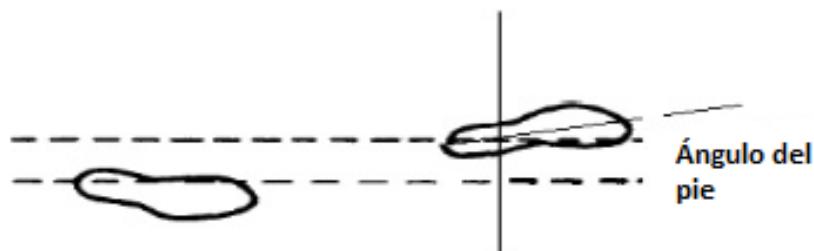


Figura 2.3: Ángulo del pie [1].

2.1.4. Aplicaciones de los sistemas de captura del movimiento humano

Los sistemas HMC (cáptura del movimiento humano) tienen un gran número de aplicaciones [2], que se han visto significativamente incrementado en los últimos años. Entre estas aplicaciones se pueden destacar las siguientes:

- **Animación de personajes virtuales en el cine:** En el cine se emplean cada vez más personajes generados por ordenador. Para conseguir que dichos personajes actúen y se muevan de forma natural, se capturan los movimientos de un actor real y luego dichos movimientos se trasladan al cuerpo del actor virtual. En la figura 2.4 tenemos un ejemplo.



Figura 2.4: Imágenes de la producción *El origen del planeta de los simios* [2].

- **Animación de personajes virtuales en videojuegos:** Los videojuegos (de ordenador y de consola) son hoy día la principal industria del ocio. Su complejidad y grado derealismo ha aumentado enormemente en los últimos años. Al igual que ocurre en el cine, para conseguir personajes que se mueven de forma fluida y natural, se capturan los movimientos de un actor real y luego se trasladan al actor virtual. La figura 2.5 muestra el proceso.

Los sistemas HMC empleados para capturar movimientos para el cine o para un videojuego son esencialmente los mismos y, aunque tradicionalmente en los videojuegos se permitían animaciones de calidad algo inferior, la brecha entre ambas aplicaciones se va cerrando conforme los videojuegos crecen en complejidad, detalle y rentabilidad.



Figura 2.5: Animación de personajes virtuales en videojuegos [2].

- **Aplicaciones médicas:** Los sistemas HMC como el mostrado en la figura 2.6 se utilizan con fines médicos, para detectar lesiones, malformaciones o problemas a partir del análisis de los movimientos de una persona, para recopilar datos sobre la evaluación del paciente en terapias de rehabilitación, o para realizar estudios de ergonomía (por ejemplo, comprobar si un determinado mobiliario de oficina fuerza posturas perjudiciales o no).



Figura 2.6: Aplicaciones médicas [2].

- **Interfaces humano-máquina avanzados:** El control de un ordenador mediante gestos puede realizarse si se utiliza un sistema HMC que capture los movimientos de una persona, sobre todo los de las manos, y en la cara los ojos para saber dónde está mirando. Como el de la figura 2.7.



Figura 2.7: Interfaces humano-máquina avanzados [2].

- **Sistemas de vigilancia:** Un sistema de seguridad que vigile una determinada zona (por ejemplo, una cámara de vigilancia en un parking) puede ver su eficacia multiplicada si se le añaden funcionalidades de captura y análisis de movimiento automáticos. La figura 2.8 muestra este tipo de sistema. Observando el patrón de movimientos de una persona, por ejemplo, se pueden detectar conductas sospechosas (por ejemplo, una persona en un parking que en lugar de entrar o salir de un coche da vueltas en torno a ellos, examinándolos) o situaciones potencialmente peligrosas (por ejemplo, una persona que deja una mochila en un rincón y se marcha).



Figura 2.8: Sistema de vigilancia [2].

- **Sistemas de interacción y aprendizaje para robots sociales:** Un robot social debe cooperar con personas en tareas cotidianas. Eso implica ser capaz de percibir a la persona y analizar sus movimientos, con un sistema HMC, para coordinar la acción del robot con esos movimientos. La figura 2.9 nos muestra un ejemplo.



Figura 2.9: Sistema de interacción y aprendizaje para para un robot [2].

2.1.5. Clasificación de los sistemas de captura del movimiento humano

Hay muchas formas distintas de capturar los movimientos de una persona. Los sistemas de captura varían en su precio, precisión, complejidad, tiempo de respuesta, robustez, versatilidad, etc.

En la figura 2.10 se pueden distinguir dos bloques principales en función del tipo de tecnología utilizada: los sistemas fundamentados en cámaras y los que no. Cuando lleguemos al apartado del *Estado del Arte* nos centraremos exclusivamente en los sistemas que funcionan con sensores inerciales (los dispositivo GaitWatch y ECnsole pertenecen a este grupo) y de manera más genérica se tratarán los distintos tipos de dispositivos de captura del movimiento con cámaras.



Figura 2.10: Clasificación de sistemas de captura del movimiento

Toda la información acerca de la clasificación de los sistemas HMC ha sido extraída en [3] donde se hace una clasificación minuciosa de este tipo de sistemas.

2.1.5.1. Sistemas de captura del movimiento no ópticos

Los sistemas HMC no ópticos se dividen en las siguientes categorías:

Sistemas mecánicos:

Consisten en una serie de brazos mecánicos como los que se muestran en la figura 2.11, adheridos mediante algún sistema de agarre o anclaje a diversas partes del cuerpo del usuario (por ejemplo, un cierre sobre la muñeca del mismo), de forma que dichos brazos sigan los movimientos de las extremidades, torso, cabeza, etc., del usuario. Los ángulos de giro de estos brazos mecánicos (que conforman lo que se conoce como exoesqueleto) se corresponden con los ángulos de giro de las articulaciones.

- Ventajas

- ◊ Se obtiene información muy precisa.
- ◊ Son muy rápidos.
- ◊ Son muy robustos frente a distorsiones producidas, por ejemplo, por el temblor de la mano.
- ◊ Son inmunes a interferencias y occlusiones.

- Inconvenientes

- ◊ Es necesario otro sistema de seguimiento para efectuar medidas de posición.
- ◊ El montaje de estos sistemas es muy aparatoso.
- ◊ Las propias limitaciones del movimiento mecánico y la necesidad de mantener fijos los puntos de unión al cuerpo del usuario disminuyen considerablemente los grados de libertad que el sistema de seguimiento es capaz de registrar, además de afectar a la movilidad del propio usuario.
- ◊ Debido a que sólo registran ángulos de giro, tienden a generar movimientos poco naturales, en los que el modelo de persona parece una marioneta que cuelga de hilos.



Figura 2.11: Exoesqueleto con monitorización de movimientos *Lokomat* [3].

Sistemas inerciales:

Estos sistemas se basan en el empleo de sensores inerciales en miniatura, usualmente giróscopos que miden la rotación de cada articulación (una tecnología similar a la usada en el mando de la Wii). Mediante estos dispositivos, por el principio de conservación del momento angular, se pueden calcular los giros en los ejes longitudinal, transversal y vertical partiendo de la resistencia inercial presentada por el giroscopio a un cambio en su orientación. Los valores de estos ángulos se envían usualmente a través de un enlace wireless a un ordenador donde los datos se filtran y procesan para conformar la pose percibida.

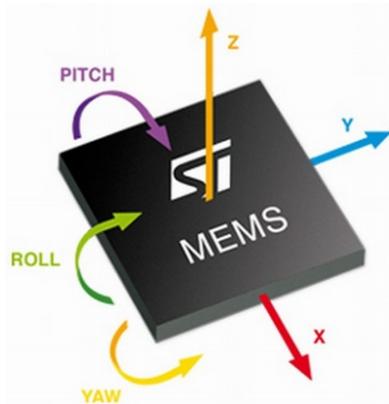


Figura 2.12: Sistema inercial. Ángulos de orientación [4].

- Ventajas

- Tienen un alcance muy amplio, por lo que su rango de acción es considerablemente elevado.
- No necesitan de un entorno controlado para ser utilizados.

- Incovenientes

- ◊ Es necesario otro sistema de seguimiento para efectuar medidas de posición.
- ◊ Mucha sensibilidad a las vibraciones.
- ◊ Tendencia a presentar derivas provocadas por acumulación de errores.

Sistemas magnéticos:

Estos sistemas (vease figura 2.13) utilizan un dispositivo transmisor de pulsos y una serie de receptores. Su principio de funcionamiento consiste en generar en el transmisor tres campos electromagnéticos perpendiculares, y cada receptor calcula la attenuación producida en cada campo. Con esta información, se calcula la posición y la orientación del receptor mediante una triangulación de los datos recibidos en relación a los tres ejes correspondientes a los campos electromagnéticos generados inicialmente.

- Ventajas

- ◊ Proporcionan información sobre los 6 grados de libertad de las articulaciones usando menos sensores que otros sistemas (por ejemplo, sólo dos tercios de la cantidad que requeriría un sistema HMC óptico que proporcionase la misma información).
- ◊ No se ven afectados por oclusiones de objetos no metálicos.

- Incovenientes

- ◊ El cableado de los sensores hace a estos sistemas aparatosos y molestos a la hora de realizar movimientos amplios y/o rápidos.
- ◊ Tienen un alcance limitado.
- ◊ Presentan latencia a la hora de presentar los datos (esto es, no son rápidos).
- ◊ Son muy sensibles a interferencias magnéticas producidas por elementos metálicos del entorno, o por dispositivos que emiten radiación electromagnética (como monitores o computadoras).
- ◊ La respuesta de estos sensores es no lineal, especialmente en los bordes del área de captura.



Figura 2.13: Sistema magnético [3].

Sistemas acústicos:

Este sistema consiste en un emisor de ondas sonoras de alta frecuencia y su sistema receptor correspondiente. El sistema receptor está formado por tres micrófonos en posición fija, y, a su vez, el sistema transmisor está formado por tres transmisores también en posición fija. Las medidas pueden efectuarse atendiendo a la coherencia de fase o a medidas de tiempos. En el primer caso, las medidas de posición y orientación se obtienen calculando el desfase entre la señal transmitida y la recibida; por tanto, siempre que el objeto al que se le hace el seguimiento no se desplace una distancia superior a la longitud de onda de la señal utilizada en el intervalo de tiempo entre una medición y otra, el resultado obtenido será correcto. En el segundo caso, se mide el tiempo que tarda el sonido emitido por el transmisor en un instante predeterminado en llegar al receptor; las medidas de posición sólo requieren un transmisor, pero para medir orientaciones sí es necesario el uso del sistema emisor formado por tres transmisores de alta frecuencia. Un ejemplo de este tipo de sistemas se muestra en la figura 2.14.

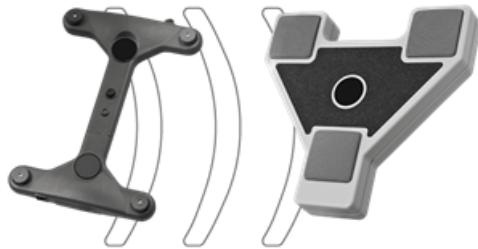


Figura 2.14: Sistema acústico [28].

- Ventajas

- ◊ Muy preciso.
- Incovenientes
 - ◊ El cableado de los sensores hace a estos sistemas aparatosos y molestos a la hora de realizar movimientos amplios y/o rápidos.
 - ◊ Tienen un alcance limitado.
 - ◊ El número de sensores de este tipo que se pueden usar simultáneamente es limitado.
 - ◊ Muy sensible a las condiciones ambientales de presión, humedad y temperatura.
 - ◊ Muy sensible a las interferencias acústicas provocadas por los objetos del entorno, lo que implica en la práctica que sólo pueden ser usados en entornos muy controlados.

2.1.5.2. Sistemas de captura del movimiento ópticos

Sistemas ópticos basados en marcadores activos

En estos sistemas, los marcadores son diodos LED (usualmente infrarrojos). Las cámaras no captan el reflejo de luz en los marcadores, sino que captan, directamente, la luz emitida por los marcadores.

Como quiera que estos marcadores han de emitir luz, necesitan disponer de una fuente de energía, a la que se conectan a través de cables. Por ello, los sistemas HMC basados en marcadores activos son más aparatosos que los basados en marcadores pasivos, ya que la persona debe llevar puestos no sólo los marcadores, sino también los codificadores, las baterías y los cables que lo conectan todo. En la figura 2.15 se muestra un sistema de estas características.

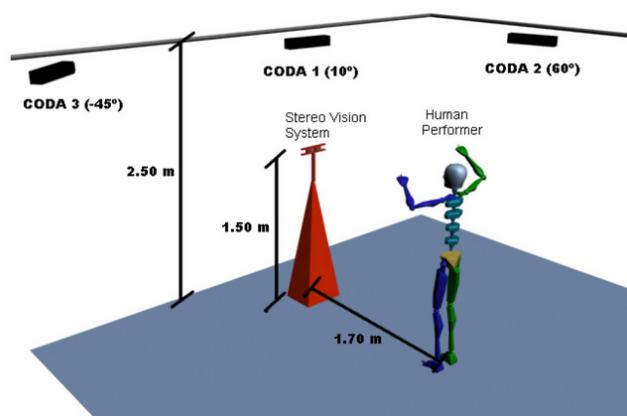


Figura 2.15: Sistema CODAMotion [3].

- Ventajas

- ◊ Son muy precisos.
- ◊ Pueden capturar movimientos muy rápidos (igual que los basados en marcadores pasivos).
- ◊ Son menos sensibles a interferencias producidas por fuentes de luz externas. Recientemente, por ejemplo, se ha empezado a comercializar un sistema de este tipo que funciona en exteriores.

- Incovenientes

- ◊ Son caros.
- ◊ Requieren un entorno controlado amplio (volumen de captura).
- ◊ Los marcadores han de colocarse cuidadosamente sobre el usuario para evitar derivas entre pruebas.
- ◊ Hay que recalibrarlos frecuentemente.
- ◊ Son sensibles a oclusiones. De hecho, utilizar muchas cámaras es una forma, sobre todo, de intentar evitar las oclusiones de los marcadores.
- ◊ No hay forma de distinguir unos marcadores de otros. Es necesario un postprocesado, a veces tedioso, para inferir qué marcador se corresponde con cada marca detectada.

Sistemas ópticos basados en marcadores pasivos

En este caso, se emplean pequeños marcadores (usualmente esféricos) cubiertos de un material retroreflector (un material que devuelve la luz que incide sobre él hacia la fuente de luz, prácticamente sin dispersión espacial).

En estos sistemas, se colocan una serie de cámaras alrededor de los marcadores que se quieren capturar. Cada cámara está a su vez emparejada con un dispositivo emisor de luz, usualmente infrarroja. De hecho, la configuración típica para estas cámaras es rodearlas de un anillo de LEDs infrarrojos, como se muestra en la figura 2.16 (donde los LEDs se ven rojos porque la emisión es en una banda de frecuencias centrada en el infrarrojo, pero que se extiende un poco al espectro visible).



Figura 2.16: Cámara rodeada de un anillo de LEDs infrarrojos [3].

Cuando la luz emitida incide sobre el marcador retrorreflector, ésta se refleja directamente hacia la cámara como en la figura 2.17, con lo que el marcador aparece en la imagen como un punto brillante, que destaca respecto al resto. Utilizando un umbral de intensidad, es posible detectar en qué píxeles 2D de la imagen se han reflejado todos los marcadores presentes.



Figura 2.17: Sistema de marcadores pasivos [3].

- Ventajas
 - ◊ Son muy precisos.
 - ◊ Pueden capturar movimientos muy rápidos (la tasa de captura suele ser de 120 a 160 imágenes por segundo, aunque algunos sistemas pueden conseguir tasas de hasta 10000 imágenes por segundo en volúmenes reducidos).
 - ◊ El usuario no tiene que portar cables, baterías ni codificadores.
- Incovenientes
 - ◊ Son caros.
 - ◊ Requieren un entorno controlado amplio (volumen de captura).
 - ◊ Los marcadores han de colocarse cuidadosamente sobre el usuario para evitar derivas entre pruebas.

- ◊ Hay que recalibrarlos frecuentemente.
- ◊ Son sensibles a oclusiones. De hecho, utilizar muchas cámaras es una forma, sobre todo, de intentar evitar las oclusiones de los marcadores.
- ◊ No hay forma de distinguir unos marcadores de otros. Es necesario un postprocesado, a veces tedioso, para inferir qué marcador se corresponde con cada marca detectada.

Sistemas ópticos sin marcadores

Los sistemas HMC anteriores son adecuados para grabar movimientos para una película o un videojuego, o para monitorizar la evolución de un paciente en procesos de rehabilitación. Sin embargo, si se quiere crear un interfaz para el ordenador basado en gestos, o controlar la televisión con movimientos de la mano, no se puede exigir al usuario que se coloque marcadores por el cuerpo, y que calibre un anillo de cámaras. Por otro lado, ¿quién demostraría nuevas tareas a un robot social, si para ello ha de acudir a una sala específica, colocarse un traje de marcadores, calibrar cámaras y postprocesar los datos?

El problema de los sistemas HMC ópticos que no usan marcadores es, precisamente, que tienen que arreglárselas para extraer la pose sin marcas. Para conseguirlo, han de ejecutar complejos algoritmos de filtrado, reconocimiento, clasificación, aprendizaje o emparejamiento. Estos algoritmos consumen recursos, y tienden a requerir bastante tiempo para completarse. Así, uno de los mayores inconvenientes de los sistemas HMC ópticos sin marcadores es el de su limitada velocidad. El otro gran inconveniente de estos sistemas es su imprecisión: por mucho que se refinen los algoritmos de extracción de pose, sus resultados no mejorarán la precisión que da un marcador óptico.

2.2. Estado del arte

Tal y como adelantamos en el apartado anterior, vamos a hacer un barrido de algunos de los dispositivos asociados a la monitorización del cuerpo humano presentes en la actualidad. Nos centraremos en dos tipos principalmente: los de dispositivos fundamentados en sensores iniciales (IMUs y MIMUs) y los dispositivos basados en cámaras.

2.2.1. Sistemas basados en sensores iniciales

Para situarnos mejor definiremos lo que es una IMU:

Una unidad de medición inercial o IMU (del inglés inertial measurement unit) [29], es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas

gravitacionales de un cuerpo que se mueve en el espacio, usando una combinación de acelerómetros y giróscopos. Además existe la posibilidad de combinarlos con magnetómetros, en ese caso el dispositivo pasa a llamarse MIMU. Las unidades de medición inercial son normalmente usadas para maniobrar aviones, incluyendo vehículos aéreos no tripulados, entre muchos otros usos, y además naves espaciales, incluyendo transbordadores, satélites y aterrizadores.

La IMU es el componente principal de los sistemas de navegación inercial usados en aviones, naves espaciales, buques y misiles guiados entre otros. En este uso, los datos recolectados por los sensores de una IMU permiten a un computador seguir la posición del aparato, usando un método conocido como navegación por estima.

A continuación se enumeran algunas de las MIMUs que pueden ser encontradas en el mercado:

2.2.1.1. 3DN-GX4-45 (*Microstrain*)

El 3DM-GX4-45 [5] es una miniatura de un tipo de Sistema de Navegación Inercial(GPS/INS), que utiliza la tecnología más avanzada de sensores MEMS. Combina un acelerómetro triaxial, un giróscopo triaxial, un magnetómetro triaxial, sensores de temperatura, un altímetro barométrico, un procesador de doble núcleo que ejecutan el *Filtro de Kalman Extendido* (EKF) para proporcionar la posición, velocidad y la estimación de la orientación de forma precisa. Su factor de forma, rango de temperatura, supervivencia a choques, bias estable, rendimiento ante las vibraciones, bajo consumo y rendimiento se combinan para ser uno de los mejores GPS/INS en su clase. La figura 2.18 muestra el dispositivo.



Figura 2.18: 3DN-GX4-45 [5].

2.2.1.2. Xsens MVN

Xsens MVN [6] es una solución para todo el cuerpo, se basa en la captura del movimiento sin cámara (MoCap). Consta de sensores iniciales adheridos al cuerpo por un traje de lycra (también disponible en tiras). Xsens MVN da libertad de movimiento porque no utiliza cámaras. Se trata de un sistema de captura portátil que se puede utilizar en interiores y al

aire libre. Se caracteriza por su facilidad de uso y un tiempo de calibración corto que permite configurar el sistema en menos de 15 minutos. En la figura 2.19 se puede ver Xsens MVN.



Figura 2.19: Xsens MVN [6].

Los datos resultantes de la captura del movimiento pueden ser utilizados para animar personajes digitales en películas, juegos, cuentos animados, anuncios,etc. También pueden ser utilizados en aplicaciones médicas y de entretenimiento para analizar el movimiento.

El traje MVN de captura del movimiento que se muestra en la figura 2.20 presenta las siguientes características:

- Seguimiento 6DOF (Degrees of Freedom) del cuerpo.
- 17 rastreadores inerciales MTx.
- Funcionamiento inalámbrico de plena libertad de circulación.
- Traje de lycra cómodo con cableado integrado.



Figura 2.20: Traje Xsens MVN [6].

2.2.1.3. 9 DOF-Sensor Stick (*SparkFun*)

Los giroscopos y los acelerómetros son bastante prácticos, pero por sí solos no dan la suficiente información para poder calcular cómodamente variables como la orientación, la posición o la velocidad. Para medir esas y otras variables se suelen combinar los dos sensores, para crear una unidad de medición inercial (IMU) que proporciona seis grados de libertad (DOF). También cabe la posibilidad de añadir un magnetómetro pasando a tener nueve grados de libertad obteniendo una información más completa. Las IMUs son ampliamente utilizados en dispositivos que requieren el conocimiento de su posición exacta, por ejemplo, los brazos róbóticos, misiles guiados y para el estudio del movimiento del cuerpo humano.

9 DOF-Sensor Stick [7] es una placa de sensores muy pequeña, con 9 grados de libertad. En la figura 2.21 se puede ver el dispositivo. Incluye acelerómetro ADXL345, el magnetómetro HMC5883L y el giróscopo MEMS ITG-3200. El "Stick" tiene una interfaz I2C simple y un agujero de montaje para instalarla en tu proyecto. Además, el tablero es un mero 0.036" de espesor, lo que le permite ser montado fácilmente en casi cualquier aplicación.



Figura 2.21: T9 DOF-Sensor Stick [7].

2.2.2. Sistemas basados en cámaras

Según la clasificación que establecimos en el apartado anterior, los sistemas basados en cámaras forman parte de los dispositivos ópticos de captura del movimiento humano. Vamos a describir a continuación algunos de ellos:

2.2.2.1. VICON 621

Vicon 621 [8] es un sistema óptico de captura de tridimensional de movimiento que utiliza marcadores esféricos reflectantes y cámaras con antorchas de luz infrarroja que recogen la reflexión infrarroja de los marcadores. Para el estudio cinematográfico es necesario colocar los marcadores sobre el sujeto, cuyo movimiento es captado por 6 cámaras, y analizado por triangulación para obtener sus posiciones 3D reales fotograma a fotograma, generando así un conjunto de datos de movimiento. El sistema es capaz de realizar simultáneamente el análisis electromiográfico de la actividad muscular. El sistema se muestra en la figura 2.22.



Figura 2.22: VICON 621 [8].

VICON 621 está compuesto por:

- 6 cámaras de captura de la imagen de 50 Hz.
- 6 canales activos de vídeo.
- Electromiógrafo de 12 canales con conexión mediante fibra óptica.
- Software BODYBUILDER, para creación, visualización y análisis de modelos cinematográficos y cinéticos.
- Software POLYGON para generación de informes.
- Estación de trabajo basada en PC de última generación e impresora.
- Sistema de captura de vídeo sincronizado.

A continuación se puede ver la figura 2.23 que muestra los marcadores del sistema VICON 621 colocados sobre una persona.



Figura 2.23: VICON 621: Marcadores [8].

2.2.2.2. Oqus Camera series (Qualisys)

La plataforma Oqus [9] ofrece cámaras de captura de movimiento adecuadas para todas las aplicaciones posibles, tanto en interiores como al aire libre, como la que se puede ver en la figura 2.24. Las cámaras Oqus están diseñadas para capturar datos *mocap* precisos con una latencia muy baja y trabaja con marcadores pasivos y activos.

La característica principal de las cámaras Oqus es la capacidad de calcular posiciones de marcador con impresionante precisión y velocidad. Cientos de marcadores se pueden medir en miles de imágenes por segundo con diez, cincuenta o incluso más cámaras - descargadas en un portátil común. No es necesario el uso de estaciones de trabajo. No se necesita ningún centro externo. El sistema aporta movilidad y es muy fácil de configurar.



Figura 2.24: Oqus Camera series [9].

Características principales:

- Captura de movimiento de alta velocidad.

- Vídeo de alta velocidad.
- Resoluciones del sensor: 0.3, 1.3, 4 y 12 megapíxeles.
- Filtro activo para mediciones en exteriores.
- Lentes motorizadas.
- Soporte para marcadores activos.
- Arquitectura de tiempo real para una baja latencia.
- Comunicación inalámbrica.
- Número de cámaras/marcadores ilimitado.
- Silencioso, sin ventilador.

2.2.2.3. Kinect

El Kinect [30] es un dispositivo de la consola de juegos Xbox de Microsoft que permite interactuar con la consola sin necesidad de tener un contacto físico con un controlador de video-juegos tradicional. Este dispositivo permite interactuar con el videojuego mediante la captura del movimiento del cuerpo, fue desarrollado por Microsoft y sus librerías son de fácil acceso para implementar este dispositivo en otras aplicaciones. La figura 2.25 muestra la imagen del dispositivo Kinect.



Figura 2.25: Kinect Xbox 360 [10].

A continuación se enumeran las características principales del Kinect Xbox 360:

Sensores

- Lentes de color y sensación de profundidad.
- Micrófono multi-arreglo.
- Ajuste de sensor con su motor de inclinación.
- Totalmente compatible con las consolas existentes de Xbox 360.

Campo de visión

- Campo de visión horizontal: 57 grados.
- Campo de visión vertical: 43 grados.
- Rango de inclinación física: ± 27 grados.
- Rango de profundidad del sensor: 1,2 - 3,5 metros.

Streams (Flujo de datos)

- 320×240 a 16 bits de profundidad @ 30fps.
- 640×480 32-bit de color @30fps.
- Audio de 16-bit @ 16 kHz.

Sistema de Seguimiento

- Rastrea hasta 6 personas, incluyendo 2 jugadores activos.
- Rastrea 20 articulaciones por jugador activo.
- Capacidad para mapear jugadores activos en Live Avatars.

Sistema de audio

- Chat en vivo y voz dentro del juego (requiere Xbox Live Gold).
- Sistema de cancelación de eco que aumenta la entrada de voz.
- Reconocimiento de voz múltiple.

Capítulo 3

Hardware de monitorización

En el Capítulo se realiza una descripción general de los sistemas GaitWatch Manager y ECnsole. Además, se especifican las características técnicas de cada uno de los dispositivos. En el caso del GaitWatch, se le añade adicionalmente una explicación exhaustiva de como se identifican los ejes de los sensores iniciales que contiene.

3.1. GaitWatch

3.1.1. Descripción general

GaitWatch es una unidad de medición inercial (IMU), diseñado con fines de supervisión de la marcha. Fue desarrollado por el Prof.Dr.med.Kai Bötzel en el Departamento de Neurología de la Universidad Ludwig-Maximilians de Munich, en conjunto con el Dr.Alberto Olivares Vicente del Departamento del departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

GaitWatch está orientado para ser utilizado en aplicaciones en las que es necesario controlar el movimiento de los pacientes. El sistema se compone de una caja que contiene la unidad de procesamiento central y un conjunto de unidades que se conectan a la misma medición. Las unidades de medición se irán colocando en los pacientes en los muslos, espinillas, brazos y tronco.

3.1.2. Descripción del Hardware

Tal y como mencionamos antes, GaitWatch tiene una caja que contiene la unidad de procesamiento central, así como un conjunto de sensores magnéticos e iniciales incrustados. El

microcontrolador que actúa como la unidad central de proceso está a cargo de la recopilación de los datos de la unidad de medida externa y la escritura en la tarjeta de memoria junto con los datos de los sensores de la caja empotrada.

Hay dos tipos diferentes de unidades externas.

La primera (de tipo A IMU) está colocada en ambos muslos y piernas. La segunda unidad (tipo B IMU) se coloca en ambos brazos. La figura 3.1 muestra un diagrama de la caja (que contiene la CPU y el tipo C IMU) junto con las unidades externas.

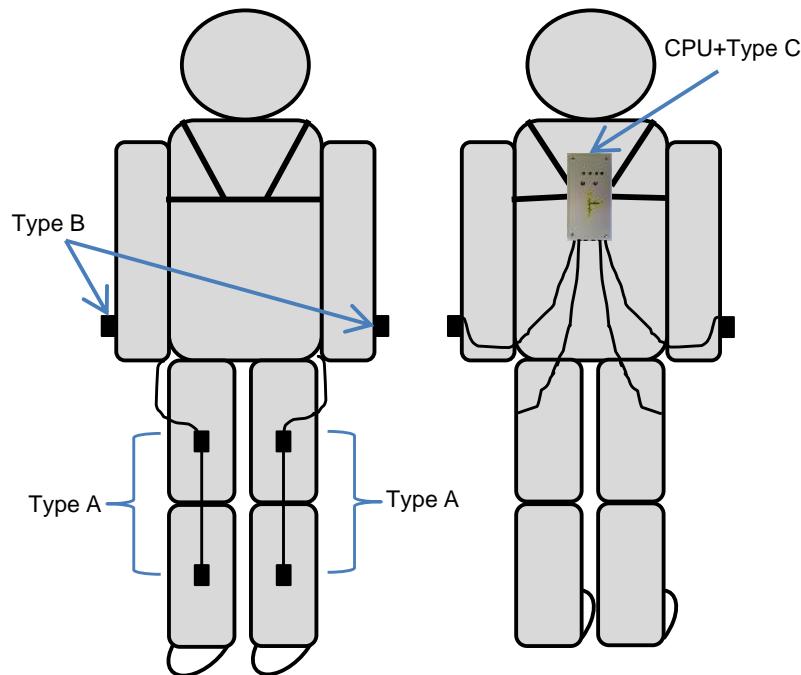


Figura 3.1: Diagrama general del GaitWatch siendo usado por un sujeto.

Los 3 tipos de diferentes IMUs que tienen los siguientes componentes:

- Tipo A(muslos y espinillas)
 - ◊ IMU 5 de Sparkfun. IMU 5 contiene un IDG500 giróscopo biaxial (que actualmente es solo usado para el eje Y) con un rango de medida ± 500 deg/s y un acelerómetro triaxial $\pm 3g$, ADXL 335.
- Tipo B(brazos)
 - ◊ IDG 500 gyroscope biaxial ± 500 deg/s

- Tipo C(tronco)
 - ◊ ADXL345 acelerómetro triaxial con un rango programable ($\pm 16g$ / $\pm 8g$ / $\pm 4g$ / $\pm 2g$)
 - ◊ IMU3000 giróscopo triaxial con un rango programable (± 250 / ± 500 / ± 1000 / ± 3000 (deg/s))
 - ◊ Micromag3 magnetómetro triaxial (± 11 Gauss)

Además, la caja del tronco contiene un soporte AL-XAVRB en cuyo interior existe un procesador AVR ATxmega que a su vez contiene el firmware embebido necesario para reunir los datos de todas las unidades de medida y almacenarlos en una tarjeta microSD.

3.1.3. Configuración inicial

3.1.3.1. Identificación de los ejes del sensor

En primer lugar se debe determinar la orientación del cuerpo de ejes que deseamos utilizar, así como la orientación de la rotación en torno a los ejes. Una de las configuraciones más populares es fijar el eje X apuntando hacia adelante, el eje Y a la derecha y el eje Z apuntando hacia abajo. Esta configuración sigue la regla de la mano derecha para la orientación de los ejes y la regla del sacacorchos para la rotación. Ambos se muestran en la figura 3.2. Estas dos estructuras de navegación estándar se emplean por lo general cuando se representa la orientación de un cuerpo en el espacio. Además, la dirección de sus ejes viene determinada de la siguiente forma: Norte-Este-Abajo (NED) y Este-Norte-Subida (ENU).

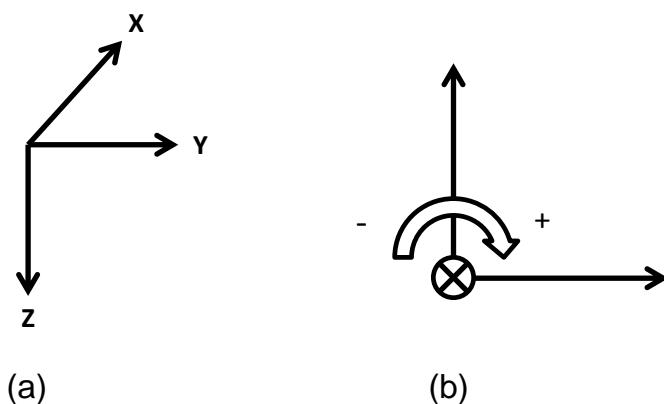


Figura 3.2: Definición del sistema de coordenadas deseado (a). Convención para la orientación de los ejes de rotación (b).

Dado que vamos a utilizar el dispositivo GaitWatch para monitorizar la marcha, necesitamos que su eje X apunte a la parte delantera del paciente y el eje Z a la planta de los pies del mismo. Vamos a empezar por la configuración de la orientación de los sensores que están dentro de la unidad de recogida de datos y que se colocará en la parte posterior del paciente. La figura 3.3 muestra la orientación deseada por la caja del GaitWatch.

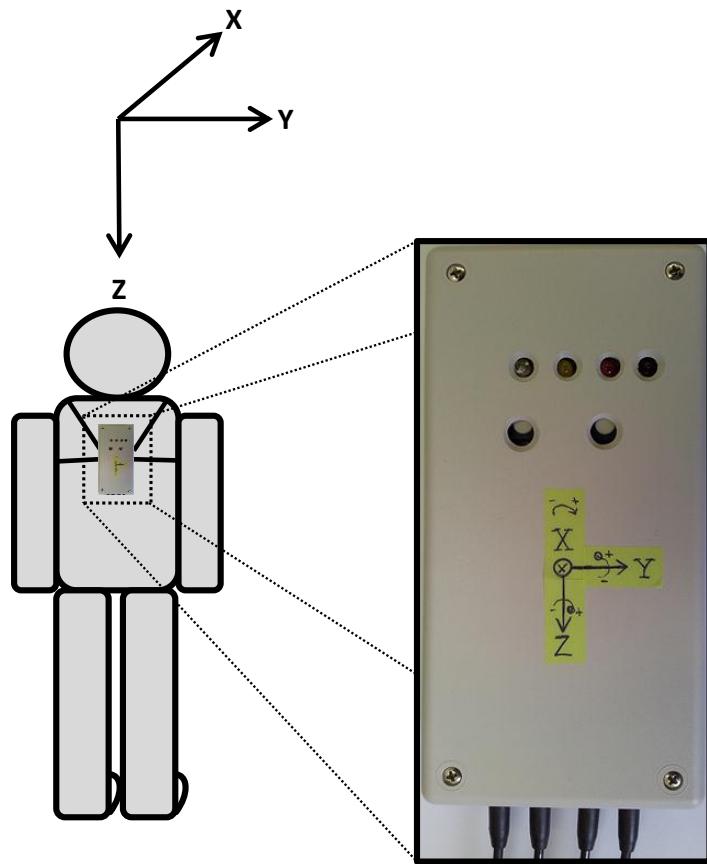


Figura 3.3: Orientación y ejes de coordenadas del GaitWatch.

Una vez que conocemos la orientación deseada, el siguiente paso es adaptar los ejes a los soportes de los sensores. Vamos a empezar por el ajuste del acelerómetro.

3.1.3.2. Identificación de los ejes del acelerómetro

El objetivo aquí es determinar la orientación de los ejes y adaptarla a la convección antes mencionada. En primer lugar, se establece el cuadro con el eje Z deseado apuntando hacia

abajo y hacia arriba, respectivamente, mientras se recogen los datos, como se muestra en la figura 3.4.

Ahora trazamos la aceleración recogida a lo largo de los tres ejes. Tenemos que identificar el canal que muestra un gran cambio en la aceleración obtenida.

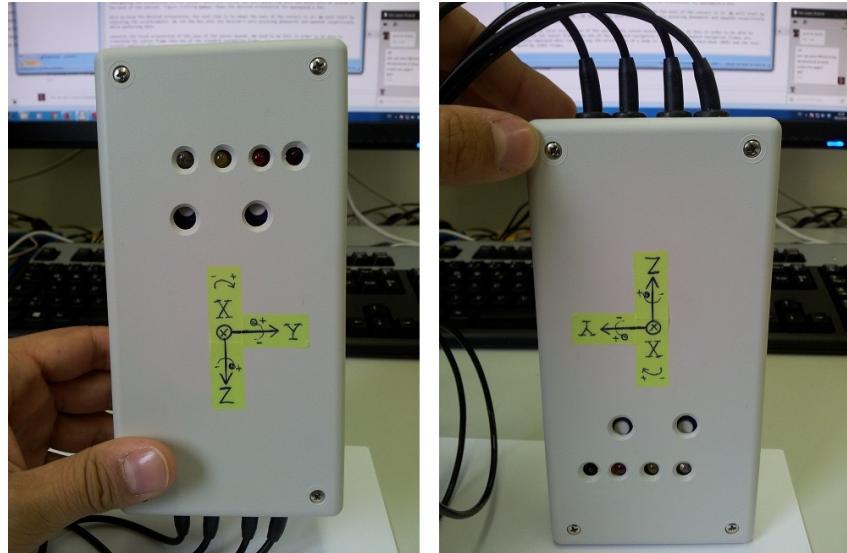


Figura 3.4: Identificación del eje Z del acelerómetro en la unidad de caja.

El canal debe ser etiquetado como Z independientemente de la etiqueta que pueda tener la placa del acelerómetro. Es decir, la placa del sensor puede indicar que el canal seleccionado es 'X' en el marco del sensor, pero vamos a etiquetarlo como 'Z' en nuestra configuración deseada. Si el eje Z de los sensores tiene la misma orientación que la de la configuración deseada, entonces, cuando apuntemos el eje Z de la caja hacia abajo, el acelerómetro medirá 1g porque su eje es paralelo al vector gravedad de la Tierra. Si se tiene la orientación opuesta, a continuación, el acelerómetro medirá -1g. Puesto que no hemos calibrado el acelerómetro hasta ahora y, por lo tanto, los datos están en unidades primitivas, la manera de identificar la orientación correcta es saber que la salida medida al establecer el eje del acelerómetro paralelo a la gravedad es más grande que su salida cuando se establece antiparalela al vector de la gravedad. La figura 3.5 muestra la aceleración triaxial recogida y el eje que debe ser etiquetado como "Z".

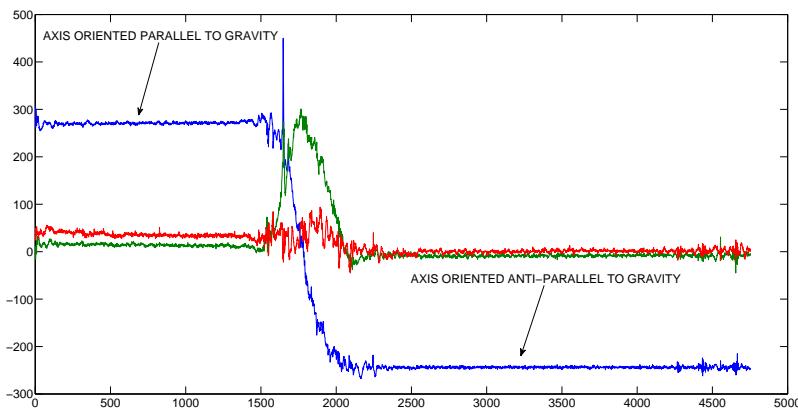


Figura 3.5: Identificación de la señal de aceleración recogida en el eje Z.

Para el caso del GaitWatch, la señal que muestra los valores de aceleración paralelos y anti-paralelos es el canal 20. La orientación del eje es correcta, ya que primero pone el eje Z deseado apuntando hacia abajo y el acelerómetro mostró el mayor valor de la aceleración en esa posición. Por lo tanto, ponemos el canal 20 como la aceleración en el eje Z. Los dos ejes restantes (X e Y) se identifica siguiendo el mismo procedimiento. La figura 3.6 muestra el cuadro que está orientado negativamente y positivamente a los largo de los ejes X e Y. Ambos ejes X e Y son orientados adecuadamente y no necesitan corrección de la orientación. Despues de inspeccionar la señales recogidas, se observa que la medida de X e Y están contenidas en los canales 22 y 21, respectivamente.



Figura 3.6: Identificación de los ejes X(arriba) e Y(abajo) de la aceleración de la caja.

Una vez identificado los ejes del acelerómetro en el interior de la caja que colocaremos

en la espalda del paciente, tenemos que aplicar el mismo procedimiento ya mencionado para identificar uno por uno los ejes de los acelerómetros biaxiales que debe ser colocados en los muslos y las espinillas. La figura 3.7 muestra el módulo de la espinilla derecha que está siendo colocado paralelo y anti-paralelo a la gravedad en los dos ejes X y Z.

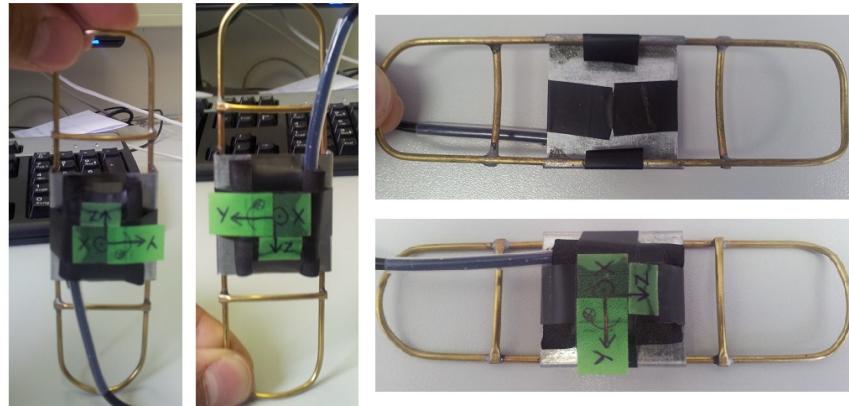


Figura 3.7: Identificación delos eje Z (izquierda) y X (derecha) del acelerómetro en la unidad de la espinilla derecha.

Tabla 3.1: Lista de acelerómetros del GaitWatch y sus canales asociados

Unidad	Eje	Compensación de la orientación	Canal
Espinilla derecha	Z	No (+1)	2
Espinilla derecha	X	No (+1)	3
Muslo derecho	Z	No (+1)	5
Muslo derecho	X	No (+1)	6
Espinilla izquierda	Z	No (+1)	8
Espinilla izquierda	X	No (+1)	9
Muslo izquierdo	Z	No (+1)	11
Muslo izquierdo	X	No (+1)	12
Tronco	Z	No (+1)	20
Tronco	Y	No (+1)	21
Tronco	X	No (+1)	22

Table 3.1: muestra los ejes de todos los acelerómetros del Gaitwatch, sus canales asociados y la compensación de la orientación (si se necesita).

3.1.3.3. Identificación de los ejes del giróscopo

Una vez que hemos identificado los ejes de los acelerómetros, ahora procedemos a identificar los ejes de los giróscopos y su orientación. Por convención, como se representa en la figura 3.2, el sentido de la rotación alrededor de un eje dado es positivo cuando el eje está apuntando hacia adelante (desde la perspectiva del usuario) y se gira hacia la derecha. Análogamente, la rotación es negativa cuando se gira a la izquierda. Sabiendo esto, para identificar el sentido de la rotación del giróscopo y adaptarlo (si es necesario) para la estructura de los ejes necesitamos proceder de la siguiente manera:

1. Coloque el eje que queremos analizar apuntando hacia delante.
2. Gire a la derecha y luego hacia la izquierda. Repita esto tres o cuatro veces.
3. Identifique el canal que contiene los datos y las trazas. Si la medida angular evaluada primero aumenta y luego disminuye, el sentido de rotación del eje del sensor es el correcto. Por otro lado, si la medida angular tiene el comportamiento opuesto tenemos que cambiar el sentido de rotación.

La figura 3.8 muestra las maniobras para identificar el sentido de rotación del eje Y del giróscopo situado en la mano izquierda. La figura 3.9 muestra la velocidad angular medida para tales maniobras. Observe cómo la primera rotación es positiva (la señal aumenta y luego disminuye), por lo que, para el caso de la unidad de la izquierda, el eje Y tiene el sentido correcto de rotación.

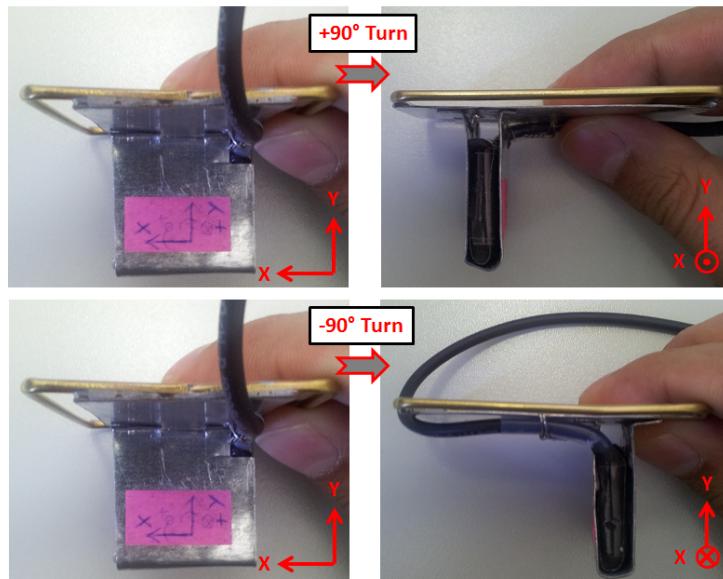


Figura 3.8: Identificación del sentido de la rotación del eje Y del giróscopo en la unidad de la mano izquierda.

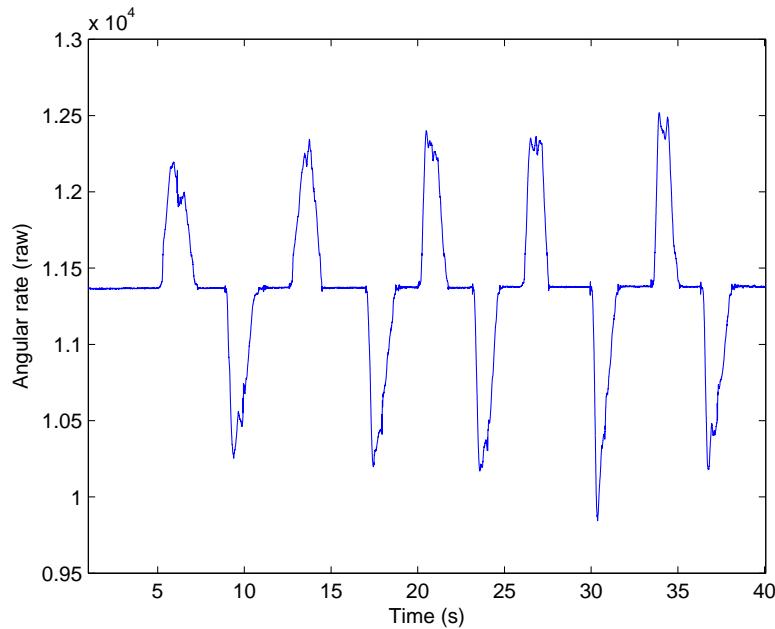


Figura 3.9: Señal de la tasa angular recogida durante las maniobras de identificación de la rotación.

Este procedimiento tiene que ser repetido para cada uno de los ejes de los giróscopos situados en todas las unidades.

Table 3.2 muestra los canales de todos los giróscopos pertenecientes al GaitWatch.

Tabla 3.2: Lista de giróscopos del GaitWatch y sus canales asociados.

Unidad	Ejes	Sentido de la compensación	Canal
Espinilla derecha	Y	Yes (-1)	1
Muslo derecho	Y	Yes (-1)	4
Espinilla izquierda	Y	Yes (-1)	7
Muslo izquierdo	Y	Yes (-1)	10
Brazo izquierdo	Y	No (+1)	13
Brazo izquierdo	X	Yes (-1)	14
Brazo derecho	Y	No (+1)	15
Brazo derecho	X	Yes (-1)	16
Tronco	Z	No (+1)	17
Tronco	Y	No (+1)	18
Tronco	X	Yes (-1)	19

3.1.3.4. Identificación de los ejes del magnetómetro

Los datos del magnetómetro triaxial embebidos en la unidad del tronco están todos contenidos en el mismo canal (canal 23). Cada eje dispone de un desplazamiento diferente por lo que es fácil separarlos. Los desplazamientos de los ejes son 32.794,20.000 y 0. Una vez que tenemos separados los datos, tenemos que identificar qué eje corresponde a cada desplazamiento, al hacerlo, ponemos el eje deseado hacia arriba o hacia abajo y giramos la caja más de 360 grados. Luego, la señal correspondiente al eje seleccionado debe ser la que muestra la menor variación (como se muestra en la figura 3.10). Esta operación debe repetirse dos veces más para identificar los dos ejes restantes.

Tabla 3.3: Desplazamiento del magnetómetro y ejes

Unidad	Ejes	Desplazamiento
Tronco	X	-20000
Tronco	Y	0
Tronco	Z	+32764

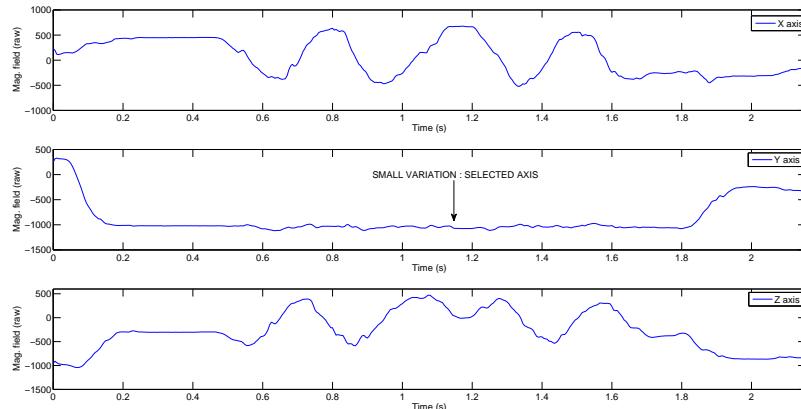


Figura 3.10: Señal de campo magnético registrada durante las maniobras de identificación del eje Y.

3.2. ECnsole

3.2.1. Descripción general

ECnsole es otra unidad de medición inercial (IMU). Desarrollada para aplicaciones de captura del movimiento humano dentro del departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

Este sistema también puede usarse en aplicaciones médicas para la monitoreo de los movimientos de los pacientes, aunque su uso puede ser bastante más amplio. El sistema está compuesto por una unidad de procesamiento central y las unidades externas de medición. Las unidades de medición se deben colocar en el pie izquierdo y en el pie derecho.

La figura 3.11 nos muestra un diagrama del dispositivo ECnsole colocado sobre una persona.

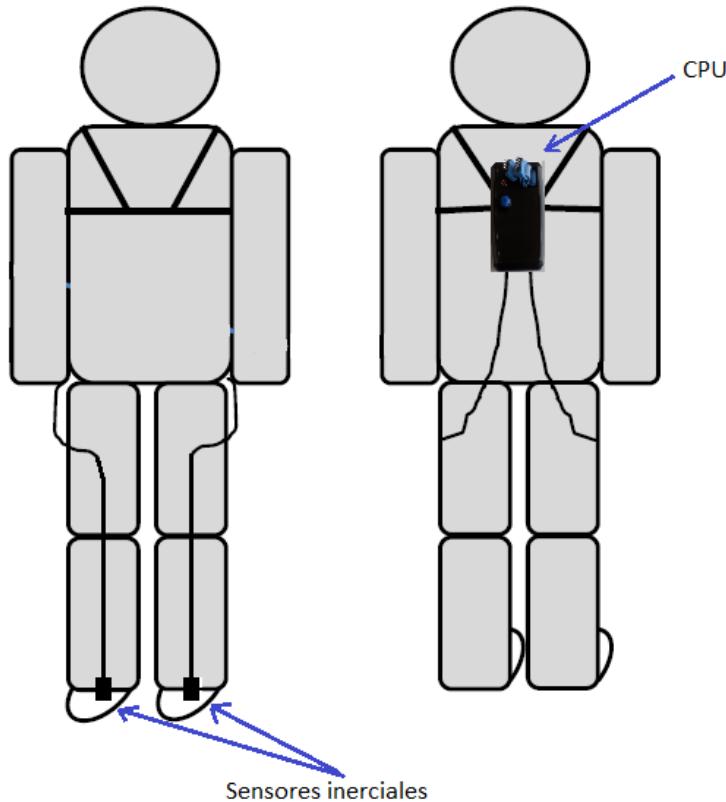


Figura 3.11: Diagrama general de ECnsole usado por un sujeto.

3.2.2. Descripción del Hardware

ECnsole posee una caja principal donde están contenidos el procesador y la tarjeta de memoria. El procesador se encarga de la recopilación y escritura de datos en la tarjeta de memoria donde se quedan almacenados. Los datos recopilados son extraídos de las unidades iniciales externas que están compuestas a su vez por los 3 tipos de sensores ya utilizados para el sistema GaitWatch (un magnetómetro, un giróscopo y un acelerómetro). Se sobreentiende que existen dos unidades iniciales, una para cada pie. La imagen de la figura 3.12 nos muestra el sistema ECnsole.



Figura 3.12: Sistema ECnsole.

A continuación se enumeran las características principales del sistema *ECnsole*:

Unidad de adquisición

- Sistema microcontrolado con PIC24FJ256GB106.
- Almacenamiento de datos en tarjeta μSD .
- Conexión USB.
- Muestreo hasta 100Hz.

Plantillas Instrumentadas

- Sustrato Plástico (PET) de 1mm de grosor.
- 4 Sensores de presión ubicados en:
 - ◊ Dedo gordo.
 - ◊ Primer metatarso.
 - ◊ Quinto metatarso.
 - ◊ Talón.
- Unidad Inercial compuesta por:
 - ◊ Acelerómetro y magnetómetro: LSM303DLHC.

- ◊ Giróscopo: L3GD20.
- Una unidad central donde se almacenan los datos en una memoria microSD.

En la figura 3.13 se pueden ver las plantillas instrumentadas:

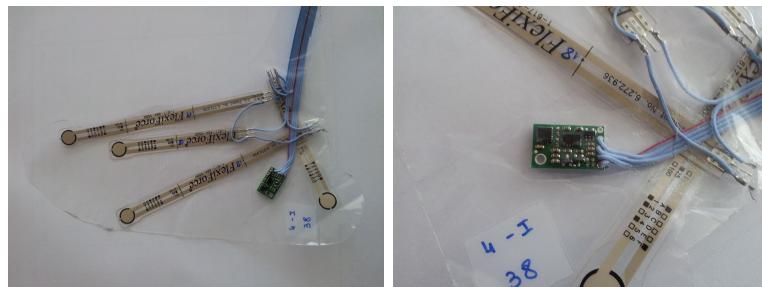


Figura 3.13: Plantillas instrumentadas.

Capítulo **4**

Fundamentos

En el Capítulo que vamos a ver a continuación, se realizará un extenso análisis a nivel teórico de los fundamentos de los sensores iniciales (acelerómetro, giróscopo y magnetómetro) utilizados en los sistemas GaitWatch y ECnsole. La segunda parte del capítulo se centrará primero en explicar el proceso de calibración, cuyo objetivo es transformar los datos originales en unidades con sentido físico. Después se describirán los algoritmos para el cálculo de la orientación. Primero el método de *Integración de la Velocidad Angular* para el giróscopo. Posteriormente, el método de *Descomposición de la Gravedad* donde el acelerómetro y el magnetómetro trabajan de manera conjunta. Después, el *Filtro de Kalman* que fusiona los datos del acelerómetro, giróscopo y manetómetro obteniendo unos resultados más precisos. Y por último, el *Filtro de Kalman Extendido* (una mejora del *Filtro de Kalman* gracias al uso de un elemento nuevo, los cuaterniones, que minimizan el cálculo computacional).

4.1. Fundamentos de sensores

A continuación vamos a realizar una descripción de los fundamentos básicos de los sensores que forman parte de los dispositivos cuyos datos son representados por la aplicación desarrollada.

4.1.1. Acelerómetros

Los acelerómetros [31] son los sensores utilizados para medir la aceleración tal y como su propio nombre indica. Es un instrumento que va unido a un objeto que permite medir la aceleración del mismo. La medida de la aceleración se consigue midiendo respecto a su masa inercial interna.

Este dispositivo nos permite saber en cada momento el valor de la segunda derivada del espacio y mide la fuerza de inercia que incide sobre una masa cuando es afectada por un cambio de velocidad.

Existen distintos tipos de tecnologías para medir la aceleración: piezo-eléctricos, piezo-resistivo, galgas extensométricas, láser, térmico, etc. Además se presentan también gran variedad de diseños en función de la aplicación a la que van a ser destinados o las condiciones de trabajo del acelerómetro.

Los dos parámetros principales que determinan la elección del medidor adecuado son los rangos de funcionamiento de temperatura y frecuencia. Existen otros parámetros a tener en cuenta, como el tamaño, la resistencia a golpes, el precio, etc.

Los acelerómetros han pasado de estar dedicados a un uso industrial (medir vibraciones y oscilaciones) y de investigación, a estar presentes en muchos aparatos cotidianos (Video-consolas, smartphones, GPS, automóviles, relojes).

Ahora vamos a describir algunos de los tipos de acelerómetros existentes. Esta información se ha extraído de [32], donde se hace un estudio en mayor profundidad.

4.1.1.1. Acelerómetro mecánico

Este tipo de acelerómetro es el más simple. Está formado por unas masas unidas a un dinamómetro que se encuentra en la misma dirección en la que queremos medir.

Haciendo uso de la segunda Ley de Newton, definida por la ecuación 4.1:

$$F = ma \quad (4.1)$$

donde F es la fuerza resultante que actúa sobre la masa, y a es la aceleración.

A través de la fórmula se puede deducir el valor de la aceleración ya que el dinamómetro mide el módulo de la fuerza.

$$a = \frac{F}{m} \quad (4.2)$$

La figura 4.1 nos muestra una acelerómetro de tipo mecánico.

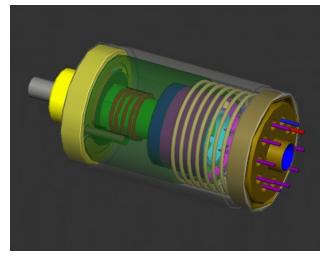


Figura 4.1: Acelerómetro mecánico [11].

4.1.1.2. Acelerómetro piezoeléctrico

El acelerómetro piezoeléctrico está basado en un retículo cristalino piezoeléctrico y el cual se comprime produciendo una carga eléctrica proporcional a la fuerza aplicada.

El material utilizado para los elementos piezoeléctricos suele ser el circonato de plomo. La estructura del acelerómetro es una caja metálica y en el interior se encuentran los elementos piezoeléctricos. Éstos están comprimidos por una masa sujetada a otro lado por un muelle. Al exponer el conjunto a una vibración, el disco piezoeléctrico se ve sometido a una fuerza variable que es proporcional a la aceleración de la masa. Gracias al efecto piezoeléctrico se genera una carga proporcional a la fuerza y por consiguiente a la aceleración. Haciendo uso de un voltímetro se puede medir el potencial.

Uno de los usos más comunes es el mantenimiento predictivo, detectando defectos en máquinas rotativas y alternativas.

La figura 4.2 muestra un acelerómetro de tipo piezoeléctrico.



Figura 4.2: Acelerómetro piezoeléctrico [12].

4.1.1.3. Otros tipos de acelerómetros

4.1.1.3.1. Acelerómetro de efecto Hall. Está fundamentado en la detección del campo magnético a través de un sensor de efecto Hall. Para ello se utiliza una masa sísmica

donde se coloca un imán y este tipo de sensor. En la figura 4.3 se puede ver un acelerómetro de efecto Hall.



Figura 4.3: Sensor de efecto Hall con salida lineal de proporción [13].

4.1.1.3.2. Acelerómetro de condensador. Mide el cambio de la capacidad de las placas de un circuito debido al movimiento de una masa sísmica interna que al desplazarse hace que cambie el valor de la corriente que pasa a través de las placas. La figura 4.4 representa un acelerómetro de este tipo.

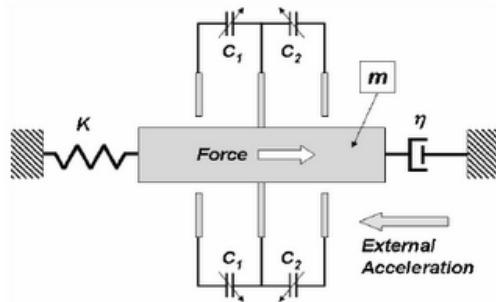


Figura 4.4: Acelerómetro de condensador [14].

4.1.1.3.3. Nuevas tecnologías. En la actualidad existen acelerómetros de 3 ejes. Son construidos en un chip de silicio sobre el que se incluye la parte electrónica encargada del procesamiento de señales.

El fundamento físico de estos sensores (tecnología MEMS) está basado en el traspaso térmico por convección natural.

Las siglas MEMS [33], son en conjunto un acrónimo para denotar a lo que actualmente se conoce como Sistemas Micro Electro Mecánicos, los cuales integran elementos mecánicos, sensores, actuadores y dispositivos electrónicos agrupados en su conjunto en un substrato u "oblea" común de silicio, mediante el uso de tecnología aplicada en micro fabricación.

Este tipo de acelerómetros son los utilizados en los sistemas GaitWatch y ECnsole.

4.1.2. Giróscopo

El giróscopo [15] es una rueda giratoria cuyo eje puede cambiar de dirección. Como vemos en la figura 4.5.

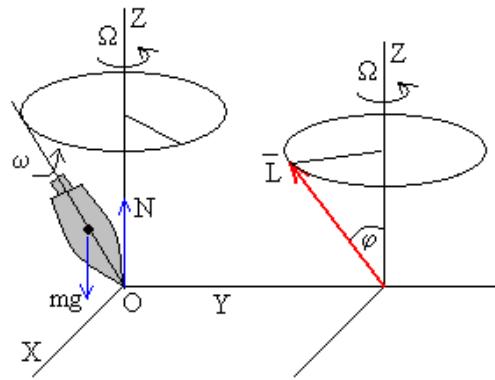


Figura 4.5: Fundamentos del giróscopo [15].

En general, el vector momento angular \mathbf{L} no tiene la dirección del eje de rotación, es decir, el vector momento angular no coincide con su proyección L_z a lo largo del eje de rotación. Cuando coinciden se dice que el eje de rotación es un eje principal de inercia. Para estos ejes existe una relación sencilla entre el momento angular \mathbf{L} y la velocidad angular $\boldsymbol{\omega}$, dos vectores que tienen la misma dirección, la del eje de rotación

$$\mathbf{L} = I\boldsymbol{\omega} \quad (4.3)$$

El término I se denomina momento de inercia y no es una cantidad característica como puede ser la masa o el volumen, sino que su valor depende de la posición del eje de rotación.

En ausencia de momento de fuerzas sobre el sólido M , el cuerpo seguirá rotando con respecto a dicho eje con velocidad angular constante.

Si el momento aplicado sobre el sólido en rotación no es nulo, el momento angular experimenta un cambio en su dirección tal como se muestra en la figura.

$$\mathbf{M} = \frac{d\mathbf{L}}{dt} \quad (4.4)$$

Las fuerzas aplicadas sobre el sólido en rotación que se muestran en la figura 4.6 son:

- El peso mg que actúa sobre el centro de masas, situado a una distancia b del punto de apoyo O.
- La reacción N en el punto de apoyo O.

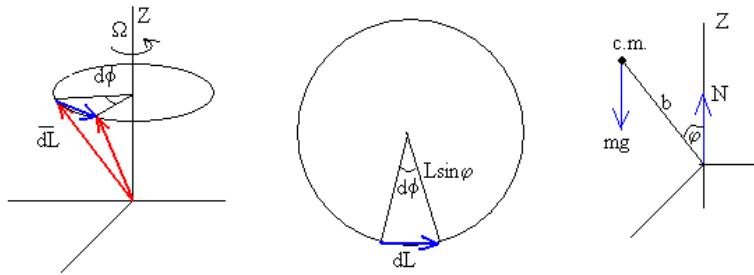


Figura 4.6: Fuerzas aplicadas sobre el sólido de rotación [15].

El momento \mathbf{M} de las fuerzas respecto del punto fijo O viene dado por la ecuación 4.5:

$$\mathbf{M} = mgb \cdot \sin\varphi \quad (4.5)$$

El cambio de momento angular $d\mathbf{L}$ tiene la dirección del momento \mathbf{M} de las fuerzas aplicadas respecto del punto de apoyo O, ya que el momento \mathbf{M} es perpendicular al momento angular \mathbf{L} , el cambio de momento angular $d\mathbf{L}$ es también perpendicular a \mathbf{L} , y el momento angular \mathbf{L} cambia de dirección pero no de módulo.

El extremo del vector momento angular \mathbf{L} describe una circunferencia de radio $L \sin\varphi$, y en un intervalo de tiempo dt se desplaza un ángulo $d\theta$; El cambio de momento angular es $dL = L \sin\varphi \cdot d\theta$, de modo que,

$$Iw \cdot \sin\varphi \frac{d\phi}{dt} = mgb \sin\varphi \quad (4.6)$$

Se denomina velocidad angular de precesión Ω a

$$\Omega = \frac{d\phi}{dt} = \frac{mgb}{Iw} \quad (4.7)$$

Cuando el punto de apoyo O coincide con el centro de masas $b=0$, el momento \mathbf{M} de las fuerzas es cero y la velocidad angular de precesión Ω es nula. El eje del giróscopo se mantiene fijo en el espacio.

Esta descripción es aproximada, siempre que la velocidad angular ω , sea grande en comparación con la velocidad angular de precesión Ω .

Un análisis más detallado del problema indica que el sólido tiene tres movimientos:

- De rotación alrededor de su eje principal de inercia con velocidad angular ω .
- De precesión alrededor del eje vertical Z, con velocidad angular Ω .
- De nutación o de oscilación del eje vertical entre dos círculos.

No solamente el movimiento de rotación contribuye al momento angular L, como hemos supuesto, sino también y en menor medida, los movimientos de precesión y de nutación, esto es lo que hace difícil el análisis detallado de este sistema mecánico.

Los fenómenos giroscópicos tienen muchas aplicaciones: la tendencia de un giróscopo a mantener el eje de rotación fijo en el espacio en ausencia de momento es utilizado en la estabilización de los barcos y en los pilotos automáticos de los aviones. En la figura 4.7 mostrámos la imagen de un giróscopo.



Figura 4.7: Giróscopo [16].

Otro ejemplo interesante es la precesión de los equinoccios. El plano del ecuador hace un ángulo de $23^{\circ} 37'$ con el plano de la órbita terrestre o eclíptica. La intersección de los dos planos es la línea de los equinoccios. La Tierra es un giróscopo gigante cuyo eje de rotación precesa alrededor del eje perpendicular al plano de la eclíptica con un periodo de 27725 años. La precesión de los equinoccios se debe al momento de las fuerzas ejercido por el Sol y la Luna sobre la Tierra.

4.1.2.1. Giróscopos MEMS

Los giróscopos MEMS [17] (sistemas microelectromecánicos) son los sensores utilizados por GaitWatch y ECnsole. Este tipo de sensores sirve para medir la velocidad angular y se

caracterizan por ser pequeños y de bajo coste. Las unidades de velocidad angular se miden en grados por segundo ($^{\circ} / s$) o revoluciones por segundo (RPS). La velocidad angular es simplemente una medida de la velocidad de rotación.



Figura 4.8: Giróscopo MEMS [17].

¿Cómo funciona el giróscopo MEMS para detectar la velocidad angular?

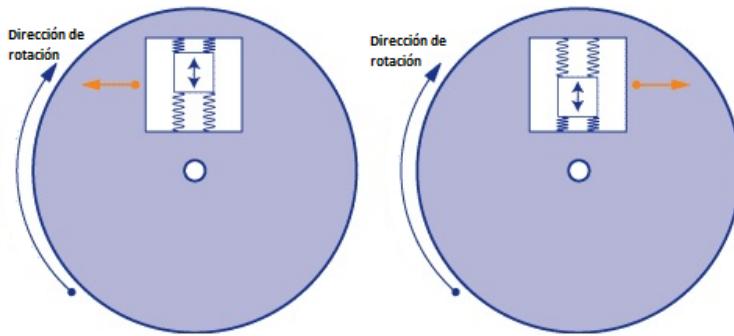


Figura 4.9: Funcionamiento interno de un sensor giroscópico MEMS [17].

El sensor MEMS dentro de un giroscopio es muy pequeño (entre 1 a 100 micrómetros , el tamaño de un cabello humano). Cuando se hace girar el giroscopio, una pequeña masa de resonancia se desplaza con los cambios de velocidad angular. Este movimiento se convierte en señales eléctricas de muy bajas corrientes que se pueden amplificar para ser leídas por un microcontrolador.

4.1.3. Magnetómetro

Un magnetómetro [34] es un sofisticado sensor que proporciona a los observadores científicos una comprensión más clara de cómo funciona el magnetismo.

La Tierra genera un campo magnético que crea disturbios magnéticos medibles en la atmósfera. Un magnetómetro es un instrumento científico que mide este fenómeno en términos de densidad de flujo magnético. La unidad científica para la lectura de la densidad del flujo magnético es el Tesla, el Gauss o As/m². Las sustancias y materiales que perturban este flujo se denominan magnéticos. Cuando hay materiales magnéticos presentes, un magnetómetro detecta la cantidad de distorsión que estos materiales causan en el campo de la Tierra. Un magnetómetro no sólo nos dice cómo afectan el flujo magnético ciertos materiales magnéticos en particular, sino que también puede medir la fuerza de los campos magnéticos. Esta información puede utilizarse para discernir la dirección, la rotación y el ángulo de los campos magnéticos, así como la ubicación de objetos específicos dentro de ellos.

4.1.3.1. Magnetómetro de precesión de protones

Un magnetómetro barato y portátil es el magnetómetro de precesión de protones. Este instrumento se utiliza principalmente en muestras cercanas a la superficie en estudios de ingeniería y medioambientales. El magnetómetro de precesión de protones requiere el uso de un líquido rico en átomos de hidrógeno para producir la señal de la precesión. Para ello, el queroseno es una de las mejores opciones. Las corrientes directas y los campos magnéticos polarizan los átomos, y el instrumento lee su frecuencia de precesión. Existen limitaciones para este tipo de magnetómetro, como su bajo nivel de sensibilidad y su alto consumo de potencia. Sin embargo, es ideal para la exploración y el mapeo de tuberías subterráneas. La figura 4.10 muestra un magnetómetro de precesión de protones.



Figura 4.10: Magnetómetro de precesión de protones [18].

4.1.3.2. Magnetómetro cuántico

Magnetómetros cuánticos son ampliamente utilizados en estudios ambientales, exploración geofísica, detección de armas y otras aplicaciones científicas. Estos instrumentos miden la magnitud específica de los campos magnéticos. Las partículas subatómicas son polarizadas por los magnetómetros cuánticos, lo que hace que se procesen alrededor de los campos magnéticos de la Tierra. Esta polarización produce un patrón reconocible que puede ser

cuantificado y medido como un momento magnético. Estos momentos magnéticos, referenciados con el campo magnético de la Tierra proporcionan información sobre la densidad del flujo magnético. La figura 4.11 muestra un magnetómetro cuántico.

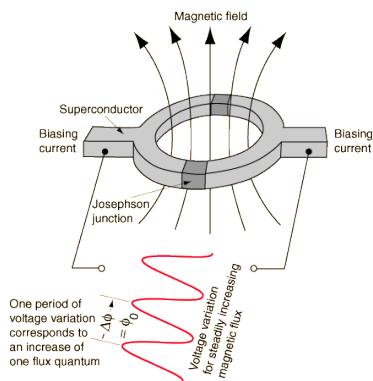


Figura 4.11: Magnetómetro cuántico [19].

4.1.3.3. Magnetómetros vectoriales

Además de tener una magnitud, el flujo magnético también tiene una dirección o vector. Los magnetómetros vectoriales miden las propiedades del campo magnético que viajan en una determinada dirección. Estos instrumentos ofrecen una lectura más precisa de la densidad del flujo magnético al eliminar la sensibilidad cruzada y funcionar con un nivel de ruido muy bajo. Los magnetómetros vectoriales se han utilizado en naves espaciales desde 1988. Mediante los magnetómetros vectoriales, los satélites pueden incluso medir los campos magnéticos de otros planetas y lunas.

Los magnetómetros pueden ser instrumentos útiles en muchas aplicaciones profesionales. Al ser ligeros y portátiles, los magnetómetros se transportan fácilmente a casi cualquier sitio. En arqueología, se pueden utilizar magnetómetros para localizar tumbas enterradas que contengan artefactos metálicos. En aplicaciones militares, los magnetómetros se utilizan para localizar los tanques, minas y los depósitos de combustible. La industria minera y otros numerosos campos científicos también miden anomalías magnéticas con magnetómetros. La figura 4.12 muestra un magnetómetro vectorial.

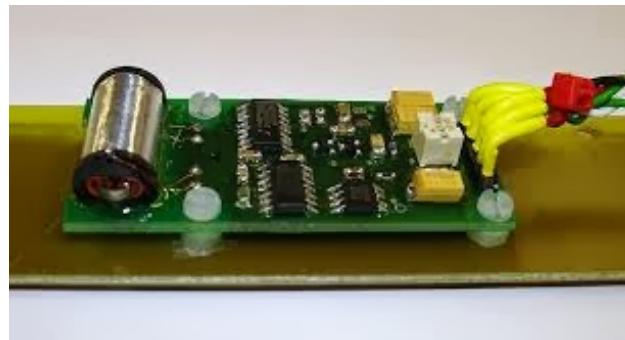


Figura 4.12: Magnetómetro vectorial [20].

4.1.3.4. Magnetómetros MEMS

Los magnetómetros MEMS son los utilizados en los dispositivos ECnsole y GaitWatch, por esta razón, se analizará su funcionamiento (en la mayoría de estos sensores predomina el uso de la fuerza de Lorentz) posteriormente. La figura 4.13 muestra un ejemplo de magnetómetro MEMS.

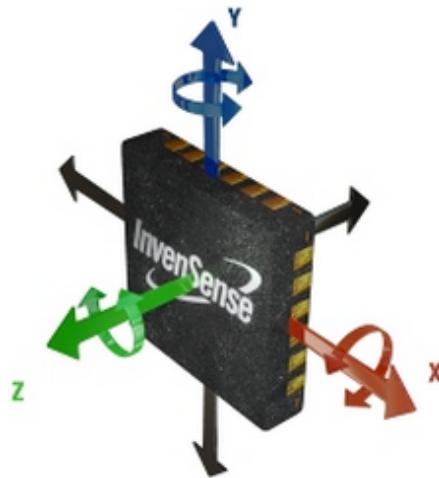


Figura 4.13: Magnetómetro MEMS [21].

¿Cómo funcionan los sensores MEMS basados en la fuerza de Lorentz?

Este tipo de sensores magnéticos [35] se basan en el movimiento mecánico de la estructura MEMS, es provocado por la fuerza de Lorentz que actúa sobre un conductor portador de la corriente situado en el interior del campo magnético. El movimiento mecánico de la microestructura se detecta electrónicamente o de forma óptica. Los métodos de transducción piezorresistivo y electroestático se pueden utilizar en la detección electrónica; y por otro lado, la medición del desplazamiento con fuente láser o fuente LED se pueden utilizar

para la detección óptica. Cabe resaltar que la estructura mecánica es inducida a alcanzar su frecuencia de resonancia a menudo con el fin de obtener la señal de salida máxima.

4.2. Fundamentos de la monitorización

Una vez presentados los fundamentos de los sensores que contienen GaitWatch y ECnsole, vamos a proceder a explicar el procedimiento de calibración de los datos medidos por los mismos.

4.2.1. Calibración

El objetivo principal del proceso de calibración consiste en transformar los datos en bruto en unidades con sentido físico.

En este apartado vamos a ver como se realiza el proceso de calibración de los sensores que forman parte de dispositivo GaitWatch. No es necesario especificar el modo de calibración para el sistema ECnsole ya que los sensores que componen el dispositivo reúnen características similares y el procedimiento de calibración es el mismo. Sin embargo difieren en el número de sensores, como se puede ver en la tabla 4.1:

Tabla 4.1: Lista de sensores que componen los sistemas GaitWatch y ECnsole

	GaitWatch	ECnsole
Acelerómetro triaxial	Sí	Sí
Acelerómetro uniaxial	Sí	No
Giróscopo triaxial	Sí	Sí
Giróscopo uniaxial	Sí	No
Magnetómetro triaxial	Sí	Sí

Las pautas para llevar a cabo el proceso de calibración vienen recogidas en [25] donde se realiza una descripción más exhaustiva del procedimiento.

4.2.1.1. Calibración del acelerómetro

La calibración de los acelerómetros está dividida en dos partes de acuerdo con el número de ejes del acelerómetro. En el caso del acelerómetro triaxial (incluido en el módulo de GaitWatch que se coloca en el tronco del paciente y en las plantillas para el sistema ECnsole) utilizaremos el algoritmo de ajuste elipsoidal de Camps et al. [36] los cuales presentaron un método teórico y experimental para calcular ganancias, bias y factores de no ortogonalidad de los acelerómetros y los magnetómetros. Las maniobras de calibración implican una serie

de rotaciones arbitrarias de la MIMU, por lo que el conjunto de maniobras para recopilar los datos es muy simple.

- **Modelado del sensor:** El modelo de salida del sensor viene dado por:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} m_x(t) \\ m_y(t) \\ m_z(t) \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{bmatrix} \quad (4.8)$$

donde $\mathbf{b}=(b_x, b_y, b_z)^T$ representa el desplazamiento, s_x, s_y, s_z son las ganancias de los sensores, $m_x(t), m_y(t), m_z(t)$ son los componentes de los actuales campos magnéticos y gravitatorios y $\epsilon_x, \epsilon_y, \epsilon_z$ son los componentes del ruido para cada eje.

Si tomamos en consideración los efectos de la no ortogonalidad, entonces la salida del sensor viene dada por:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{xy} & s_{yy} & s_{yz} \\ s_{xz} & s_{yz} & s_{zz} \end{bmatrix} \quad (4.9)$$

donde s_{ij} para $i \neq j$ representa los errores de ortogonalidad entre los ejes i y j del sensor.

- **Procedimiento de calibración:** Usando el hecho de que la norma del vector de entrada $\mathbf{m} = [m_x(t)m_y(t)m_z(t)]^T$ es constante, la siguiente relación es derivada:

$$\|\mathbf{m}\|^2 = m_x(t)^2 + m_y(t)^2 + m_z^2 \quad (4.10)$$

$$\|\mathbf{h}\|^2 = \left(\frac{v_x(t) - b_x}{s_x}\right)^2 + \left(\frac{v_y(t) - b_y}{s_y}\right)^2 + \left(\frac{v_z(t) - b_z}{s_z}\right)^2 \quad (4.11)$$

donde $\mathbf{m}=(m_x(t), m_y(t), m_z(t))^T$ es el vector de campo gravitatorio $m_x(t), m_y(t), m_z(t)$ son los componentes de los actuales campos gravitatorios, $\mathbf{h}=(h_x(t), h_y(t), h_z(t))^T$ es el vector del campo gravitatorio sin tener en cuenta los componentes del ruido para cada eje, $v_x(t), v_y(t), v_z(t)$ son las componentes de la salida del sensor, $\mathbf{b}=(b_x, b_y, b_z)^T$ representa el desplazamiento, y s_x, s_y, s_z son las ganancias de los sensores.

La ecuación 4.11 es la ecuación paramétrica de una elipsoide con centro en \mathbf{b} y semi-ejes s_x, s_y y s_z . Usando el sistema de ecuaciones formado por varias medidas de tiempo t , estimamos los parámetros a través de la minimización no lineal cuadrática del error de la función

$$e_p(t) = \|\mathbf{m}\|^2 - (\mathbf{v}(t) - \mathbf{b})^T (S^{-1})^2 (\mathbf{v}(t) - \mathbf{b}) \quad (4.12)$$

donde

$$S^{-1} = \begin{bmatrix} 1/s_{xx} & 1/s_{xy} & 1/s_{xz} \\ 1/s_{xy} & 1/s_{yy} & 1/s_{yz} \\ 1/s_{xz} & 1/s_{yz} & 1/s_{zz} \end{bmatrix} \quad (4.13)$$

El coste de la función $e_p(t)$ es cuadrático, y es minimizado iterativamente por el algoritmo de Levenberg-Marquardt [37, 38].

En resumen, si el acelerómetro se sitúa en múltiples posiciones casi-estáticas y aleatorias, los datos recogidos deben describir idealmente una esfera de radio igual a la magnitud del vector gravedad.

Por lo tanto, el primer paso de calibración es recoger los datos de la aceleración que serán usados para encontrar los parámetros de calibración óptimos. Como se dijo antes, necesitamos situar el módulo que contiene el acelerómetro en múltiples posiciones casi-estáticas y aleatorias intentando cubrir todas las orientaciones. Ya que la transición de una posición casi-estática es la causa de que el acelerómetro mida aceleraciones lineales que interrumpen la aceleración de la gravedad, no podemos tomar cada punto medido. Para evitar seleccionar valores de aceleración lineal tenemos que aplicar un algoritmo que se capaz de detectar los instantes casi-estáticos. La figura 4.14 muestra la aceleración triaxial medida en cada posición casi-estática, además de la aceleración lineal no deseada. Esta figura también muestra la salida del algoritmo de detección de los instantes casi-estáticos.

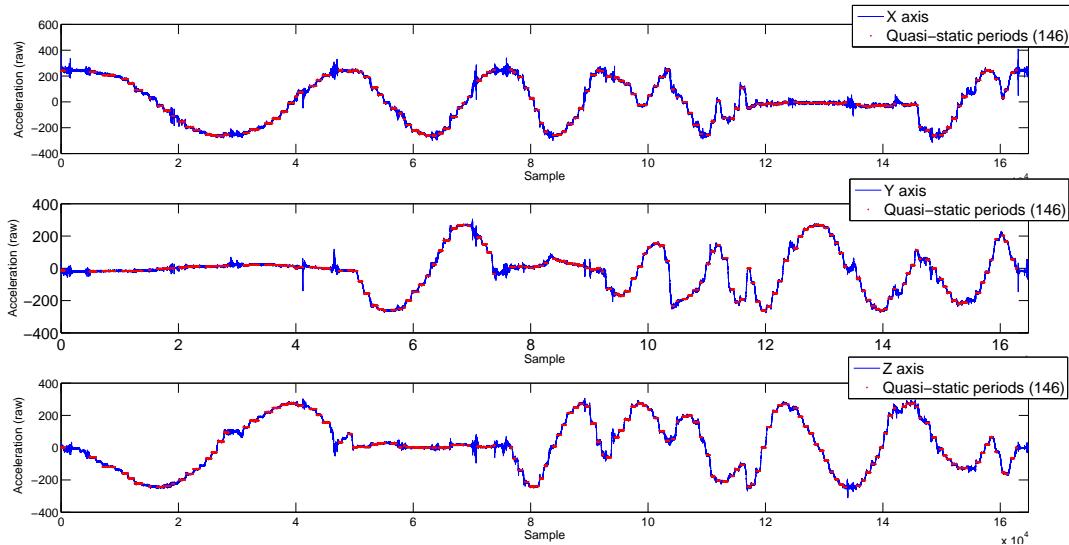


Figura 4.14: Aceleración recogida en las posiciones casi-estáticas y salida del algoritmo de detección

Si trazamos las aceleraciones casi-estáticas detectadas en 3D, debemos cubrir el lugar de una esfera como se muestra en la figura 4.15

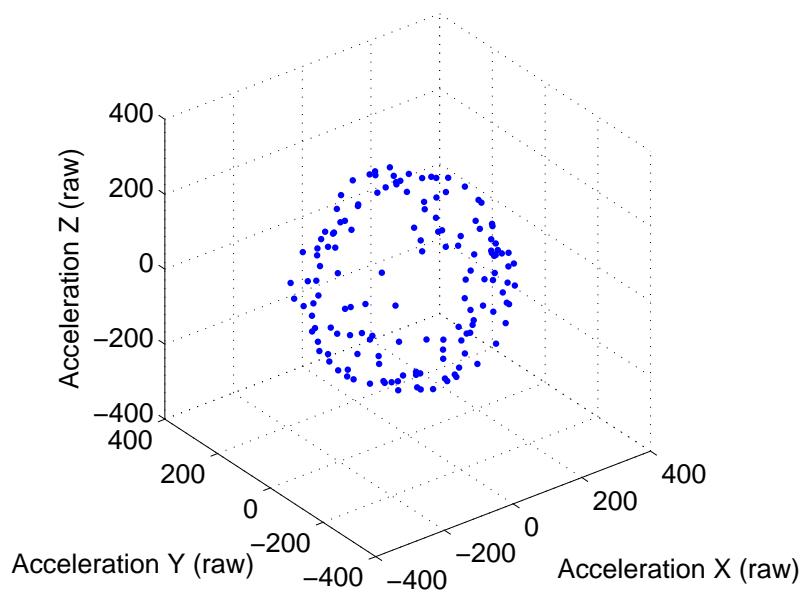


Figura 4.15: Representación 3D de la aceleración recogida durante las posiciones casi-estáticas.

Entonces alimentamos el algoritmo de minimización con esos datos y encontramos los parámetros de calibración óptimos. La ecuación 4.14 muestra la estimación de los parámetros de calibración para el acelerómetro triaxial que se encuentra en el módulo de GaitWatch que se coloca en el tronco del paciente.

$$\begin{bmatrix} 256,68 & -4,01 \cdot 10^{-5} & -2,41 \cdot 10^{-5} \\ -4,01 \cdot 10^{-5} & 262,74 & -3,98 \cdot 10^{-7} \\ -2,41 \cdot 10^{-5} & -3,98 \cdot 10^{-7} & 263,04 \end{bmatrix} \quad (4.14)$$

$$[\mathbf{b} = -23,69 \quad -6,95 \quad 22,85]$$

La figura 4.16 muestra la aceleración casi-estática calibrada. Observe como los puntos definen ahora una esfera centrada en el origen con un radio de 1(g). Posteriormente veremos como se calcula.

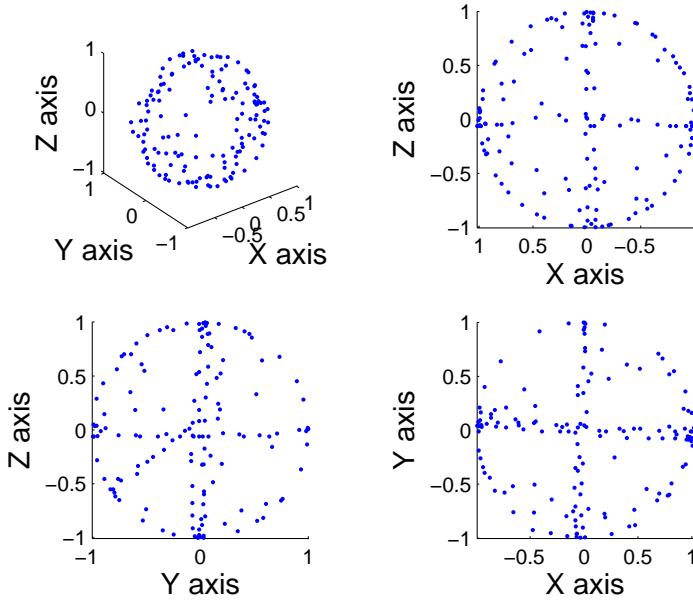


Figura 4.16: Representación 3D de las posiciones casi-estáticas calibradas

Los parámetros calculados anteriormente son incluidos a la siguiente ecuación que son utilizados para calibrar la aceleración en bruto y transformarla a unidades físicas (la figura 4.16 nos mostraba los resultados), además de paliar algunos de los errores no deseados presentes a la salida del sensor:

$$\mathbf{a}_{cal} = S^{-1}(\mathbf{a}_{raw} - \mathbf{b}) \quad (4.15)$$

donde \mathbf{a}_{cal} es un vector de tres dimensiones que contiene la aceleración calibrada, S es la matriz de ortogonalidad calculada, \mathbf{a}_{raw} es el vector de tres dimensiones que contiene la aceleración original y \mathbf{b} es el vector bias triaxial calculado.

Para el acelerómetro biaxial podemos adaptar los ya mencionados métodos de dos dimensiones. Sin embargo, las maniobras de calibración serían mucho más complicadas porque necesitaríamos bloquear la unidad incluyendo el acelerómetro en el plano XZ y después describir una rotación lenta y completa para recoger las posiciones casi-estáticas.

Optamos por llevar a cabo un procedimiento de calibración de un solo eje que sólo requiere recoger datos de dos posiciones en cada eje. Este procedimiento es muy simple; primero ponemos el eje deseado paralelo al vector gravedad y lo dejamos en esa posición un par de segundos. Seguidamente, le damos la vuelta y lo colocamos de forma anti-paralela al vector de la gravedad donde lo volvemos a dejar estático otros dos segundos. Procediendo de esta manera, conocemos los valores originales que corresponden a $+1g$ y $-1g$ respectivamente. Ya que los acelerómetros incluidos en las unidades del GaitWatch son lineales, podemos entonces

encontrar la ecuación de calibración en una forma muy simple. La ecuación de calibración está definida como sigue,

$$a_{cal} = k \cdot a_{raw} + b \quad (4.16)$$

donde a_{cal} es la aceleración calibrada, a_{raw} es la aceleración original, k es el factor de escala y b es el offset. Si sustituimos los dos valores recogidos tendríamos el siguiente sistema de ecuaciones de donde es sencillo encontrar los valores de k y b ,

$$\begin{aligned} 1 &= k \cdot a_{raw,+} + b \\ -1 &= k \cdot a_{raw,-} + b \end{aligned} \quad (4.17)$$

donde $a_{raw,+}$ y $a_{raw,-}$ son las aceleraciones originales recogidas en las posiciones paralela y anti-paralela respectivamente.

Las unidades de GaitWatch situadas en las espinillas y los muslos contienen un acelerómetro biaxial(X,Z) por lo tanto ponemos la unidad en las cuatro posiciones requeridas (dos en cada eje) y entonces aplicamos el detector de posiciones casi-estáticas para extraer los valores en esas posiciones.

La figura 4.17 muestra la aceleración original recogida en las cuatro posiciones y los valores casi-estáticos extraídos para el acelerómetro en la espinilla izquierda.

Usando este procedimiento los parámetros de calibración son encontrados en todas las unidades. La tabla 4.2 los muestra.

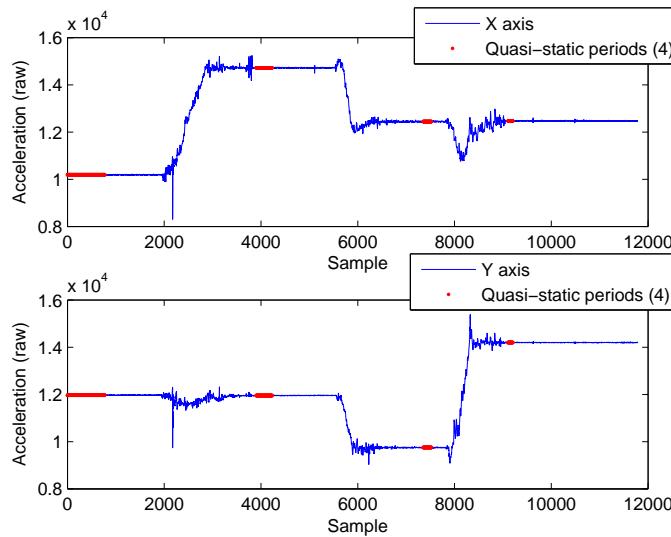


Figura 4.17: Aceleración original recogida cuando ambos ejes X y Z son colocados paralelo y anti-paralelo al vector gravedad

Tabla 4.2: Parámetros de calibración de los acelerómetros biaxiales

Unidad	Eje	Factor de escala	Sesgo
Espinilla izquierda	X	4.45e-04	-5.59
Espinilla izquierda	Z	4.49e-04	-5.36
Muslo izquierdo	X	4.58e-04	-5.76
Muslo izquierdo	Z	4.74e-04	-5.70
Espinilla derecha	X	4.38e-04	-5.50
Espinilla derecha	Z	4.77e-04	-5.72
Muslo derecho	X	4.36e-04	-5.44
Muslo derecho	Z	4.43e-04	-5.34

4.2.1.2. Calibración del magnetómetro

La calibración de un magnetómetro triaxial está también basada en el algoritmo de Camps et al. usado para la calibración del acelerómetro. En este caso, solo necesitamos cambiar los datos recogidos en las maniobras y el valor de la magnitud del vector de referencia (que en este caso es el vector de campo magnético de la Tierra).

Las maniobras en este caso son más simples porque el magnetómetro no se ve afectado por la aceleración lineal. Por lo tanto, podemos mover libremente el magnetómetro

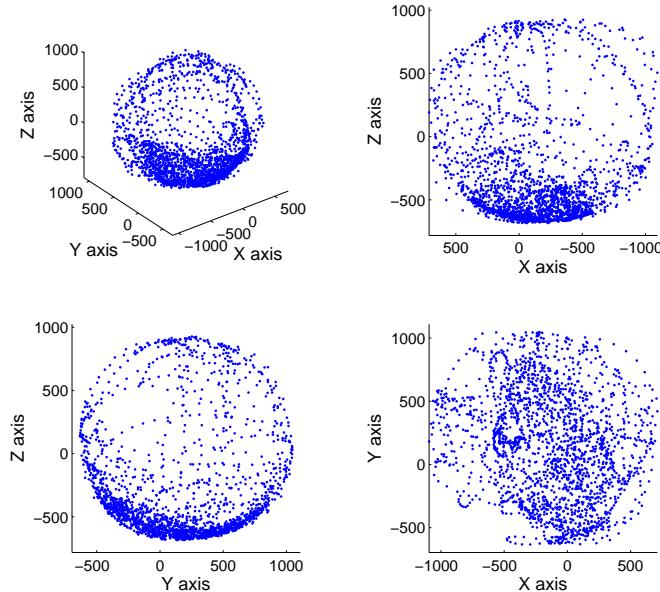


Figura 4.18: Campo magnético original recogido por el movimiento aleatorio de la unidad del tronco que contiene el magnetómetro

a cualquier velocidad intentando cubrir el mayor espacio posible. La figura 4.18 muestra la representación 3D del campo magnético original medido llevando a cabo esas maniobras.

Idealmente, esos datos deben de describir una esfera centrada en el origen con un radio dado por la magnitud del campo magnético local de la Tierra. Al comprobar el campo magnético en la calculadora del American National Geophysical Data Center, obtenemos el valor para Munich (lugar donde fueron realizadas las maniobras de calibración), que es de 0.482352 Gauss. Entonces proporcionamos al algoritmo [39] los datos mostrados en la figura 4.18 y los parámetros de calibración son calculados. Análogamente al acelerómetro, el campo magnético calibrado es encontrado aplicando la siguiente ecuación,

$$\mathbf{h}_{cal} = S^{-1}(\mathbf{h}_{raw} - \mathbf{b}) \quad (4.18)$$

donde otra vez \mathbf{h}_{cal} es un vector tridimensional que contiene el campo magnético calibrado, S es la matriz de ortogonalidad calculada, \mathbf{h}_{raw} es el vector de tres dimensiones que contiene el campo magnético original y \mathbf{b} es el vector bias triaxial calculado. Si aplicamos esta ecuación a los valores originales usados para calcular los parámetros de calibración, podemos ver como ahora definen una esfera centrada en 0 y con un radio de 0.482352 Gauss (figura 4.19)

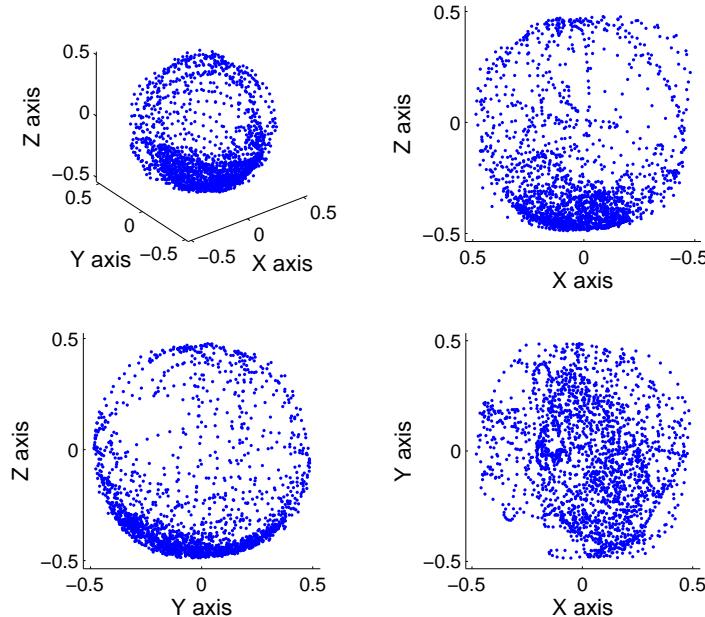


Figura 4.19: Campo magnético calibrado recogido moviendo aleatoriamente la unidad del tronco que contiene el magnetómetro.

La figura 4.20 muestra las señales originales y calibradas para los tres ejes del magnetómetro incluidos en la unidad del tronco.

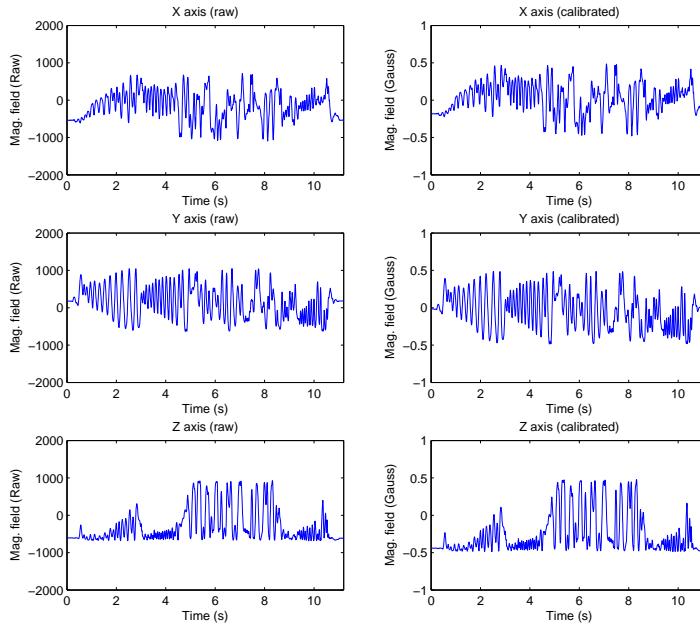


Figura 4.20: Campo magnético original (izquierda) vs. Campo magnético calibrado (derecha) recogidos con movimientos aleatorios de la unidad del tronco que contiene el magnetómetro

Los parámetros de calibración del magnetómetro en la unidad del tronco se enumeran a continuación

$$S = \begin{bmatrix} 1,75 \cdot 10^3 & -3,46 \cdot 10^{-8} & -3,04 \cdot 10^{-6} \\ -3,46 \cdot 10^{-8} & 1,97 \cdot 10^3 & 1,42 \cdot 10^{-5} \\ -3,04 \cdot 10^{-5} & 1,42 \cdot 10^{-6} & 1,83 \cdot 10^3 \end{bmatrix} \quad (4.19)$$

$$\mathbf{b} = [14,84 \quad 102,43 \quad -42,04]$$

4.2.1.3. Calibración del giróscopo

El giróscopo es el último tipo de sensor que queda por ser calibrado. La forma más precisa para calibrar un giróscopo es someterlo a diferentes rotaciones conocidas sus velocidades y luego asociarlos a los valores originales recogidos para obtener la ecuación de calibración. Si no tenemos mesa de rotación de velocidad variable, podemos usar un equipo más sencillo y sujetar el someter al giróscopo a rotaciones conocidas en lugar de velocidades angulares conocidas. Para calibrar cada uno de los ejes necesitamos construir un dispositivo parecido al que se muestra en la figura 4.21. Este dispositivo debe permitir la rotación de la unidad que contiene el giróscopo alrededor de 180° (otras configuraciones del dispositivo que permiten diferentes rotaciones conocidas también son válidas).

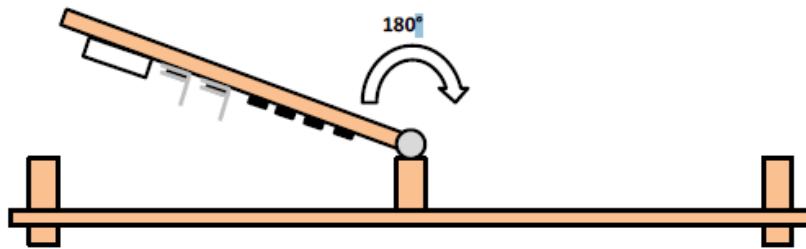


Figura 4.21: Dispositivo de calibración del giróscopo

Una vez que el dispositivo de calibración está listo, las maniobras de calibración no son complicadas. Cada una de las unidades debe estar adherida a la superficie del dispositivo y entonces necesitamos llevar a cabo la rotación completa de 180° una serie de veces para posteriormente promediar los datos calculados.

Para calcular los parámetros de calibración vamos a utilizar el hecho de que la integración de la velocidad angular conduce a un desplazamiento angular. Por lo tanto, si sometemos al giróscopo a una rotación conocida (180° en este caso), podemos aplicar las siguientes ecuaciones para encontrar el factor de escala,

$$\mu_{raw} = k \cdot \omega + b \quad (4.20)$$

$$\omega = \frac{\mu_{raw} - b}{k} \quad (4.21)$$

$$\mu_{raw,corr} = \mu_{raw} - b \quad (4.22)$$

$$\int \omega dt = \frac{\int \mu_{raw,corr} dt}{k} \quad (4.23)$$

$$\Omega = \int \omega dt \quad (4.24)$$

$$\theta = \int \mu_{raw,corr} dt \quad (4.25)$$

$$\Omega = \frac{\theta}{k} \quad (4.26)$$

$$k = \frac{\theta}{\Omega} \quad (4.27)$$

donde μ_{raw} es la velocidad angular original recogida, ω es la velocidad angular real (en unidades físicas), k es el factor de escala y b es el offset. En este caso $\Omega = 180$ y θ es la velocidad angular original integrada (corregida en offset) durante la rotación de 180° .

Podemos ver que antes de la integración de la velocidad angular necesitamos sustraer el offset.

El offset se encuentra fácilmente dejando las unidades en una posición estática antes de empezar la rotación. Para tener en cuenta los efectos del ruido, calculamos la moda de todas las muestras recogidas mientras los sensores estaban estáticos.

Desde el punto de vista práctico, la rutina de calibración de Matlab muestra por primera vez la señal de la velocidad angular original recogida y pide al usuario seleccionar los puntos inicial y final del período estático como muestra la figura 4.22.

Una vez encontrado el offset, necesitamos indicarle a la rutina cada uno de los puntos de inicio y final de las rotaciones, así sabremos los instantes inicial y final de la integración. Para lograrlo utilizaremos la herramienta *datacursor* de Matlab. Es importante seleccionar únicamente las rotaciones positivas (como se indica en la figura 4.23), es decir, cuando la velocidad angular incrementa y después decrementa. La rutina entonces integrará los segmentos seleccionados de la señal y promediará el factor de escala de cada uno de ellos.

La calibración de la velocidad angular es finalmente encontrada sustiyendo los parámetros de calibración calculados en la ecuación 4.21.

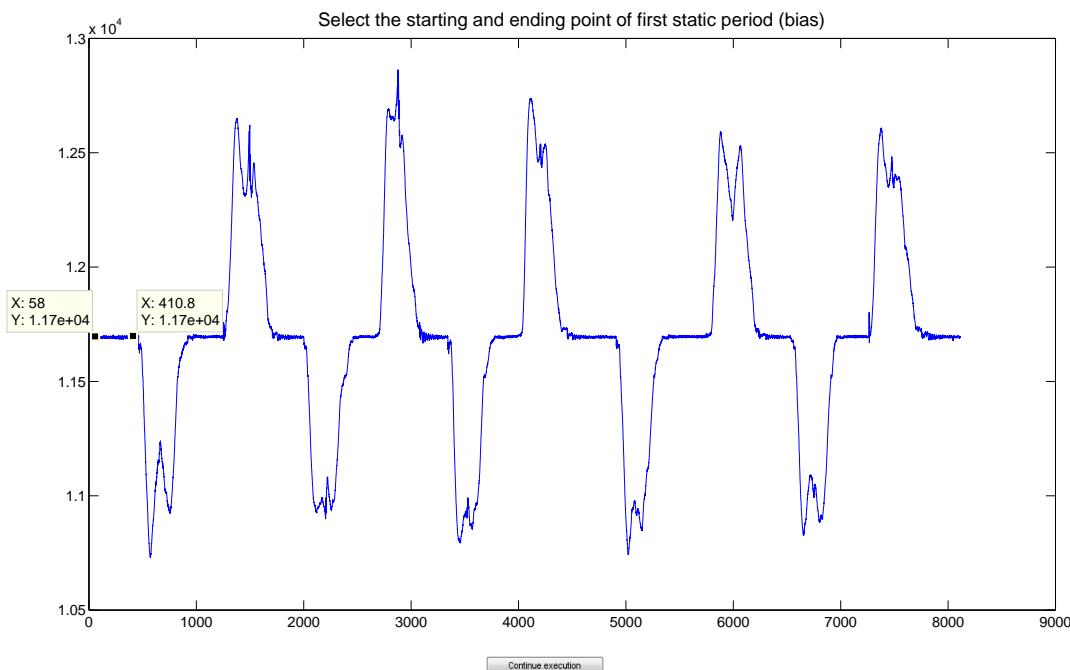


Figura 4.22: Rutina de calibración del giróscopo: selección del período para calcular el offset.

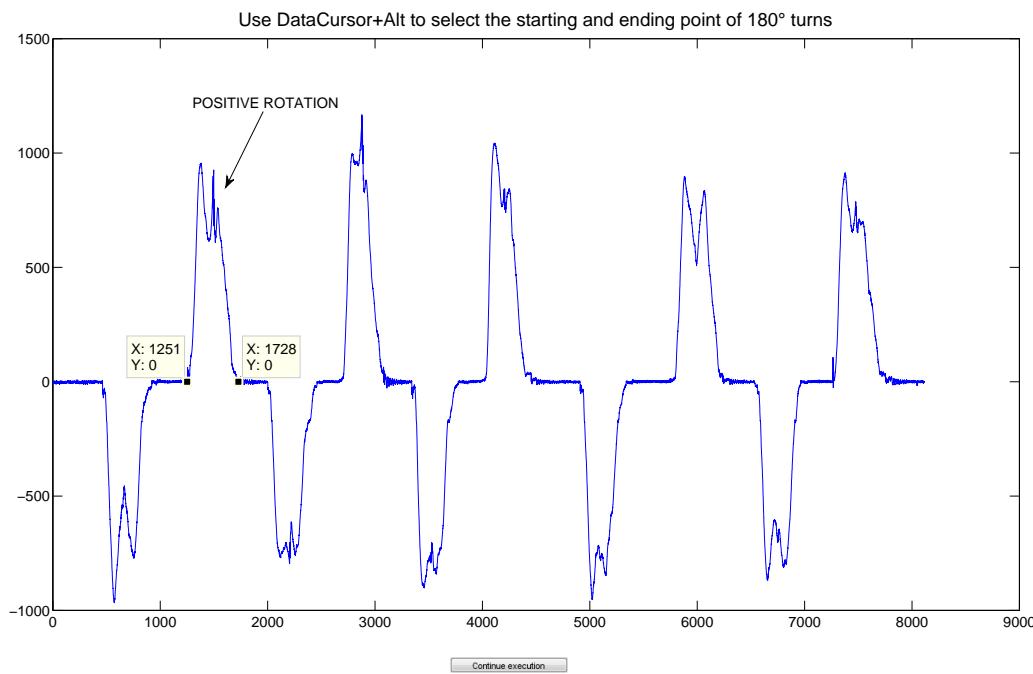


Figura 4.23: Rutina de calibración del giroscopio: selección de las rotaciones positivas para el cálculo del factor de escala

La Tabla 4.3 enumera los parámetros de calibración calculados para todos los giroscopos del GaitWatch.

Tabla 4.3: Parámetros de calibración de los giroscopos

Unidad	Eje	Factor de escala	Sesgo
Espinilla izquierda	Y	-5.84	11591
Muslo izquierdo	Y	-6.36	11720
Espinilla derecha	Y	-5.95	11618
Muslo derecho	Y	-6.40	11740
Brazo izquierdo	X	-5.89	12087
Brazo izquierdo	Y	5.75	10937
Brazo derecho	X	-6.02	11543
Brazo derecho	Y	5.64	11415
Tronco	X	-16.15	-10
Tronco	Y	16.02	62
Tronco	Z	16.23	31

4.2.1.4. Recapitulación

Como resumen final, una vez que todos los parámetros de calibración de todos los sensores están calculados, sólo necesitamos aplicar las siguientes ecuaciones,

- **Acelerómetro:**

- Acelerómetro triaxial:

$$\mathbf{a}_{cal} = S^{-1}(\mathbf{a}_{raw} - \mathbf{b}) \quad (4.28)$$

- Acelerómetro biaxial:

$$a_{cal} = k \cdot a_{raw} + b \quad (4.29)$$

- **Magnetómetro:**

$$\mathbf{h}_{cal} = S^{-1}(\mathbf{h}_{raw} - \mathbf{b}) \quad (4.30)$$

- **Giróscopo:**

$$\omega_{cal} = \frac{u_{raw} - b}{k} \quad (4.31)$$

4.2.2. Cálculo de orientación

Esta subsección comienza con una introducción a los conceptos que serán usados a lo largo de la misma, posteriormente se continuará describiendo los algoritmos utilizados en el cálculo de orientación para el sistema GaitWatch. No es necesario explicar la parte acociada al dispositivo ECnsole ya que los algoritmos aplicados son idénticos (vuelve a ocurrir lo mismo que en la parte de calibración).

A continuación se enumeran a modo de resumen los algoritmos de cálculo de orientación común a los dos dispositivos:

- Integración de la velocidad angular.
- Descomposición de la gravedad.
- El Filtro de Kalman.
- El Filtro de Kalman Extendido.

De forma equivalente al proceso de calibración, las pautas para llevar a cabo el proceso de cálculo de los algoritmos de orientación vienen recogidas en [25] donde se realiza una descripción más minuciosa del procedimiento.

4.2.2.1. Conceptos básicos

4.2.2.1.1. Sistemas de coordenadas

Un sistema de coordenadas no es más que un conjunto de vectores y números sin sentido. Para que estos valores adquieran significado, es necesario relacionarlos con una referencia conocida. A continuación definiremos los principales sistemas de coordenadas son recogidos en [23] como sigue,

- **El sistema de coordenadas inercial (i-frame).** Este sistema de coordenadas tiene su origen en el centro de masas de la Tierra y se supone que no rota respecto al espacio inercial, ya que se mueve con el planeta. Sabemos que eso no es del todo cierto, porque la Tierra gira con respecto al Sol y además de su propia rotación Ω , pero se tomará este sistema como inercial.

Los ejes del sistema ECI están fijados en las estrellas: El eje Z coincide con el eje polar y el plano perpendicular al eje Z coincide con el Ecuador. El eje X e Y no rotan con la tierra, apuntando X directamente al equinoccio Vernal.

- **El sistema de coordenadas de Navegación (n-frame).** Tiene su origen en la localización del sistema inercial (longitud latitud). Es un sistema local con sus ejes X Y en el plano tangente al punto de la Tierra donde está el origen. Típicamente el eje X apuntará al norte, el eje Y al Este y el eje Z abajo, aunque debe ser especificado. También se le conoce como NED (North, East, Down) ya que sus ejes apuntan a estas direcciones. Otra posible configuración, sería con el eje X apuntando al este, el eje Y apuntando al norte y el eje Z apuntando hacia arriba, también conocido como ENU (East, North, Up). La figura 4.24 representa el esquema de coordenadas de navegación ENU.

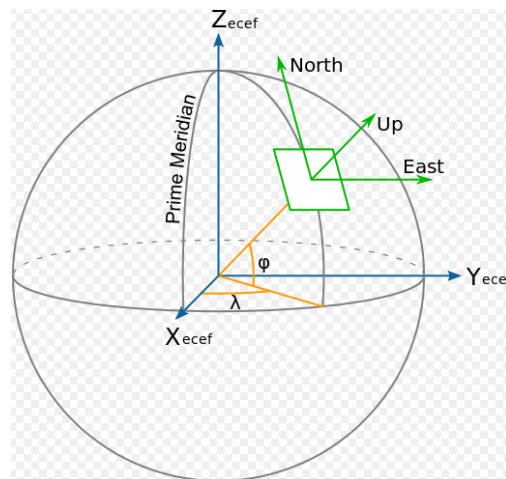


Figura 4.24: Esquema de coordenadas de navegación ENU [22].

- **El sistema de coordenadas Body (b-frame).** Este sistema tiene su origen en el centro de masas del vehículo. Típicamente usado en plataformas Strapdown, es decir, cuando los sensores tienen también como centro de masas el vehículo y sus ejes se mueven con él. La figura 4.25 nos muestra este sistema de coordenadas.

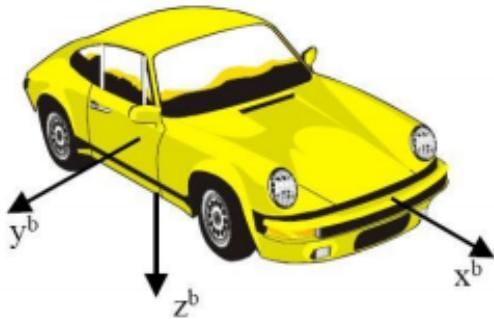


Figura 4.25: Esquema de coordenadas de Body en un vehículo [23].

4.2.2.1.2. Representación de la posición

4.2.2.1.2.1. Ángulos de Euler

Otro método bastante popular para especificar la orientación angular de un sistema de coordenadas respecto a otro es el uso de los tres ángulos de Euler. Tres ángulos, que mediante una sucesión ordenada de giros, definen el cambio de un sistema de coordenadas a otro. En este caso, utilizaríamos como referencia *el sistema de coordenadas de navegación inercial*.

Los tres ángulos de Euler son definidos en [25] como sigue,

- El ángulo *roll* (Φ) determina la rotación alrededor del eje X. Está normalmente acotado en el intervalo $(-\pi, \pi]$.
- El ángulo *pitch* (θ) determina la rotación alrededor del eje Y. Está normalmente acotado en el intervalo $[-\frac{\pi}{2}, \frac{\pi}{2}]$.
- El ángulo *yaw* (ψ) determina la rotación alrededor del eje Z. Está normalmente acotado en el intervalo $(-\pi, \pi]$.

La figura 4.26 representa los ángulos de Euler:

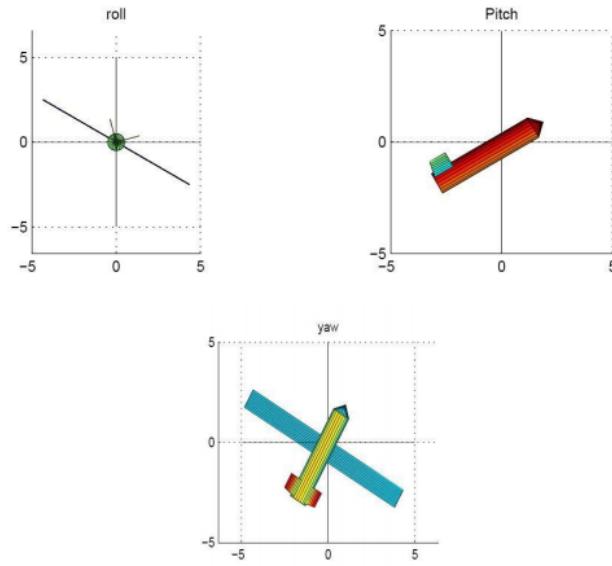


Figura 4.26: Representación bidimensional de los giros en roll, pitch y yaw [23].

4.2.2.1.3. Matrices de rotación

Las matrices de rotación [25] son utilizadas para llevar a cabo rotaciones en el espacio Euclídeo. De esta manera se usarán las matrices para poder mapear las rotaciones de Euler en el interior de las matrices. Así, si multiplicamos un vector por la matriz de rotación obtendremos su versión rotada en el espacio.

$$\mathbf{v}' = R\mathbf{v} \quad (4.32)$$

La obtención de las matrices implica una serie de operaciones trigonométricas. A continuación vamos a ir viendo cada uno de los giros por separado (asumiendo que estamos usando el sistema de coordenadas NED). Finalmente se obtendrá la matriz de rotación como resultado de la multiplicación de las tres matrices de giro.

Giro en Yaw

Es el primero de los giros que se hará. Con esta matriz de cambio se puede pasar del sistema de coordenadas (X, Y, Z) al nuevo sistema (X' , Y' , Z').

$$C_\psi = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.33)$$

Giro en Pitch

Giramos el sistema (X', Y', Z') para conseguir (X'', Y'', Z'')

$$C_\theta = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (4.34)$$

Giro en Roll

El último de los giros: de (X'', Y'', Z'') a (x, y, z)

$$C_\Phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & \sin \Phi \\ 0 & -\sin \Phi & \cos \Phi \end{bmatrix} \quad (4.35)$$

Entonces, definimos la matriz de cambio de coordenadas C_n^b así:

$$C_n^b = C_\psi C_\theta C_\Phi \quad (4.36)$$

$$C_n^b = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \phi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (4.37)$$

La ecuación 4.37 corresponde a la matriz de cambio de coordenadas de navegación a *body*. Si quisieramos la transformación inversa, es decir, cambio de *body* a navegación se obtiene aplicando

$$C_b^n = (C_n^b)^{-1} = (C_n^b)^T \quad (4.38)$$

que conduce a

$$C_b^n = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \phi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (4.39)$$

4.2.2.1.4. Cuaterniones

Los cuaterniones [25] (también llamados cuaternios) son una extensión de los números reales, similar a la de los números complejos. Mientras que los números complejos son una extensión de los reales por la adición de la unidad imaginaria i , tal que $i * i = -1$, los cuaterniones son una extensión generada de manera análoga añadiendo las unidades imaginarias: i, j y k a los números reales y tal que $i^2 = j^2 = k^2 = ijk = -1$.

¿Cómo está compuesto un cuaternion?

Un cuaternion es un número definido de la siguiente manera:

$$q = a + bi + cj + dk \quad (4.40)$$

donde a, b, c y d son números reales arbitrarios. Normalmente, a es un escalar y $bi + cj + dk$ es el vector del cuaternion. No podemos pensar que (a, b, c) como un vector típico en 3D, es un vector en 4D, aunque es muy poco intuitivo de ver.

Existen otras representaciones para los cuaterniones y se van a mostrar a continuación:

$$\mathbf{q} = (a, b, c, d) = [a, (b, c, d)] = (q_0, q_1, q_2, q_3) \quad (4.41)$$

Además, vamos a resaltar los elementos neutros o elemento identidad para las operaciones suma y resta:

$\mathbf{q} = [1, (0, 0, 0)]$. es el cuaternion identidad para la multiplicación

$\mathbf{q} = [0, (0, 0, 0)]$ es el cuaternion identidad para la suma.

Cuaterniones y las rotaciones

Un cuaternion [24] es una forma alternativa de representar rotaciones a través del cualquier eje. Comparados con los ángulos de Euler, son más simples de componer y evitan el problema del bloqueo del cardán (es la pérdida de un grado de libertad que ocurre cuando 2 de los 3 ejes necesarios para las rotaciones en 3D, son llevados a la misma dirección). Comparados con las matrices de rotación, son más eficientes y más estables numéricamente. El elemento utilizado son los cuaterniones unitarios y son aquellos que cumplen que

$$a^2 + b^2 + c^2 + d^2 = 1 \quad (4.42)$$

Los cuaterniones unitarios son especiales porque representan la orientación en el espacio 3D (permiten representar rotaciones en 3D de manera muy sencilla). Si \mathbf{q} es un cuaternion unitario, éste puede pensarse como una esfera de radio 1 en el espacio 4D, aunque resulte poco intuitivo de ver. A través de la figura 4.27 nos podemos hacer una idea.

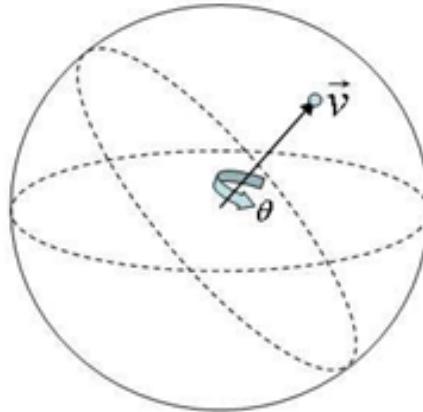


Figura 4.27: Representación en el espacio de un cuaternion unitario [24].

donde $\mathbf{q} = [\cos \frac{\theta}{2} + \sin \frac{\theta}{2} \vec{\mathbf{v}}]$

con

$$\|\vec{\mathbf{v}}\| = 1$$

Así, una rotación se puede representar con el cuaternion $\mathbf{q} = (a, (b, c, d))$, donde (b, c, d) es el eje de rotación y a es el valor del ángulo de rotación.

Conversión desde cuaterniones

Para usar los cuaterniones en la orientación, es necesario convertirlos a otras representaciones (como matrices) y volverlos a cuaterniones.

- De cuaternion a matriz.

$\mathbf{q} = (q_0, q_1, q_2, q_3)$ equivale a

$$M = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (4.43)$$

- De cuaternion a ejes/ángulo.

$\mathbf{q} = (q_0, q_1, q_2, q_3)$ equivale a

$$\Theta = 2 \cos(q_0)/s \quad (4.44)$$

$$V = (q_1, q_2, q_3)/s \quad (4.45)$$

donde $s = \sqrt{q_1^2 + q_2^2 + q_3^2}$

- De cuaternion a ángulos de Euler.

$\mathbf{q} = (q_0, q_1, q_2, q_3)$ equivale a

$$\phi = \arctan\left(\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \quad (4.46)$$

$$\theta = \arcsin(2(q_0q_2 - q_3q_1)) \quad (4.47)$$

$$\psi = \arctan\left(\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \quad (4.48)$$

Puesto que arctan y arcsin son funciones cuyo resultado se mueve en el intervalo $[-\pi/2, \pi/2]$, no podemos definir todas las posibles orientaciones teniendo tres rotaciones con con dicho intervalo. Como antes mencionamos, para cubrir todas las posibles orientaciones, roll(ϕ) y yaw(ψ) están generalmente definidos dentro de un rango $(-\pi, \pi]$ mientras pitch(θ) está delimitado entre $[-\pi/2, \pi/2]$. Por lo tanto, para definir todas las posibles posiciones y rotaciones de un cuerpo en las tres dimensiones del espacio necesitamos sustituir la función arctan en 4.46 y 4.48 por la función atan2 que está definida en el intervalo $(-\pi, \pi]$.

Conversión a cuaterniones

- De ejes/ángulo a cuaterniones.

Θ, V equivale a

$$\mathbf{q} = (\cos(\Theta/2), u_x \sin(\Theta/2), u_y \sin(\Theta/2), u_z \sin(\Theta/2)) \quad (4.49)$$

donde $\mathbf{u} = (u_x, u_y, u_z)$ es un vector en \mathbb{R}^3 de magnitud igual a la unidad.

- De ángulos de Euler a cuaternión.

Sea (ϕ, θ, ψ) hay que obtener 3 cuaterniones independientes

$$q_\phi = [\cos(\phi/2), \sin(\phi/2), 0, 0] \quad (4.50)$$

$$q_\theta = [\cos(\theta/2), 0, \sin(\theta/2), 0] \quad (4.51)$$

$$q_\psi = [\cos(\psi/2), 0, 0, \sin(\psi/2)] \quad (4.52)$$

4.2.2.2. Cálculo de la orientación usando los ángulos de Euler

4.2.2.2.1. Integración de la velocidad angular

El giróscopo mide la velocidad angular alrededor de cada uno de los ejes Cartesianos relativa a su sistema corporal. Dicha velocidad angular puede ser integrada para estimar la rotación del ángulo entre los instantes t_0 a t_1 . Por lo tanto, nosotros podemos calcular la posición del cuerpo relativa al marco de navegación si, y solo si, la orientación inicial del sistema corporal es conocida respecto al marco de navegación. Una sencilla solución es usar la aproximación basada en la proyección de la gravedad y los vectores del campo magnético de la tierra para calcular tal orientación inicial cuando el cuerpo está parado.

Existen muchos métodos numéricos que pueden ser usados para integrar la velocidad angular. En este caso nosotros usaremos la integración numérica basada en la regla trapezoidal que calcula el área debajo de la función realizando una aproximación trapezoidal como sigue:

$$\alpha(n) = \alpha_0 + \frac{T}{2} \sum_{k=1}^n [\omega_g(k) + \omega_g(k - 1)] \quad (4.53)$$

Donde T es el período de muestreo, n es el número de muestras, ω_g la medida del giróscopo en el instante n y α_0 el ángulo cuando $n=0$. El ángulo puede ser calculado también de forma recursiva usando

$$\alpha(n) = \alpha(n - 1) + \frac{T}{2}[\omega_g(n) + \omega_g(n - 1)] \quad (4.54)$$

donde $\alpha(0)=\alpha_0$.

La ventaja principal de este método es que una vez se conoce la orientación inicial,

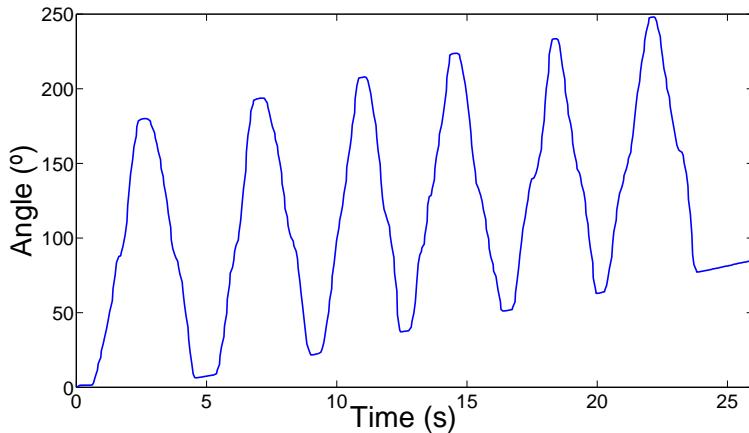


Figura 4.28: Efectos del offset dinámico en la salida integrada de la señal del giróscopo

puede ser usado para movimientos de baja intensidad o de alta intensidad mientras las medidas del giróscopo no estén afectadas por la aceleración lineal. Sin embargo, la integración del ruido y el offset dinámico conducen a un desplazamiento que crece rápidamente que tiene efectos críticos en el cálculo del ángulo de orientación. Este efecto hace que la estimación sea absolutamente errónea después de unos pocos segundos como se puede ver en la figura 4.28.

Por tanto, el uso individual de la velocidad angular para el cálculo de la orientación es inviable si se usan giróscopos MEMS los cuales presentan gran cantidad de ruido en los datos medidos.

4.2.2.2.2. Descomposición de la gravedad

Podemos usar la proyección del vector gravedad de la Tierra sobre los ejes del acelerómetro para estimar el ángulo de orientación de la estructura corporal respecto al marco de referencia. En el caso de GaitWatch, para las unidades del muslo y las espinillas sólo está disponible un acelerómetro biaxial. Por lo tanto, sólo podemos calcular uno de los dos ángulos de orientación (roll y pitch) que pueden ser estimados por medio de estas técnicas. Ya que estamos principalmente interesados en el ángulo definido en el plano XZ, esto es, el ángulo alrededor del eje es decir, el pitch. El cálculo de pitch sólo puede ser válido si el ángulo roll de las piernas tiende a 0. Debido a que las piernas se mueven principalmente en el plano sagital durante la marcha, consideraremos la siguiente expresión válida para el cálculo del ángulo pitch,

$$\theta_k = \text{atan2}(a_{z,k}, a_{x,k}) \quad (4.55)$$

donde atan2 es la función arcotangente con corrección de cuadrante.

4.2.2.2.3. Fusión de la velocidad angular con el ángulo de orientación calculado con el acelerómetro

4.2.2.2.3.1. El filtro de Kalman

El filtro de Kalman, es un conjunto de ecuaciones matemáticas que proporciona un medio para estimar el estado de un proceso minimizando el error cuadrático medio.

Si se considera un modelo de tiempo lineal, invariante y continuo:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= F\mathbf{x}(t) + C\mathbf{u}(t) + I_n\mathbf{w}(t) \\ \mathbf{z}(t) &= H\mathbf{x}(t) + I_q\mathbf{v}(t) \end{aligned} \quad (4.56)$$

donde,

- el operador $\dot{\mathbf{x}}$ denota la primera derivada \mathbf{x} .
- $\mathbf{x}(t) \in \mathbb{R}^n$ es el estado ($n \times 1$) del vector incluyendo los n estados de las variables que describe el sistema.
- $\mathbf{z}(t) \in \mathbb{R}^q$ es la salida ($q \times 1$) del vector incluyendo las observaciones (medidas) reales de los estados de las variables del proceso.
- $\mathbf{u}(t) \in \mathbb{R}^p$ es el ($q \times 1$) vector de control incluyendo las p entradas de las señales de control.
- F es el ($n \times n$) estado de la matriz de transición.
- C es la ($n \times p$) matriz de entrada.
- H es la ($q \times n$) matriz de salida.
- I_n es la ($n \times n$) matriz identidad.
- I_q es la ($q \times q$) matriz identidad.
- $\mathbf{w}(t) \in \mathbb{R}^n$ es el ($n \times 1$) proceso del vector ruido.

- $\mathbf{v}(t) \in \mathbb{R}^q$ es la ($q \times 1$) observación del vector ruido.

En ausencia de una entrada de control tenemos:

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) + I_n\mathbf{w}(t) \quad (4.57)$$

$$\mathbf{z}(t) = H\mathbf{x}(t) + I_q\mathbf{v}(t)$$

$$\mathbf{z}(t) = H\mathbf{x}(t) + I_q\mathbf{v}(t)$$

Si aplicamos la aproximación del método de Euler:

$$\dot{\mathbf{x}}(t) \simeq \frac{\mathbf{x}(t+dt) - \mathbf{x}(t)}{dt} \quad (4.58)$$

obtenemos:

$$\frac{\mathbf{x}(t+dt) - \mathbf{x}(t)}{dt} \simeq F\mathbf{x}(t) + I_n\mathbf{w}(t) \quad (4.59)$$

$$\mathbf{x}(t+dt) \simeq [F\mathbf{x}(t) + I_n\mathbf{w}(t)]dt + \mathbf{x}(t)$$

$$\mathbf{x}(t+dt) \simeq F\mathbf{x}(t)dt + I_n\mathbf{w}(t)dt + \mathbf{x}(t)$$

$$\mathbf{x}(t+dt) \simeq [Fdt + I_n]\mathbf{x}(t) + I_n\mathbf{w}(t)dt$$

si ahora discretizamos el modelo:

$$\mathbf{x}_{k+1} = [Fdt + I_n]\mathbf{x}_k + I_n\mathbf{w}_k dt \quad (4.60)$$

$$\mathbf{x}_{k+1} = \Phi\mathbf{x}_k + I_ndt\mathbf{w}_k$$

$$\mathbf{z}_k = H\mathbf{x}_k + I_q\mathbf{v}_k$$

o como es normalmente representado:

$$\mathbf{x}_k = \Phi\mathbf{x}_{k-1} + B\mathbf{w}_{k-1}dt \quad (4.61)$$

$$\mathbf{z}_k = H\mathbf{x}_k + I_q\mathbf{v}_k \quad (4.62)$$

donde,

- $\mathbf{x}(t) \in \mathbb{R}^n$ es de nuevo el $(n \times 1)$ estado del vector del sistema dinámico lineal.
- $\mathbf{z}(t) \in \mathbb{R}^q$ es el $(q \times 1)$ vector de observaciones.
- Φ es ahora el $(n \times n)$ estado de la matriz de transición del sistema discreto lineal dinámico.
- H es de nuevo $(q \times n)$ la matriz de medición de la sensibilidad definiendo la relación entre el estado de un sistema dinámico y las medidas que pueden hacerse.
- $B = I_n dt$, donde dt es el período de muestreo.
- $\mathbf{w}(t) \in \mathbb{R}^n$ es otra vez el $(n \times 1)$ proceso del vector ruido.
- $\mathbf{v}(t) \in \mathbb{R}^q$ es otra vez la $(q \times 1)$ observación del vector ruido.

Por lo tanto, cuando lo aplicamos a señales discretas, el filtro de Kalman pretende resolver el problema de intentar estimar el estado $\mathbf{x} \in \mathbb{R}^n$ de un proceso de tiempo discreto que es definido por el modelo de las ecuaciones 4.61 y 4.62. Además \mathbf{w}_k y \mathbf{v}_k son procesos de ruido blanco considerados independientes entre sí y con una distribución normal de probabilidad

$$p(\mathbf{w}) \sim N(0, Q) \quad (4.63)$$

$$p(\mathbf{v}) \sim N(0, R) \quad (4.64)$$

donde, a su vez, Q y R son las matrices de covarianza del ruido del proceso y del ruido de la medida respectivamente y pueden considerarse invariantes en el tiempo en la práctica. Cuando derivamos las ecuaciones del filtro de Kalman, el objetivo es encontrar una ecuación que calcule un estado (actualizado y corregido con la medición) a posteriori estimado como una combinación lineal de una estimación (prevista) a priori $\hat{\mathbf{x}}_k^-$ y una diferencia ponderada entre una medida actual \mathbf{z}_k y una medida de predicción $H\hat{\mathbf{x}}_k^-$,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H\hat{\mathbf{x}}_k^-) \quad (4.65)$$

La diferencia en 4.65 es conocida como medida de innovación o el residuo. El residuo indica la discordancia entre la predicción de la medida $H\hat{\mathbf{x}}_k^-$ y la medida \mathbf{z}_k .

La matriz K_k en 4.65 es conocida como la ganancia de Kalman y es elegida para ser la ganancia que minimizará el error de covarianza a posteriori. Otra forma de ver la ponderación por K_k es que a medida que el error de covarianza tienda a cero, se confía más en la medida \mathbf{z}_k y se confía menos en la predicción de la medida. $H\hat{\mathbf{x}}_k^-$. Análogamente, cuando el error de covarianza estimado a priori tiende a cero se confía menos en la medida y más en la predicción de la medida.

El filtro de Kalman controla un proceso usando una forma de control de realimentación: el filtro estima el estado del proceso en un momento y luego obtiene de forma realimentada

las medidas del ruido. Por lo tanto, las ecuaciones del filtro de Kalman se dividen en dos grupos: las ecuaciones de actualización de tiempo y las ecuaciones de actualización de la medición.

Las ecuaciones de actualización de tiempo pueden ser consideradas también como ecuaciones de predicción, mientras las ecuaciones de actualización de la medición pueden también ser consideradas como ecuaciones de corrección. Si nosotros suponemos que Φ , Q y R son constantes, la ecuaciones de actualización de tiempo son como siguen:

$$\hat{\mathbf{x}}_k^- = \Phi \hat{\mathbf{x}}_{k-1}^- \quad (4.66)$$

$$P_k^- = \Phi P_{k-1}^- \Phi^T + Q \quad (4.67)$$

mientras las ecuaciones de actualización de la medición son

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (4.68)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H \hat{\mathbf{x}}_k^-) \quad (4.69)$$

$$P_k = (I - K_k H) P_k^- \quad (4.70)$$

donde P_k^- y P_k son la estimación del error de covarianza a priori y a posteriori y K_k es un factor conocido como la ganancia de Kalman.

Así, para contruir un algoritmo usando el filtro de Kalman se deben implementar los siguientes pasos:

- Parámetros conocidos:
 1. Φ : Matriz de transición de estado.
 2. H : Matriz de medición.
 3. Q : Matriz de covarianza del ruido del proceso.
 4. R : Matriz de covarianza del ruido de medición.
- Cálculos.
 1. Cálculo de \hat{P}_k^- en sustituyendo P_{k-1} , Φ y Q en 4.67.
 2. Cálculo K_k sustituyendo \hat{P}_k^- , H y R en 4.68.
 3. Cálculo de P_k sustituyendo K_k y \hat{P}_k^- en 4.70.
 4. Cálculo de los valores sucesivos de $\hat{\mathbf{x}}_k^-$ de forma recursiva usando 4.66 y 4.69 y el cálculo de K_k . Se comienza con las estimaciones iniciales $\hat{\mathbf{x}}_0$ y P_0 .

La figura 4.29 muestra el diagrama de flujo de un filtro de Kalman de propósito general

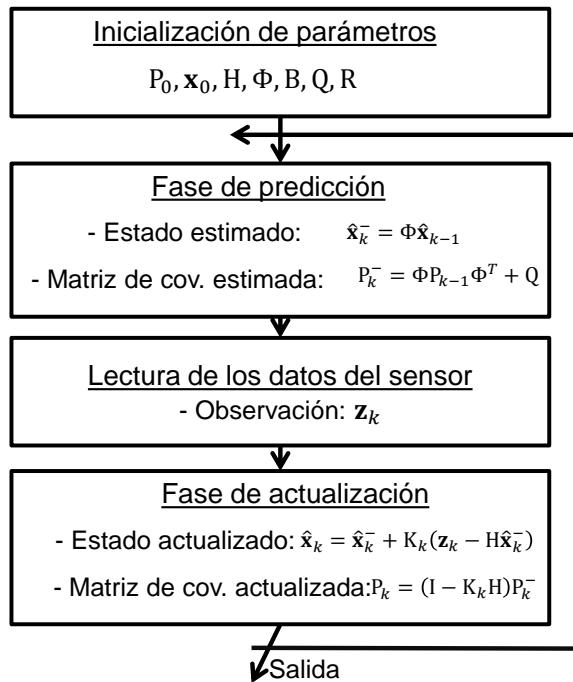


Figura 4.29: Diagrama de flujo de un filtro de Kalman de propósito general

4.2.2.2.3.2. El filtro de Kalman aplicado a la estimación de la orientación

Después de esta breve introducción a los fundamentos básicos del filtro de Kalman, explicaremos cómo emplearlo en aplicaciones de fusión, y de manera más específica cómo aplicarlo para fusionar la información que viene de los distintos sensores utilizados.

Sea el vector de estado que queremos estimar sea el siguiente

$$\mathbf{x}(t) = \begin{bmatrix} \alpha(t) \\ bias(t) \end{bmatrix} \quad (4.71)$$

donde $\alpha(t)$ es alguno de los ángulos de orientación (pitch, roll o yaw) y $bias(t)$ es la diferencia entre la velocidad angular medida y la velocidad angular real.

$$bias(t) = \omega_{meas}(t) - \omega_{actual}(t) \quad (4.72)$$

Sea además $\mathbf{w}(t)$ el vector compuesto del ruido de la estimación del ángulo y el ruido de la velocidad angular respectivamente. La observación del proceso, $\mathbf{z}(t)$ es la estimación del ángulo calculado usando la aproximación del acelerómetro-magnetómetro.

$$\mathbf{z}(t) = z(t) = \alpha(t) \quad (4.73)$$

Procedemos ahora a obtener la expresión de las derivadas de las variables de estado:

$$\frac{d\alpha(t)}{dt} = \omega_{actual}(t) \quad (4.74)$$

conocida $w_{actual} = w_{meas} - bias(t)$ que es la velocidad angular medida que presenta un desplazamiento con respecto a su valor real:

$$\frac{d\alpha(t)}{dt} = \omega_{medido}(t) - bias(t) \quad (4.75)$$

Si consideramos w_{medido} como una parte del proceso del ruido, $w_{medido} = w_\alpha$, obtenemos

$$\frac{d\alpha(t)}{dt} = -bias(t) + \omega_\alpha(t) \quad (4.76)$$

Así, el primer componente de la potencia del ruido del proceso, w_α debería ser la varianza de la señal del ángulo calculado con la aproximación del acelerómetro-magnetómetro.

Por otro lado, aplicamos la derivada a la segunda variable de estado, $bias(t)$, constituye la segunda componente del proceso del ruido y como resultado obtenemos:

$$\frac{dbias(t)}{dt} = \omega_{bias}(t) \quad (4.77)$$

Entonces, el modelo del sistema es como sigue:

$$\mathbf{x}(t) = \begin{bmatrix} \alpha(t) \\ bias(t) \end{bmatrix};$$

$$\begin{aligned}\frac{\mathbf{x}(t)}{dt} &= \frac{d}{dt} \begin{bmatrix} \alpha(t) \\ bias(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha(t) \\ bias(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_\alpha(t) \\ \omega_{bias}(t) \end{bmatrix}; \\ z(t) &= [1 \ 0] \begin{bmatrix} \alpha(t) \\ bias(t) \end{bmatrix} + v(t)\end{aligned}\quad (4.78)$$

Si ahora aplicamos la aproximación para discretizar el sistema obtenemos:

$$\begin{aligned}\mathbf{x}_{k+1} &= \begin{bmatrix} \alpha_{k+1} \\ bias_{k+1} \end{bmatrix} = \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -dt \\ 0 & 0 \end{bmatrix} \right] \begin{bmatrix} \alpha_k \\ bias_k \end{bmatrix} + \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} \omega_{\alpha,k} \\ \omega_{bias,k} \end{bmatrix}; \\ \mathbf{z}_{k+1} &= [1 \ 0] \begin{bmatrix} \alpha_{k+1} \\ bias_{k+1} \end{bmatrix} + v_{k+1}\end{aligned}\quad (4.79)$$

o equivalentemente:

$$\begin{aligned}\mathbf{x}_k &= \begin{bmatrix} \alpha_k \\ bias_k \end{bmatrix} = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{k-1} \\ bias_{k-1} \end{bmatrix} + \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} \omega_{\alpha,k-1} \\ \omega_{bias,k-1} \end{bmatrix}; \\ &\vdots \\ \mathbf{z}_k &= [1 \ 0] \begin{bmatrix} \alpha_k \\ bias_k \end{bmatrix} + v_k\end{aligned}\quad (4.80)$$

Entonces, identificando términos tenemos:

$$\Phi = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}; \quad B = \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix}, \quad H = [1 \ 0] \quad (4.81)$$

Una vez que tenemos definidas las ecuaciones de estado del sistema discreto, repetimos las ecuaciones del filtro de Kalman y su orden de aplicación para más claridad:

ECUACIONES DE PREDICCIÓN

$$\hat{\mathbf{x}}_k^- = \Phi \hat{\mathbf{x}}_{k-1} \quad (4.82)$$

$$P_k^- = \Phi P_{k-1} \Phi^T + Q \quad (4.83)$$

ECUACIONES DE ACTUALIZACIÓN

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (4.84)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (z_k - H \hat{\mathbf{x}}_k^-) \quad (4.85)$$

$$P_k = (I - K_k H) P_k^- \quad (4.86)$$

donde

- $\hat{\mathbf{x}}_k^-$ es el estado estimado a priori.
- P_k^- es la matriz de covarianza a priori del sistema.
- Q es la matriz de covarianza del ruido del proceso y está definida como

$$Q = \begin{bmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\omega \end{bmatrix} \quad (4.87)$$

donde σ_α es la varianza del ángulo calculado con la aproximación del acelerómetro-magnetómetro y σ_ω es la varianza de la velocidad angular medida por el giróscopo.

- R , que en este caso es un escalar, es la potencia del ruido observado y está definida como la varianza del ángulo estimado por el acelerómetro-magnetómetro.
- K_k es la ganancia del filtro de Kalman.
- $\hat{\mathbf{x}}_k^-$ es el estado estimado a posteriori.
- P_k matriz de covarianza a posteriori del sistema.

El lector, se estará preguntando dónde exactamente en las ecuaciones se ha utilizado la medida de la velocidad angular. Lo cierto es que las ecuaciones 4.82 y 4.86 no implementan todavía un sistema de fusión de sensores porque sólo estamos usando los datos que vienen del acelerómetro y el magnetómetro. La clave para incluir la fusión de sensores en el sistema definido es sustituir el cálculo de la predicción de los estados $\hat{\mathbf{x}}_k^-$ como sigue:

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} \alpha_k^- \\ bias_k^- \end{bmatrix} = \begin{bmatrix} \alpha_{k-1} + (\omega_{meas,k} - bias_{k-1})dt \\ bias_{k-1} \end{bmatrix} \quad (4.88)$$

De esta forma, el estado estimado a priori está siendo calculado utilizando la información de ambas aproximaciones, el cálculo del ángulo usando el acelerómetro y el magnetómetro y la integración de la velocidad angular.

Para finalizar la explicación del enfoque clásico del filtro de Kalman aplicado a fusionar la información del acelerómetro-magnetómetro y el giróscopo, incluimos debajo paso a paso las operaciones que se necesitan para llevar a cabo el algoritmos completo:

1. Inicializar la matriz de covarianza como una matriz identidad de dimensiones 2×2 , $P = I_{2 \times 2}$.
2. Definir la matriz de covarianza usando la ecuación 4.87. Esta matriz permanece constante durante la ejecución del algoritmo.

3. Definir la varianza del ruido medido, R , que en este caso es la varianza del ángulo de estimación del acelerómetro-magnetómetro, $R = \sigma_\alpha$. Este valor permanece constante durante la ejecución del algoritmo.
4. Definir el estado de las matrices de transición y de medida, Φ y H usando la ecuación (4.81).
5. Leer los datos del acelerómetro(\mathbf{a}_k), giróscopo (ω_k) y magnetómetro(\mathbf{h}_k).
6. Usando (\mathbf{a}_k) y (\mathbf{h}_k), calcular el ángulo de estimación del acelerómetro-magnetómetro, α_k mediante cualquiera de los métodos de estimación de la orientación de no fusión explicado en las subsecciones anteriores.
7. Calcular la estimación a priori del vector de estados usando la ecuación 4.88
8. Calcular la matriz de covarianza a priori del sistema usando la ecuación 4.83.
9. Calcular la ganancia de Kalman usando la ecuación 4.84, donde $z_k = \alpha_k$.
10. Actualizar el vector de estados usando la ecuación 4.85.
11. Actualizar la matriz de covarianza del sistema usando la ecuación 4.86.
12. Volver al paso 5.

La figura 4.30 muestra el diagrama de flujo del enfoque estándar del filtro de Kalman con fusión de sensores.

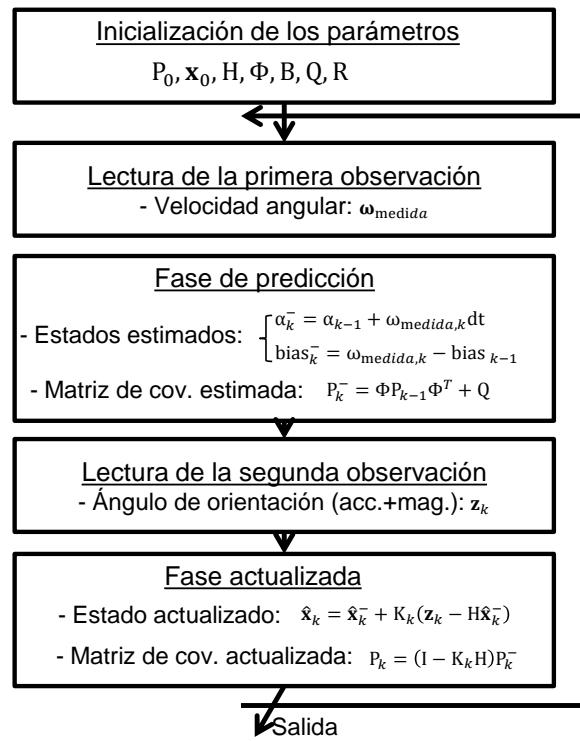


Figura 4.30: Diagrama de flujo del enfoque estándar del filtro de Kalman con fusión de sensores

Software de monitorización

Este capítulo comienza con una introducción teórica acerca de Matlab GUI para que el lector se pueda familiarizar con la creación de interfaces gráficas a través de Matlab. Inicialmente se ofrece la posibilidad de uso del entorno de desarrollo GUIDE para el diseño de aplicaciones relativamente sencillas, GUIDE es una herramienta que permite facilitar el diseño de mismas. Posteriormente se dan las pautas a seguir si decidimos prescindir de dicha herramienta, es decir, si se opta por la creación de las GUIs de forma programática (el proyecto ha sido desarrollado de esta manera). Una vez finalizado este bloque, se continúa con la descripción de la aplicación (la interfaz gráfica desarrollada que constituye el proyecto). Aquí se van metiendo en distintas secciones cada una de las partes de la aplicación con sus capturas y explicaciones correspondientes.

5.1. GUI en Matlab

Existen dos posibilidades a la hora de crear una interfaz gráfica de usuario en Matlab y las vamos a describir a continuación.

En la primera se hace uso de GUIDE el cual es un entorno de desarrollo gráfico que nos va a permitir distribuir en el espacio los elementos de la interfaz sin necesidad de aplicar un código asociado a la creación del objeto. Se elige el elemento deseado y se sitúa en el lugar correspondiente de la interfaz arrastrando con el ratón. El código del elemento se genera automáticamente cuando compilamos y tan sólo tenemos que limitarnos a darle funcionalidad.

La otra forma sería escribir el script a mano, sin ninguna ayuda externa. Implica ciertas desventajas en cuestiones tales como dedicarle más tiempo para situar los objetos (se sitúan a través de coordenadas) o el tiempo para crear las funciones correspondientes a los componentes gráficos. En el proyecto se optó por esta opción, la razón mas importante que justifica su elección es la complejidad del programa. GUIDE está pensado para la creación de pequeñas interfaces gráficas, en el momento que fue necesario añadir nuevos elementos como las *pestañas*, ir creando nuevos objetos dinámicamente o hacer aparecer y desaparecer un número indefinido de objetos, la herramienta resultó bastante limitada.

Inicialmente se partió de la idea de utilizar GUIDE y se usó hasta que comenzaron las dificultades. A continuación, vamos a explicar detalladamente las dos formas de creación de interfaces gráficas para matlab en los siguientes apartados.

5.1.1. GUI en Matlab utilizando GUIDE

Una interfaz de usuario (GUI) es una pantalla gráfica de una o más ventanas con controles (denominados componentes) que permite al usuario realizar tareas interactivas. El usuario de la interfaz gráfica no tiene que crear un script o escribir sentencias en la línea de comandos para realizar las tareas. A diferencia de los programas de codificación, que sí necesitan entender los detalles de cómo se realizan las tareas.

Los componentes GUI pueden incluir: menús, toolbars, push buttons, radio buttons, list boxes, sliders, etc (sólo por nombrar unos pocos). Por otro lado, las GUIs fueron creadas haciendo uso de las herramientas de MATLAB y pueden realizar tareas como: cálculo, leer, escribir archivos de datos, comunicarse con otras interfaces gráficas de usuario, mostrar datos como tablas, etc.

A modo de ejemplo, les vamos a mostrar como crear una interfaz gráfica simple describiendo el proceso y los distintos componentes:

Inicialmente ejecutamos desde la ventana de comandos de MATLAB el comando *guide*. Como muestra la figura 5.1:

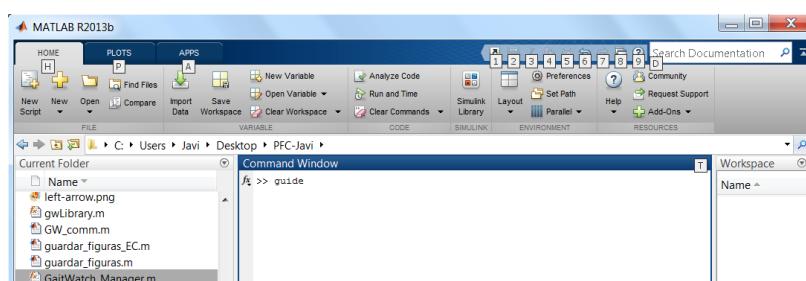


Figura 5.1: Ventana de comandos en *MATLAB*.

La respuesta que nos ofrece el programa es la que se muestra en la figura 5.2:

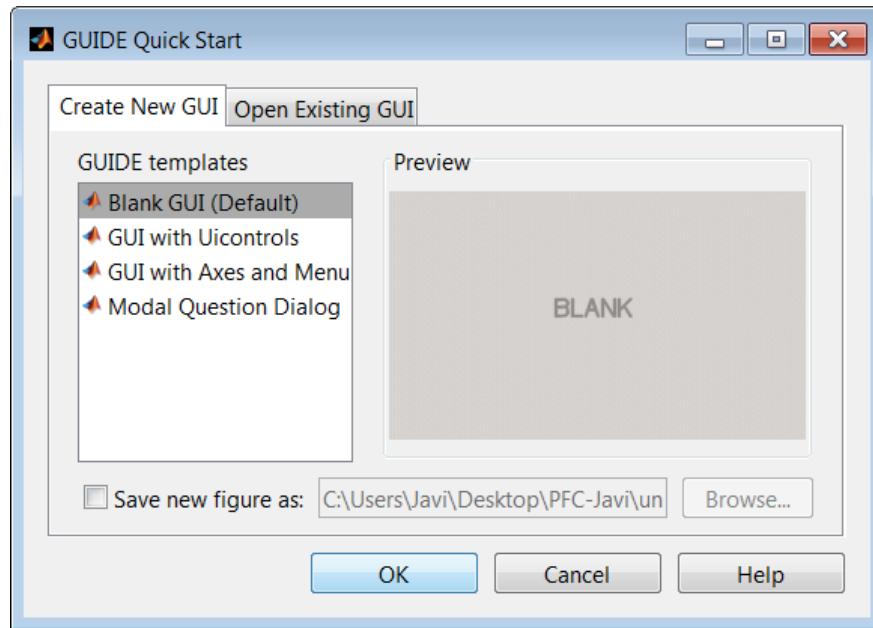


Figura 5.2: Plantillas para MATLAB GUI.

Vemos que nos muestra 4 tipos de plantillas diferentes que vamos a describir detalladamente a continuación:

- **Blank GUI.** Es la opción que viene por defecto. Genera una interfaz de usuario en blanco y es conveniente elegirla siempre que las otras plantillas no sean un punto de partida adecuado para la interfaz que se quiera crear. O que más bien se busque empezar desde cero.
- **GUI with Uicontrols.** Esta plantilla incluye: push buttons, sliders, radio buttons, check boxes, editables, static texts, list boxes y toggle buttons. Es básicamente un ejemplo de programa.
- **GUI with Axes and Menu.** Este tipo de plantilla ya viene predefinida con un axes, un popup menu y un button. Viene dado como otro ejemplo.

- **Question Dialog.** Como su nombre propiamente indica, la plantilla genera directamente un cuadro de diálogo. Está compuesta por una imagen, un static text y dos botones (*Yes* y *No*).

La opción que nos va a interesar es la primera (*Blank GUI*). Por lo tanto, la elegimos y se nos genera una ventana de desarrollo para GUI como la de la figura 5.3:

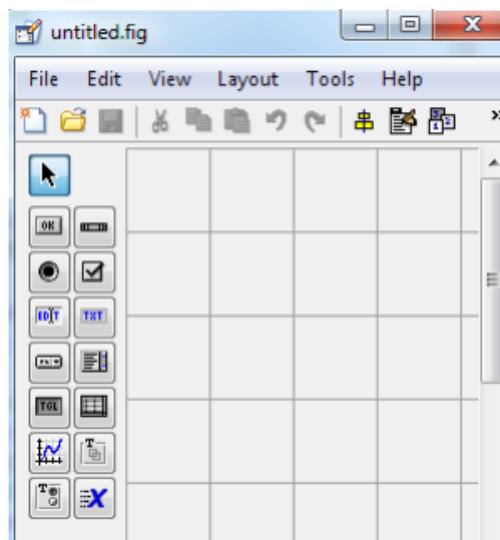


Figura 5.3: Ventana de desarrollo de GUI.

Cuando creamos un GUIDE se generan dos tipos de archivos. Por un lado el .fig que contiene el diseño gráfico y por otro el archivo.m. Este último es el script asociado al .fig.

El archivo de extensión .m está compuesto por una cabecera y por una serie de subrutinas. El nombre que se le atribuye es exactamente el mismo que al archivo.fig.

A la hora de ejecutar el programa tan sólo es necesario escribir el nombre en la consola de comandos.

Si atendemos nuevamente a la figura 5.3, nos va a llamar la atención la paleta donde están ubicados los componentes gráficos. Nos sirven de gran utilidad, con tan sólo pinchar y soltar en la plantilla podemos empezar a construir una interfaz gráfica de usuario. En la tabla 5.1 se describe uno a uno los distintos componentes gráficos:

Tabla 5.1: Tabla de componentes en GUIDE

Componente	Icono	Descripción
Push Button		Con este componente, el usuario puede hacer click y disparar una acción específica.
Slider		Son objetos gráficos que permiten al usuario seleccionar valores dentro de un rango continuo. Además de poder moverse entre un valor mínimo y máximo especificado a través de una barra de menú.
Radio Button		Al igual que pasa con el <i>check box</i> , este tipo de botón posee dos estados diferentes: activo e inactivo. Presenta unas características parecidas al Toggle Button diferenciándose en la forma.
Check Box		La funcionalidad de este tipo de botón es idéntica a la del <i>toggle button</i> , es decir, se puede elegir entre 2 estados, activo e inactivo, pero la forma es diferente.
Edit Text		Un <i>edit text</i> es un objeto gráfico que permite al usuario insertar una cadena de texto.
Static Text		Un <i>static text</i> es un objeto gráfico que representa una cadena de texto. El texto no puede ser modificado por el usuario.
Pop-Up Menu		Los <i>popup menus</i> son listas desplegables que permiten al usuario seleccionar los distintos elementos de la misma.
List Box		Son objetos gráficos que contienen una lista de elementos y posibilitan la elección de cualquiera de ellos.
Toggle Button		Es un tipo de botón que presenta dos estados: activo o inactivo.
Table		Permite generar una tabla.
Axes		Crea un nuevo conjunto de ejes para mostrar los datos activos.
Panel		Crean un marco que representa una caja rectangular sin ninguna figura dentro. Los marcos son usados para agrupar otros componentes.
Button Group		Supervisa el estado de un grupo de botones y garantiza que tan solo un botón esté activo en cada momento.
ActiveX Component		Permite mostrar controles ActiveX en la GUI.

Para finalizar, si queremos familiarizarnos con el desarrollo de interfaces gráficas a través de GUIDE deberíamos leernos algún manual de iniciación como el que tenemos en [40]. Puede resultar demasiado extenso pero expone en mayor profundidad todos los puntos analizados anteriormente.

5.1.1.1. Sumador

A modo de ejemplo, vamos a crear un programa simple. El programa va a ser un sumador [41]. Consta de tres componentes: un *pushbutton* (figura 5.4), un *static text* y un *edit text*. Se podría complicar más pero no es el objetivo que se persigue en este apartado. Lo único que se pretende es dar una breve explicación del funcionamiento de la herramienta.

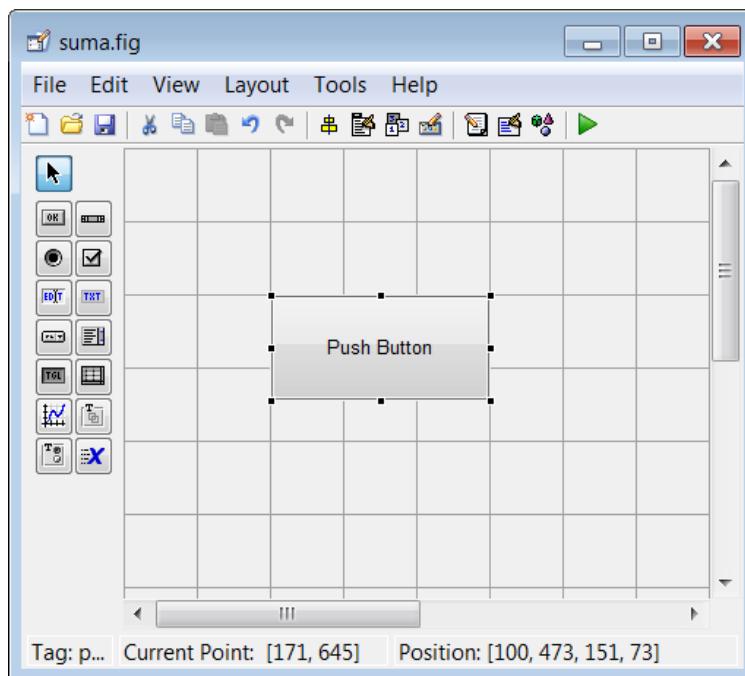


Figura 5.4: Elemento Push Button. Sumador.

Cada vez que pulsamos sobre el botón derecho del ratón encima de un componente gráfico, en este caso *un pushbutton*, se nos despliega un menú con las distintas opciones (figura 5.5).

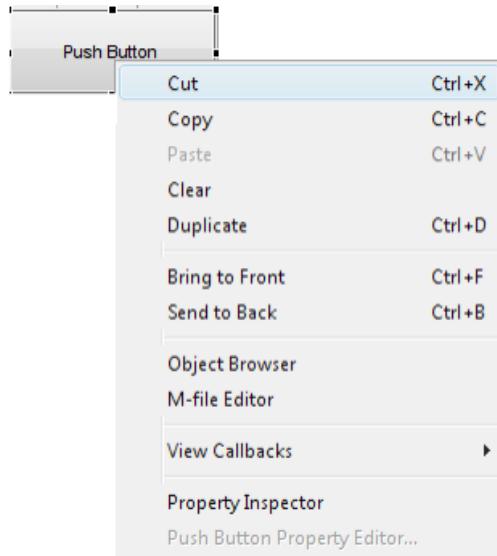


Figura 5.5: Opciones para un componente gráfico.

Si elegimos la opción *Property Inspector*, nos aparece un cuadro como el de la figura 5.6 que nos da la posibilidad de modificar las propiedades del componente gráfico.

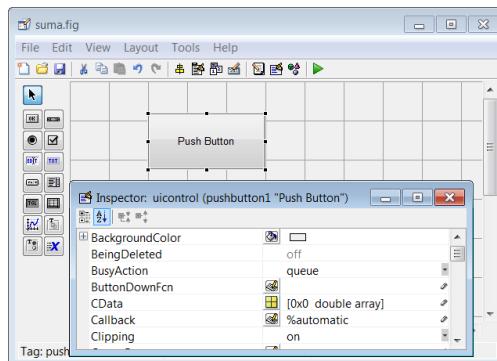


Figura 5.6: *Property Inspector*.

Una vez que nos dispongamos a editar las propiedades de un componente gráfico será porque se nos ha desplegado una consola (variará en función del elemento que estemos editando) como la de la figura 5.7. Si el usuario lo desea, podrá modificar las propiedades. Cada objeto es diferente al igual que sus propiedades. En este caso, tenemos un *pushbutton* y le podríamos cambiar propiedades tales como el color, la posición, el string, el nombre, etc.

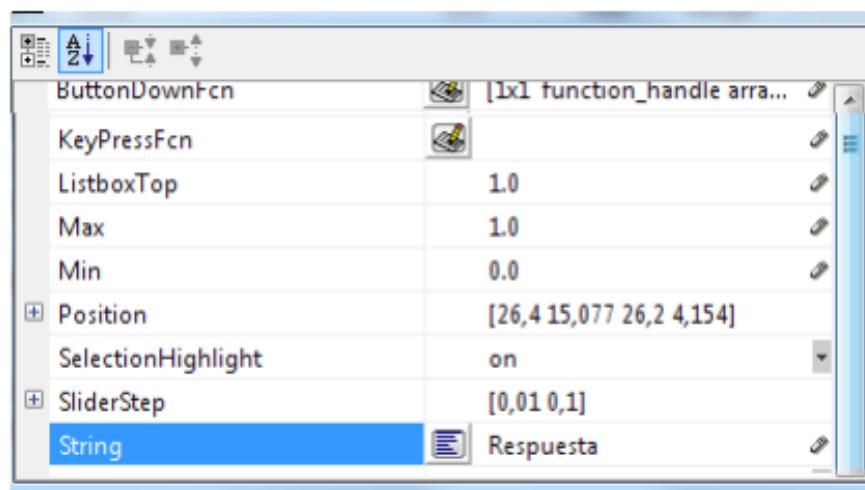


Figura 5.7: Propiedades. *PushButton*.

Conviene destacar alguna propiedad importante:

Tag nos permite modificar el nombre que va a presentar la subrutina ligada al componente gráfico que estamos configurando.

Dejando de un lado el diseño gráfico y la configuración de las propiedades el objeto nos situamos sobre el script. Un recurso bastante útil, que se encuentra en la barra de menu es la opción *show functions* (figura 5.8). Facilita en gran medida situarse en la subrutina asociada al objeto deseado con tan solo un click.

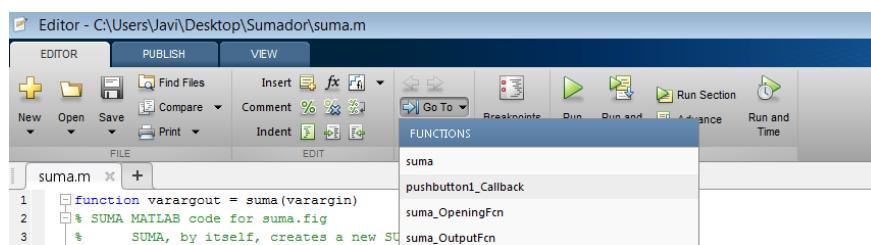


Figura 5.8: Opción. *Show functions*.

Por otra parte, no paramos de hablar de las funciones *callbacks* asociadas a los objetos que se generan automáticamente cuando compilamos el archivo .fig del diseño gráfico correspondiente. Sin embargo, no hemos mostrado ningún ejemplo del código todavía. A continuación, se muestra la función callback para el *pushbutton*:

```
%—— Executes on button press in pushbutton1.
function pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Volvemos a repetir todo el proceso anterior, en este caso para un *edit text* (Figura 5.9):

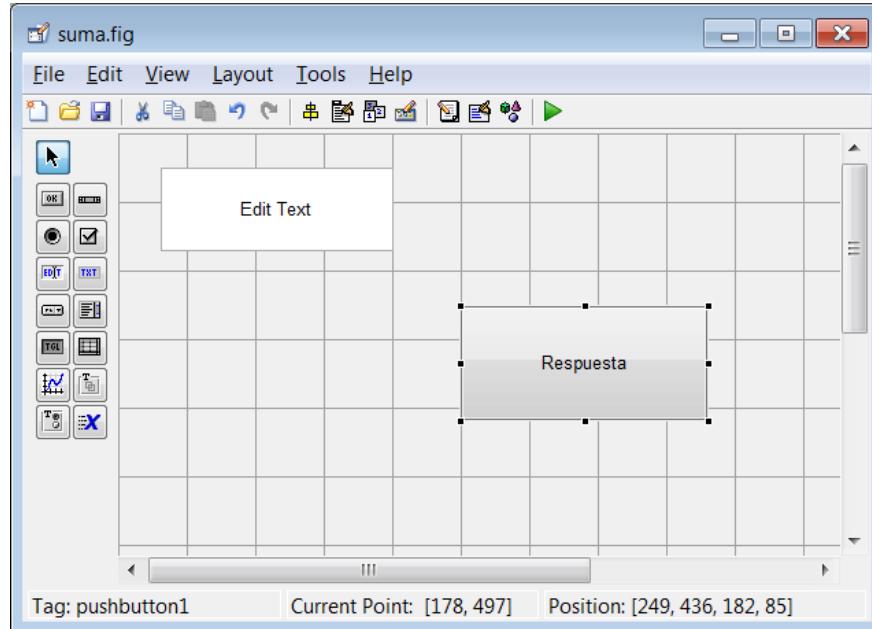


Figura 5.9: Primer *Edit Text*. Sumador.

y el código para la función callback perteneciente al *edit text* vendría dado como:

```
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Pasamos a la parte más tediosa, la implementación de la función. Vamos a explicar todas las sentencias por separado. El fragmento de código se muestra a continuación:

```
function edit1_Callback(hObject, eventdata, handles)
Val=get(hObject, 'String'); % Almacenar valor ingresado
NewVal = str2double(Val); % Transformar a formato double
handles.edit1=NewVal; % Almacenar en identificador
guidata(hObject,handles); % Salvar datos de la aplicación
```

En la primera sentencia extraemos el valor que el usuario introduzca en el *edit text* y lo almacenamos en la variable *Val*. Para una mejor compresión, es interesante saber como funciona el comando *get*:

- En el campo *hObjet* se especifica el handles correspondiente al objeto donde estamos trabajando.
- *String* es la propiedad elegida. En este caso pedimos que nos devuelva el campo *String* del *Edit Text*

La setencia *str2double* hace una conversión del valor de string a double, o lo que es lo mismo, de cadena de texto a número.

Posteriormente, se guarda en el identificador correspondiente al objeto para poder guardar los datos de la aplicación gracias al comando *guidata*.

Procedemos a realizar el mismo proceso para el segundo *edit text* (figura 5.10).

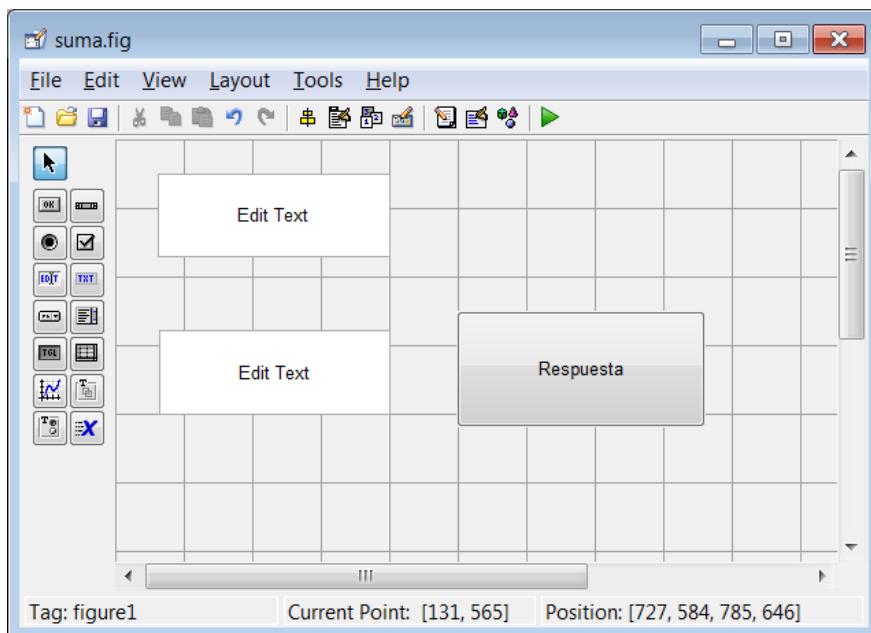


Figura 5.10: Segundo *Edit Text*. Sumador.

Lo único que hay que en tener cuenta es cambiar el identificador correpondiente al nuevo objeto. El código para *edit2* es el siguiente:

```
function edit2_Callback(hObject, eventdata, handles)
Val=get (hObject, 'String'); % Almacenar valor ingresado
```

```

NewVal = str2double (Val); % Transformar a formato double
handles.edit2=NewVal; % Almacenar en identificador
guidata(hObject, handles); % Salvar datos de la aplicación

```

Finalmente, estamos acercándonos a nuestro objetivo. Para conseguirlo es necesario crear un *static text* como el de la figura 5.11, que nos muestre el resultado de la suma.

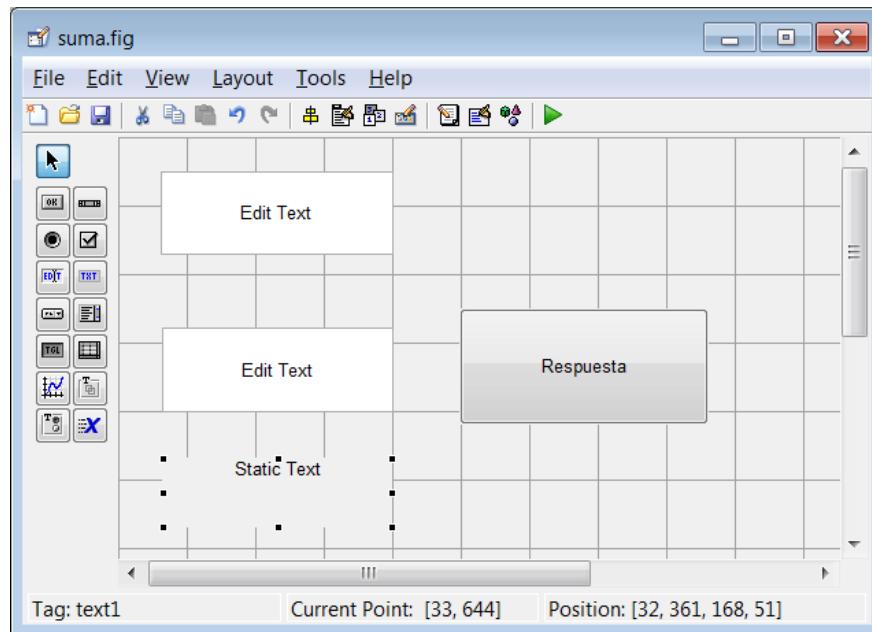


Figura 5.11: Static Text. Sumador.

Hasta ahora, tenemos almacenados los dos sumandos en los identificadores *handles.edit1* y *handles.edit2*. Queremos que cuando el usuario pulse el botón aparezca el resultado a través del *static text*. La solución se consigue dándole funcionalidad al *pushbutton*.

```

function pushbutton1_Callback(hObject, eventdata, handles)
A=handles.edit1;
B=handles.edit2;
suma=A+B;
set(handles.text1, 'String', suma);

```

En primer lugar, se extrae el valor de los dos sumandos. Después se procede a sumarlo. Finalmente hacemos uso del comando *set*. Este comando nos permite modificar las propiedades de un objeto.

El funcionamiento del comando *set* sigue las siguientes pautas:

- Se especifica el objeto usando la referencia asociada al *static text*, *handles.text1*.
- En el segundo campo fija la propiedad a modificar. En este caso, el String por donde se mostrará el resultado de la suma.
- Finalmente le pasamos el valor de la suma.

Una vez que tenemos implementado el programa queda ejecutarlo (figura 5.12) y ver si funciona correctamente (figura 5.13).

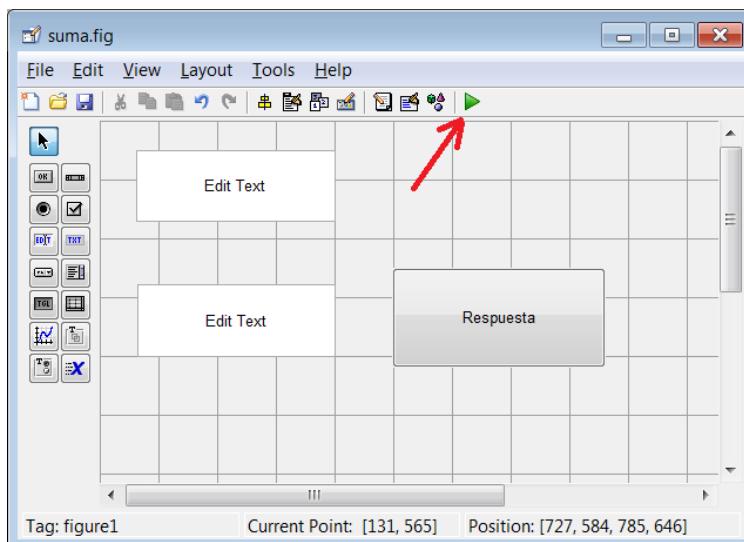


Figura 5.12: Componentes GUI. Sumador.

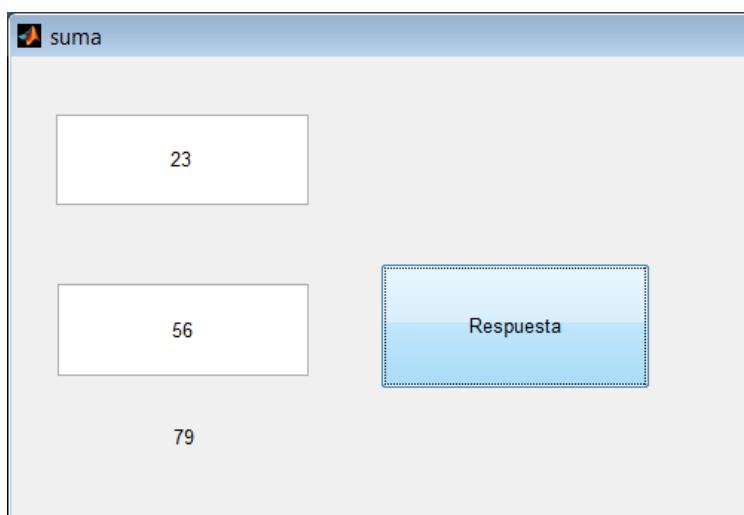


Figura 5.13: Ventana gráfica para el programa *Sumador*.

5.1.2. GUI en Matlab sin hacer uso de GUIDE. Programando

La manera de construir una interfaz gráfica de usuario sin hacer uso del entorno de desarrollo integrado (GUIDE) es servirnos del comando *uicontrol*. Este comando nos crea los elementos necesarios para diseñar las interfaces y nos permite caracterizar un elemento inicializando las propiedades en el momento que es utilizado.

A modo orientativo vamos a ver el código de un programa simple [42], sería demasiado engorroso poner un fragmento de código del proyecto. Además, en la siguiente sección se hará un estudio detallado del mismo. El programa es el siguiente:

```
function Simple_ProgGUI
f=figure('Visible', 'on', 'MenuBar', 'none', 'resize', 'off',...
    'NumberTitle', 'off', 'Name', 'Ejemplo 1',...
    'position', [300,400,150,150]);
entrada=uicontrol('style','edit','string','','position',[30,120,90,20]);
salida=uicontrol('style','text','string','','position',[30,70,90,20]);
Ver=uicontrol('style','pushbutton','string','Ver','Callback',@ver,...
    'position',[30,20,90,20]);

    function ver(hObject, eventdata)
        texto=get(entrada,'string');
        set(salida,'string',texto);
    end
end
```

La estructura que presenta posee una función principal *Simple_ProgGUI* donde se define la ventana con todos sus elementos. Para darle funcionalidad a los componentes gráficos que se han ido creando es necesario establecer las subrutinas asociadas a cada uno dentro de la función principal (la función *ver*, perteneciente al botón, ha sido implementada en el interior de la función principal del programa ejemplo). En definitiva, se puede considerar una elección del programador, los scripts implementados en el proyecto se caracterizan por dejar fuera de la función principal los *callbacks* de los componentes gráficos. Veamos más detenidamente cada una de las sentencias.

Inicialmente se crea un *figure* con las siguientes propiedades:

- 'Visible' → 'on'. Mantiene la ventana visible al usuario. Es posible ocultarla sin necesidad de cerrarla.
- 'MenuBar' → 'none'. Inhabilita la barra de menú de la ventana.
- 'Resize' → 'off'. Desactiva la posibilidad de restaurar la ventana.

- 'NumberTitle' → 'off'. Desactiva el título de la ventana. El número determina la posición que ocupa la figura entre todas las que han sido creadas.
- 'Name' → 'Ejemplo 1'. Se estable un nombre concreto a la figura.
- 'Position' → [300,400,150,150]. Se fija la posición de la ventana en la pantalla, además de la anchura y altura de la misma.

Al igual que con el comando *figure*, para crear cualquier elemento se fijan las propiedades de la misma forma (se hace uso del comando *uicontrol*). Para poder diferenciar entre elementos se utiliza la propiedad 'style' y cuando se quiere especificar el objeto al que pertenecen se recurre a la propiedad 'parent'.

En la figura vienen definidos tres tipos de elementos: un *edit text*, un *static text* y un *pushbutton*.

El callback del *pushbutton* llamado *ver*, simplemente, lo que hace es recoger la cadena de texto que el usuario introduce en el *edit text* y mostrarla a través del *static text*. Se ejecuta cada vez que el usuario hace un click con el ratón en el botón de la figura.

La imagen resultante de la interfaz gráfica la tenemos en la figura 5.14:



Figura 5.14: GUI sin hacer uso de GUIDE. Programa ejemplo.

5.2. Descripción de la aplicación

El propósito inicial que se marcó fue la realización de una aplicación Matlab que permitiera la visualización y configuración de los datos recogidos por los sistemas de monitorización del movimiento del cuerpo humano *GaitWatch Manager* y *ECnsole*. La implementación del software de los dispositivos difiere en cierta medida por el ajuste a las características propias de cada uno de los sistemas. Se va a comenzar con la descripción detallada del GaitWatch, esta decisión viene determinada al tener en cuenta la mayor complejidad respecto al dispositivo ECnsole. Sin embargo, no es especialmente relevante ya que las aplicaciones de los dos dispositivos presentan características muy similares.

En un primer instante se pensó que iba a tener acceso al sistema de monitorización de la marcha *GaitWactch* y extraer los datos necesarios para poder calcular los parámetros de calibración de manera correcta, pero finalmente no fue así. Cabe destacar que la aplicación está preparada para obtener los resultados óptimos en el momento que poseamos los datos. Sin embargo, para el dispositivo ECnsole no surgió ningún problema en ese sentido. Por razones obvias se va a hacer más hincapié en cuestiones referentes a los resultados (no así en cuanto a la funcionalidad que ofrece la aplicación) cuando se llegue al apartado correspondiente a ECnsole.

5.2.1. Pantalla principal (*GaitWatch Manager*)

La presentación de las imágenes se produce de manera secuencial de la misma forma que se ha configurado el programa. La pantalla inicial (figura 5.15) está constituida por un menú principal con las opciones *File*, *Calibration* y *Preferences*. Se pretende que cuando el usuario haya configurado adecuadamente el dispositivo y elegido las señales que desea mostrar, los resultados sean mostrados en la pantalla de inicio.



Figura 5.15: Apariencia de la pantalla principal de la aplicación del GaitWatch.

5.2.2. File (*GaitWatch Manager*)

Una vez que tenemos elegido *File* (figura 5.16) nos aparece la opción *Load Data* con la que podemos cargar todos los datos recogidos de cada uno de los segmentos corporales.



Figura 5.16: Apariencia de la pantalla principal de la aplicación del GaitWatch. Opción: *File*.

Da paso a otra ventana y nos da la posibilidad de elegir si queremos que los datos sean cargados directamente del GaitWatch por medio de *Load data from GaitWatch* o, por el contrario, si ya tenemos almacenados los datos en el disco duro, no tener que acceder al dispositivo físicamente. Viene indicado como *Load data from hard drive* (figura 5.17).

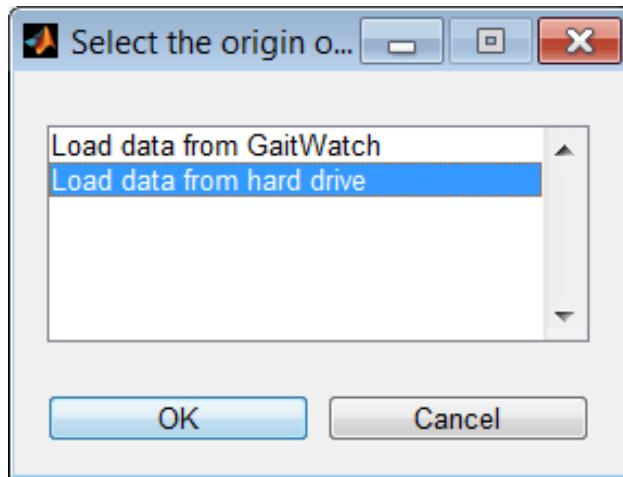


Figura 5.17: Selección del origen de los datos.

Al pinchar en *Load data* ha de aparecer una ventana de selección de archivo como la que se ve en la figura 5.18. Esto se consigue usando la función *uigetfile*. Seguidamente, los archivos se extraerían de la carpeta *data* asociada al GaitWatch. Cabe destacar que los datos recogidos fueron tomados en distintas condiciones: andando rápido, andando normal, saltando,etc. Por otro lado, sabemos que el interior de los archivos está compuesto por todos los canales tomados con los tres tipos de sensores iniciales: Acelerómetro, Giroscopo y Magnetómetro.

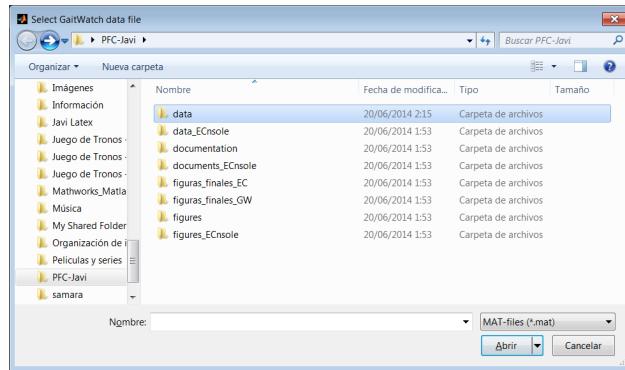


Figura 5.18: Selección del archivo de datos.

Entre todos los segmentos corporales, el magnetómetro solo es usado para captar los movimientos del tronco. Las características que posee el magnetómetro propician tener la información referente a los tres ejes: X, Y y Z. La figura 5.19 está formada por 3 ventanas para que el usuario pueda determinar los errores del magnetómetro y el programa proceda a su corrección inmediata.

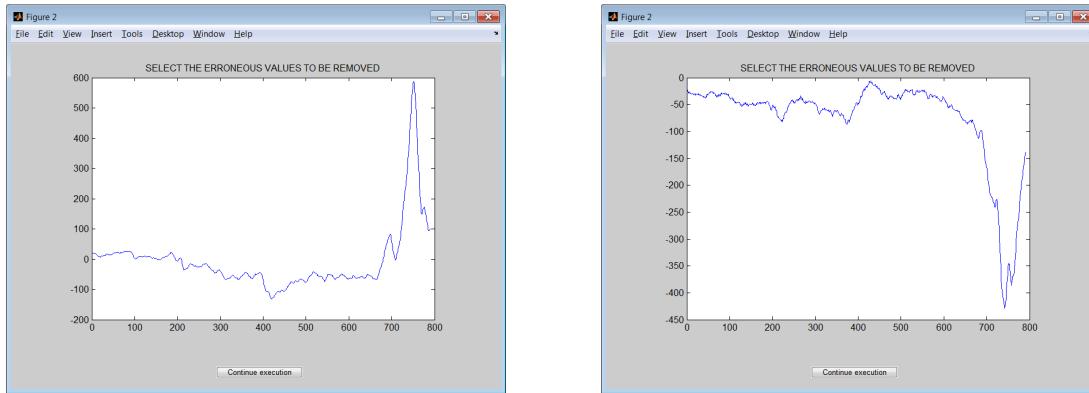


Figura 5.19: Selección de los errores del magnetómetro para su eliminación. Ejes: X,Y y Z.

5.2.2.1. Select the unit

En un principio se creó una ventana que mostrase el listado de los segmentos corporales donde se habían colocado los sensores con la posibilidad de elección de las partes que el usuario considerase convenientes. Sin embargo, queda más intuitivo colocar una imagen que represente el cuerpo de una persona consiguiendo el mismo resultado como se puede ver en la figura 5.20.

Los 7 segmentos corporales monotorizados por el GaitWatch son:

- Espinilla derecha.
- Espinilla izquierda.
- Muslo derecho.
- Muslo izquierdo.
- Brazo derecho.
- Brazo izquierdo.
- Tronco.

En el *listbox* de la derecha de la figura 5.20 quedan acumulados cada uno de los segmentos elegidos por el usuario. Se consigue pinchando en los distintos *checkboxs* que hace que se vayan añadiendo y quitando los segmentos. De esta forma se puede crear una lista personalizada.

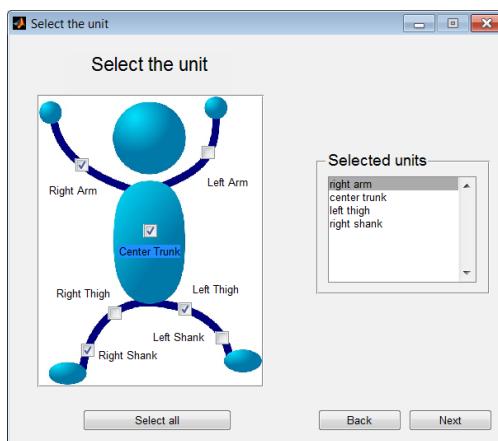


Figura 5.20: Pantalla de selección de los componentes corporales.

Existe la opción de seleccionar todas las unidades a la vez pinchando el botón *Select all*. Automáticamente, adquiere un nuevo nombre *Unselect all* y, por tanto, una nueva funcionalidad que permite deseleccionar todos los segmentos al completo. En la figura 5.21 se puede visualizar.

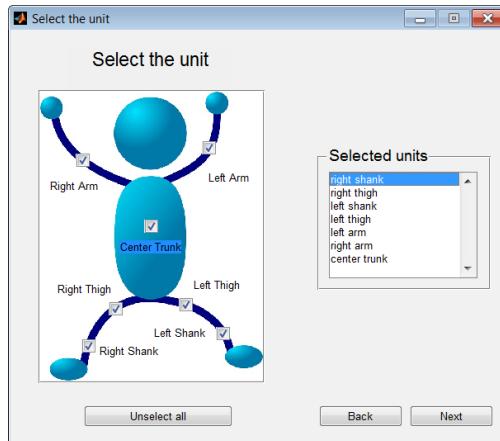


Figura 5.21: Pantalla de selección de los componentes corporales. Select all/Unselect all.

En el caso de que ninguno de los elementos del cuerpo esté seleccionado y el usuario desee pasar a la *pantalla de selección de señales*, está programado para que nos aparezca el mensaje de error "You haven't selected any body part" (figura 5.22). Se impediría el acceso a la siguiente ventana, obviamente no tendría sentido que se nos dé la posibilidad de elegir las señales si no existe ninguna.

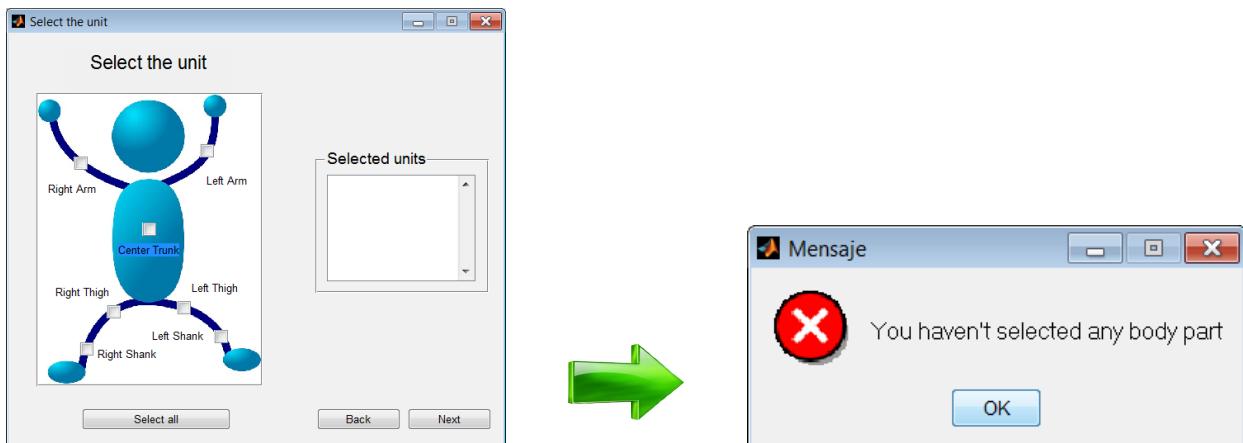


Figura 5.22: Pantalla de selección de los componentes corporales. Error.

Otro de los cuadros de diálogo que pueden aparecer por pantalla es un mensaje de tipo pregunta, cuyo mensaje sería "Do you want to return to the previous window" (figura 5.23) marcando por defecto la opción *No*. En ocasiones el usuario puede querer cancelar y

volver a la pantalla anterior. Pero no siempre se genera el mensaje, si es la primera vez que accedemos a la ventana *Select de unit* directamente se posiciona sobre la pantalla inicial. Si, por el contrario, se viene desde la ventana de *selección de señales*, para no perder la configuración establecida, se efectuará la pregunta correspondiente.

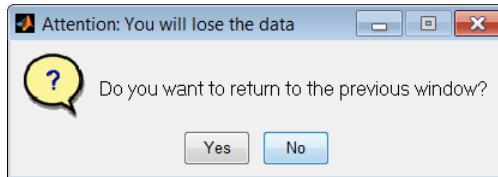


Figura 5.23: Pantalla de selección de los componentes corporales. Aviso.

5.2.2.2. Select the data you wish to be displayed

A priori, esta ventana parece solamente la pantalla de *selección de señales* previa a la ventana de configuración de las figuras. De manera no transparente al usuario cuando éste último hace click en el botón *Next*, se realizará el proceso de calibración y se le aplicarán los algoritmos de orientación a las señales resultantes. Los procesos pueden permanecer con los valores de los parámetros fijados por defecto o por el contrario, el usuario puede modificarlos en las opciones de Calibración y Orientación. Si atendemos a la imagen de la figura 5.24 vemos que la pantalla *Select the data you wish to be displayed* está formada por tres listas principalmente denominadas:

- **Raw data** → Son los datos originales extraídos directamente del sistema de sensores magnéticos e iniciales GaitWatch Manager.
- **Calibrated** → Una vez que ya tenemos los datos, se le aplica el procedimiento de calibración por medio de los algoritmos correspondientes. Son necesarios los parámetros de calibración calculados en la opción de *Calibration* de la pantalla inicial, o no necesariamente si ya estaban calculados con anterioridad. Los datos vendrán expresados en sus unidades correspondientes.
- **Orientation** → De manera equivalente al proceso de calibración se aplican los algoritmos de Orientación. Las señales resultantes son las que ibamos buscando desde un principio y las que se ajustan más a la realidad.

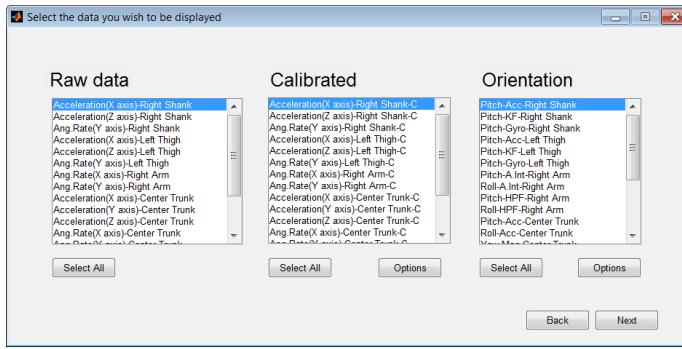


Figura 5.24: Pantalla de selección de señales.

A continuación, vamos a mostrar la tabla 5.2 con el conjunto de señales existentes para cada una de las listas, si se hubiesen escogido todos los segmentos corporales:

Tabla 5.2: Señales del sistema GaitWatch Manager

Raw data	Calibrated	Orientation
Acceleration(X axis)-Right Shank	Acceleration(X axis)-Right Shank-C	Pitch-Acc-Right Shank
Acceleration(Z axis)-Right Shank	Acceleration(Z axis)-Right Shank-C	Pitch-KF-Right Shank
Ang.Rate(Y axis)-Right Shank	Ang.Rate(Y axis)-Right Shank-C	Pitch-Gyro-Right Shank
Acceleration(X axis)-Left Thigh	Acceleration(X axis)-Left Thigh-C	Pitch-Acc-Left Thigh
Acceleration(Z axis)-Right Thigh	Acceleration(Z axis)-Right Thigh-C	Pitch-KF-Left Thigh
Ang.Rate(Y axis)-Right Thigh	Ang.Rate(Y axis)-Right Thigh-C	Pitch-Gyro-Left Thigh
Acceleration(X axis)-Right Arm	Acceleration(X axis)-Right Arm-C	Pitch-A.Int-Right Arm
Ang.Rate(Y axis)-Right Arm	Ang.Rate(Y axis)-Right Arm-C	Pitch-HPF-Right Arm
Acceleration(X axis)-Center Trunk	Acceleration(X axis)-Center Trunk-C	Roll-A.Int-Right Arm
Acceleration(Y axis)-Center Trunk	Acceleration(Y axis)-Center Trunk-C	Pitch-HPF-Right Arm
Acceleration(Z axis)-Center Trunk	Acceleration(Z axis)-Center Trunk-C	Roll-HPF-Right Arm
Ang.Rate(X axis)-Center Trunk	Ang.Rate(X axis)-Center Trunk-C	Pitch-Acc-Center Trunk
Ang.Rate(Y axis)-Center Trunk	Ang.Rate(Y axis)-Center Trunk-C	Roll-Acc-Center Trunk
Mag.Field(X axis)-Center Trunk		Yaw-Mag-Center
Mag.Field(Y axis)-Center Trunk		Pitch-KF-Center Trunk
Mag.Field(Z axis)-Center Trunk		Roll-KF-Center Trunk
		Pitch-Gyro-Center Trunk
		Roll-Gyro-Center Trunk
		Yaw-Gyro-Center Trunk
		Pitch-EKF-Center Trunk
		Roll-EKF-Center Trunk
		Yaw-EKF-Center Trunk

Del mismo modo que en la ventana *Select the unit* las tres listas, poseen un botón con doble funcionalidad (*Select All/Unselect All*), para poder seleccionar todos los elementos y una vez que lo aplicamos podemos conseguir el efecto totalmente contrario eliminando la selección de todas las señales. En la figura 5.25 tenemos un ejemplo.

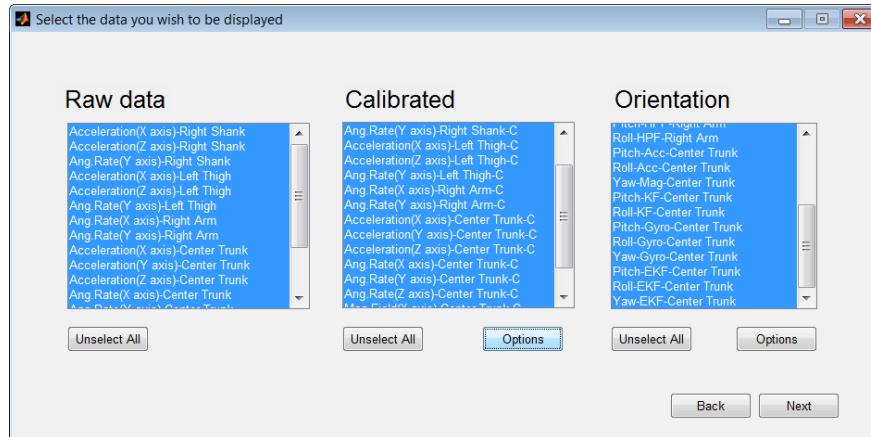
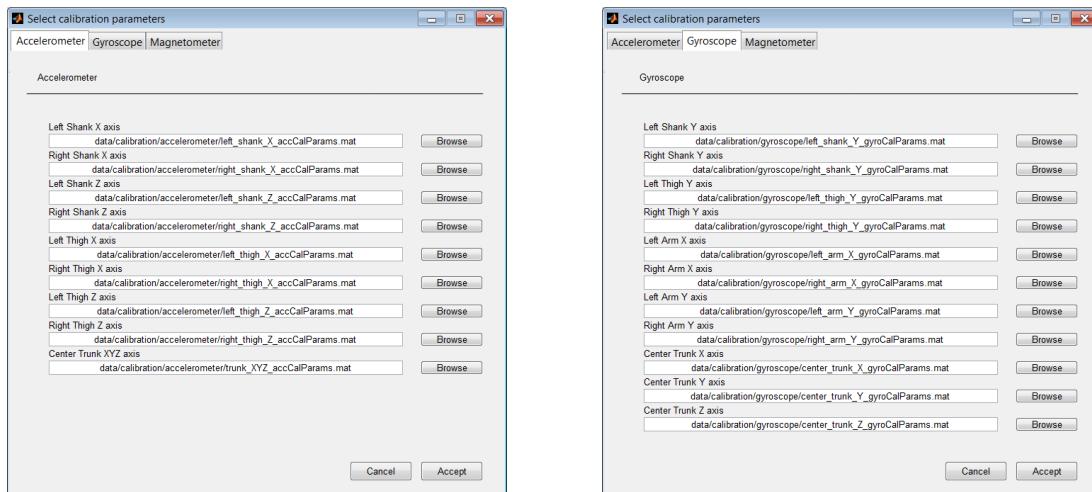


Figura 5.25: Pantalla de selección de señales. Todas las señales seleccionadas.

Al seleccionar las opciones de calibración nos aparece una nueva ventana dividida a su vez en tres por un nuevo elemento, la pestaña. Cabe destacar que las pestañas inicialmente nos ayudaron a resolver el problema de aglutinar toda la información que no cabía en una sola página. Cada pestaña está asociada a los sensores inerciales utilizados, es decir, la primera pestaña a los acelerómetros de un eje y tres ejes, la segunda pestaña al giróscopo de un eje y la tercera pestaña al magnetómetro de tres ejes. La figura 5.26 nos muestra la ventana *Select calibration parameters*.



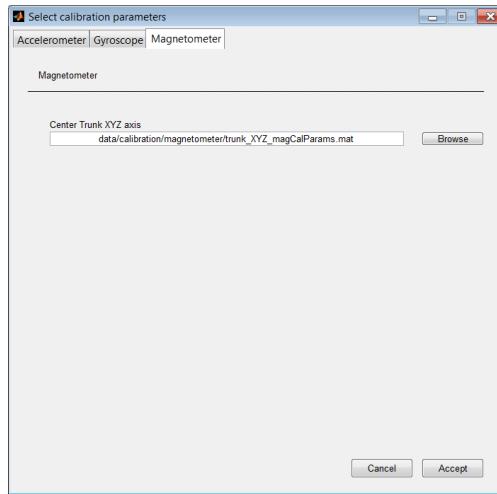


Figura 5.26: Selección de parámetros: Acelerómetro, Giróscopo y Magnetómetro.

Si pinchamos en el botón *Browse* de cualquiera de las parte del cuerpo ha de parecer una ventana de selección de archivo como las de las figura 5.27, independientemente del sensor al que pretendamos configurar sus parámetros de calibración. Cada segmento corporal tiene asociado los ejes que han sido medidos con ese sensor. Además, la ventana se abre directamente por donde están guardados los archivos de los parámetros de calibración del sensor correspondiente facilitando la búsqueda.

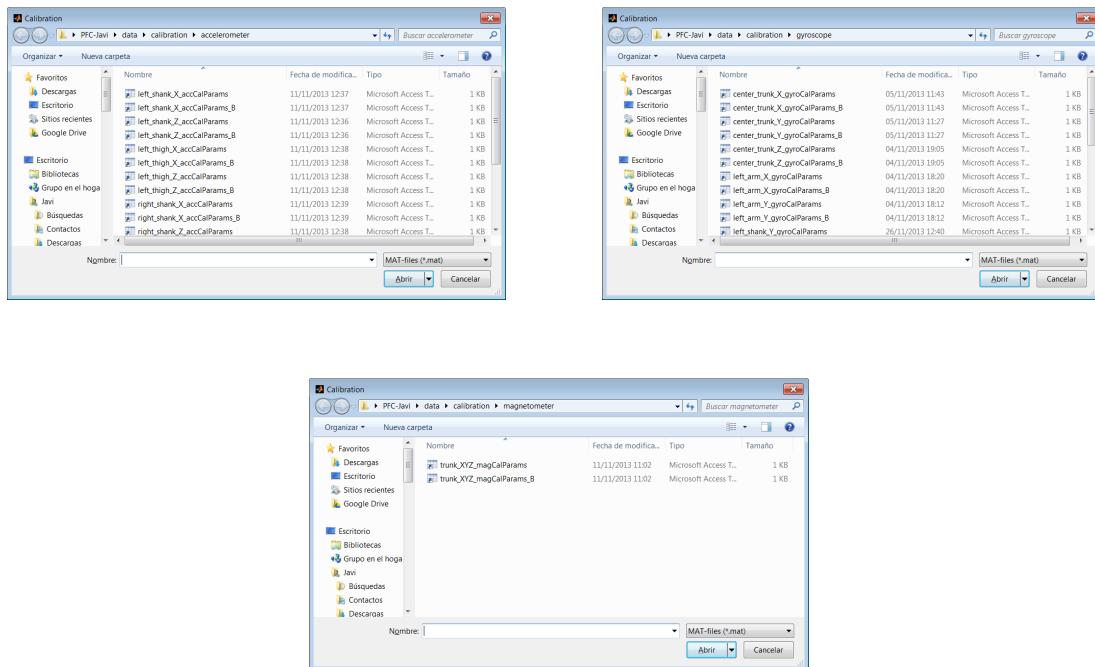


Figura 5.27: Selección de parámetros: Acelerómetro, Giróscopo y Magnetómetro.

El resultado de cambiar los archivos que contienen los parámetros de calibración por otros distintos sería el que se muestra en la figura 5.28. Estos cambios se harán efectivos en el momento que se proceda a pulsar en botón *Next* de la pantalla de selección de señales.

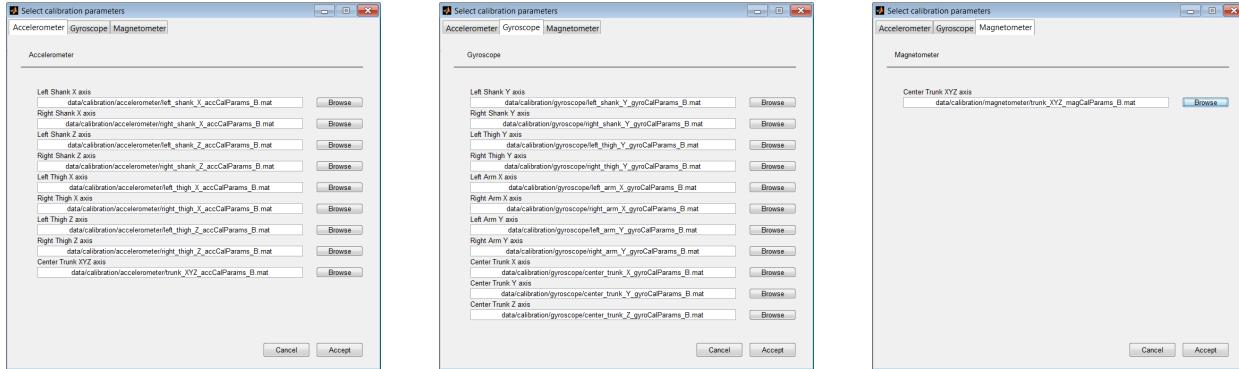


Figura 5.28: Modificación de archivos de parámetros de calibración: Acelerómetro, Giróscopo y Magnetómetro.

Por otro lado, al pinchar en el botón *options*, debajo de la lista *Orientation*, aparecerá una ventana (figura 5.29) en la que podremos configurar algunas opciones del proceso de cálculo de la orientación. De tal manera que se podrá seleccionar cualquier método perteneciente a uno de los segmentos corporales (cuyos parámetros se permitan cambiar) para proceder a modificar sus valores.

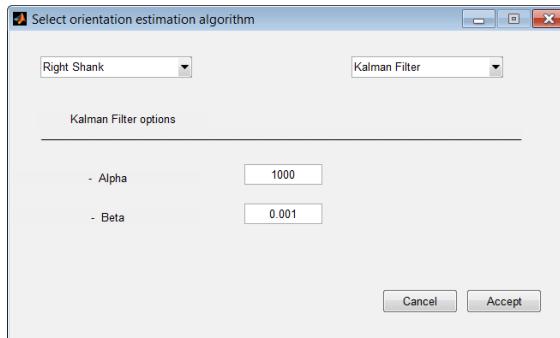


Figura 5.29: Configuración de los algoritmos de estimación de la orientación.

Seguidamente se enumera una lista de los segmentos corporales pertenecientes a las piernas, incluyendo los algoritmos del cálculo de la orientación que pueden ser modificados y sus respectivos parámetros. La figura 5.30 nos vale de ejemplo.

- Right Shank → Kalman Filter → Alpha y Beta
- Left Shank → Kalman Filter → Alpha y Beta

- Right Thigh → Kalman Filter → Alpha y Beta
- Left Thigh → Kalman Filter → Alpha y Beta

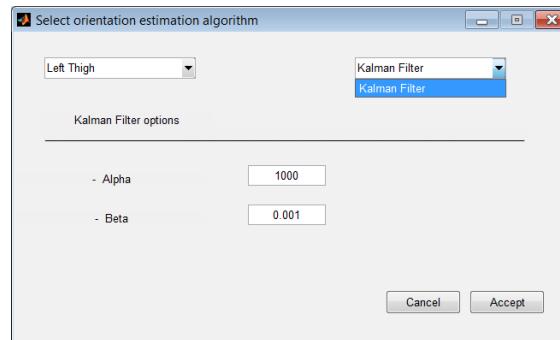


Figura 5.30: Configuración de los algoritmos de estimación de la orientación. Piernas.

Del mismo modo, se enumera una lista de los segmentos corporales pertenecientes a las brazos, incluyendo los algoritmos del cálculo de la orientación que pueden ser modificados y sus respectivos parámetros. La figura 5.31 nos vale de ejemplo.

- Right Arm → High Pass Filter options → Lower Freq. Limit
- Left Arm → High Pass Filter options → Lower Freq. Limit

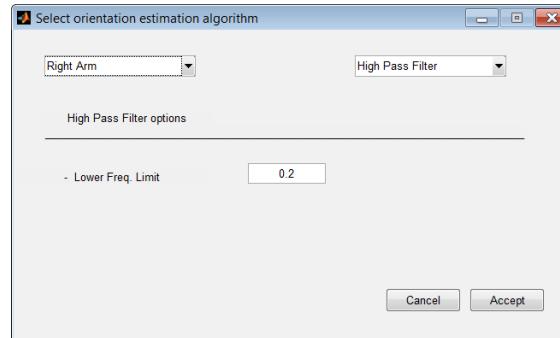


Figura 5.31: Configuración de los algoritmos de estimación de la orientación. Brazos.

Por último, se cita al tronco incluyendo los algoritmos del cálculo de la orientación que pueden ser modificados y sus respectivos parámetros. La figura 5.32 nos vale de ejemplo.

- Center Trunk

- ◊ Kalman Filter
 - Alpha
 - Beta
- ◊ Kalman Filter Extended
 - GyroVarX
 - GyroVarY
 - GyroVarZ
 - Mu. Gain
 - Alpha

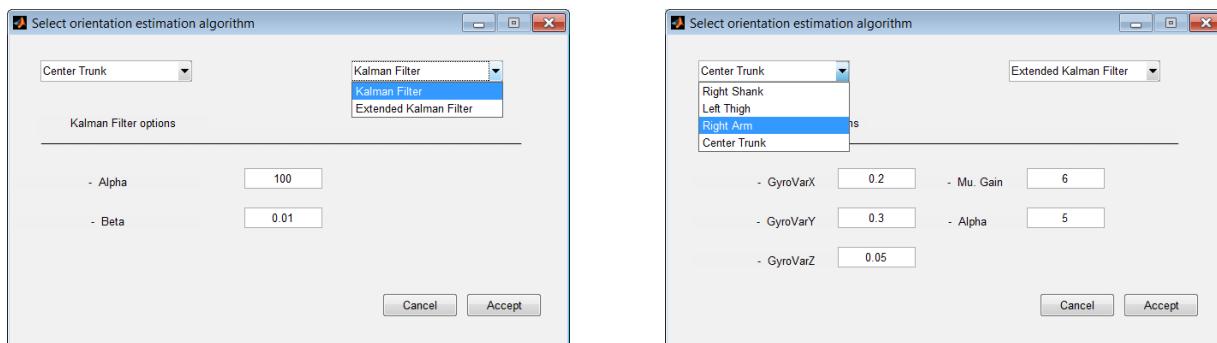


Figura 5.32: Configuración de los algoritmos de estimación de la orientación. Tronco.

A diferencia del dispositivo ECnsole, se le ha introducido una barra de progreso (figura 5.33) mientras se ejecutan los procesos de calibración y de cálculo de los algoritmos de orientación. El retardo de procesamiento es aproximadamente de unos 10 segundos. La barra de progreso facilita al usuario tener una referencia de tiempo en el paso de la ventana *Select the data you wish to be displayed* a la ventana *Set structure of figures*.

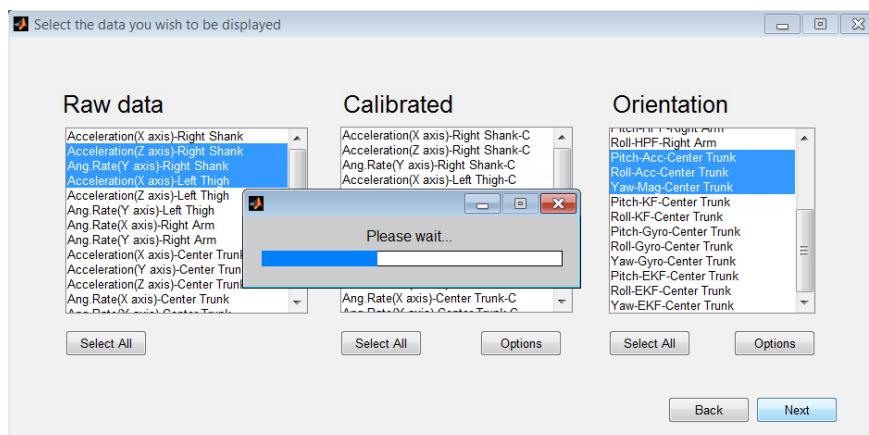


Figura 5.33: Barra de progreso.

5.2.2.3. Set structure of figures

A partir del nombre de la ventana, *Set structure of figures*, se pueden intuir sus características. Esta ventana permite al usuario configurar las figuras a su antojo, es decir, podrá establecer el número de gráficas con sus respectivas señales. Las señales seleccionadas en la pantalla anterior se aglutanán de manera secuencial (manteniendo el siguiente orden: Raw data, Calibrated y Orientation) en el listbox de la parte superior-izquierda de la figura 5.34.

Las figuras se van organizando en pestañas, las pestañas se pueden generar de forma ilimitada en función de lo que el usuario vaya requeriendo. Cada figura o pestaña se puede configurar con una de 9 combinaciones posibles (después se verá con más detalle) y la de 3x3 es la que permitiría aglutinar un mayor número de gráficas (9).

Una de las cuestiones que nos dió más problemas fue la creación de las pestañas. Se generan a través de las funciones *uitab* (pestaña) y *uitabgroup* (contendor de pestañas). Las funciones están semi-documentadas, lo que significa que no tienen apoyo o documento de página, pero si tienen secciones de ayuda legibles dentro de sus archivos -m.

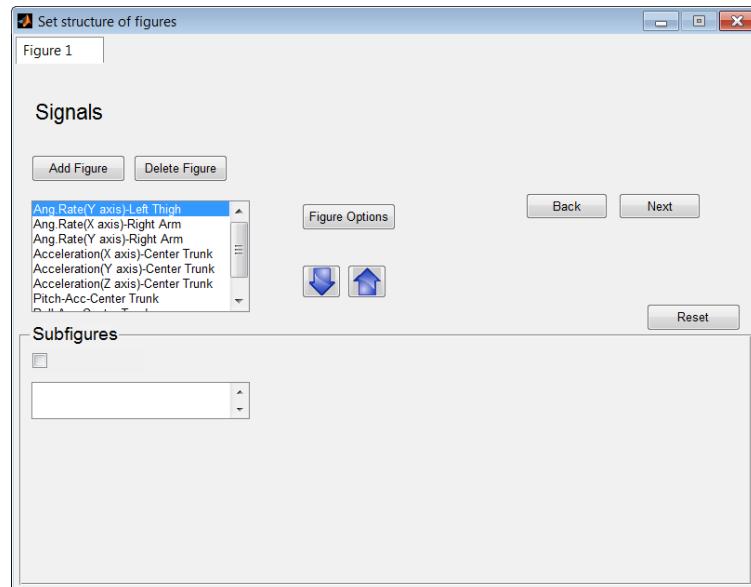


Figura 5.34: Componentes *Set structure figures*.

La tabla 5.3 recoge los componentes gráficos de la ventana *Set structure of figures*. Se muestra a continuación:

Tabla 5.3: Componentes *Set structure of figures.*

Componente	Icono	Descripción
Pestaña		El elemento gráfico permite cambiar entre figuras de forma rápida. Se referencia como <i>Figure</i> e inmediatamente se le añade el número con la posición que ocupa. La forma de las pestañas es rectangular y se ubican unas al lado de las otras distribuidas en fila. Cuando un usuario accede a una de las pestañas (con un clic) se activan todos los elementos asociados a misma.
Añadir figura		Crea una pestaña o figura nueva.
Borrar figura		Elimina la figura deseada. Posteriormente se mostrará un ejemplo.
Opciones de figura		Despliega una nueva ventana de opciones. El usuario podrá determinar las dimensiones de cada una de los <i>figures</i> . Luego veremos el diseño de la ventana.
Reiniciar		El botón permite eliminar todas las configuraciones de las figuras realizadas y partir nuevamente de la fase inicial.
Añadir señales		El botón incorpora a las gráficas (listboxs) seleccionadas en ese momento las señales que haya elegido.
Quitar señales		El botón elimina de las gráficas (listboxs) que estén seleccionadas en ese momento las señales que haya elegido de cada una.
Selección de gráfica		El elemento sirve para referenciar la gráfica que va a ser modificada en ese momento.
Volver		Nos da la opción de volver a la ventana anterior (Select the data you wish to be displayed).
Siguiente		El botón <i>Siguiente</i> nos lleva a la pantalla principal mostrándonos los resultados de las gráficas.

En la parte inferior de la ventana aparecen los listboxs donde se irán incluyendo las señales deseadas. Inicialmente sólo se puede visualizar uno, el correspondiente a la configuración de **1x1**. Las nueve configuraciones posibles serían de: **1x1** (1 gráfica), **1x2** (2 gráficas), **1x3** (3 gráficas), **2x1** (2 gráficas), **2x2** (4 gráficas), **2x3** (6 gráficas), **3x1** (3 gráficas), **3x2** (6 gráficas)

y **3x3** (9 gráficas). Veremos en las imágenes siguientes como la configuración establecida de los listboxs se corresponde con una estructura gráfica matricial de filas y columnas.

Al pulsar sobre el botón *Figure Options* se despliega una ventana con el mismo nombre (figura 5.35). Aquí el usuario puede determinar las dimensiones de las figuras.

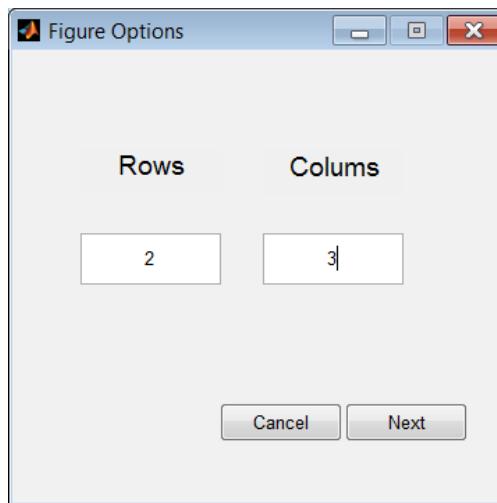


Figura 5.35: Figure Options.

Los listboxs y checkboxs activados de la figura irán acorde a las dimensiones establecidas en este momento, es decir, mantendrán la estructura matricial indicada.

Además, existen dos posibilidades de error (figura 5.36):

- La primera se da cuando las dimensiones introducidas no se corresponden con ninguna de las combinaciones citadas anteriormente. Aparecería el mensaje de error: *Figure's dimensions are invalid*.
- La segunda cuando el usuario olvida introducir al menos una de las dimensiones. El mensaje de error que se generaría es el siguiente: *You must enter values for all required fields*.



Figura 5.36: Mensajes de error: Figure Options.

Por otro lado, la representación gráfica de los resultados va a tener también el mismo tipo de estructura matricial. A través de un ejemplo vamos a corroborar que cada gráfica se corresponderá con su listbox asociado según la posición que ocupa. Para ello, *Set structure of figures* se configura de la siguiente forma:

- 9 pestañas
- Todas las configuraciones de gráficas posibles en una figura por medio de las 9 pestañas (1×1 , 1×2 , 1×3 , 2×1 , 2×2 , 2×3 , 3×1 , 3×2 y 3×3).
- Las señales se introducirán de la siguiente forma. Tomamos como ejemplo la combinación 2×3 .
 - ◊ Gráfica(1,1) → 1 señal
 - ◊ Gráfica(1,2) → 2 señales
 - ◊ Gráfica(1,3) → 3 señales
 - ◊ Gráfica(2,1) → 4 señales
 - ◊ Gráfica(2,2) → 5 señales
 - ◊ Gráfica(2,3) → 6 señales

Así con cada una de las combinaciones. Observamos las figuras 5.37, 5.38 y 5.39.

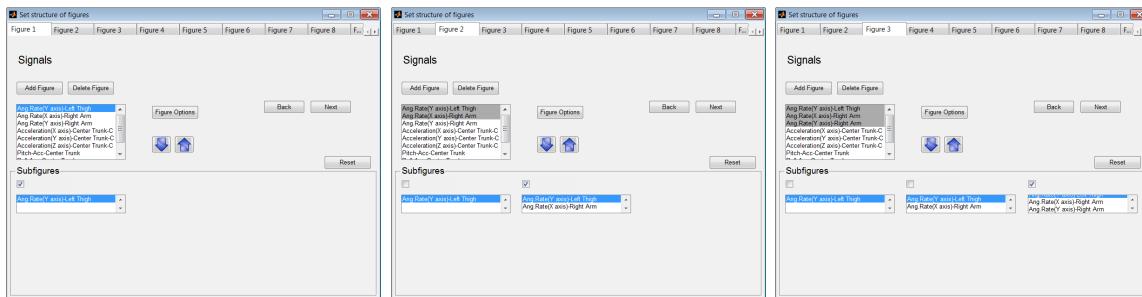


Figura 5.37: Set structure of figures: 1×1 , 1×2 y 1×3 .

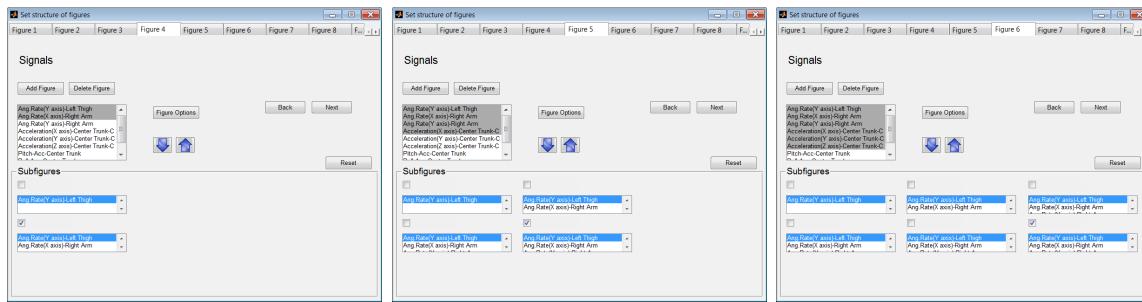


Figura 5.38: Set structure of figures: 2x1, 2x2 y 2x3.

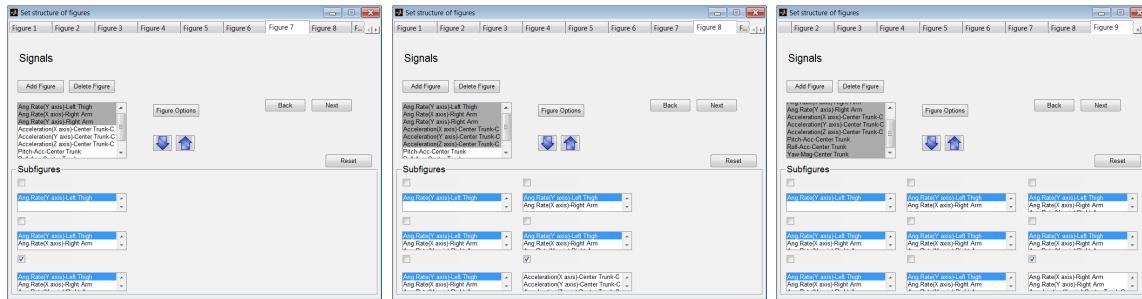


Figura 5.39: Set structure of figures: 3x1, 3x2 y 3x3.

Posteriormente, al pulsar el botón *Next* damos paso de nuevo a la pantalla principal, pero mostrando como resultado las gráficas configuradas anteriormente (figuras 5.40, 5.41, 5.42, 5.43, 5.44, 5.45, 5.46, 5.47 y 5.48).

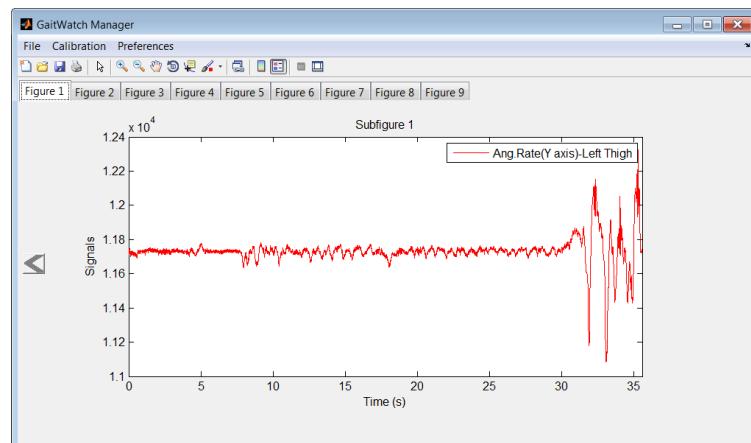


Figura 5.40: GaitWatch Manager: 1x1.

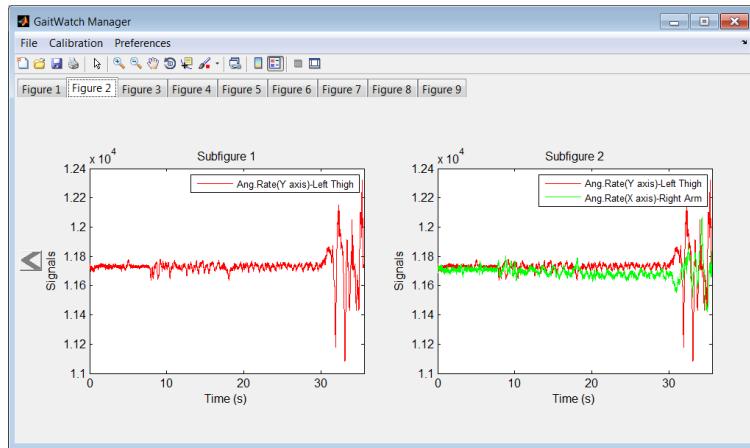


Figura 5.41: GaitWatch Manager: 1x2.

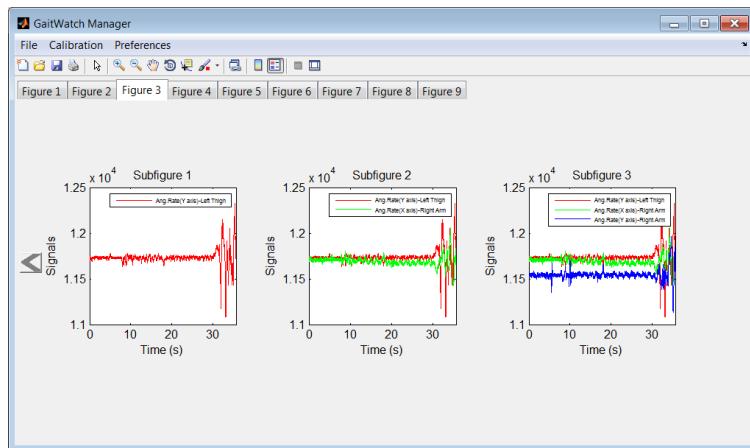


Figura 5.42: GaitWatch Manager: 1x3.

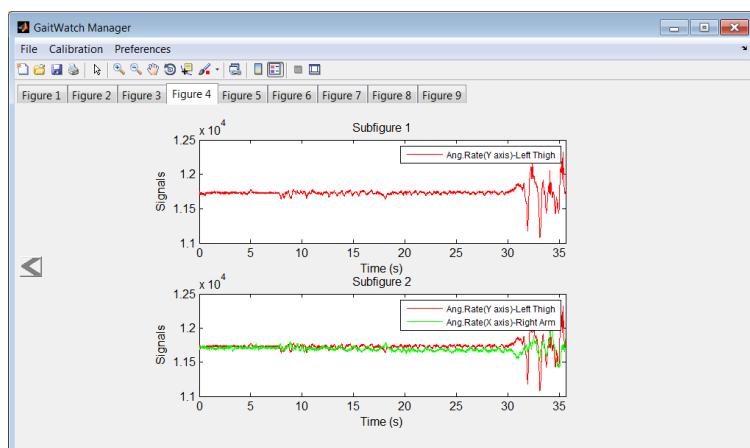


Figura 5.43: GaitWatch Manager: 2x1.

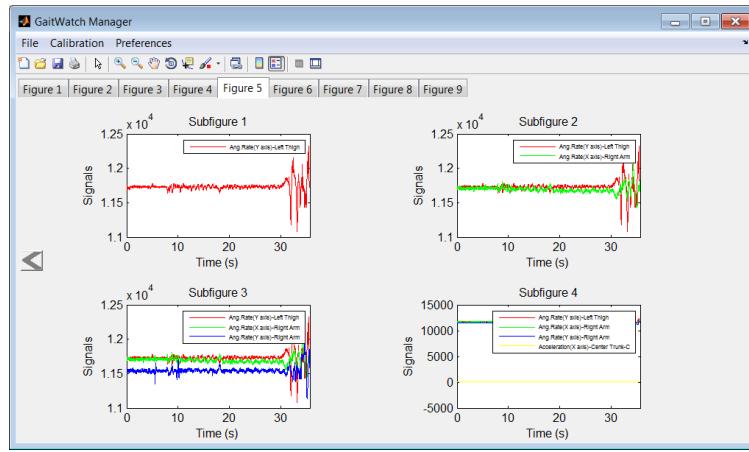


Figura 5.44: GaitWatch Manager: 2x2.

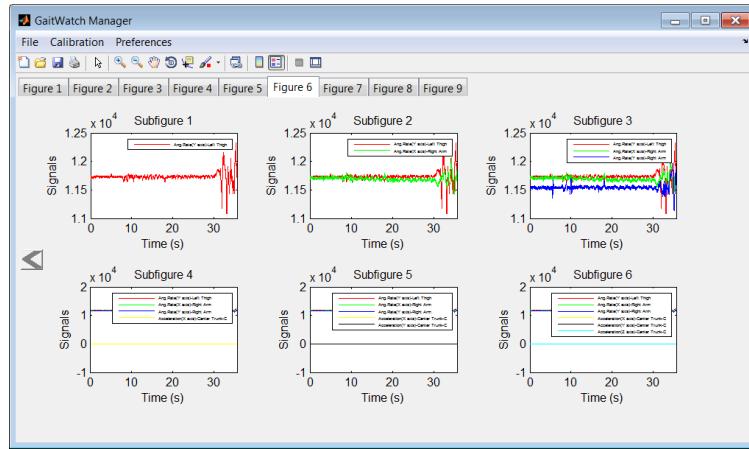


Figura 5.45: GaitWatch Manager: 2x3.

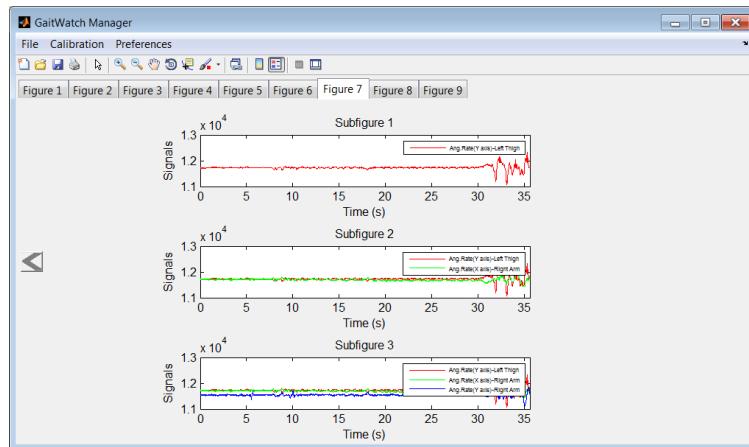


Figura 5.46: GaitWatch Manager: 3x1.

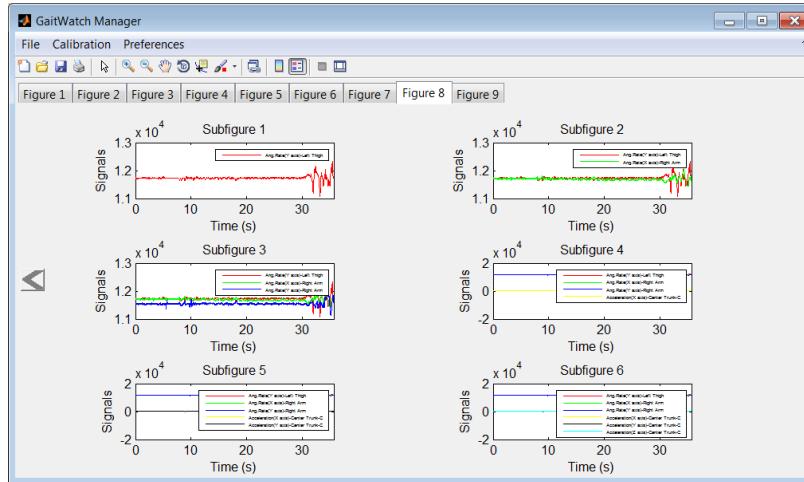


Figura 5.47: GaitWatch Manager: 3x2.

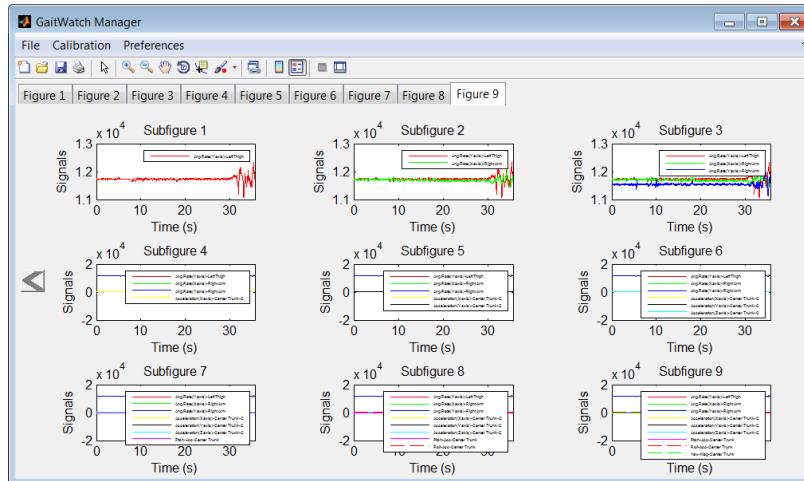


Figura 5.48: GaitWatch Manager: 3x3.

Siempre que el usuario lo considere necesario existe la posibilidad de eliminar una figura por medio del botón *Delete* de la ventana *Set structure of figures*. La pestaña se queda anulada automáticamente cuando está marcada y pulsamos el botón.

En el ejemplo de la imagen de abajo (figura 5.49) se ve que la forma de contabilizar las figuras se sigue manteniendo, es decir, las figuras 1 y 3 siguen activas mientras la figura 2 queda desactivada. Se había mencionado anteriormente pero no lo habíamos mostrado:

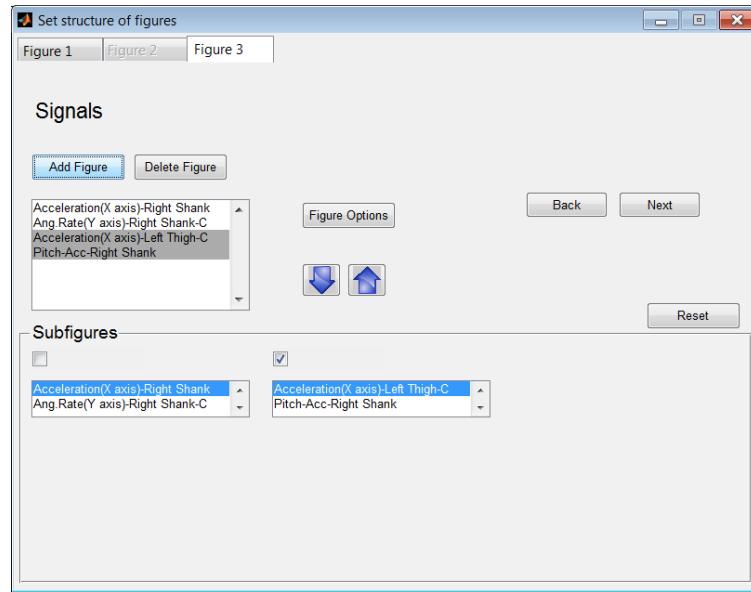


Figura 5.49: Figura eliminada.

Por lo tanto, la representación gráfica solo aparece con las figuras 1 y 3, tal y como se puede ver en la figura 5.50.

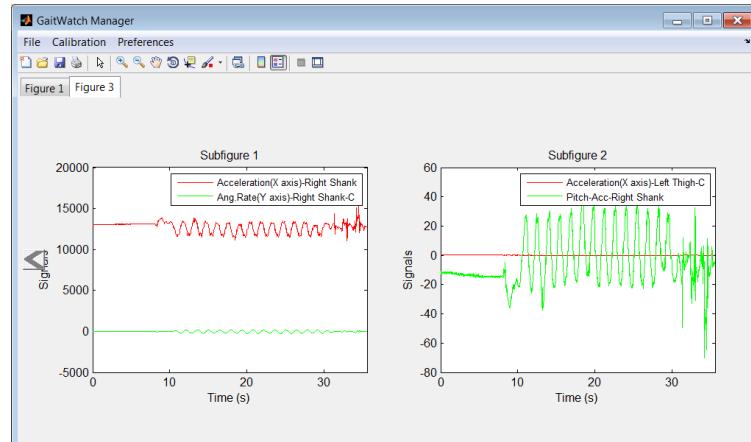


Figura 5.50: Figura eliminada: Representación gráfica.

Además, a las gráficas ya representadas se les han añadido las funcionalidades propias de la barra de herramientas de un *figure* en Matlab. Seguidamente se describen algunos de los elementos más destacados en la tabla 5.4:

Tabla 5.4: Barra de herramientas Matlab.

Componente	Icono	Descripción
Figura nueva		Su funcionalidad es crear una nueva figura.
Abrir archivo		Nos permite abrir cualquier figura almacenada.
Guardar Figura		Guarda todas las figuras actuales. No es aconsejable guardar las figuras con esta opción, la calidad es mala y no se realiza de manera individual.
Imprimir Figura		Imprime las figuras que estamos viendo.
Editar gráfica		Permite editar la figura. No posee un funcionamiento óptimo.
Aumentar zoom		Amplia la imagen sobre la zona que le especifiquemos.
Disminuir zoom		Produce el efecto contrario, dismuye la imagen todo lo que se quiera hasta quedarse en su estado inicial.
Panorámica de la imagen		Realiza un recorrido de la imagen en función de las necesidades del usuario.
Rotación 3D		Produce la rotación de la figura en 3 dimensiones.
Cursor de datos		Permite seleccionar el punto que indiquemos en la imagen. Nos da sus coordenadas.
Pintar/Dato seleccionado		Cubre con una capa de color el punto indicado.

Cabe destacar la falta de eficacia de la opción para guardar figuras. Se ha optado por añadir a *File* una nueva funcionalidad denominada *guardar como*(figura 5.51) que resuelva los problemas enumerados a continuación:

- Las figuras nos se pueden guardar individualmente.
- La calidad de las figuras es bastante deficiente. El efecto de guardado crea un momento de inestabilidad sobre la imagen y las pestañas se desajustan de su sitio coincidiendo que la captura de pantalla se toma en ese mismo instante.
- Una vez que la figura ha quedado guardada ya no se puede volver a hacer uso de la opción *Guardar como*. Se vuelve a guardar en el lugar especificado anteriormente sobreescribiendo el archivo.

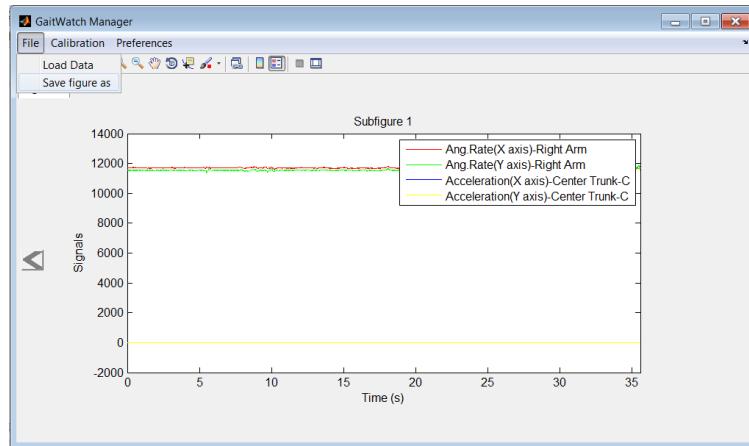


Figura 5.51: Save figure as.

Además, se puede guardar en una gran variedad de formatos. Los formatos vienen especificados ahora pero también los tenemos en la figura 5.52.

- MATLAB Figure (*.fig)
- Bitmap file (*.bmp)
- EPS file (*.eps)
- Enhanced metafile (*.emf)
- JPEG image (*.jpg)
- PaintBrush 24-bit file (*.pcx)
- Portable Bitmap file (*.pbm)
- Portable Document Format (*.pdf)
- Portable Graymap file (*.pgm)
- Portable Network Graphics file (*.png)
- Portable Pixmap file (*.ppm)
- TIFF image (*.tif)

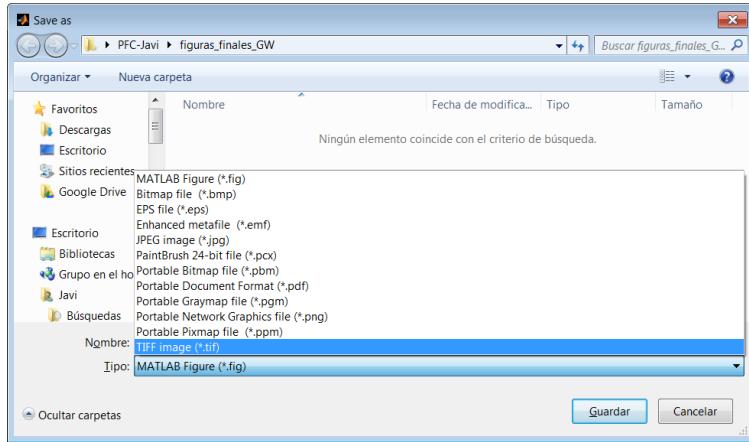


Figura 5.52: Formatos de guardado.

5.2.3. Calibration (GaitWatch Manager)

En esta parte no se va a mostrar ninguna gráfica adecuada al proceso de cálculo de los parámetros de calibración. Los resultados obtenidos son erróneos debido a que el sistema de monitorización del movimiento GaitWatch está en Alemania y por motivos obvios tuvimos grandes dificultades para extraer los datos necesarios. Sin embargo, no quita que los algoritmos utilizados no cumplan con su objetivo con unos datos óptimos. Cuando lleguemos a la parte de ECnsole visualizaremos algunas de las gráficas pero tampoco profundizaremos demasiado ya que el proyecto se centra en el desarrollo del software que permita mostrar los resultados.

Nuevamente nos vamos a centrar en el dispositivo GaitWatch y vamos a ver que para acceder al cálculo de los parámetros de calibración es necesario recurrir al menú, más en concreto, a la opción *Calibration* (figura 5.53), en donde se puede elegir entre los distintos sensores: *Calibrate accelerometer*, *Calibrate gyroscope* y *Calibrate magnetometer*.

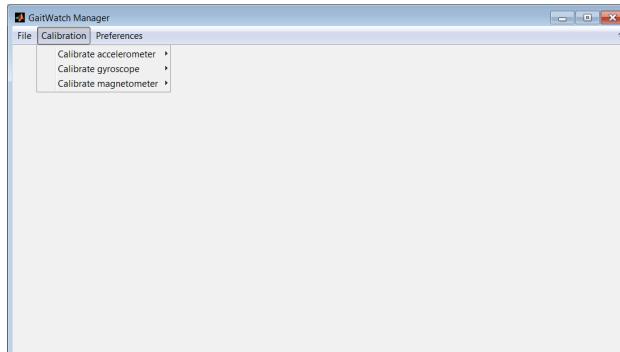


Figura 5.53: Apariencia de la pantalla principal de la aplicación del GaitWatch. Opción: *Calibration*.

Teniendo en cuenta que existen dos tipos de acelerómetro (figura 5.54) vamos a pasar a describirlos por separado.

Cuando el usuario se decanta por la opción del acelerómetro de un eje implica el cálculo de los parámetros de calibración con los datos recogidos por los acelerómetros uniaxiales que están localizados en las espinillas y los muslos. La calibración utilizada es bastante simple. El acelerómetro necesita colocar su eje de forma paralela y antiparalela al vector gravedad de la Tierra. Por esta razón, tendrímos 2 valores diferenciados (1g y -1g) para cada acelerómetro determinando así el factor de escala y la bias.

En el caso que el usuario elija el acelerómetro triaxial implica el cálculo de los parámetros de calibración que estarían incluidos dentro del tronco haciendo uso del GaitWatch. Se usa el algoritmo de ajuste de elipsoide propuesto por Camps et al. (F. Camps, S. Harasse, and A. Monin. Numerical calibration for 3-axis accelerometers and magnetometers. En la IEEE Conferencia Internacional *Electro/Information Technology*, 2009. EIT 09, páginas de 217 a 221. IEEE, June 2009). Aquí se presentan los pasos teóricos y experimentales del método de cálculo de ganancias, bias y factores de no ortogonalidad del los magnetómetros y los acelerómetros. La calibración llevada a cabo implica poner los acelerómetros en múltiples posiciones casi estáticas.



Figura 5.54: Calibración de parámetros del acelerómetro.

En la figura 5.55 se opta por el cálculo de los parámetros de calibración del giróscopo. Se basa en la rotación de un ángulo conocido del giróscopo en vez de someterlo a una velocidad angular que también sería conocida. Es necesario que el usuario intervenga. Más concretamente, al usuario se le mostrará la señal de velocidad angular que contiene los giros positivos y negativos de 180 °. Él debe seleccionar los puntos inicial y final del periodo estático al comienzo de la señal. La selección se realiza con el *datacursor* que se activa en cuanto se genera la figura. Para seleccionar varios puntos el usuario ha de pulsar la letra *Alt* mientras hace click con el ratón sobre los puntos. Una vez que el usuario ha elegido los puntos inicial y final se pulsa el botón *continue execution* y se calculará la bias estática para los valores recogidos en la selección del período estático. Después, al usuario se le muestra la señal de velocidad angular obtenida del GaitWatch con la bias corregida. En esta ocasión, el debe seleccionar el punto inicial y final de las rotaciones positivas. Para determinar esos puntos el usuario debe usar la herramienta *Zoom* contenida en *View→Figure toolbar* junto con el cursor de datos para los valores 0 que son más próximos al comienzo y finalización de las vueltas. A veces, debido al ruido estos valores no son 0, por lo tanto, el usuario debe seleccionar entonces valores cercanos a 0 (-2, -1, 1 o 2).

Para incrementar la precisión de la rotación, el usuario debe seleccionar los puntos inicial

y final, al menos 4 o 5, en rotaciones positivas. Antes de seleccionar hay que asegurarse que el número de puntos es par. Cuando se esté listo se hace click en *continue execution* y el script calculará el factor de escala. Usando este valor y el valor bias calculado al comienzo del script, la señal original que contiene los giros de 180 ° es calibrada y posteriormente integrada. Si el cálculo de los parámetros es exitoso entonces el cálculo de la rotación debe de ser próximo a los 180 °(o -180 ° en el caso de que la rotación del giróscopo fuese invertida). Al final del script los parámetros de la bias y el factor de escala son guardados y se almacenan en una carpeta que será usada en la calibración de los datos.



Figura 5.55: Calibración de parámetros del giróscopo.

La figura 5.56 nos muestra la elección del cálculo de los parámetros de calibración del magnetómetro que, al igual que el acelerómetro de 3 ejes, está situado en el tronco. También está basado en el algoritmo de ajuste de elipsoide propuesto por Camps et al. Como ya comentamos antes, se presentan los pasos teóricos y experimentales del método de cálculo de ganancias, bias y factores de no ortogonalidad de los magnetómetros y los acelerómetros. La calibración implica rotaciones arbitrarias de la IMU con las que se consiguen los datos necesarios, el proceso es bastante simple.



Figura 5.56: Calibración de parámetros del magnetómetro.

5.2.4. Preferences (GaitWatch Manager)

Como tercera opción se le ofrece la posibilidad al usuario de poder elegir el sistema de monitorización del movimiento deseado. Al pinchar en *Preferences* nos aparece un submenú con la opción *Select Device* (figura 5.57).



Figura 5.57: Selección de dispositivo.

La siguiente ventana (figura 5.58) nos muestra la lista de dispositivos disponibles. En nuestro caso: *GaitWatch Manager* y *ECnsole*. La selección es posible gracias al elemento gráfico *checkbox* que acompaña al nombre del dispositivo.



Figura 5.58: Selección de dispositivo.

La ventana no permite la selección de más de un dispositivo. Cuando el usuario se decanta por uno, automáticamente desaparece la selección anterior. Una vez pulsado el botón *Accept* se pasa a la pantalla principal del dispositivo que corresponda. Los datos y gráficas que tuviesemos representados se perderán.

5.2.5. Pantalla principal (ECnsole)

En los apartados anteriores se ha explicado en profundidad la composición de la interfaz gráfica desarrollada para el dispositivo *GaitWatch Manager*. Ahora nos centraremos en el otro sistema de monitorización de la marcha que tenemos, denominado *ECnsole*. La interfaz gráfica diseñada (figura 5.59) comparte en gran medida las características del sistema anterior. Por lo tanto, destacaremos exclusivamente las cuestiones que los diferencian, o más bien, las peculiaridades propias del dispositivo *ECnsole*.

Para empezar, la pantalla principal en su estado inicial es similar, exceptuando el nombre de la cabecera que hace referencia al dispositivo actual *ECnsole*.



Figura 5.59: Pantalla principal: ECnsole.

5.2.6. File (ECnsole)

En primer lugar, la opción *File* nos facilita la carga de los archivos que el usuario solicite. El comportamiento es similar al dispositivo GaitWatch. Nos aparece la ventana de selección de archivos (figura 5.60) y debemos de pinchar sobre la carpeta *data_ECnsole* si queremos encontrar los datos correspondientes al dispositivo.

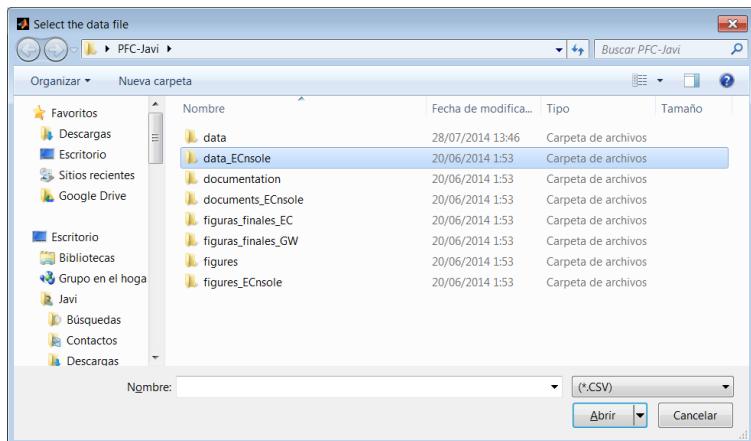


Figura 5.60: Pantalla: Select the data file.

ECnsole también hace uso de los tres tipos de sensores iniciales: un acelerómetro, un giroscopio y un magnetómetro.

En el sistema anterior, una vez que finalizaba la carga nos aparecían tres ventanas para corregir los errores del magnetómetro antes de pasar a la siguiente sección. En este caso concreto no ocurre así.

El hecho más significativo es la abolición de la ventana Select the Unit. En el GaitWatch teníamos hasta 7 segmentos corporales, recordemos: espinilla izquierda, espinilla derecha, muslo izquierdo, muslo derecho, brazo izquierdo, brazo derecho y tronco. Para ECnsole el número de segmentos se reduce considerablemente a 2 (pie izquierdo y pie derecho), no siendo necesaria la implementación de la pantalla. Todas las señales existentes aparecerán directamente en la pantalla de selección de señales.

5.2.6.1. Select the data you wish to be displayed

Otra vez nos aparecen las 3 listas que contienen las señales recogidas por los distintos sensores iniciales (Raw data, Calibrated y Orientation).

Nuevamente, los procesos de calibración y del cálculo de los algoritmos de orientación para las señales calibradas pasarán inadvertidos para el usuario en la transición a la ventana *Set structures of figures*.

A continuación, se muestra la ventana en la figura 5.61:

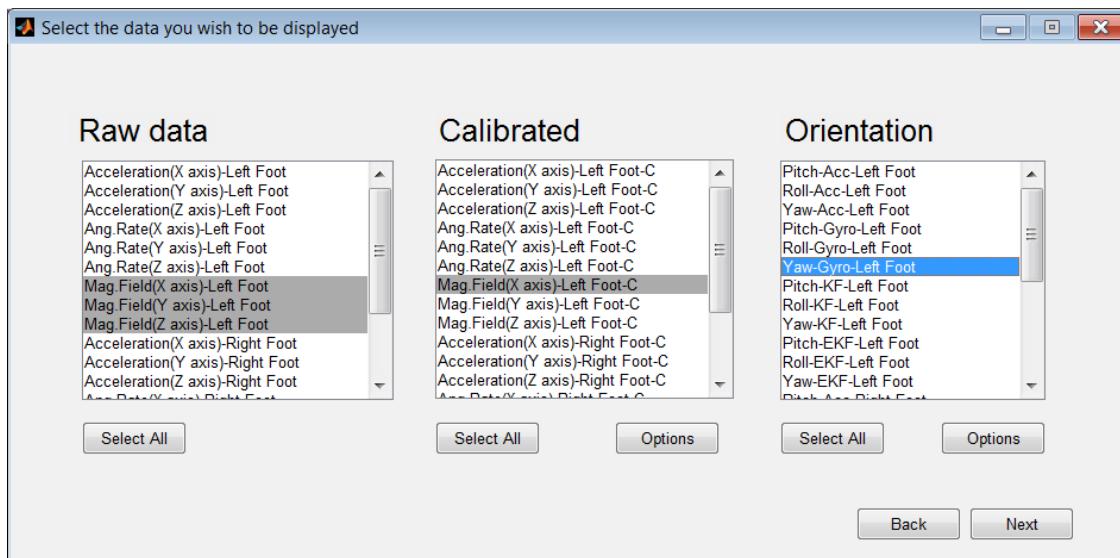


Figura 5.61: Pantalla: Select the data you wish to be displayed.

Conserva todas las funcionalidades que poseía su homóloga.

De nuevo volvemos a tener la capacidad de cambiar la configuración establecida de la calibración. Al pulsar sobre el botón *Options* se nos despliega una ventana (figura 5.62) que engloba a los tres tipos de sensores: el acelerómetro, el giróscopo y el magnetómetro.

En la tabla 5.5 se aglutan las señales asociadas a los 2 segmentos corporales medidos (pie izquierdo y pie derecho):

Tabla 5.5: Señales del sistema ECnsole

Raw data	Calibrated	Orientation
Acceleration(X axis)-Left Foot	Acceleration(X axis)-Left Foot-C	Pitch-Acc-Left Foot
Acceleration(Y axis)-Left Foot	Acceleration(Y axis)-Left Foot-C	Roll-Acc-Left Foot
Acceleration(Z axis)-Left Foot	Acceleration(Z axis)-Left Foot-C	Yaw-Acc-Left Foot
Ang.Rate(X axis)-Left Foot	Ang.Rate(X axis)-Left Foot-C	Pitch-Gyro-Left Foot
Ang.Rate(Y axis)-Left Foot	Ang.Rate(Y axis)-Left Foot-C	Roll-Gyro-Left Foot
Ang.Rate(Z axis)-Left Foot	Ang.Rate(Z axis)-Left Foot-C	Yaw-Gyro-Left Foot
Mag.Field(X axis)-Left Foot	Mag.Field(X axis)-Left Foot-C	Pitch-KF-Left Foot
Mag.Field(Y axis)-Left Foot	Mag.Field(Y axis)-Left Foot-C	Roll-KF-Left Foot
Mag.Field(Z axis)-Left Foot	Mag.Field(Z axis)-Left Foot-C	Yaw-KF-Left Foot
Acceleration(X axis)-Right Foot	Acceleration(X axis)-Right Foot-C	Pitch-EKF-Left Foot
Acceleration(Y axis)-Right Foot	Acceleration(Y axis)-Right Foot-C	Roll-EKF-Left Foot
Acceleration(Z axis)-Right Foot	Acceleration(Z axis)-Right Foot-C	Yaw-EKF-Left Foot
Ang.Rate(X axis)-Right Foot	Ang.Rate(X axis)-Right Foot-C	Pitch-Acc-Right Foot
Ang.Rate(Y axis)-Right Foot	Ang.Rate(Y axis)-Right Foot-C	Roll-Acc-Right Foot
Ang.Rate(Z axis)-Right Foot	Ang.Rate(Z axis)-Right Foot-C	Yaw-Acc-Right Foot
Mag.Field(X axis)-Right Foot	Mag.Field(X axis)-Right Foot-C	Pitch-Gyro-Right Foot
Mag.Field(Y axis)-Right Foot	Mag.Field(Y axis)-Right Foot-C	Roll-Gyro-Right Foot
Mag.Field(Z axis)-Right Foot	Mag.Field(Z axis)-Right Foot-C	Yaw-Gyro-Right Foot
		Pitch-KF-Right Foot
		Roll-KF-Right Foot
		Yaw-KF-Right Foot
		Pitch-EKF-Right Foot
		Roll-EKF-Right Foot
		Yaw-EKF-Right Foot
		Pitch-Acc-Center Trunk

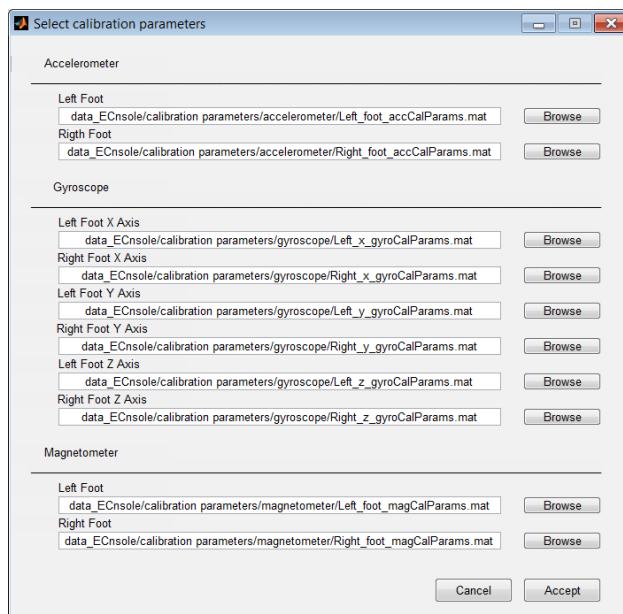


Figura 5.62: Pantalla: Select calibration parameters.

La distribución de la figura cambia respecto a su homólogo. No ha sido necesario dividir la información en pestañas, se aglutina toda en la misma.

Cuando se decide cambiar la configuración en las opciones de calibración nos aparece de nuevo una ventana de selección de archivos (figura 5.63) con la ruta especificada según el sensor que corresponda.

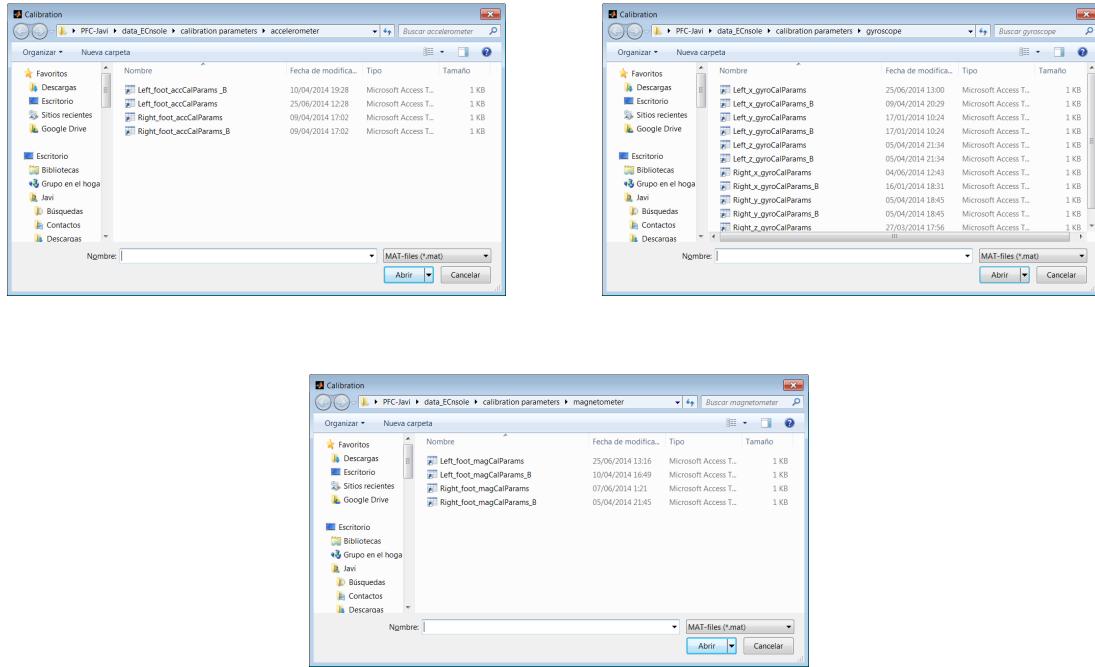


Figura 5.63: Selección de parámetros: Acelerómetro, Giroscopio y Magnetómetro.

Ahora bien, si el usuario decide cambiar la configuración para las estimación de la orientación, conviene puntualizar que tan solo algunos de los algoritmos utilizados permiten modificaciones de sus parámetros. La lista de señales contiene otras señales calculadas con algoritmos alternativos que no requieren modificaciones.

La recogida de datos de los dos segmentos corporales (pie izquierdo y pie derecho) posee características similares, por lo tanto, los algoritmos de orientación utilizados son equivalentes.

- Pie izquierdo/derecho
 - ◊ Kalman Filter
 - ◊ Extended Kalman Filter

En la figura 5.64 se muestran las opciones para la estimación de la orientación del pie izquierdo:

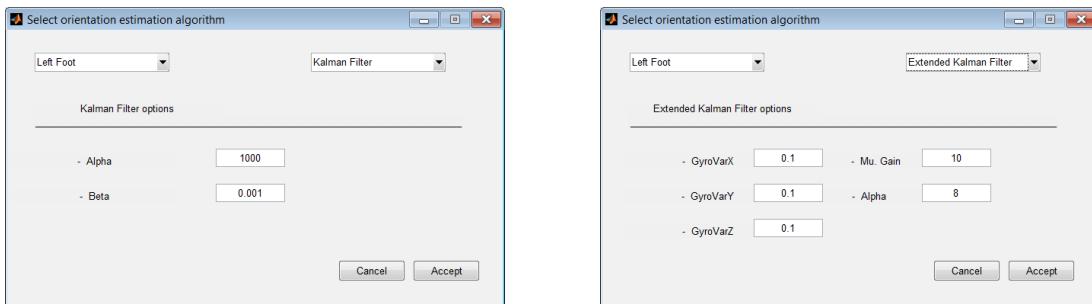


Figura 5.64: Ventana *Select orientation estimation algorithm* para el pie izquierdo.

En la figura 5.65 se muestran las opciones para la estimación de la orientación del pie derecho:

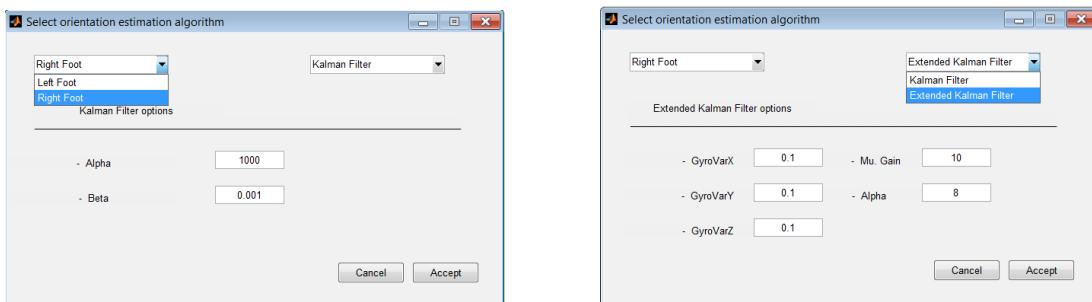


Figura 5.65: Ventana *Select orientation estimation algorithm* para el pie derecho.

Una vez finalizada la configuración de los procesos de calibración y orientación, el usuario puede dar paso a la ventana siguiente denominada *Set structure of figures*. En este caso, no existe la necesidad de creación de una barra de progreso. El tiempo de procesamiento de dichos procesos es ínfimo e inperceptible para el usuario.

Sin embargo, si se decide volver a la pantalla principal y anteriormente ya nos habíamos situado sobre la ventana *Set structure of figures*, nos aparecerá un mensaje de aviso (figura 5.66). Nos advierte de que toda la configuración establecida hasta el momento se puede perder. Difiere de la ubicación del mensaje en el dispositivo GaitWatch que se encontraba en la ventana *Select the unit*. En este caso, era la ventana siguiente a la pantalla principal.

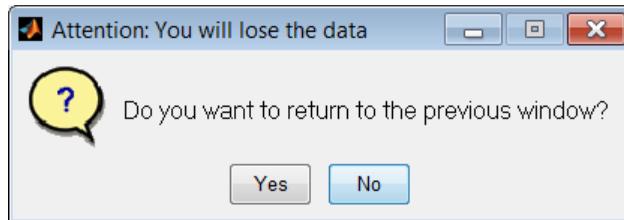


Figura 5.66: Aviso: Pérdida de datos.

5.2.6.2. Set structure of figures

La pantalla de configuración de las figuras posee exactamente las mismas características que la del sistema GaitWatch. En consecuencia, se requiere volver a realizar una descripción exhaustiva de la misma. Pero sí vamos a poner un ejemplo gráfico que:

- Aparezcan varias figuras.
- Se aprecie la eliminación de una ellas.
- Se pueda visualizar una de las configuraciones existentes.

Para ello, la configuración se muestra a través de la figura 5.67:

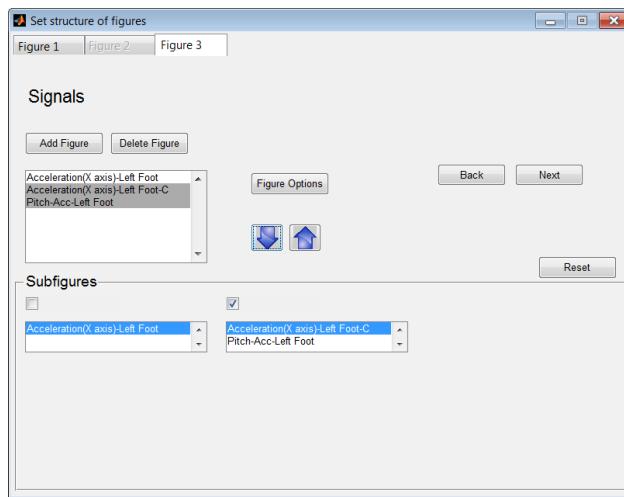


Figura 5.67: Pantalla principal: Set structure of figures.

La representación gráfica de la configuración establecida se muestra en la figura 5.68:

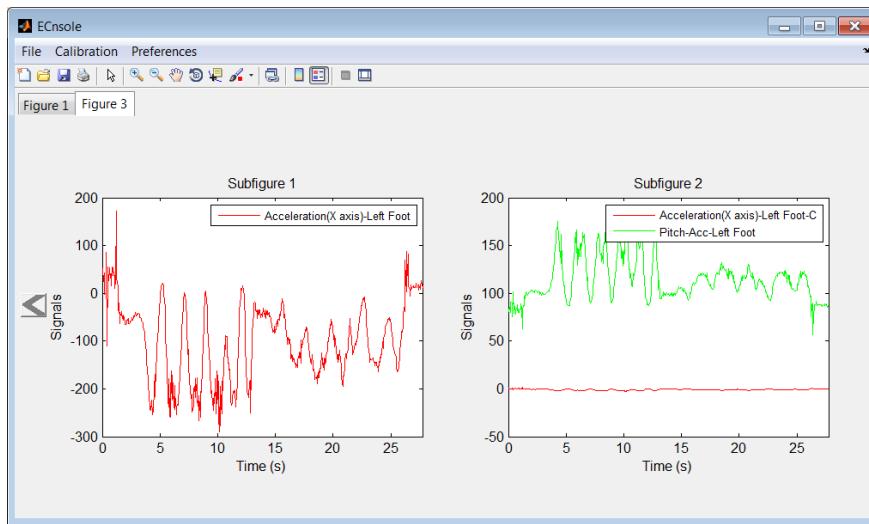


Figura 5.68: Aviso: Pantalla principal: Representación gráfica.

Además, la representación gráfica de las señales presenta exactamente las mismas funcionalidades explicadas anteriormente.

5.2.7. Calibration (ECnsole)

El fundamento teórico de la calibración de los sensores se detalló anteriormente (ver sección 5.2.3). Tan sólo tener presente que el sistema de monitorización del movimiento ECnsole carece de acelerómetro uniaxial al desplegar menú del *acelerómetro* (figura 5.69).



Figura 5.69: Pantalla principal: Representación gráfica.

La calibración correcta de los parámetros a través del GaitWatch no se pudo realizar como consecuencia de no tener el dispositivo. Sin embargo, los datos recogidos con el sistema ECnsole si nos permiten un cálculo eficiente de los parámetros de calibración y mostrar algunas de las gráficas resultantes más representativas a modo de ejemplo. La representación de los resultados se lleva a cabo en la pantalla principal, tal y como hicimos con las señales recogidas por el dispositivo.

La figura 5.70 muestra la aceleración original de períodos casi-estáticos. Los datos son representados en 3D y en los 3 posibles planos de proyección (XY, XZ, YZ).

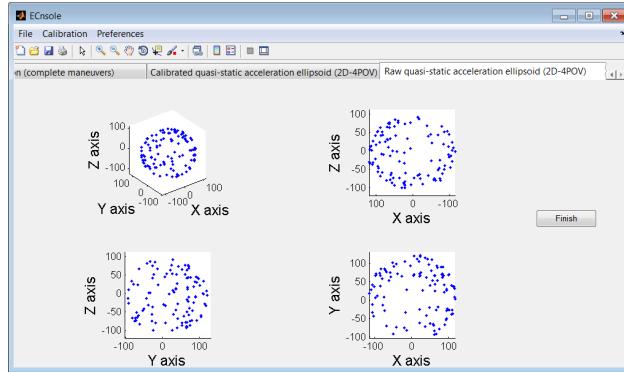


Figura 5.70: Acelerómetro: Cálculo de los parámetros de calibración. Aceleración original.

La figura 5.71 muestra la aceleración calibrada de períodos casi-estáticos. Los datos son representados en 3D y en los 3 posibles planos de proyección (XY, XZ, YZ).

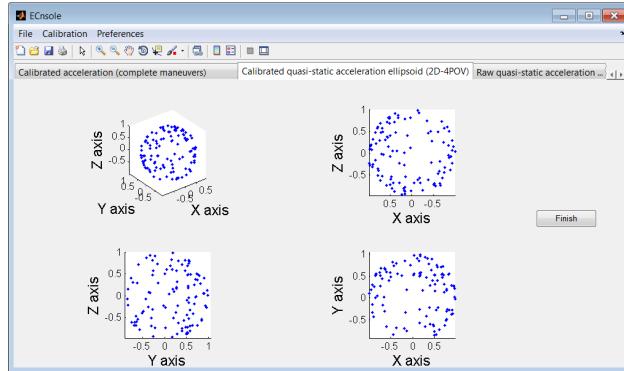


Figura 5.71: Acelerómetro: Cálculo de los parámetros de calibración. Aceleración calibrada.

El cálculo de parámetros para el acelerómetro está basado en el algoritmo de ajuste elipsoidal propuesto por Camps et al.

La figura 5.72 representa las rotaciones calculadas del giroscopio. Si el cálculo de los parámetros (bias y factor de escala) es exitoso entonces el cálculo de las rotaciones tiene que ser próximo a 180° (o -180° si la rotación del giroscopio fuese invertida).

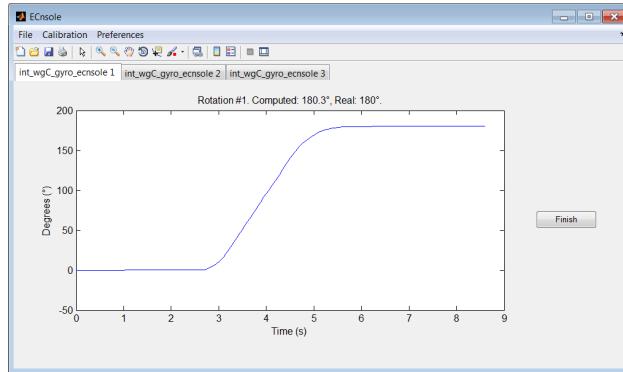


Figura 5.72: Giroscopio: Cálculo de los parámetros de calibración. Aceleración calibrada.

El valor de rotación de la figura es de 180.3° , por lo tanto, el cálculo de parámetros se puede considerar óptimo. Cabe destacar que los puntos elegidos son los correspondientes al tramo de rotación positiva del giróscopo.

La figura 5.73 muestra los datos originales del magnetómetro desde 4 puntos de vista (3D y los tres planos de proyección):

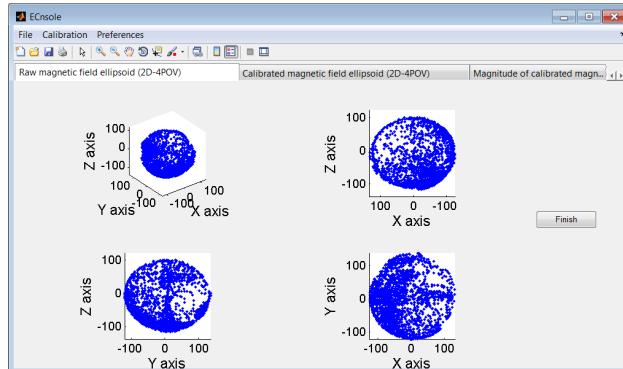


Figura 5.73: Magnetómetro: Cálculo de los parámetros de calibración. Campo magnético original.

La figura 5.74 muestra los datos calibrados del magnetómetro desde 4 puntos de vista (3D y los tres planos de proyección):

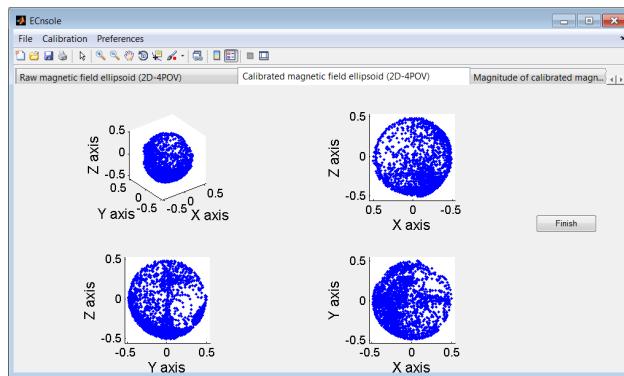


Figura 5.74: Magnetómetro: Cálculo de los parámetros de calibración. Campo magnético calibrado.

El cálculo de parámetros para el magnetómetro está basado en el algoritmo de ajuste elipsoidal propuesto por Camps et al.

Por último, destacamos que la organización de la representación gráfica de los resultados, nuevamente, se llevan a cabo mediante pestañas. Cada pestaña recoge una de las gráficas del proceso de cálculo de parámetros.

Se escogió este modo de implementación por sus múltiples ventajas. Las alternativas que se barajaban se citan a continuación:

- Representar todas las gráficas juntas en la misma pantalla. Sería difícil discernir los resultados.
- Resfrescar la pantalla cada vez que el usuario solicitase una gráfica nueva.

Ninguna de las opciones resultaba tan eficiente, ni permitía ver los datos con tanta claridad.

5.2.8. Preferences (ECnsole)

La ventana *Preferences* (figura 5.75) es común a los sistemas *GaitWatch Manager* y *ECnsole*. Su utilidad es la de dar paso de un dispositivo a otro. La apariencia gráfica ya se mostró en la misma sección correspondiente al GaitWatch.



Figura 5.75: Select device.

Conclusiones y trabajo futuro

6.1. Conclusiones

La aplicación desarrollada cumple con los requisitos exigidos inicialmente. El software permite la configuración y visualización de los datos recogidos por los sistemas de medida de parámetros GaitWatch y ECnsole.

Inicialmente, se llevó a cabo una revisión del estado del arte de los dispositivos existentes en la actualidad en el mercado. Se ha dejado constancia de la existencia de dos clases de dispositivos de monitorización de la marcha principalmente, los sistemas basados en cámaras y los sistemas basados en sensores iniciales. Algunos de estos dispositivos como el Kinect de la Xbox nos resultan cotidianos, ya que en la actualidad los videojuegos forman parte de nuestras vidas. Sin embargo, se enumeran otros para los cuales se requirió una búsqueda más exhaustiva dirigidos a personas más específicas.

Posteriormente se ha procedido a estudiar los fundamentos teóricos de los sistemas de monitorización de la marcha utilizados. De hecho, se describen los principios básicos por los que se rigen los diferentes tipos de sensores iniciales. Además se dedican varios apartados para dar a conocer los detalles del hardware de los sistemas GaitWatch y ECnsole; formando parte del análisis de los sistemas de monitorización de la marcha. En este apartado se establecen los requisitos necesarios del software a implementar en función del hardware empleado. En nuestro caso particular, los dispositivos se caracterizan por una unidad de procesamiento central y un conjunto de unidades periféricas. Los dos sistemas difieren respecto a la zona de colocación de las unidades de medición en las diferentes pruebas efectuadas. Para el sistema GaitWatch, se han colocado en los muslos, espinillas, brazos y tronco; sin embargo para ECnsole se han situado en el pie derecho y en el pie izquierdo. Como consecuencia, el número de señales generadas en el sistema GaitWatch es superior respecto al sistema ECnsole y los algoritmos de estimación de la orientación aplicados varían en concordancia a la zona del

cuerpo donde se coloca la unidad de medición.

En el siguiente punto se ha continuado con el desarrollo del software, el cual es el bloque que compone el grueso del trabajo de este proyecto. Consiste en la creación del programa que se ajuste a los objetivos fijados inicialmente. La aplicación está constituida por una serie de ventanas que adoptan una funcionalidad específica. A continuación se enumeran las más destacables: carga de archivos de datos, selección de señales, opciones de calibración y orientación, configuración de la estructura de las figuras, muestra de los resultados, cálculo de parámetros de calibración y cambio de dispositivo.

Para finalizar, se han integrado en nuestro programa los algoritmos, ya desarrollados, de calibración y cálculo de la orientación. Es importante resaltar que la aplicación desarrollada puede ser operada por personal no especializado y sin conocimiento de los algoritmos que se ejecutan de manera transparente.

6.2. Trabajo futuro

La aplicación desarrollada puede ser ampliada para añadirle un mayor número de funcionalidades. De hecho, podríamos crear una aplicación equivalente aumentando el número de dispositivos o incluso capaz de adaptarse a las peculiaridades de cualquier sistema.

Ante una nueva IMU (unidad de medición inercial) el programa facilitaría caracterizar el formato de trama, el número y tipo de sensores, etc. Una posible solución es crear una pequeña interfaz gráfica para poner en marcha el proceso de transmisión de información entre distintos sistemas (la IMU y el ordenador donde se ejecuta el programa). Entonces, ¿cómo lograr que los sistemas, siendo diferentes uno del otro, puedan transmitirse datos con la información requerida acerca de los sensores? Sencillo, gracias a los formatos de representación de información. Es aquí donde entran estándares como XML, JSON y YAML. Estos formatos se caracterizan por ser robustos, transmitir y compartir información bastante compleja entre sistemas diferentes, con estructuras jerárquicas y atributos variables pero a su vez fáciles de leer por un ser humano.

Adicionalmente, se podría incluir la posibilidad de añadir nuevos algoritmos de calibración y cálculo de la orientación en forma de plugins. Los algoritmos vendría recogidos en un archivo.m (archivo de texto donde se introducen comandos y/o instrucciones de Matlab). Para conseguir una mayor adaptación respecto al dispositivo, se crearía una pantalla donde fuese posible introducir código y seleccionar las entradas del algoritmo.

Otra posibilidad, referente a la caracterización del *análisis de la marcha* estudiado en el capítulo 2; sería integrar algoritmos de cálculo de parámetros de la marcha, tales como longitud de zancada, cadencia, distancia recorrida, etc. Así, de esta manera servirnos de las distintas señales obtenidas y transformarlas en datos que nos permitan caracterizar más exhaustivamente la marcha humana.

Además, como se ha visto en otros trabajos, pueden diseñarse otros tipos de algoritmos diferentes a los estudiados, profundizando sobre los basados en cuaterniones o los clasificadores. No suponen un gran salto de calidad aunque su desarrollo puede ser complejo. Asimismo, no se corresponde con una mejora de la aplicación visual pero sí del sistema en conjunto.

Bibliografía

- [1] Definición de la marcha humana. <http://goo.gl/yEYtYI>.
- [2] Aplicaciones. <http://goo.gl/tiuJm6>.
- [3] Clasificación de sistemas hmc. <http://goo.gl/D2OqJM>.
- [4] Sensor inercial. <http://goo.gl/mJrgaR>.
- [5] 3dm-gx4-45. <http://goo.gl/Btv9gI>.
- [6] Xsens mvn. <http://goo.gl/Bfu2Gp>.
- [7] Sparkfun. <http://goo.gl/gdkAn>.
- [8] Vicon. <http://goo.gl/Wzl43x>.
- [9] Qualisys. <http://goo.gl/7naTK2>.
- [10] Imagen kinect. <http://goo.gl/udkT8x>.
- [11] Acelerómetro mecánico. <http://goo.gl/8ZuwBD>.
- [12] Acelerómetro piezoeléctrico. <http://goo.gl/licOq8>.
- [13] Acelerómetro de efecto hall. <http://goo.gl/4887Ri>.
- [14] Acelerómetro de condensador. <http://goo.gl/V1i9UJ>.
- [15] Finn E. J Alonso M. *Física. Mecánica*. 1970.
- [16] Giróscopo. <http://goo.gl/t3wkzn>.
- [17] Giróscopo mems. <http://goo.gl/CVGMEK>.
- [18] Magnetómetro de precesión de protones. <http://goo.gl/ald4dS>.

- [19] Magnetómetro cuántico. <http://goo.gl/Y8whjE>.
- [20] Magnetómetros vectoriales. <http://goo.gl/t9Zp6u>.
- [21] Magnetómetro mems. <http://goo.gl/mmjLfA>.
- [22] Sistema de coordenadas (n-frame). <http://goo.gl/g667oR>.
- [23] Sistema de coordenadas (b-frame). <http://goo.gl/XrDcp6>.
- [24] Douglas García y Victor González. Animación ii.
- [25] A. Olivares. *"Signal Processing of Magnetic and Inertial Sensor's Signals applied to Human Body Motion Monitoring"*. 2013.
- [26] Biomecánica. <http://goo.gl/3b5jHU>.
- [27] Historia. <http://goo.gl/WnQWbP>.
- [28] Sistema acústico. <http://goo.gl/U7cBMD>.
- [29] Imu. <http://goo.gl/NEpmJ0>.
- [30] Kinect. <http://goo.gl/rIhpqM>.
- [31] Sensor medidor de aceleración. <http://goo.gl/93uoS>.
- [32] Tipos de acelerómetro. <http://goo.gl/gkGd3I>.
- [33] Mems. <http://goo.gl/IISqdC>.
- [34] Magnetómetro. <http://goo.gl/UBMd3X>.
- [35] Definición magnetómetro mems. <http://goo.gl/YTq4XQ>.
- [36] S. Harasse F. Camps and A. Monin. "numerical calibration for 3-axis accelerometers and magnetometers". In *IEEE International Conference on Electro/Information Technology*, pages 217–221. EIT'09, 2009.
- [37] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.
- [38] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [39] National Geophysical Data Center. NGDC geomagnetic calculators-magnetic field. <http://www.ngdc.noaa.gov/geomag-web/#igrfwmm>. [Accessed January 14th,2013].
- [40] Matlab gui. <http://goo.gl/8kYHDn>.
- [41] Sumandor en matlab gui. <http://goo.gl/Ywz14z>.
- [42] Ejemplo de gui sin guide. <http://goo.gl/LzSWIk>.