



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño de un Sistema
Inalámbrico de Goniometría
para Análisis de Marcha
Humana**

TESIS

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

José Arturo Ruiz Díaz

DIRECTORA DE TESIS

Dra. María del Pilar Corona Lira



Ciudad Universitaria, Cd. Mx., Ingresada 2018

Dedicatorias

Es un hecho que nada de lo que ocurre en nuestras vidas es arbitrario. Cada suceso que experimentamos, persona que conocemos, lugar que visitamos y decisión que tomamos tienen una razón para presentarse en nuestro camino. La mayoría de las veces no somos capaces de identificarlo en un principio o por simple desconocimiento a lo que depara el futuro, lo ignoramos. Sin embargo, una vez que pasa el tiempo suficiente y somos capaces de reflexionar, con retrospectiva volteamos la mirada hacia el pasado y nos hacemos conscientes de lo que nos llevó hasta el punto en el que estamos situados. Finalmente, somos capaces de visualizar lo que cada elemento deja a su paso y permanencia en nosotros mismos.

Muchas personas han formado parte de mi historia y ciertamente son también muchas las que han contribuido a mi vida de una forma significativa. El trabajo que presento a continuación es fruto de mucha dedicación y esfuerzo, que con la guía, apoyo y cariño de aquellas personas que han causado un verdadero impacto en mi vida, pudo convertirse en una realidad.

A mi padre, Cutberto Ruiz, quien desde pequeño me inculcó una grande pasión por la Ciencia y la tecnología, quien me ha mostrado los frutos de la perseverancia, el temple, el pensamiento crítico, y la entrega y hasta el día de hoy continúa guiándome en mi camino hacia convertirme en un profesionista exitoso y una persona de valor. A mi madre, Ana Gabriela Díaz, quien ha sido un apoyo incondicional en todos y cada uno de mis proyectos de vida, quien me mostró el gran valor de mis compromisos, quien grabó en mí un enorme sentido de la responsabilidad y de quien aprendí a sensibilizarme por las necesidades de las personas que amo. A mi hermano Antonio Ruiz, ciertamente mi mejor amigo y la persona con quien he compartido gran parte de mis gustos y pasiones, de quien aprendí que uno mismo forja su propio destino, mi auténtica *Rosa de los Vientos*. Me siento orgulloso de la familia en la que nací, los amo profundamente y comparto con ustedes este logro.

A todos aquellos miembros de mi familia que de alguna forma u otra, contribuyeron en mi proceso de alcanzar este logro y siempre mostraron interés en el trabajo que aquí se plasma: Francisco Lojero, Belén Vivar, la familia Díaz Cedillo, la Gran Familia Ruiz y mi primo Alonso.

A Astrid Argueta, mi persona, quien me impulsó a dedicar mente y corazón a la culminación de este trabajo, quien me ha inspirado siempre a superarme, de quien aprendí que siempre existe la posibilidad de alcanzar los sueños y metas de uno mismo y junto con quien me aventuré a conocer el amor verdadero. Con su cariño y apoyo incondicional y el de sus padres José Argueta y Magdalena Ramos, fueron una pieza fundamental en este proceso.

A mis amigos, quienes me han acompañado en este y muchos otros procesos de mi vida. Todos forman parte de distintos círculos de mi vida, y de ellos he aprendido el verdadero sentido de la amistad y el compañerismo: Miguel Zavala, Rafael Nava, David Negrete, Kazu Ueji, Erick Maravillas, Héctor Don Pablo, Arturo Yañez, Aaron Posadas, Daniel López, Ricardo y Alejandro Ordoñez y todos quienes han dejado huella en mí.

A todas aquellas personas que de manera directa e indirecta participaron en la culminación de este proyecto.

A mis compañeros de proyectos del Centro de Ingeniería Avanzada, quienes fueron una fuente de inspiración y ejemplo a seguir durante el desarrollo de este proyecto. Al M.I. Carlos Canchola.

A mis profesores de la Facultad de Ingeniería, quienes me proporcionaron las herramientas necesarias para desempeñarme adecuadamente como Ingeniero Mecatrónico, de quienes en su mayoría aprendí mucho más de lo que dictan los temarios de mi plan curricular. En especial agradezco a mi directora de tesis y a los miembros de mi sínodo, ya que orgullosamente todos fueron mis profesores durante la carrera: Dra. María del Pilar Corona, Dr. Alejandro Ramírez, M.I. Itzel Flores, M.A. Yair Bautista, Ing. Liliana Flores.

A mi universidad, mi *alma máter*, la prestigiosa Universidad Nacional Autónoma de México y a la Facultad de Ingeniería, por ser la sede madre de muchos de mis logros e impulsora por excelencia de mis más grandes sueños y pasiones.

Este logro también es de ustedes. Gracias por todo su apoyo, enseñanzas y cariño, por formar parte de este momento y por todos los que en el futuro nos aguarden.

«Por mi raza hablará el espíritu».

Índice general

1. Introducción	7
2. Objetivos	9
2.1. Objetivo General	9
2.2. Objetivos Particulares	9
3. Antecedentes	10
3.1. Goniometría	10
3.2. Sistemas de Instrumentación	10
3.2.1. Sistema de Captura de Movimiento	10
3.2.2. Clasificación de los Sistemas de Captura de Movimiento [2]	11
3.2.3. Estado del Arte: Sistemas de Captura de Movimiento con Sensores IMU y MARG	13
3.3. Sensores Inerciales	15
3.3.1. Tecnología MEMS	15
3.3.2. El Acelerómetro	16
3.3.3. El Giroscopio	17
3.3.4. El Magnetómetro	18
3.3.5. Sensores MEMS: MARG vs IMU	19
3.4. Representación de la Rotación en el Espacio	19
3.4.1. Notación	19
3.4.2. Sistema de coordenadas	20
3.4.3. Tramas y Matrices de Rotación	20
Conclusiones respecto a la Representación de la Orientación por Matrices de Rotación	22
Gimbal Lock	22
3.4.4. Cuaterniones	23
Relación entre el espacio de Cuaterniones y los Vectores \mathbb{R}^3	24
Cuaterniones Unitarios	24
3.4.5. Representación de la orientación por medio de Cuaterniones	25
Producto entre cuaterniones reales [23]	25
Producto entre un cuaternión real y un cuaternión vectorial [23]	25
Producto entre dos cuaterniones vectoriales [23]	25
Producto entre un cuaternión unitario y un cuaternión vectorial perpen- dicular [23]	25
Rotación de un vector arbitrario mediante el producto de cuaterniones unitarios [23]	26
Relación entre los Cuaterniones y las Matrices de Rotación	28
Relación entre los Cuaterniones y los Ángulos de Euler	28
3.5. Algoritmos de Orientación Espacial	29

3.5.1.	Problemas de Medición en los Sensores	29
	Acelerómetro	29
	Magnetómetros	30
	Giroscopio	32
3.5.2.	Filtro de Madgwick [19]	33
3.6.	Biomecánica	34
3.6.1.	Planos y Ejes Anatómicos	35
3.6.2.	Vectores Biomecánicos [3]	36
3.6.3.	Biomecánica de Miembro inferior	37
3.6.4.	Movimientos en Articulaciones de Miembro Inferior [31]	40
	Articulaciones	40
	Movimientos Angulares de Articulaciones Sinoviales de Miembro Inferior	41
3.6.5.	Análisis de la Marcha Humana	43
	Componentes Análisis Clínico de la Marcha [30]	44
4.	Metodología	47
4.1.	Planteamiento del problema	47
4.1.1.	Pregunta de investigación	47
4.1.2.	Justificación	48
4.2.	Propuesta de solución: Sistema de Goniometría	
	Propuesto	48
4.3.	Diseño a Nivel Sistema	48
4.3.1.	Sistemas funcionales	48
	Transmisor	49
	Receptor	50
4.3.2.	Generación y selección de conceptos	51
	Transmisor	51
	Receptor	54
4.4.	Diseño de detalle	55
4.4.1.	Selección de componentes	55
	Sensor inercial tipo MARG	55
	Microcontrolador	57
	<i>Transceptor</i> por Radiofrecuencia	58
	Interfaz de usuario	59
	Suministro de energía del transmisor	60
4.4.2.	Transmisión de Datos	61
	Vinculación	63
	Transmisión - Recepción de datos	67
	Datos a transmitir	68
4.4.3.	Administración de Energía	72
	Microcontrolador	72
	MPU-9250 (Sensor MARG) y nRF42L01 (<i>Transceptor</i>)	74
	Carga de la batería	74
4.4.4.	Interfaces de Usuario	75
	Receptor	75
	Transmisor	78
4.4.5.	Programación de transmisores	79
4.5.	Prototipo	80

5. Protocolo de Validación	82
5.1. Protocolo de Pruebas	83
5.1.1. Objetivo	83
5.1.2. Lugar de pruebas	83
5.1.3. Fecha	83
5.1.4. Instrumentos de medición	83
5.1.5. Módulos	84
Antropometría	84
Prueba de marcha	86
6. Resultados	88
7. Análisis de Resultados	113
8. Conclusiones	129
9. Trabajo Futuro	131
10. Anexos	133
10.1. Representación de la orientación de un cuerpo en el espacio mediante matrices de rotación	133
Ángulos fijos X - Y - Z	134
Ángulos d Euler Z - Y - X	135
Ángulos de Euler Z - Y - Z	136
Ángulo - Eje Equivalente	137
10.2. Propiedades algebraicas de los cuaterniones	138
10.3. Filtro de Madgwick[19]	139
Orientación a partir de la velocidad angular	139
Orientación a partir de la observación de vectores	140
Algoritmo de fusión de filtros	143
Compensación de distorsión magnética	144
Compensación del <i>drift</i> del Giroscopio	145
Ganancias del Filtro [19]	145
10.4. Código del Receptor	146
10.4.1. Receptor de datos sin procesar	146
10.4.2. Receptor de cuaterniones	149
10.5. Códigos del Transmisor	153
10.5.1. Transmisor de datos sin procesar	153
10.5.2. Transmisor de cuaterniones	161
10.6. Código de Interfaz de Usuario y Sistema de Conversión de Datos del Receptor	171
10.7. Código de Post-procesamiento	175
10.8. Carta de Consentimiento Informado	179
10.9. Datos antropométricos recolectados durante las pruebas	183

Capítulo 1

Introducción

En el presente trabajo se desarrolla el proceso de análisis, diseño y construcción de un prototipo de un sistema inalámbrico de goniometría; con el objetivo de capturar el desplazamiento angular del movimiento de flexo-extensión en la articulación de rodilla durante el ciclo de marcha humana.

En la actualidad existe una gran variedad de sistemas de captura de movimiento dedicados al análisis de los movimientos del cuerpo humano; y dichos sistemas de instrumentación, implementan distintos tipos de tecnologías para lograr su propósito. Una de estas tecnologías es la integración de acelerómetros, magnetómetros y giroscopios dentro de una sola unidad de medición inercial conocida como sensor tipo *MARG* (por sus siglas en inglés: *Magnetic Angular Rate Gravity*). Para este tipo de unidades se toman como referencias el campo magnético y el campo gravitatorio terrestres, por lo que ofrece la oportunidad de medir, capturar y/o analizar los movimientos del cuerpo de una persona en cualquier lugar (con la debida configuración inicial y calibración). Son sensores de tamaño reducido, de bajo consumo energético, configurables, y ofrecen una considerable cantidad de ventajas más con respecto a los sistemas actualmente en el mercado.

Reduciendo el enfoque de análisis del movimiento humano al ciclo de marcha, una de las variables biomecánicas de interés es el ángulo de flexo-extensión de rodilla en ámbitos como la rehabilitación clínica o mejora del desempeño deportivo. Por tal motivo es de suma importancia el papel de la instrumentación implantada a los sujetos de estudio en cuestiones como: comodidad, niveles de invasión y perturbación, certeza y velocidad de muestreo. En el presente proyecto se propone el diseño de un sistema de goniometría que permita mejorar cada uno de los puntos antes mencionados abordando las ventajas de los sensores *MARG* incorporados a una red inalámbrica de sensores.

Considerando lo anterior se analizan los requerimientos del sistema, se definen las especificaciones necesarias, y se divide el proyecto en los sistemas funcionales que lo integran. Posteriormente, se realiza una búsqueda de las posibles soluciones tanto para el algoritmo de orientación espacial como para los componentes necesarios de cada sistema funcional; y por consiguiente que se cumplan con las especificaciones definidas. De manera subsecuente, se realiza el diseño de detalle de software y hardware para proceder así con la construcción de un prototipo.

Finalmente, al ser una propuesta de solución, se planea un protocolo como opción de validación del sistema diseñado, implementando un sistema comercial de la misma naturaleza. Se comparan, por medio de un análisis estadístico de correlación directa, las señales obtenidas mediante el sistema propuesto con respecto del comercial. De esta manera, se determina de forma cuantitativa la funcionalidad del sistema desarrollado en congruencia con las especifica-

ciones definidas y los objetivos del proyecto; y se identifican áreas de oportunidad para posibles mejoras o correcciones a futuro.

Capítulo 2

Objetivos

2.1. Objetivo General

Diseñar y construir un prototipo funcional de un sistema de goniometría inalámbrica que permita la medición de desplazamientos angulares de la articulación de rodilla involucrados en la marcha humana.

2.2. Objetivos Particulares

- Medir los desplazamientos angulares entre el fémur y el segmento tibia-peroné en movimiento de flexo-extensión de rodilla durante la marcha.
- Diseñar el sistema de forma que se obtenga una velocidad de muestreo de mínimo 25 [Hz] para la orientación en el espacio de cada sensor (transmisor).
- Determinar la validez de los datos obtenidos por el sistema propuesto comparándolos con datos recabados por un sistema de medición inercial comercial.
- Diseñar el sistema de tal manera que permita la sincronización de los sensores (transmisores) con el receptor de manera sencilla y modular.

Capítulo 3

Antecedentes

3.1. Goniometría

La Goniometría es una rama de la Metrología cuyo principal objetivo es cuantificación del desplazamiento angular. El término nace posterior a los instrumentos que en sí son los responsables de realizar las mediciones pertinentes: **los goniómetros**. Si se desea conocer el ángulo existente entre dos cuerpos rígidos, para la mayoría de los goniómetros no es necesario conocer la orientación ni la posición en el espacio de dichos cuerpos. Los goniómetros, dependiendo de la tecnología implementada, realizan una medición directa del ángulo en sí. Sin embargo, es posible conocer dicho ángulo al conocer la orientación de cada uno de los cuerpos en el espacio y la diferencia entre ambas.

La Goniometría tiene aplicaciones diversas. Es posible encontrar goniómetros en campos como la Astronomía, la Mineralogía, las Matemáticas así como las Ciencias de la Salud y la Biomecánica.

3.2. Sistemas de Instrumentación

3.2.1. Sistema de Captura de Movimiento

En ciertas disciplinas de la Ingeniería (como la Robótica, Aeronáutica, realidad virtual y aumentada o Ingeniería Biomédica), se requiere conocer **la orientación y posición de un cuerpo rígido en el espacio**. Ya sea para capturar el movimiento de dicho cuerpo, analizarlo o para controlarlo, se han implementado distintas técnicas de sensado a lo largo del tiempo. Existen sistemas especializados para la obtención de la posición de un objeto en el espacio así como otros enfocados a sensar la orientación del mismo y finalmente aquellos que son capaces de realizar ambas tareas en paralelo. Para la medición de variables involucradas en la marcha humana, un sistema de captura de movimiento se compone principalmente de cinco elementos [1]:

1. Sensores, marcas y/o fuentes colocadas sobre el sujeto de estudio
2. Fuentes, marcas y/o sensores externos al sujeto de estudio
3. Interfaz electrónica sistema computadora
4. Computadora
5. Modelo computacional y datos

Dependiendo de la tecnología empleada así como la configuración del sistema en sí, alguno de los elementos antes mostrados podría ser omitido. Sin embargo, para sensar el movimiento de una serie de objetos articulados (así como las extremidades del cuerpo humano) siempre se realiza con respecto a un sistema de referencia ya sea externo al objeto o incorporado a una sección del mismo.

3.2.2. Clasificación de los Sistemas de Captura de Movimiento [2]

Una clasificación de los sistemas de captura de movimiento, tomando en consideración el principio anteriormente mencionado, fue propuesta por Axel Mulder en 1994. A diferencia de una clasificación basada en el principio de funcionamiento de los elementos del sistema (e.g. óptico, electrónico, mecánico, etc.), dicha clasificación agrupa a todos aquellos sistemas de medición del movimiento en función de la disposición de los sensores y fuentes antes mencionados. Los grupos son:

- **Medición Interna (*inside-in*):** en este caso, tanto el sensor como la fuente se encuentran posicionados sobre el sujeto de estudio. Son mayormente implantados en aquellos sistemas que miden la posición relativa de los eslabones de un tren cinemático, por lo que el rango de medición se considera reducido.

Un ejemplo de este tipo de sistemas son aquellos con **goniómetros y torsiómetros** incorporados [3]. Los goniómetros son instrumentos que cuantifican el desplazamiento angular entre dos eslabones unidos a las terminales del dispositivo. Actualmente en el mercado existen goniómetros capaces de medir uno o dos planos de movimiento.

Otro ejemplo de los sistemas *inside-in*, son aquellos que incorporan componentes **piezoresistivos**. Aprovechando las propiedades de la tela piezoresistiva se reconstruye el movimiento [4] de las juntas de una cadena cinemática al sensar el cambio de la resistencia eléctrica en puntos estratégicos del componente. Una ventaja de este tipo de sensores es la notable disminución de estructuras mecánicas y eléctricas en la zona de la articulación (en ocasiones, la zona de interés).

- **Medición del exterior al interior (*outside-in*):** para esta configuración en particular, los sensores se encuentran fijos en el exterior, mientras detectan fuentes colocadas sobre el cuerpo. Dichas fuentes pueden ser activas (emisores) o pasivas (reflectores). Los **sistemas ópticos** son los que cubren la mayoría de los dispositivos desarrollados dentro de esta categoría, principalmente los que implementan **visión** en el procesamiento de las señales adquiridas. Un conjunto de cámaras instaladas en un área específica captan las marcas posicionadas en el cuerpo y tras procesar la información se estima la posición de cada una. Una ventaja considerable es la precisión que ofrece, además de la capacidad de realizar tareas de sensado prolongadas. Mas, cabe resaltar que las mediciones se deben confinar al área de visión de las cámaras (tomando en cuenta que la instalación de cada debe ser precisa y calibrada), además que el sistema en sí es sensible a la oclusión (no visibilidad de alguna marca por la mera presencia del cuerpo de estudio).
- **Medición del interior al exterior (*inside-out*):** en esta configuración, los sensores se fijan sobre las partes del cuerpo a estudiar, mientras que existe una fuente exterior que genera una señal perceptible por los sensores unidos al cuerpo. En esta clasificación, los sistemas proveen información sobre las posiciones de los sensores relativas al ambiente; son mayormente utilizados para sensar regiones más amplias del cuerpo, o en su defecto, su totalidad. La desventaja de estos sistemas, se presenta como la limitación en el espacio de trabajo por cuestiones de ruido ambiental, zona de efectividad, etc.

Los sistemas de captura **electromagnéticos** son un ejemplo de esta clasificación. Un transmisor genera un campo electromagnético de baja frecuencia (ya sea mediante una señal sinusoidal o un tren de pulsos cuadrados) de tal forma que los receptores, colocados sobre el sujeto de estudio, detectan y transmiten la información a una unidad electrónica de control donde se filtra y amplifica. Por lo cual, al enviar la información a una computadora, se procesa y se obtiene la posición de los sensores, así como su orientación [3]. Sin embargo, estos instrumentos son muy sensibles al ruido ambiental ocasionado principalmente dentro de estructuras metálicas por cuestión de las corrientes parásitas inducidas en la estructura por medio de la fuente electromagnética.

Otro ejemplo de este tipo de sistemas, son los **sistemas micro-electromecánicos** (**MEMS**, por sus siglas en inglés: *Microelectromechanical Systems*). Para estos sistemas, son dos las principales configuraciones que se han desarrollado y se encuentran actualmente en el mercado. Las **Unidades de Medición Inercial** (**IMU**, por sus siglas en inglés: *Inertial Measurement Unit*) se conforman de acelerómetros y giroscopios en su mayoría; ambos sensores miden aceleración lineal en un eje y velocidad angular alrededor de un eje respectivamente. Por otro lado, los sensores denominados **MARG** (por sus siglas en inglés: *Magnetic Angular Rate Gravity*) son una extensión de las IMU. Esto debido a que se le añade un tercer sensor fundamental: el magnetómetro, sensor capaz de medir la intensidad de campo magnético en un eje. Para ambos casos, el sensor es colocado en el cuerpo mientras que el mismo capta las fuentes externas (para este caso, naturales) antes mencionadas. Las desventajas de este tipo de sensores, es la constante descalibración del mismo, por lo que se utiliza en mediciones cortas. Sin embargo, el rango aumenta considerablemente en comparación con las configuraciones *inside-in* y *outside-in*.

En adición a la clasificación, es importante añadir los sistemas que de forma externa funcionan como fuentes y sensores de la señal, dejando al sujeto de estudio como un elemento pasivo de la medición. Podrían ser clasificados dentro de una categoría diferente:

- **Medición externa (*outside-out*):** para esta configuración se tiene una fuente externa la cual envía una señal la cual se transforma al interactuar con el cuerpo y finalmente es recuperada por medio de sensores igualmente externos. La gran ventaja de estos dispositivos es la invasión nula (o casi nula) al sujeto de estudio. Un ejemplo de estos sistemas son los algoritmos de **reconstrucción virtual por visión**, los cuales recuperan imágenes del sujeto y tras procesarlas se reconstruye el cuerpo, entonces así es posible realizar mediciones.

Otro ejemplo son los dispositivos acústicos, los cuales permite el reconocimiento de posturas del cuerpo humano. El dispositivo se coloca en las cercanías del cuerpo de manera estratégica. Un emisor envía una señal al ambiente y al reflejarse la onda sonora en el cuerpo, una serie de sensores receptores capturan las ondas reflejadas. Finalmente, tras procesarlas, se obtienen matrices de puntos que reconstruyen el cuerpo de forma virtual. Es posible capturar grandes movimientos como se muestra en el trabajo [5] de Kaustubh K. y Bhiksha, R en 2009, o pequeñas expresiones como en el proyecto *Soli*, actualmente desarrollado por Google.

La configuración que se pretende implementar para la obtención de variables relacionadas con la marcha humana, serán los dispositivos MEMS, en específico los sensores tipo MARG; debido a las ventajas que ofrece en cuestiones de espacio de trabajo disponible y la naturaleza misma de las mediciones a realizar. Dichas variables en concreto son los ángulos relativos.

3.2.3. Estado del Arte: Sistemas de Captura de Movimiento con Sensores IMU y MARG

Existen hoy en día, distintos métodos para aplicar un análisis inercial a un sujeto de estudio implementando cinemática inversa. Sin embargo, al avocarse al desarrollo de sistemas de captura de movimiento implementando IMU, se pueden hallar diferentes trabajos que convergen en el objeto de investigación del presente proyecto.

En el año de 1996, se presentó un artículo por parte de Anton J. Van den Bogert, *et al* en *Journal of Biomechanics*. En éste se detalla un método desarrollado para el cálculo de la fuerza resultante y momento en la cadera inferior humana durante una rutina de trabajo y al correr. Se utilizaron acelerómetros montados en la sección superior del cuerpo de los participantes. Los resultados de dicho estudio se cotejaron con los aquellos obtenidos de un sistema de visión y medición de la fuerza de reacción en tierra. Se hace énfasis de las ventajas que los sistemas con acelerómetros pueden ofrecer en cuanto a procesamiento de las señales y de la implementación de los dispositivos al aire libre.[6]

Seis años más tarde, en la misma revista arbitrada se publicó un trabajo de investigación a cargo de Ruth E. Mayagoitia, Anand V. Nene y Peter H. Veltink, donde se especifica al uso de acelerómetros y giroscopios como una herramienta alternativa para la obtención de la cinemática en el plano sagital durante el ciclo de marcha humana. Se montaron cuatro acelerómetros uniaxiales por cada giroscopio en diez voluntarios sanos y se tomaron registros de cinco velocidades de marcha en cada uno. Los resultados fueron comparados con mediciones realizadas con sistemas ópticos y uso de marcadores pasivos. Los autores muestran los bajos errores que se obtuvieron como producto de dicha comparación, recalcando la eficiencia de los sensores inerciales sin dejar a un lado sus ventajas económicas, de portabilidad y ergonomía.[7]

Por otro lado, la Universidad de Twente en los Países Bajos, publicó un artículo en 2005, en el cual se describe el desarrollo y la validación de un método para la adecuada medición de la orientación del cuerpo humano en movimiento. Detalla el uso de acelerómetros y giroscopios en conjunto para estimar la posición en el espacio de tres secciones del cuerpo humano: pelvis, tronco y antebrazo.[8]

En 2008, se presentó ante el IEEE (*Institute of Electrical and Electronics Engineers*) un proyecto nombrado como «*GaitShoe*»: Dispositivo inalámbrico adaptable a cualquier calzado, y con el cual se realizaba un análisis de marcha implementando acelerómetros y giroscopios con tres ejes ortogonales. En sincronía con estos dos elementos se integraron sensores de presión dinámica, fuerza, doblez bidireccional, así como sensores de campo eléctrico. Con validación del *Biomotion Laboratory* en el *Massachusetts General Hospital*, se probó que el dispositivo es capaz de detectar el momento de apoyo del talón hasta el momento de despegue de los dedos así como la posición y orientación del pie, *inter alia*.[9]



Figura 3.1: *Gait-Shoe*: Sensores inerciales montados en un zapato

En ese mismo año, se abrió la posibilidad de implementar las unidades de medición inercial en métodos capaces de asistir de forma eficiente a un deportista; se tiene como antecedente que científicos de la Universidad de Massey en Nueva Zelanda, desarrollaron un sistema de captura de movimiento para analizar las carreras de esquí.[10] En su artículo, publicado por *Sports Technology* se explica que se hizo uso de las IMUs en sincronía con cuatro técnicas aisladas: Sistema de Posicionamiento Global (GPS, por sus siglas en inglés), sensores de presión en plantillas, captura de video y mediciones con teodolitos. Se menciona al uso de las IMUs y el GPS como Fusión de Captura de Movimiento (FMC), lo que a su vez amplía el paradigma establecido, al complementar el flujo de datos resultados de las mediciones hechas por cada unidad con otro tipo de dispositivos de posicionamiento con la finalidad de mejorar la precisión de cada medición.

En 2009, Se presentó ante la revista *Journal of Biomechanics*, el trabajo de estudio de Ryo Takeda, Shingeru Tadano, *et al.* En dicho escrito se explica la obtención de los desplazamientos angulares en rodillas y cadera, para la reconstrucción tridimensional de figuras esquemáticas del cuerpo humano en un ambiente virtual *stick figures*. Se hizo uso de acelerómetros triaxiales en conjunto con giroscopios triaxiales con la validación de los resultados obtenidos implementando un sistema de visión. [11]

En 2013, se otorgó la patente US8512415B2[12] a Hugh M. Herr, *et al.* por la invención de una prótesis activa para tobillo y pie. En dicha prótesis se integra una IMU, la cuál monitorea continuamente la orientación tridimensional del tobillo así como la aceleración traslacional de la prótesis misma. Se identifica un error de divergencia de la posición en el tiempo, producto de aplicar la integral a la aceleración, llamado *drift*. Dicha patente resulta útil para la identificación de la ascensión y descenso de escaleras y rampas.

Durante ese mismo año, Camiglio, *et al.*, presentaron durante la serie de conferencias de la revista *Journal of Physics*, una alternativa al análisis biomecánico mediante la implementación de electrogoniómetros. La ventaja de estos componentes es que ofrecen una alta precisión y elevada velocidad de muestreo, sin mencionar que se conectan por medio de una red inalámbrica disminuyendo complicaciones en el momento en que se practica la medición. Si bien, es evidente que no se ocupan sensores inerciales en dicho proyecto, es un ejemplo de la existente posibilidad de incursionar en el análisis biomecánico al monitorizar exclusivamente la variación espacio-temporal de los ángulos relativos a los eslabones involucrados en la marcha humana. [13]

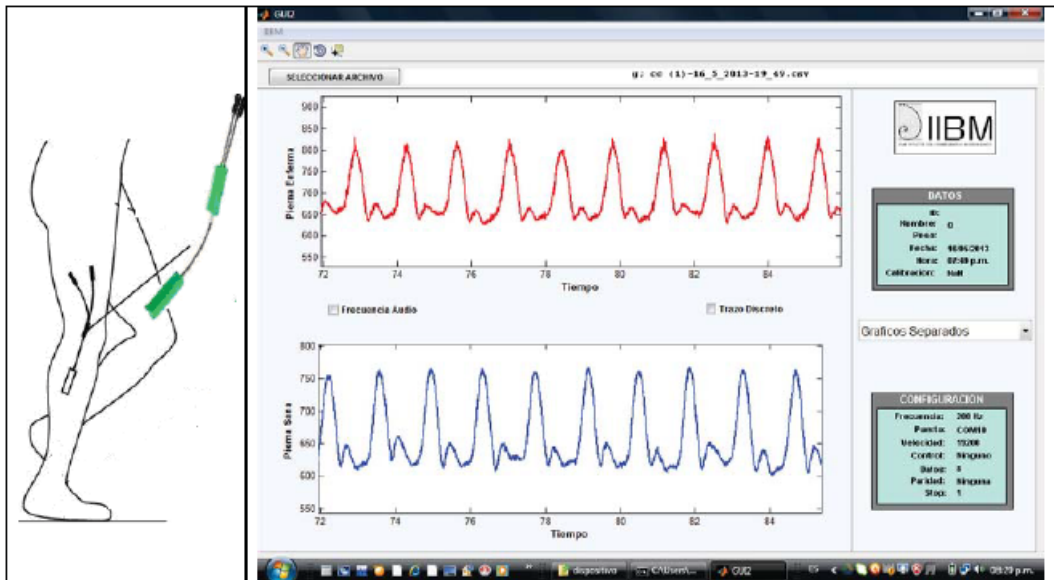


Figura 3.2: Análisis de la variación del ángulo de flexo-extensión de rodilla implementando goniómetros inalámbricos. [13]

Por otro lado, la Universidad de Cambridge en Reino Unido presentó en 2014 durante la conferencia de *International Sports Engineering Association* un artículo donde se propuso la posibilidad de implementar las IMUs como una alternativa potencial para situaciones de rehabilitación y en el deporte. En este trabajo se recalca la importancia del tamaño de dichos sensores, debido a que en la actualidad pueden adaptarse al calzado del usuario y se logra a la par un monitoreo continuo de la marcha del mismo. De igual manera brinda un panorama más extenso de la fiabilidad de estos dispositivos, al comparar los resultados obtenidos con pruebas que se realizaron con sistemas de captura de movimiento que utilizan visión. [14]

En el mismo año, la revista *Sensors* publicó un artículo de Seel, Thomas, *et al* donde se realiza la medición de los ángulos de articulaciones en tren inferior para el análisis de marcha. En esta publicación se pueden identificar dos aspectos relevantes: En primer lugar, se detalla la importancia de la colocación de los dispositivos para la correcta medición de los ángulos en articulaciones. Y en segundo lugar, se trabajó con un sujeto de estudio al que se le amputó la pierna izquierda donde se observó la clara implementación de las unidades de medición inercial para la rehabilitación. La prótesis que el individuo utilizó, se sometió a pruebas de carácter biomecánico para garantizar el funcionamiento óptimo de la misma.

En la revista *IEEE Potentials*, se publicó en julio de 2016 un artículo en el que se menciona la importancia del uso de las IMUs en los últimos años para el monitoreo tanto de las actividades físicas que desempeña el ser humano, como de la salud. Incorpora dos importantes elementos a su sistema: la electromiografía (técnica popular en el análisis de la activación de los músculos), y la funcionalidad inalámbrica en los dispositivos incorporados. [15]

3.3. Sensores Inerciales

3.3.1. Tecnología MEMS

El acrónimo MEMS se forma de las palabras en inglés: *Microelectromechanical Systems* (Sistemas microelectromecánicos), *i.e.* tecnología que incorpora componentes mecánicos y electrónicos dentro de una escala de entre 1 a 100 micrómetros.

3.3.2. El Acelerómetro

Un acelerómetro MEMS puede ser considerado como un **sistema simplificado masa-resorte**, como lo muestra la Figura 3.3. Consiste en una masa de prueba con placas primarias (estructura roja), que en conjunto se encuentra unido a un sistema de suspensión por medio de un resorte fijo a un marco de referencia. En este marco se encuentra un arreglo de placas secundarias (verdes) polarizadas y adyacentes a las placas primarias con un espaciado nominal. Cuando el dispositivo se acelera sobre el eje de trabajo del acelerómetro, la masa de prueba se mueve, cambiando así el espacio entre las placas. Al medir la variación de las capacitancias ($C1$ y $C2$) en las placas primarias con respecto a las secundarias, se puede calcular una equivalencia de la aceleración de la masa [16].

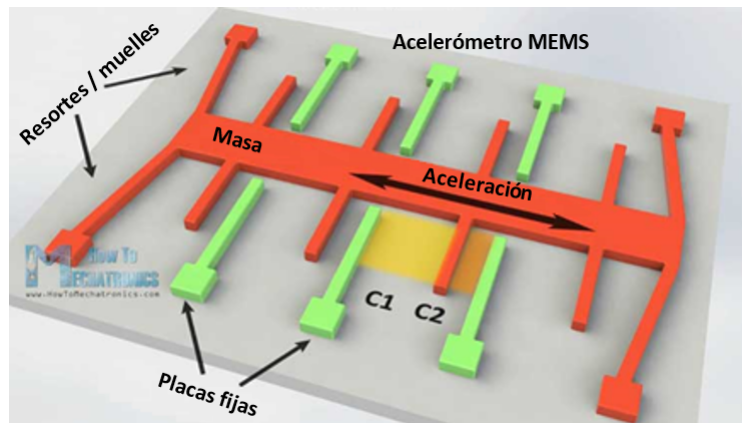


Figura 3.3: Esquema de un acelerómetro MEMS [17]

Es importante recalcar que en realidad no se mide la aceleración lineal del cuerpo del sensor, sino que **el sensor reacciona a la fuerza aplicada a la masa de prueba, y al medir el desplazamiento causado por dicha fuerza se calcula la aceleración involucrada** en el movimiento. Por este motivo, estos dispositivos son sensibles a la fuerza gravitatoria. En diversas aplicaciones este fenómeno se aprovecha para calcular la orientación de cuerpos (generalmente restringidos a movimientos lineales o con leves aceleraciones lineales), tomando como referencia el campo gravitatorio. Sin embargo, en aquellas aplicaciones en las que no es de relevancia la fuerza de atracción gravitatoria, es necesario sustraer dicha medición, de lo contrario los pequeños errores de medición en la aceleración, serán integrados en errores de velocidad progresivamente más grandes y aún mayores errores en la posición.

A continuación se detallan las ventajas y desventajas de los acelerómetros:

Ventajas

- Ofrece un vector de referencia para determinar la orientación del sensor (dirección del campo gravitatorio terrestre)
- Es preciso a largo plazo al no presentar divergencia en las mediciones de aceleración
- Mediciones a altas frecuencias

Desventajas

- Mediciones susceptibles al ruido, debido a la alta sensibilidad del sensor
- Baja confianza en mediciones a corto plazo, debido a detección de vibraciones (ruido)

3.3.3. El Giroscopio

El giroscopio es un dispositivo que mide la velocidad angular de un cuerpo aprovechando el **efecto Coriolis** el cuál se manifiesta cuando una masa se mueve en una dirección en particular con una velocidad determinada y un par externo de fuerzas provoca una velocidad angular en el cuerpo, y por ende aparece una segunda fuerza perpendicular a la dirección del movimiento. En la Figura 3.4 se muestran las fuerzas involucradas en el efecto Coriolis. [17]

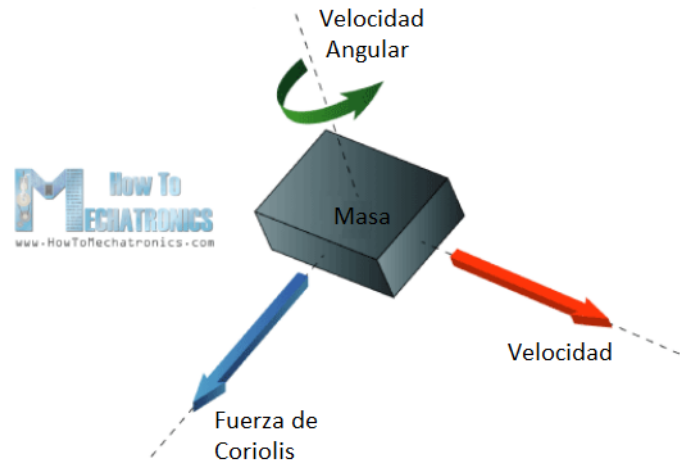


Figura 3.4: Diagrama de cuerpo libre del Efecto Coriolis. [17]

La microestructura de un giroscopio MEMS se muestra en la Figura 3.5. Las estructuras verdes son elementos flexibles que permiten el movimiento del marco amarillo (en conjunto con el rojo) en una sola dirección (Dirección de vibración). Las paletas azules, tanto las internas como las externas permanecen fijas en el marco inercial del sensor. Si una velocidad angular es aplicada como se muestra en la imagen, los elementos flexibles internos permiten el desplazamiento de la masa de prueba en dirección perpendicular al marco de referencia (Dirección del Cambio de la Capacitancia). Finalmente, de la misma manera que se realiza para el acelerómetro, se mide la variación de las capacitancias en las placas primarias con respecto a las secundarias, de tal forma que se calcula el desplazamiento perpendicular de la masa y - por ende - es posible determinar la velocidad angular causante de dicho desplazamiento [17].

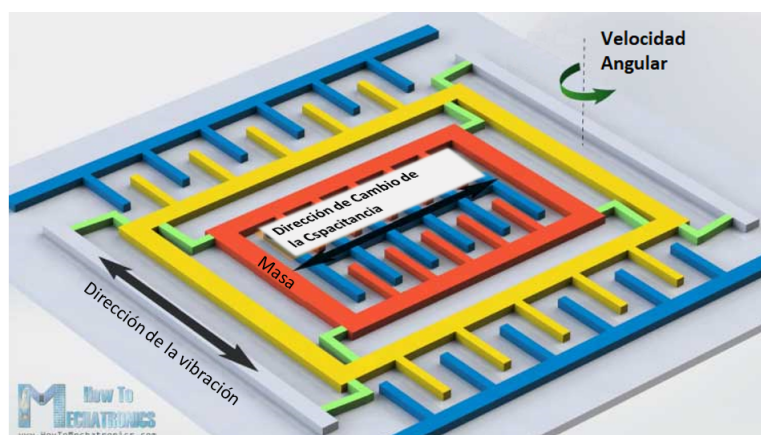


Figura 3.5: Esquema de un Giroscopio MEMS [17].

A continuación se detallan las ventajas y desventajas de los giroscópios:

Ventajas

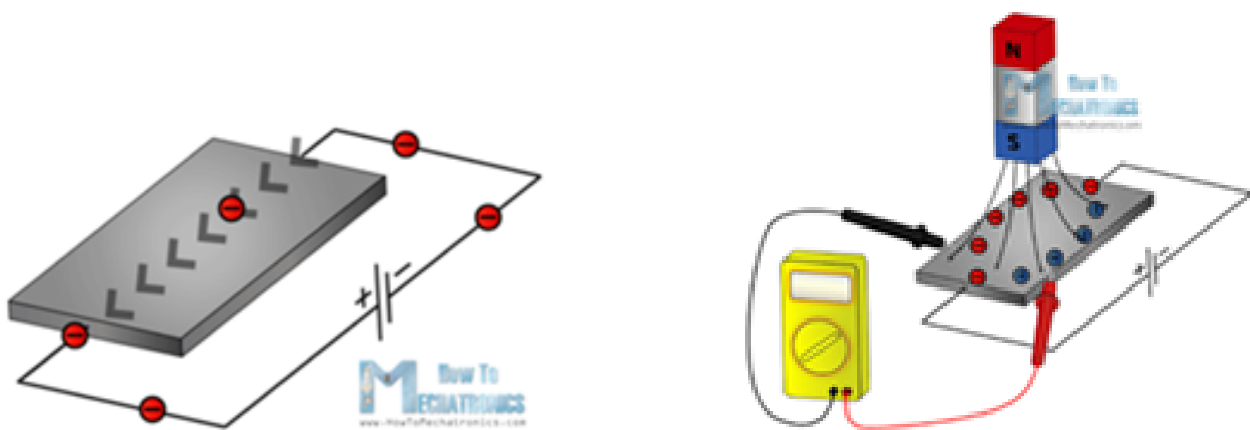
- Dispositivo implementado para calcular orientación de forma precisa en lapsos de tiempo cortos (tomando en cuenta que se estima la orientación mediante algún algoritmo); Complementa las mediciones realizadas por sensores secundarios

Desventajas

- Constante de «descentramiento» (bias) dependiente de la temperatura
- Estimación inestable de la orientación para movimientos no lineales

3.3.4. El Magnetómetro

Cerca de un 90 % de los magnetómetros en el mercado operan bajo el principio conocido como **Efecto Hall** (por otro lado, existen magnetómetros que funcionan mediante el efecto magneto-resistivo). Al conectar una fuente eléctrica a través de una placa conductora, como se muestra en la Figura 3.6a, los electrones se desplazan de manera uniforme. Sin embargo, al acercarse un campo magnético a la placa, los electrones son desplazados hacia un extremo de las placas ocasionando que las cargas positivas se agrupen al lado opuesto de la placa; como se muestra en la Figura 3.6b. Dicha polarización en la placa se refleja a modo de una **diferencia de potencial en el eje perpendicular a la corriente eléctrica y sobre el plano de la placa, y por ende se calcula la intensidad de campo magnético involucrado en el voltaje medido.** [17]



(a) Flujo de electrones sin presencia de campo magnético [17]

(b) Diferencia de potencial en la placa, en presencia de un campo magnético [17]

Figura 3.6: Efecto Hall

A continuación se detallan las ventajas y desventajas de los magnetómetros:

Ventajas

- Ofrece un vector de referencia para determinar la orientación del sensor (dirección del campo magnético terrestre)

Desventajas

- Mediciones susceptibles al ruido ocasionado por estructuras metálicas y distorsiones en el campo magnético
- Es necesario calibrar el dispositivo para acondicionarlo a las condiciones ambientales iniciales, el cual incluso varía con la ubicación geográfica

El sensor AK8963 implantado en la tarjeta de desarrollo de *SparkFun Electronics* funciona bajo el principio descrito anteriormente. Incorpora un arreglo de tres sensores ortogonales entre sí de tal forma que se miden las intensidades de campo magnético en tres ejes.

3.3.5. Sensores MEMS: MARG vs IMU

Un sensor **MARG** es aquel dispositivo que mide la posición y orientación de un cuerpo que lo transporta, utilizando la combinación de las señales de acelerómetros, giroscopios y magnetómetros, y en ocasiones barómetros y termómetros. Por el contrario, las **IMUs** se constituyen únicamente mediante el conjunto de acelerómetros y giroscopios, permitiendo el seguimiento del movimiento lineal y rotacional de un cuerpo respectivamente [18]. Si bien, ambos dispositivos están compuestos por una combinación de sensores, la diferencia fundamental entre ellos es el número de **grados de libertad (GDL)**. Los GDL de un sistema es el número de parámetros (mediciones) independientes que se requieren para definir de manera única su condición en cualquier instante de tiempo. Por lo tanto, dependiendo de las diferentes variables que mida el sensor y el número de ejes que posea para medir dicha variable, será el número de GDL que posea el sensor. **La mayoría de las IMUs son de 6 GDL**, debido a que se compone de arreglos ortogonales de tres acelerómetros y tres giroscopios. Los instrumentos desarrollados con esta tecnología son capaces de medir por su cuenta (aplicando algún algoritmo de procesamiento existente) la orientación espacial relativa a la dirección del campo gravitatorio terrestre. [3] [19]

Para la medición tridimensional, son necesarios tres sensores independientes y capaces de realizar mediciones en tres ejes ortogonales llevando a cabo 9 mediciones independientes: GDL. Por tal motivo, **la mayoría de los sensores MARG son de 9 GDL** ya que se implantan los mismos arreglos que en las IMUs y se añade un arreglo más de magnetómetros ortogonales en la misma unidad. Los MARG también son conocidos como **AHRS** (Sistema de Referencia de Actitud y Sentido, por sus siglas en inglés para: *Attitude and Heading Reference System*, por su capacidad de proveer una medición completa de su orientación con respecto a las direcciones de los campos gravitatorio y magnético terrestres. [19]

3.4. Representación de la Rotación en el Espacio

3.4.1. Notación

Para este proyecto, se utiliza la notación de Craig [20], la cual consta de las siguientes convenciones:

1. Los subíndices y superíndices a la izquierda identifican en qué sistema de coordenadas está escrita esa cantidad. Por ejemplo, ${}^A P$ representa un vector de posición escrito en el sistema de coordenadas $\{A\}$. Por otro lado, ${}^A_B R$ es una matriz de rotación que especifica la orientación del sistema de coordenadas $\{B\}$ respecto al sistema $\{A\}$.
2. Los superíndices a la derecha se utilizan (según su amplia aceptación) para indicar la inversa o transpuesta de una matriz (por ejemplo, R^{-1} , R^T).
3. Los subíndices a la derecha no están sujetos a ninguna convención estricta, pero pueden indicar el componente de un vector (por ejemplo, x , y o z).
4. Para funciones trigonométricas, la notación se simplifica como se muestra para el coseno de un ángulo θ_1 . Puede tomar cualquiera de las siguientes formas: $\cos(\theta_1) = c\theta_1 = c_1$.
5. Los vectores se consideran como vectores columna, por lo cual los vectores renglón tendrán la transpuesta de forma explícita.

3.4.2. Sistema de coordenadas

Se entiende por sistema de coordenadas a un **sistema mediante el cual se define la posición y orientación de un cuerpo en el espacio**. En particular, el sistema de coordenadas cartesiano (que se utiliza ampliamente en Robótica y Biomecánica) se forma estructuralmente por tres ejes coordenados ortonormales entre sí los cuales parten de un punto conocido como origen [18]. Para conocer la posición de un cuerpo rígido basta con conocer la **posición** de un punto fijo sobre el mismo referente a un sistema de coordenadas. Comúnmente, se adjunta un segundo sistema de coordenadas al cuerpo, con la finalidad de describir su **orientación** con respecto a un sistema de referencia [20].

Si es necesario definir la posición de un sistema de coordenadas $\{B\}$ con respecto a un sistema de referencia $\{A\}$, se necesita un vector que indique la posición del origen sistema $\{B\}$ en $\{A\}$; tomando en cuenta que un punto referente al sistema $\{A\}$ ${}^A P$, se representa por medio de un vector columna (3×1).

$${}^A P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (3.1)$$

Para un sistema de Goniometría es meramente necesario conocer los ángulos de un cuerpo al que esté fijo, respecto a otro. Si los sensores MARG propuestos son capaces de proveer la información necesaria para estimar dichas variables, el procesamiento a desarrollar se centrará únicamente en la obtención de la orientación.

3.4.3. Tramas y Matrices de Rotación

Las matrices de rotación son ampliamente utilizadas en distintas disciplinas para definir la orientación de un cuerpo con respecto a una referencia en el espacio. Al igual que en las posiciones, es necesario asociar un sistema de coordenadas al cuerpo.

Para definir la orientación de un sistema de coordenadas $\{B\}$ con respecto a $\{A\}$, se definen los vectores unitarios del sistema $\{B\}$ como \hat{X}_B , \hat{Y}_B y \hat{Z}_B . Al escribirse en referencia al sistema $\{A\}$ se nombran como: ${}^A \hat{X}_B$, ${}^A \hat{Y}_B$ y ${}^A \hat{Z}_B$. Al formar una matriz cuyas columnas sean los vectores descritos anteriormente, se obtiene la **matriz de rotación**, la cual describe la orientación de $\{B\}$ respecto a $\{A\}$. en la Ecuación 3.2 se observa dicha matriz.

$${}^A R_B = \begin{bmatrix} {}^A \hat{X}_B & {}^A \hat{Y}_B & {}^A \hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.2)$$

En la Ecuación 3.2 se pueden definir expresiones para los escalares r_{ij} teniendo en cuenta que los componentes de cualquier vector son simplemente las proyecciones de ese vector en las direcciones unitarias de su sistema de referencia. Por lo tanto, cada componente de ${}^A R_B$ en (3.2) puede escribirse como el producto punto de un par de vectores unitarios [20]:

$${}^A R_B = \begin{bmatrix} {}^A \hat{X}_B & {}^A \hat{Y}_B & {}^A \hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} \quad (3.3)$$

Como el producto punto de dos vectores unitarios es el coseno del ángulo entre ellos, queda claro porqué a los componentes de las matrices de rotación se les llama comúnmente cosenos de dirección. Una inspección más detallada de la Ecuación 3.3 nos muestra que las filas de la

matriz son los vectores unitarios de $\{A\}$ expresados en $\{B\}$.

De tal manera que es posible identificar la posición y orientación de un sistema de coordenadas con respecto a otro por medio de una matriz de 3×3 y un vector de 3×1 . De esta premisa surge el concepto de la **trama**.

Partiendo del hecho que en un inicio los sistemas de coordenadas A y B ocupan un mismo origen y orientación de ejes; al aplicar las transformaciones descritas anteriormente, el sistema B rota y se traslada en relación al sistema A.

La trama puede resumirse como aquella entidad matemática que se compone principalmente por la matriz de rotación y el vector de traslación. Se trata de una matriz de 3×4 , en cuyas columnas se encuentra la información sobre la posición y la orientación de la trama secundaria. Por ejemplo, la trama $\{B\}$ se describe mediante ${}^A_B R$ y $A P_{BORG}$, en donde $A P_{BORG}$ es el vector que ubica el origen de la trama $\{B\}$. Por lo tanto, al sistema de referencia «de principal interés», se le puede asociar con el nombre de *trama*, debido a que es su propia definición:

$$\{B\} = {}^A_B R, {}^A P_{BORG} \quad (3.4)$$

En la Figura 3.7 se muestra una representación gráfica de la trama. Es necesario, además de trazar los vectores unitarios (tres ejes) de la trama, dibujar una flecha que representa a un vector que va de un origen hacia el otro. Este vector representa la posición del origen en la punta de la flecha, en términos de la trama en la cola de la flecha. Las posiciones podrían representarse mediante una trama cuya parte relacionada a la matriz de rotación es la matriz identidad y cuya parte correspondiente al vector de posición ubica el punto que se está describiendo. De igual forma, una orientación podría representarse mediante una trama cuya parte correspondiente al vector de posición sea al vector cero.

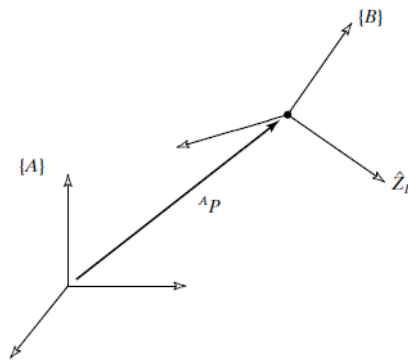


Figura 3.7: Representación gráfica de una trama

En Robótica, las tramas ayudan a posicionar y orientar sistemas de coordenadas asociados a eslabones de una cadena cinemática. Por medio de transformaciones (rotaciones y traslaciones, derivadas de las tramas), se reconstruyen uno a uno los eslabones que forman la cadena, actualizándose en cada iteración. De hecho resultan de gran utilidad para ubicar en el espacio tanto los eslabones propios de una cadena, como cuerpos rígidos libres.

En el presente proyecto, las traslaciones no figuran dentro de los objetivos del mismo, por lo tanto, aunque las tramas resultan de gran utilidad en la Ingeniería, las matrices de rotación podrían bastar para obtener la orientación de un cuerpo rígido en el espacio.

Existen distintas formas de representar la orientación de un cuerpo rígido en el espacio mediante matrices de rotación. De las mayormente utilizadas se encuentran las **rotaciones mediante**

Ángulos de Euler, además de las rotaciones mediante **Ángulo - Eje equivalente** detallados en el anexo 10.1.

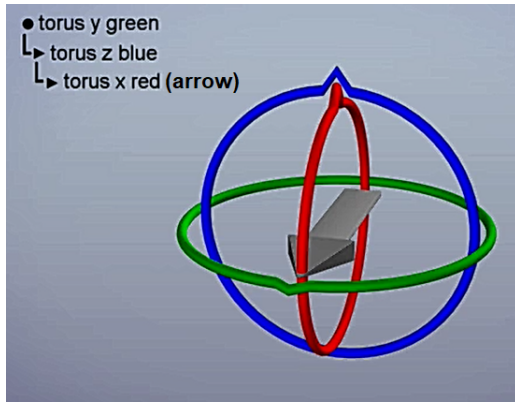
Conclusiones respecto a la Representación de la Orientación por Matrices de Rotación

En perspectiva, **las aplicaciones mayormente viables para la implementación de rotaciones por medio de Matrices de Rotación, son aquellas en las que las rotaciones estén acotadas a cierta región en el espacio; e.g.** el vuelo de un helicóptero, el cual comúnmente no realiza movimientos de balanceo e inclinación pronunciados (Ver Figura 10.2). Sin embargo **para sistemas de mayor complejidad en el que las rotaciones de los cuerpos involucrados resulta de gran inconveniencia** debido a las indeterminaciones anteriormente analizadas.

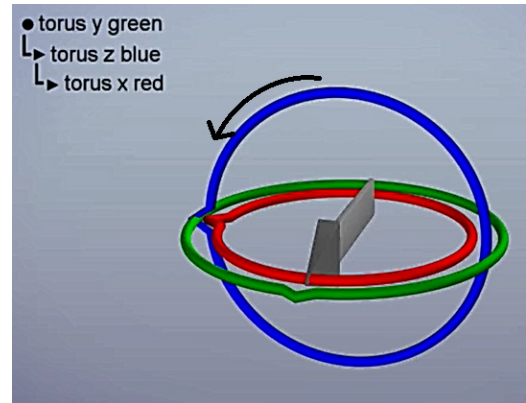
Por otro lado, los **recursos computacionales** para el procesamiento de las matrices de rotación y el cálculo de los ángulos solución implicados, resultan altos. Debido a las diferentes combinaciones en el orden de rotación alrededor de los ejes coordenados, se obtiene un conjunto de 24 parametrizaciones de una matriz de rotación, es decir 24 formas de representar de forma matricial una rotación. Sin mencionar que este tipo de representaciones sufren de un problema conocido como «*Gimbal Lock*». [16]

Gimbal Lock

El Gimbal Lock (Bloqueo de Cardán o Bloqueo de Eje) es una singularidad matemática propia de las rotaciones sucesivas alrededor de ejes coordenados. Es evidente que no existe una singularidad en el grupo de rotaciones, dado que es posible girar un cuerpo rígido en el espacio sin caer en la singularidad alguna. Sin embargo existe la posibilidad que **se pierda un grado de libertad en rotaciones características**. A este decremento en los grados de libertad de la representación en sí misma, es llamada singularidad coordenada [21]. De forma gráfica, esto ocurre en el caso en que dos ejes coordenados (tras girar de forma sucesiva) resulten paralelos. En la Figura 3.8 se muestra de forma gráfica el fenómeno del Gimbal Lock. Si se tiene una flecha en un sistema coordenado y se acopla dicha flecha al eje X, tras definir el orden de giros como X, seguido por Y, seguido por Z como en la Figura 3.8a se manifiesta la singularidad al girar 90° el eje Z. En la Figura 3.8b se observa la alineación de los ejes Y y X; al girar un ángulo θ alrededor de X, se obtiene el mismo efecto si se girara con respecto a Y, por lo que los grados de libertad se reducen en uno.



(a) Estructura de giro mediante ángulos de Euler Z - Y - X [22]



(b) Al girar alrededor de Z (azul) se aprecia la alineación de los ejes X (rojo) y Y (verde) [22]

Figura 3.8: Visualización del bloqueo de eje «*Gimbal Lock*»

Con las conclusiones presentadas se da pie a una opción alternativa para representar las rotaciones de un cuerpo rígido en el espacio: **los Cuaterniones**.

3.4.4. Cuaterniones

Creados por William Rowan Hamilton en 1843 [16] los Cuaterniones se definen como la suma de cuatro términos: [23]

$$q = 1 \cdot q_1 + i \cdot q_2 + j \cdot q_3 + k \cdot q_4$$

en donde q_1, q_2, q_3, q_4 , pertenecen al dominio de los **Números Reales**, 1 es la identidad multiplicativa e i, j, k son la **representación simbólica de tres componentes en el dominio de los Números Imaginarios** los cuales poseen las siguientes propiedades:

$$\begin{aligned} i^2 &= -1, & j^2 &= -1, & k^2 &= -1, \\ ij &= k, & ji &= -k, \\ jk &= i, & kj &= -i \\ ki &= j, & ik &= -j \end{aligned} \tag{3.5}$$

Las propiedades algebraicas se analizan en el anexo 10.2 y se aprovechan en análisis posteriores.

Relación entre el espacio de Cuaterniones y los Vectores \mathbb{R}^3

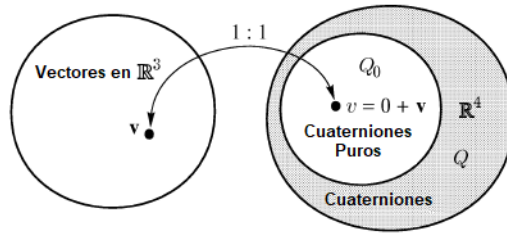


Figura 3.9: Correspondencia entre Vectores \mathbb{R}^3 y Cuaterniones [24]

Sea un cuaternión $q = q_1 + iq_2 + jq_3 + kq_4$, aquel componente correspondiente a q_1 se le conoce como la **parte escalar del cuaternión**. Por otro lado, a la parte correspondiente a los componentes iq_2 , jq_3 y kq_4 se le conoce como la **parte vectorial** del cuaternión q .

Un **cuaternión vectorial** se define como aquel cuya parte escalar es cero; es decir: $q_v = 0 + iq_2 + jq_3 + kq_4$. Mientras que un **cuaternión real o escalar** se define como aquel cuya parte vectorial es nulo; es decir: $q_r = q_1 + 0i + 0j + 0k$

Para una correspondencia uno a uno entre los vectores en \mathbb{R}^3 y su correspondiente cuaternión puro, el significado del producto entre un vector y un cuaternión se puede entender como la multiplicación de dos cuaterniones, en donde uno de ellos es un cuaternión puro cuya parte vectorial corresponde en sus componentes con el vector.

Por lo que, si se tiene un vector $v = a \cdot \hat{i} + b \cdot \hat{j} + c \cdot \hat{k}$ definida en \mathbb{R}^3 , su cuaternión puro correspondiente sería $q_v = 0 + ia + jb + kc$ y puede se le puede aplicar cualquier operación propia de los cuaterniones.

Cuaterniones Unitarios

Aquel cuaternión $q_u = \hat{q}$ será unitario si cumple con la condición: $|\hat{q}| = 1$. Cabe resaltar que los cuaterniones unitarios poseen propiedades particulares, por ejemplo de la ecuación 4:

$$\hat{q}^{-1} = \hat{q}^* \quad (3.6)$$

Para los cuaterniones unitarios existe una propiedad en específico que resulta de gran utilidad, la cual es:

$$\hat{q} = q_1 + iq_2 + jq_3 + kq_4 = r_u \cos \theta + p_u \sin \theta = \cos \theta + p_u \sin \theta \quad (3.7)$$

En donde $r_u = 1 + 0r_{u,2} + 0r_{u,3} + 0r_{u,4} = 1$ es un cuaternión unitario de parte escalar igual a uno, mientras que $p_u = 0 + ip_2 + jp_3 + kp_4$ es un cuaternión unitario de parte escalar igual a cero y paralelo a la parte vectorial del cuaternión \hat{q} . La demostración de lo anterior mencionado es:

$$\begin{aligned} |\hat{q}|^2 &= \hat{q} \otimes \hat{q}^* = (r_u \cos \theta + p_u \sin \theta) \otimes (r_u \cos \theta + p_u \sin \theta)^* \\ &= r_u \otimes r_u^* \cos^2(\theta) + (r_u \otimes p_u^* + p_u \otimes r_u^*) \sin \theta \cos \theta + p_u \otimes p_u^* \sin^2(\theta) \\ &= \cancel{r_u \otimes r_u^*}^1 \cos^2(\theta) + (\cancel{r_u \otimes p_u^* + p_u \otimes r_u^*})^0 \sin \theta \cos \theta + \cancel{p_u \otimes p_u^*}^1 \sin^2(\theta) \\ &= \cos^2(\theta) + \sin^2(\theta) = 1 \end{aligned}$$

Hasta este punto es posible interpretar a θ como una manera de cuantificar la relación de la representación real del cuaternión con la magnitud de su representación vectorial. Se representa geoméricamente como el ángulo de rotación de un vector. [23]

3.4.5. Representación de la orientación por medio de Cuaterniones

Para comprender como se representa la orientación de un cuerpo en el espacio a través de las operaciones que se implementan con los cuaterniones, es necesario detallar ciertas características más de operaciones específicas de los cuaterniones.

Producto entre cuaterniones reales [23]

El producto de cuaterniones reales da como resultado un cuaternión real y la operación es conmutativa:

$$p_r \otimes q_r = p_1q_1 = q_1p_1 = q_r \otimes p_r$$

E incluso es asociativa también:

$$(p_r \otimes q_r) \otimes r_r = (p_1q_1)r_1 = p_1(q_1r_1) = p_r \otimes (q_r \otimes r_r)$$

Producto entre un cuaternión real y un cuaternión vectorial [23]

El producto entre un cuaternión real y uno vectorial da como resultado un cuaternión vectorial. Y dicha operación es conmutativa:

$$p_r \otimes q_v = 0 + p_1q_2i + p_1q_3j + p_1q_4k = 0 + q_2p_1i + q_3p_1j + q_4p_1k = q_v \otimes p_r$$

Producto entre dos cuaterniones vectoriales [23]

El producto entre dos cuaterniones vectoriales tiene una propiedad de gran importancia:

$$\begin{aligned} p_v \otimes q_v &= -(p_2q_2 + p_3q_3 + p_4q_4) + (p_3q_4 - p_4q_3)i + (p_4q_2 - p_2q_4)j + (p_2q_3 - p_3q_2)k \\ &= -p_r \cdot q_r + p_v \times q_v \end{aligned}$$

Donde las operaciones « \cdot » y « \times » corresponden al producto punto y producto vectorial del álgebra vectorial clásica. Por lo que se concluye que el resultado claramente es un cuaternión exceptuando dos casos:

- Si $p_v \parallel q_v$ el producto entre los cuaterniones p y q es únicamente la parte real del resultado: $-p_r \cdot q_r$
- Si $p_v \perp q_v$ el producto entre los cuaterniones p y q es únicamente la parte vectorial del resultado: $p_v \times q_v$

Producto entre un cuaternión unitario y un cuaternión vectorial perpendicular [23]

En este caso específico, resulta de gran relevancia para el entendimiento del cómo es que los cuaterniones se utilizan para rotar un vector en tres dimensiones. Se parte de suponer un cuaternión vectorial s_v y un cuaternión unitario $\hat{q} = q$ (cuya parte vectorial se denota como q_v) y que las partes vectoriales de ambos son perpendiculares. De acuerdo con la ecuación 3.7 se puede asumir que:

$$t = q \otimes s_v = (\cos \theta + q_v \sin \theta) \otimes s_v = \cos \theta s_v + \sin \theta (q_v \otimes s_v) \quad (3.8)$$

En donde q_v es la parte vectorial del vector q . Para el primer término resultante $t_{v,(1)}$ ($= \cos \theta s_v$) es un cuaternión vectorial paralelo a s_v . El segundo término $t_{v,(2)}$ ($= \sin \theta(q_v \otimes s_v)$) también es un cuaternión vectorial y ya que $q_v \perp s_v$, entonces implica que $t_{v,(2)}$ es perpendicular tanto a s_v como a q_v . Tanto $t_{v,(1)}$ como $t_{v,(2)}$ pertenecen a un plano perpendicular a q . En este caso, $t = t_{v,(1)} + t_{v,(2)}$ puede ser interpretado geoméricamente como un desplazamiento angular de s_v con magnitud θ en este plano (*i.e.* alrededor de un eje paralelo a q_v).

Ahora, si se considera el siguiente producto:

$$u_v = t_v \otimes q^{-1} = t_v \otimes q^* = (\cos \theta)t_v + \sin \theta(t_v \otimes q_v^*) = (\cos \theta)t_v - \sin \theta(t_v \otimes q_v)$$

Posteriormente se considera la identidad $t_v \otimes q_v = -q_v \otimes t_v$ para reescribir la ecuación anterior como:

$$u_v = (\cos \theta)t_v + \sin \theta(q_v \otimes t_v) \quad (3.9)$$

De las ecuaciones 3.8 y 3.9 se puede distinguir que en la segunda se realiza una segunda rotación de magnitud θ alrededor de \hat{q} . Debido a que el desplazamiento angular θ es el mismo para ambos productos, la ecuación 3.10 define una rotación de 2θ al vector s_v a rededor de \hat{q} :

$$u_v = \hat{q} \otimes s_v \otimes \hat{q}^{-1} = \hat{q} \otimes s_v \otimes \hat{q}^* \quad (3.10)$$

Rotación de un vector arbitrario mediante el producto de cuaterniones unitarios [23]

En este caso se analiza el comportamiento de un vector arbitrario al aplicarle la operación descrita en la ecuación 3.10. Para este caso se considera un vector v_v y un cuaternión unitario q (cuya parte vectorial se denota como q_v), el cual puede ser descrito como la suma de dos vectores:

$$v_v = w_v + s_v : w_v \parallel q_v, s_v \perp q_v$$

Dado este caso, entonces:

$$\begin{aligned} q \otimes v_v \otimes q^{-1} &= q \otimes (w_v + s_v) \otimes q^{-1} \\ &= q \otimes w_v \otimes q^{-1} + q \otimes s_v \otimes q^{-1} \\ &= q \otimes w_v \otimes q^{-1} + u_v \end{aligned}$$

Donde u_v es el vector resultante de la rotación de s_v alrededor de q_v una magnitud de 2θ . Para evaluar el primer término, es necesario considerar que $w_v \parallel q_v$, por lo que es posible considerar que: $w_v = z(q_v) : z \in \mathbb{R}$. Por lo tanto:

$$\begin{aligned} q \otimes w_v \otimes q^{-1} &= q \otimes z(q_v) \otimes q^{-1} \\ &= z(q \otimes q_v \otimes q^{-1}) \\ &= z(q_v \otimes \cancel{q \otimes q^{-1}} \rightarrow 1) \\ &= z(q_v) = w_v \end{aligned}$$

Finalmente se tiene que:

$$q \otimes v_v \otimes q^{-1} = w_v + u_v$$

Geoméricamente, se interpreta a esta operación como la rotación de v_v alrededor de q con un desplazamiento angular de 2θ , como se muestra en la Fig. 3.10.

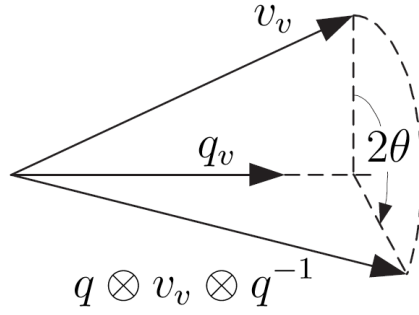


Figura 3.10: Representación gráfica de la rotación de un vector arbitrario v_v por medio de un cuaternión q un ángulo de 2θ

La ventaja de esta operación es que puede aplicarse a cualquier tipo de vector en un espacio de \mathbb{R}^3 . Incluidos los vectores unitarios que describen un sistema de coordenadas. Es decir, **Si se aplica la operación a cada eje (\hat{i} , \hat{j} , \hat{k}) coordenado que defina la orientación de un cuerpo en el espacio, se tiene como resultado un sistema coordenado desplazado de forma angular y por ende la definición del mismo cuerpo orientado de manera distinta.**

Tomando como referencia la notación anteriormente descrita en la sección de Matrices de Rotación aplicada a la introducción de los Cuaterniones se muestra la forma de representar una rotación en el espacio por medio de a transformación de una trama a otra. [19]

Sea la trama $\{B\}$ relativa a $\{A\}$ por medio de una rotación, el cuaternión unitario ${}^A_B\hat{q}$ describe la orientación descrita en la ecuación 3.11 en donde r_x , r_y y r_z definen las componentes de un vector unitario (sus componentes en los ejes x , y y z , respectivamente) ${}^A\hat{r}$ descrito en la trama $\{A\}$. Por otro lado, θ representa a la magnitud de dicha rotación. De manera gráfica se observa dicha transformación en la Fig. 3.11. Es posible distinguir cierta **coincidencia con el fenómeno que manifiesta la aplicación de una Matriz de transformación de una trama mediante un Ángulo - Eje Equivalente.**

$$\begin{aligned} {}^A_B\hat{q} &= [q_1 \quad q_2 \quad q_3 \quad q_4] \\ &= \left[\cos \frac{\theta}{2} \quad -r_x \sin \frac{\theta}{2} \quad -r_y \sin \frac{\theta}{2} \quad -r_z \sin \frac{\theta}{2} \right] \end{aligned} \quad (3.11)$$

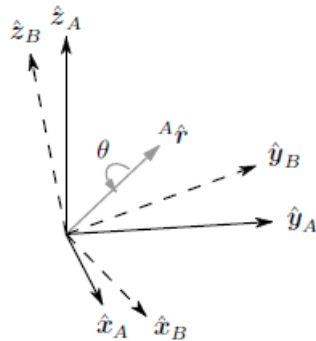


Figura 3.11: Representación Gráfica de la Rotación de la B , con respecto de A [19]

Para este caso, **el conjugado del cuaternión, y tomando en cuenta la ecuación 3.6 puede utilizarse para intercambiar las tramas de referencia del mismo.** Es decir:

$${}^A_B\hat{q}^* = {}^B_A\hat{q} = [q_1 \quad -q_2 \quad -q_3 \quad -q_4]$$

Por otro lado, **el producto entre cuaterniones se utiliza para definir orientaciones compuestas**. Es decir, que para dos orientaciones descritas como ${}^A_B\hat{q}$ y ${}^B_C\hat{q}$, la orientación compuesta será descrita por ${}^A_C\hat{q}$ mediante:

$${}^A_C\hat{q} = {}^B_C\hat{q} \otimes {}^A_B\hat{q}$$

De la definición de la multiplicación de cuaterniones (ecuación 10.7) es posible obtener una representación matricial como:

$$\begin{aligned} p \otimes q &= \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix} \otimes \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \\ &= \begin{bmatrix} p_1q_1 - p_2q_2 - p_3q_3 - p_4q_4 \\ p_1q_2 + p_2q_1 + p_3q_4 - p_4q_3 \\ p_1q_3 - p_2q_4 + p_3q_1 + p_4q_2 \\ p_1q_4 + p_2q_3 - p_3q_2 + p_4q_1 \end{bmatrix}^T \end{aligned} \quad (3.12)$$

Tras definir la matriz en la ecuación 3.12, un vector de dimensión 3 puede ser rotado mediante un cuaternión utilizando la relación descrita en la ecuación 3.13. En donde Av y Bv son el mismo vector, descrito primero en la trama $\{A\}$ y posteriormente en la trama $\{B\}$ respectivamente. Sin embargo para ser congruentes con el producto entre dos cuaterniones, es necesario considerar a dicho vector de componentes x_v , y_v y z_v como los elementos de la parte vectorial de un cuaternión y considerar que la parte real es igual a cero[19]. Es decir: $v = [0 \quad x_v \quad y_v \quad z_v]$

$${}^Bv = {}^A_B\hat{q} \otimes {}^Av \otimes {}^A_B\hat{q}^* \quad (3.13)$$

Relación entre los Cuaterniones y las Matrices de Rotación

La orientación descrita mediante el cuaternión ${}^A_B\hat{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]$ puede ser representada como una Matriz de Rotación ${}^A_B R$ definida en la ecuación 3.14.[19]

$${}^A_B R = \begin{bmatrix} 2q_1^2 - 1 + 2q_2^2 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & 2q_1^2 - 1 + 2q_3^2 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & 2q_1^2 - 1 + 2q_4^2 \end{bmatrix} \quad (3.14)$$

Relación entre los Cuaterniones y los Ángulos de Euler

Como se describió anteriormente, los Ángulos de Euler ψ , θ y ϕ describen la orientación de una trama $\{B\}$ tras aplicar una secuencia de rotaciones desde su orientación alineada con la trama $\{A\}$. Considerando que ψ provoca una rotación alrededor de \hat{z}_B , θ alrededor de \hat{y}_B y ϕ alrededor de \hat{x}_B . Por lo tanto, la obtención de dichos ángulos a partir de un cuaternión ${}^A_B\hat{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]$ se logra mediante las ecuaciones 3.15 .[19]

$$\begin{aligned} \psi &= \arctan\left(\frac{2q_2q_3 - 2q_1q_4}{2q_1^2 + 2q_2^2 - 1}\right) \\ \theta &= -\arcsin(2q_2q_4 + 2q_1q_3) \\ \phi &= \arctan\left(\frac{2q_3q_4 - 2q_1q_2}{2q_1^2 + 2q_4^2 - 1}\right) \end{aligned} \quad (3.15)$$

3.5. Algoritmos de Orientación Espacial

Un algoritmo de estimación de la orientación toma las señales de cada uno de los sensores mencionados en la sección de Sensores Inerciales y fusiona dichas señales de forma estratégica para estimar la orientación del cuerpo al cual se implante [19]. Éste es un componente fundamental de cualquier sistema que implementen una IMU o MARG. Los algoritmos implementados en la actualidad hacen uso de las herramientas y entidades matemáticas descritas en la sección anterior, mediante la interpretación de las señales de los sensores, se obtienen vectores, matrices y cuaterniones que posteriormente son procesados. Existen ciertos algoritmos que utilizan técnicas distintas, basados en teorías diferentes. Por mencionar algunos:

- Filtros Complementarios
- Filtro de Kalman
- Algoritmo de Madgwick
- Algoritmo de Mahony

A pesar de que el origen de este tipo de sensores es de tipo bélico (Primeras IMU creadas para guiar misiles, orientar submarinos y aviones de combate), en la actualidad tienen varios usos civiles [19]. Una de las aplicaciones de estos dispositivos, es precisamente la realización de mediciones inerciales del cuerpo humano; entre las cuales, la orientación relativa entre dos eslabones propios del cuerpo.

Tomando en consideración que cada sensor presenta ventajas y desventajas, los algoritmos de estimación de orientación lo que pretenden sacar provecho de las ventajas de cada uno. **A través de la fusión de las señales de los tres sensores, se contrarrestan los posibles errores de orientación espacial para obtener la mejor estimación posible.** Por lo que a continuación se analizan los problemas a los que uno se enfrenta al tratar de obtener la orientación con cada uno de los sensores funcionando de manera aislada. Se ocupan como representación de la orientación, los ángulos de Euler, debido a la facilidad con la que se visualizan en comparación con otro tipo de representaciones.

3.5.1. Problemas de Medición en los Sensores

Acelerómetro

Los acelerómetros son dispositivos que pueden llegar a ser muy precisos a largo plazo. Es posible determinar incluso los ángulos de *pitch* y *roll* cuando el sensor se encuentra estático. Aplicando las ecuaciones 3.16, es posible determinar dichos ángulos. [25]

$$\begin{aligned}\tan(\theta) &= \frac{a_y}{a_z} \\ \tan(\phi) &= -\frac{a_x}{\sqrt{(a_y^2 + a_z^2)}}\end{aligned}\tag{3.16}$$

Donde a_x , a_y y a_z son las señales recabadas en los ejes inerciales del sensor x , y y z respectivamente.

Es importante hacer notar que un acelerómetro de tres ejes no es suficiente para conocer la orientación total del sensor en el espacio, ya que aún es necesario conocer el ángulo *yaw*. Suponiendo que el plano $x - y$ del sensor se encuentre alineado con el piso, no importa cuanto

rotemos el sensor alrededor de su eje z , seguirá marcando la misma dirección. Por tal motivo es necesario un segundo sensor para obtener dicha rotación.

Cuando el sensor se encuentra en movimiento, la aceleración del mismo ocasiona una alteración en el cálculo de la orientación [25]. Si se analiza la estructura del acelerómetro mostrada en la Fig. 3.3, es posible deducir que una aceleración en una dirección diferente a la del campo gravitatorio terrestre, podría llevar a la **interpretación de una orientación errónea** derivado a la suma de las componentes de aceleración debida al movimiento y la aceleración debido al campo gravitatorio.

Otro problema notorio de estos dispositivos, es la **alta sensibilidad al ruido**. Debido a la estructura que lo compone, los acelerómetros son capaces de captar incluso las mas ligeras vibraciones, es decir movimientos de alta frecuencia. En la práctica se utilizan filtros de tipo *Pasa-baja* para que el ruido ocasionado por fuerzas externas y vibraciones sea reducido lo mayormente posible y así captar movimientos de menor frecuencia.

Magnetómetros

Para obtener la dirección que nos hace falta mediante los acelerómetros, los magnetómetros parecieran ser la opción más adecuada. Mediante el mismo principio que las brújulas convencionales, la intensidad del campo magnético terrestre provoca una respuesta en cada eje de un magnetómetro. Si el plano $x - y$ se encuentra alineado con el piso, al medir esa respuesta (eléctrica) de dichos ejes podemos deducir el ángulo *yaw* mediante la ecuación 3.17. [25]

$$\psi = \tan^{-1} \left(\frac{m_y}{m_x} \right) \quad (3.17)$$

Donde m_y y m_x son las señales de intensidad de campo magnético en los ejes y y x del magnetómetro respectivamente.

Sin embargo, este tipo de sensores tienen una latencia alta, es decir que el tiempo en el que se obtiene una señal nueva es más alta en comparación con los otros sensores; es más lento. Sin mencionar que la orientación debe ser calibrada de acuerdo a la zona geográfica a la que se encuentre el sensor, debido a la declinación magnética. Debido a que el campo magnético terrestre no es uniforme, es preciso conocer la corrección pertinente.

Al ser sensores magnéticos, los magnetómetros son susceptibles a los cambios en el campo eléctrico del ambiente. Por lo que existe la posibilidad de encontrar gran cantidad de ruido en la cercanías de estructuras metálicas grandes, antenas de telecomunicaciones, instalaciones eléctricas, etc. Existen en particular dos fenómenos que provocan distorsiones en las mediciones del magnetómetro. Conocidas en inglés como «*hard - iron distortion*» y «*soft - iron distortion*» (Distorsiones de acero-duro y acero-blando respectivamente), provocan alteraciones en un patrón ideal de mediciones. Si no estuvieran presentes ninguna de ambas distorsiones, las lecturas de un magnetómetro que gira 360° alrededor de su eje z tendrían como resultado una gráfica similar a la mostrada en la Fig. 3.12. En donde los ejes x y y resultan en un círculo centrado en el origen $(0, 0)$. [26]

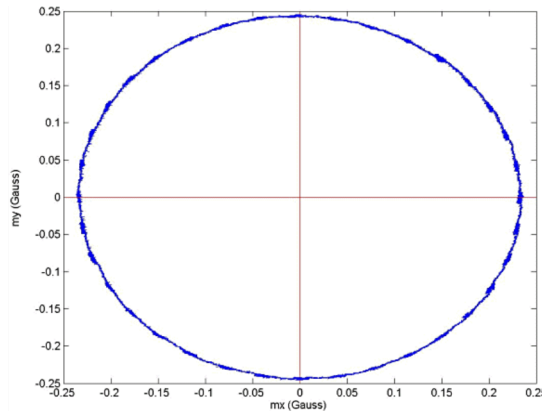


Figura 3.12: Lecturas de un magnetómetro libre de distorsiones magnéticas [26]

Las **distorsiones de acero-duro** son provocadas por materiales que producen de manera constante un campo magnético aditivo al terrestre y por lo tanto generan un valor aditivo a las mediciones de los sensores en cada uno de los ejes de un magnetómetro. Un claro ejemplo puede ser una bocina magnética que se encuentre en las cercanías del sensor. La distorsión por acero-duro podría ser identificado como un desfase del origen de un círculo, como puede apreciarse en la Fig. 3.13. Las correcciones a este tipo de distorsiones se logra realizando una lectura similar en cada uno de los planos del sensor para identificar el desfase en el origen de cada gráfica para posteriormente añadir dichos valores a las mediciones en cada eje. La corrección de la Inclinación del campo magnético terrestre, debe ser considerada anterior a la corrección de distorsiones por acero-duro. [26]

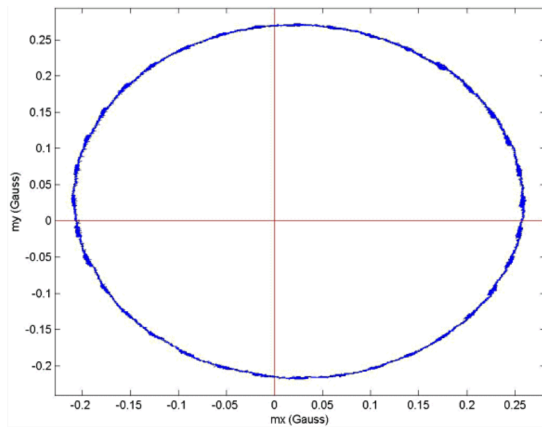
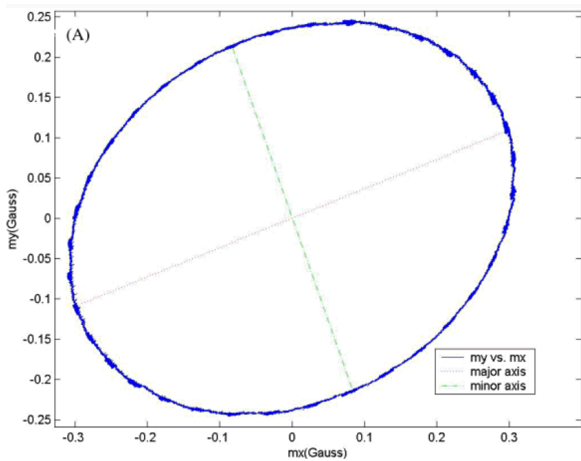
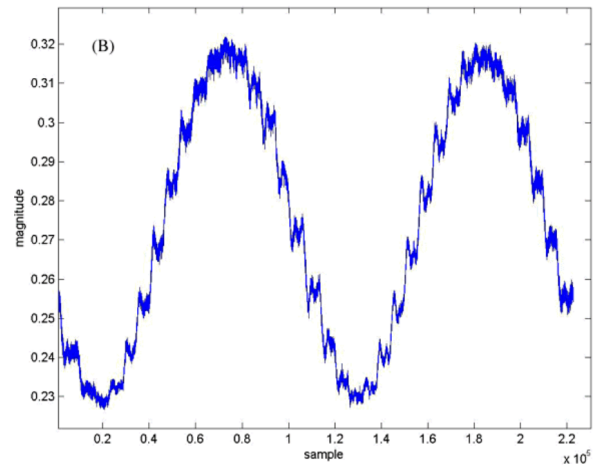


Figura 3.13: Distorsión de acero-duro en las lecturas de un magnetómetro [26]

A diferencia de la distorsión de acero-duro, las **distorsiones de acero-blando** son el resultado de la influencia o distorsión propia de materiales al campo magnético en las cercanías del sensor. No significa que estos materiales generen un campo magnético propio, por lo tanto no son aditivas las alteraciones que provocan. El hierro y el níquel son ejemplos de éstos. Mientras que la distorsión de acero-duro al campo magnético terrestre permanece constante (siempre y cuando dichos materiales no se desplacen), la distorsión de acero-blando es dependiente de la orientación del material que la provoca con respecto al magnetómetro y al campo magnético terrestre. De manera gráfica, la distorsión por acero-blando se visualiza como una transformación del círculo ideal mostrado en la Fig. 3.12 en una elipse como se muestra en la Fig. 3.14a. E incluso, si se grafican las magnitudes en el tiempo, es posible distinguir una onda compuesta de varias ondas a diferentes frecuencias (en particular dos) como se muestra en la Fig. 3.14b.



(a) Distorsión de acero-blando en las lecturas de un magnetómetro



(b) Distorsión en la magnitud del campo magnético sentido

Figura 3.14: Distorsiones de acero-blando [26]

Existen correcciones a este tipo de distorsiones que suelen ser complejas tomando en cuenta que se necesitan realizar operaciones matriciales para rotar y cambiar la excentricidad de las lecturas para acercarse a la lectura real. Sin embargo una manera de lograrlo e incluso ahorrar procesamiento es identificar los ejes que componen la elipse y asignar una escala a cada eje del sensor que multiplique la medición realizada de manera que se corrija la excentricidad de la elipse en sus ejes principales. [27]

Giroscopio

Los giroscopios nos permiten la velocidad angular del cuerpo al que se encuentre sujeto. Por lo tanto, si se desea conocer la posición angular resultante a dicha velocidad conocida es necesario conocer tanto la posición angular inicial, como el tiempo en el que se tarda en sensar la velocidad. Por consiguiente se integra de forma numérica para conocer la posición actualizada del sensor. En términos de los ángulos de Euler, se puede estimar la orientación mediante las ecuaciones simplificadas 3.18. [25]

$$\theta = \omega_{\theta} \times \Delta t, \quad \phi = \omega_{\phi} \times \Delta t, \quad \psi = \omega_{\psi} \times \Delta t \quad (3.18)$$

Donde ω_{θ} , ω_{ϕ} y ω_{ψ} son las velocidades angulares medidas por el giroscopio en las direcciones de los ángulos de Euler y Δt es el tiempo entre una medición y otra.

Sin embargo, el mas ligero error de medición, a medida que se realiza la integración numérica, se amplifica. A este fenómeno se le conoce como «*drift*». Incluso si el giroscopio regresa a un estado de reposo, el error acumulado continúa incrementando y jamás regresa a cero.[25] En la Fig. 3.15 se puede apreciar el resultado de una prueba estática a la que se sometió un giroscopio. Se obtuvieron las lecturas en un eje y posteriormente se integró el valor de manera numérica. Al tener un error de medición en el giroscopio de $0.0014 \left[\frac{rad}{s} \right]$, al cabo de media hora el error se vuelve bastante obvio. [28]

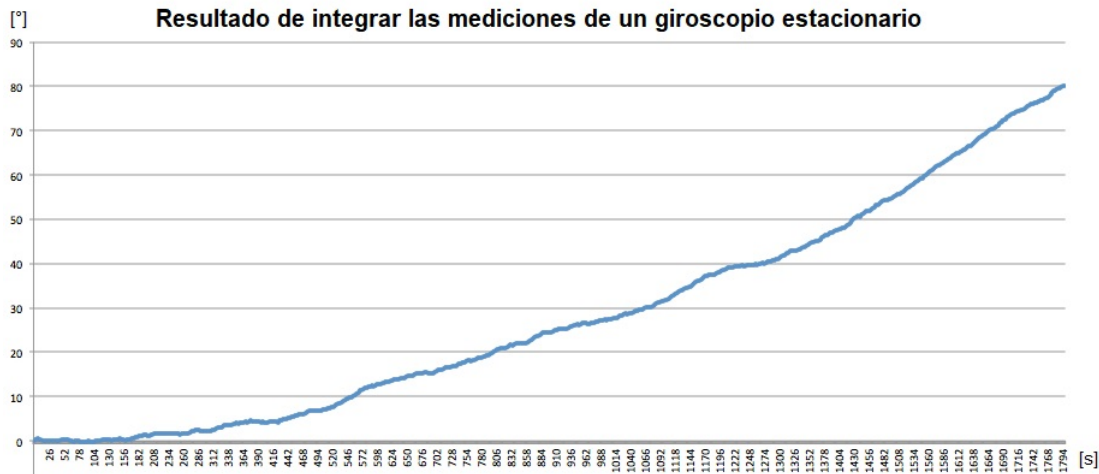


Figura 3.15: Ejemplo del *drift* en un giroscopio [28]

Estos dispositivos, son muy precisos en muestreos a corto plazo. Es decir que las señales de alta frecuencia que recaba son muy precisas, por tal motivo es muy común que se aplique un filtro de tipo *Pasa-altas* posterior a que se obtenga la velocidad angular.

3.5.2. Filtro de Madgwick [19]

Como una solución a los problemas que cada uno de los sensores presentan de forma aislada, Madgwick *et al.* propusieron un algoritmo de estimación de la orientación de un sensor tipo MARG aplicando los cuaterniones como herramienta fundamental del método. Dicho algoritmo es iterativo y depende tanto de las señales de salida del sensor como del tiempo de muestreo entre cada iteración. El algoritmo lo divide en seis secciones.

- **Orientación a partir de la velocidad angular:** En donde se aprovechan la razón de cambio de los desplazamientos angulares medidos por los giroscopios del sensor para obtener una razón de cambio del cuaternión equivalente.
- **Orientación a partir de la observación de vectores:** Aquí se toman tanto al vector unitario que representa la dirección de la fuerza gravitatoria terrestre (medido por los acelerómetros) como al vector unitario que representa la dirección del campo magnético terrestre (medido por los magnetómetros) como vectores de referencia para procesamientos posteriores.
- **Algoritmo de fusión de filtros:** Para esta sección se aprovechan las mediciones de las secciones anteriores para estimar un cuaternión que represente la orientación del sensor en el espacio.
- **Compensación de distorsión magnética:** En conjunto con el algoritmo de fusión de filtros se realiza una compensación de la distorsión magnética local, aprovechando parámetros de una calibración previa del magnetómetro en el sitio de estudio.
- **Compensación del *drift* del Giroscopio:** Al igual que la sección anterior, se realizan compensaciones al error conocido como *drift* que presentan los giroscopios, mediante parámetros de caracterización de los mismos.
- **Ganancias del Filtro:** A las ganancias del filtro se les vincula con el proceso de compensación del *drift* del giroscopio debido a que éstas ganancias son: el promedio de los errores de las mediciones en los giroscopios y la razón de convergencia a eliminar dichos errores.

movimiento humano [3]. De hecho se considera que el estudio del movimiento humano en sí se descompone en el área del análisis de fenómenos físicos catalogados en la biomecánica y por otro lado en el análisis del área clínica conocida como *Kinesiología*. Por lo cual, uno de los principales objetivos de la biomecánica es entender las funciones motoras del cuerpo humano. [21]

Duane Knudson propone que existen nueve principios fundamentales para la aplicación de la biomecánica en el análisis del movimiento humano. Uno de ellos es el principio de **Interacción Segmentada**. Dicho principio propone que las fuerzas que actúan en un sistema de cuerpos rígidos unidos pueden ser transferidas a través de juntas y eslabones.[30] Por lo cual es posible interpretar los movimientos del cuerpo humano como una serie de eslabones y juntas con diferentes propiedades a las cuales se les aplican una serie de fuerzas para reproducir el movimiento a analizar. De esta manera, se reduce el análisis a la interacción de cadenas cinemáticas abiertas o cerradas.

Aún al considerar este principio en particular, es importante conocer los conceptos clave de la anatomía humana relacionados con el movimiento humano.

3.6.1. Planos y Ejes Anatómicos

Para la identificación de la posición del cuerpo en un espacio de tres dimensiones en relación a una *posición anatómica*, se utilizan los **tres planos anatómicos: frontal, sagital y transversal**. [30] Al conjunto de estos planos ortogonales se le conoce como **planimetría**.

El plano *sagital*, es un plano vertical que divide el cuerpo u órgano en un lado derecho y uno izquierdo. El plano *frontal* (también llamado *ccoronal*), es un plano que divide el cuerpo u órgano en un lado anterior (frente) y uno posterior (espalda). Y el plano *transversal*, es un horizontal plano que divide el cuerpo u órgano en una porción superior y una inferior. [3]

Los ejes de intersección de los planos anatómicos forman los ejes alrededor del cual un cuerpo u órgano puede rotar. Al existir tres ejes de intersección, existen **tres ejes anatómicos: mediolateral, anteroposterior y superoinferior**.

El eje *mediolateral* (también llamado *bilateral, frontal, frontal-transversal, frontal-horizontal y transversal*) posee un recorrido angular *i.e.* de lado a lado, al ser un eje perpendicular al plano sagital. El eje *anteroposterior* (también llamado *sagital, sagital-horizontal y sagital-transversal*) posee un recorrido angular del frente hacia atrás, al ser un eje perpendicular al plano frontal. Y el eje *superoinferior* (también llamado *longitudinal, frontal-sagital y vertical*) posee un recorrido angular del frente hacia atrás, al ser un eje perpendicular al plano transversal. [3]

En la Fig. 3.17 se aprecian los planos de movimiento y ejes de rotación anteriormente mencionados. Es importante señalar que el movimiento no se restringe a el cuerpo en su totalidad, sino que los movimientos y rotaciones que se manifiesten en cualquier segmento del cuerpo humano puede ser referenciado a los planos y ejes anatómicos. En este sentido, y retomando el principio de Interacción segmentada, la cadena cinemática y los movimientos que describan su posición en el espacio, pueden ser descritos mediante las mismas referencias.

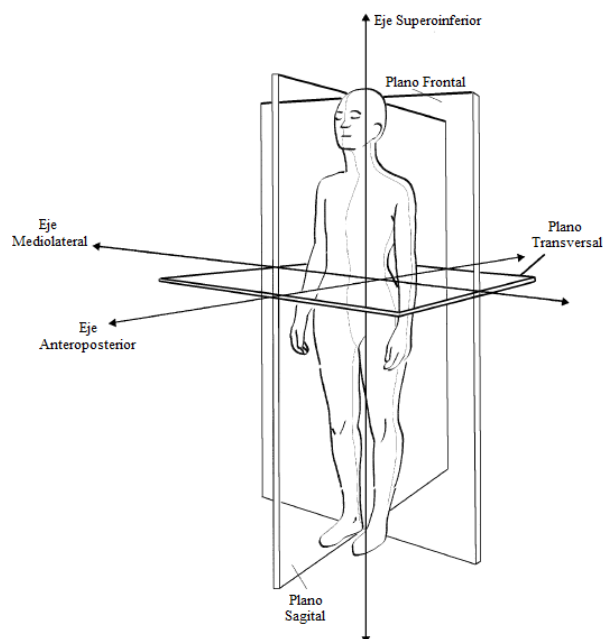


Figura 3.17: Diagrama de Planos y Ejes Anatómicos [30]

3.6.2. Vectores Biomecánicos [3]

En adición a los planos y ejes anatómicos en La Anatomía existen términos de dirección que nos permiten describir la posición de estructuras en relación con la posición anatómica. Dichas direcciones son conocidos como los **vectores biomecánicos**. Existen seis vectores biomecánicos paralelos y coincidentes con los ejes anatómicos, que se aprecian en el diagrama de la Fig. 3.18. [12 pt] Paralelos al eje superoinferior, existen los vectores **superior** e **inferior**. Con dirección a la cabeza se encuentra el vector superior (también llamado *cefálico*), mientras que con dirección a los pies se encuentra el vector inferior (también llamado *caudal*).

Paralelos al eje anteroposterior, existen los vectores **anterior** e **posterior**. Con dirección al frente del cuerpo se encuentra el vector anterior (también llamado *ventral*), mientras que con dirección a espaldas del cuerpo se encuentra el vector posterior (también llamado *dorsal*).

Paralelos al eje mediolateral, existen los vectores **medial** e **lateral**. Con dirección al interior del cuerpo se encuentra el vector medial (ya que apunta a la línea media del cuerpo), mientras que con dirección exterior del cuerpo se encuentra el vector lateral.

Existen también otros términos que describen la dirección de estructuras, aunque estos en particular no sean coincidentes a los ejes anatómicos:

- Con respecto del punto de inserción de un miembro al tronco: **Proximal** (más cerca del origen o del tronco) y **Distal** (más distante al tronco).
- Con respecto a la superficie del cuerpo humano: **Superficial** (hacia o sobre la superficie) y **Profundo** (más distante a la superficie).

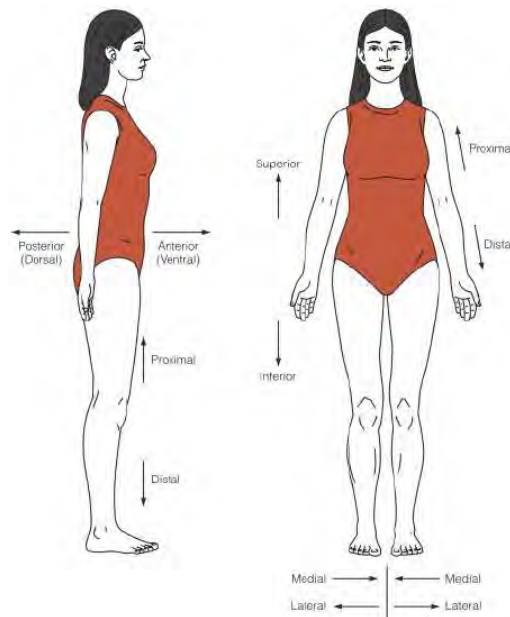


Figura 3.18: Diagrama de los Vectores Biomecánicos [3]

3.6.3. Biomecánica de Miembro inferior

En Anatomía, se considera que el Sistema Esquelético tiene dos divisiones principales: el esqueleto axial y el apendicular. El esqueleto axial tiene como función principal la protección de los órganos internos. Por otro lado, el esqueleto apendicular adopta un papel importante en el movimiento humano. [31]

El esqueleto apendicular incluye huesos que forman los miembros superiores e inferiores, así como los huesos de las dos cinturas (escapular y pélvica) que fijan los miembros al esqueleto axial, como se aprecia en la Fig. 3.19. Los huesos que conforman al esqueleto apendicular están conectados entre sí y en función conjunta con los músculos esqueléticos permite realizar actividades tales como escribir, nadar, tocar un instrumento musical, bailar y caminar. [31]

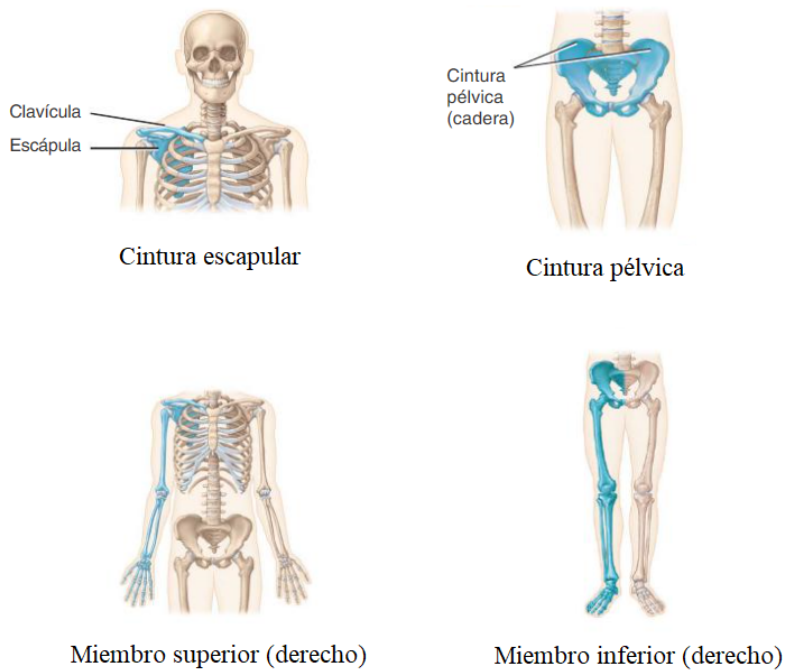


Figura 3.19: Cinturas del Esqueleto Apendicular [31]

Cada miembro inferior (extremidad inferior o «pierna») tiene 30 huesos en cuatro localizaciones como se muestra en la Fig. 3.20:

- Fémur en el muslo
- Rótula
- Tibia y peroné en la pierna
- Los 7 huesos tarsianos del tarso (tobillo), los 5 metatarsianos del metatarso y las 14 falanges (huesos de los dedos)

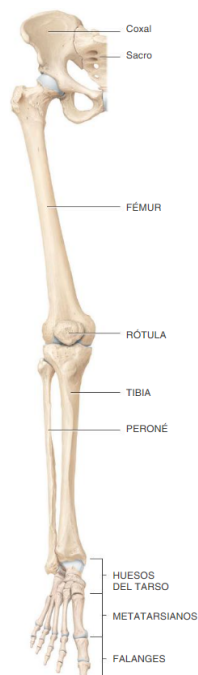


Figura 3.20: Huesos del miembro inferior derecho [31]

En el campo de la biomecánica se desarrolló un modelo de la absorción y distribución de fuerzas en las actividades en las cuales las articulaciones se someten a distintos tipos de cargas. Para este modelo se consideran a los huesos como cuerpos rígidos, isotrópicos, homogéneos y que las fuerzas transmitidas a través de ellos siguen la trayectoria más óptima. [3]

En ocasiones esta trayectoria suele coincidir de forma muy aproximada con el eje longitudinal del hueso en cuestión. A éste eje se le denomina **eje anatómico**.

Sin embargo, existen ocasiones en el que el eje anatómico del hueso, no coincide con la trayectoria óptima de transmisión de fuerza. Para conseguir dicha trayectoria, se identifican los centros de rotación de los extremos distal y proximal del hueso. Posteriormente se traza una línea recta entre ambos centros y se obtiene un eje conocido como el **eje mecánico**.

En el caso particular del miembro inferior, es posible distinguir que para la tibia, el eje anatómico y el eje mecánico coinciden. No obstante, para el fémur el eje mecánico del fémur discrepa alrededor de 6° con respecto del eje anatómico como se aprecia en la Fig. 3.21. [3] Si se prolongara el eje mecánico del fémur hasta el eje mecánico de la tibia, de forma ideal se observaría una línea recta compuesta por ambas.

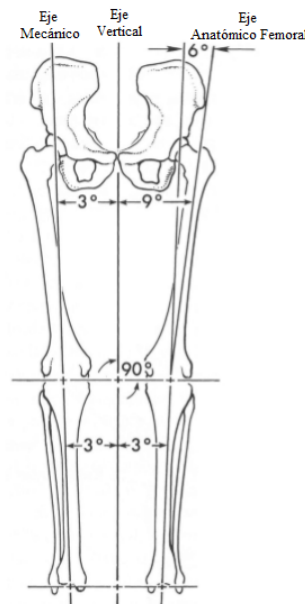


Figura 3.21: Ejes Mecánico y Anatómico en fémur y tibia [32]

El estudio de la variación de este ángulo es de gran importancia debido a que puede ser un factor determinante de patologías de la rodilla y por ende movimientos atípicos de la *marcha humana*. Tales como el *genu varo* (rodillas arqueadas en «O») y el *genu valgo* (rodillas arqueadas en «X») mostradas en la Fig. 3.22.

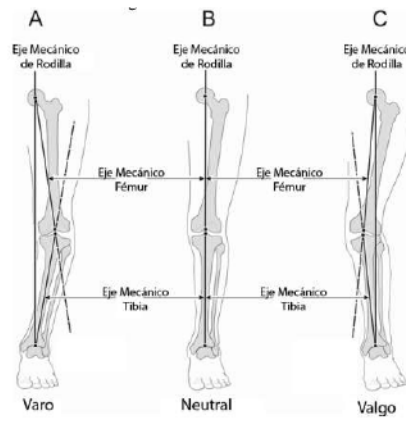


Figura 3.22: Definición de rodilla neutra, en varo y valgo acorde al eje mecánico de rodilla [3]

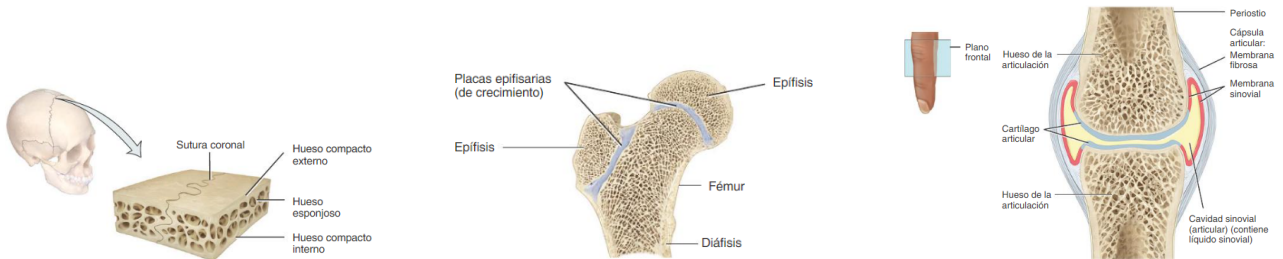
3.6.4. Movimientos en Articulaciones de Miembro Inferior [31]

Articulaciones

Los huesos son demasiado rígidos para doblarse sin dañarse. Por fortuna, las articulaciones que mantienen unidos a los huesos están compuestas por tejido flexible que, en la mayoría de los casos permite cierto grado de movimiento.

Las articulaciones se clasifican de acuerdo con su estructura; es decir, sobre la base de las características anatómicas y con su función, de acuerdo con el tipo de movimiento que permiten:

- **Articulaciones fibrosas:** No existe cavidad sinovial y los huesos se mantienen unidos por tejido conectivo denso irregular, rico en fibras de colágeno. (Por ejemplo, las suturas que se aprecian en el cráneo. Obsérvese la Fig. 3.23a)
- **Articulaciones cartilagosas:** Tampoco existe cavidad sinovial y los huesos se mantienen unidos mediante cartílago. (Por ejemplo las placas epifisarias entre las epífisis y el fémur. Obsérvese la Fig. 3.23b)
- **Articulaciones Sinoviales:** Los huesos que forman la articulación tienen una cavidad sinovial y están unidos por una cápsula articular de tejido conectivo denso irregular y a menudo por ligamentos accesorios. (Por ejemplo las articulaciones entre las falanges del dedo. Obsérvese la Fig. 3.23c)



(a) Articulación Fibrosa: Sutura entre los huesos del cráneo.

(b) Articulación Cartilaginosa: Placas epifisarias.

(c) Articulación Sinovial: Articulaciones entre las falanges del dedo.

Figura 3.23: Clasificación de las articulaciones [31]

A su vez, existe la clasificación funcional que se relaciona con el grado de movimiento que permiten: [31]

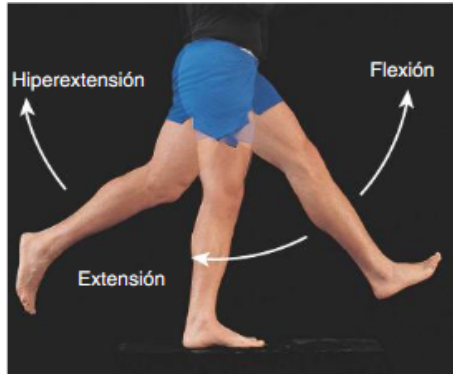
- **Sinartrosis:** Articulación inmóvil
- **Anfiartrosis:** Articulación con movimiento limitado
- **Diartrosis:** Articulación con gran movimiento. Todas las diartrosis son articulaciones sinoviales. Tienen una gran variedad de formas y permiten muchos tipos de movimientos diferentes.

Debido al principio de Interacción Segmentada propuesto por Duane Knudson, el tipo de articulaciones que permiten analizar el movimiento humano reduciéndolo a un movimiento de una cadena cinemática, son las articulaciones sinoviales.

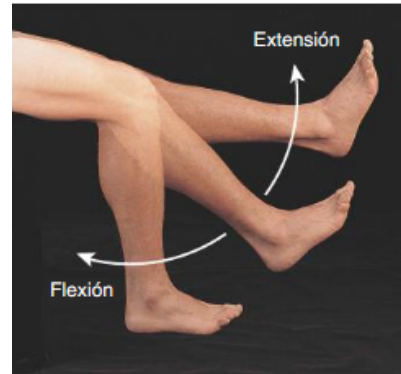
Movimientos Angulares de Articulaciones Sinoviales de Miembro Inferior

En los movimientos angulares, de manera generalizada se produce un incremento o una disminución del ángulo entre los huesos de la articulación. Sin embargo, la disposición de los huesos en los segmentos de miembros inferiores, no permiten en ocasiones la amplia gama existente de estos movimientos. Para miembros inferiores, los movimientos angulares más importantes son:

- **Flexión:** Disminuye el ángulo entre los huesos de la articulación (Fig. 3.24a y 3.24b).
- **Extensión:** Es el movimiento opuesto a la flexión. Se incrementa el ángulo entre los huesos de la articulación (Fig. 3.24a y 3.24b).
- **Hiperextensión:** Es la continuación de la extensión más allá de la posición anatómica (Fig. 3.24a y 3.24b).
- **Abducción:** Es el desplazamiento angular en el cual el hueso se aleja de la línea media del cuerpo (o del plano sagital) (Fig. 3.24d).
- **Aducción:** Es el movimiento contrario a la abducción en donde el hueso se aproxima a la línea media del cuerpo (Fig. 3.24d).
- **Circuducción:** Es el movimiento circular del extremo distal de un segmento. No se considera un movimiento aislado, sino como una secuencia continua de flexión, abducción, extensión, aducción y rotación de la articulación (o en orden contrario) (Fig. 3.24c).
- **Rotación:** Es el movimiento en donde un hueso gira alrededor de su eje longitudinal. Se pueden considerar los vectores biomecánicos para indicar la dirección de rotación tomando en cuenta la porción distal del hueso que rota (*e.g.* medial o lateral) (Fig. 3.24e).
- **Inversión:** Es aquel movimiento angular en sentido medial de las partes de las plantas de los pies a nivel de las articulaciones intertarsianas (*i.e.* rotar hacia dentro). En ocasiones los fisioterapeutas denominan a este movimiento como *supinación* (Fig. 3.24f).
- **Eversión:** Movimiento opuesto a la Inversión (*i.e.* rotar hacia fuera). En ocasiones los fisioterapeutas denominan a este movimiento como *pronación* (Fig. 3.24f).
- **Dorsiflexión:** Es el movimiento del pie en el tobillo o articulación talocrural (entre la tibia, peroné y astrálogo) en la dirección del dorso (empeine) (Fig. 3.24g).
- **Flexión plantar:** Implica doblar el pie en la articulación del tobillo en dirección de la superficie plantar. Es el movimiento opuesto a la dorsiflexión (Fig. 3.24g).



(a) Flexión, extensión e hiperextensión en cadera



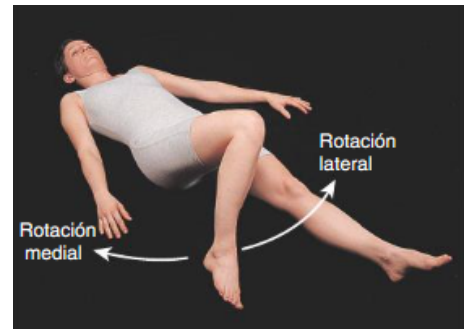
(b) Flexión y extensión en rodilla



(c) Circunducción en cadera



(d) Abducción y aducción en cadera



(e) Rotaciones media y lateral en cadera



(f) Inversión y eversión en articulaciones intertarsianas



(g) Dorsiflexión y flexión plantar en articulación del tobillo

Figura 3.24: Movimientos Angulares de Miembros Inferiores [31]

3.6.5. Análisis de la Marcha Humana

Movimientos de locomoción humana que puede incluir el caminar, correr, ascenso y descenso de escaleras, entre otros son bien conocidos y estudiados. La **marcha humana** es una actividad cíclica para la cual ciertos eventos discretos han sido identificados como significativos. Típicamente el **Ciclo de Marcha** es definido como el período de tiempo desde el punto de contacto inicial (referido como *contacto del pie*) del pie de un individuo con el suelo hasta el siguiente punto de contacto inicial para la misma extremidad. [30] Como se muestra en la Fig. 3.25, el ciclo se divide al ciclo en dos fases: la fase de apoyo y la fase de balanceo. Una extremidad se encuentra en fase de apoyo cuando está en contacto con el suelo y posteriormente se encuentra en fase de balanceo cuando no lo está. Adicionalmente, se identifica el apoyo sencillo cuando sólo una extremidad se encuentra en contacto con el suelo y el apoyo doble ocurre cuando ambas extremidades lo están.

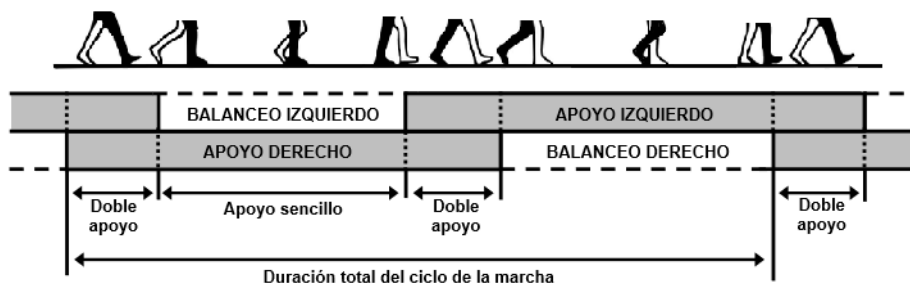


Figura 3.25: El ciclo de marcha y sus fases

Según el Instituto de Biomecánica de Valencia, a una velocidad normal de 100 a 115 pasos por minuto, se divide el tiempo que conlleva completar un ciclo de marcha en el porcentaje correspondiente a un ciclo completo. Esta distribución se observa en la Fig. 3.26 que a la fase de apoyo toma un 60 %, para la fase de balanceo toma un 40 % y ya completado el ciclo se considera un 20 % adicional a la fase de doble apoyo (*i.e* durante el 60 % correspondiente a la fase de apoyo, el 20 % es de un apoyo en ambos pies al suelo).

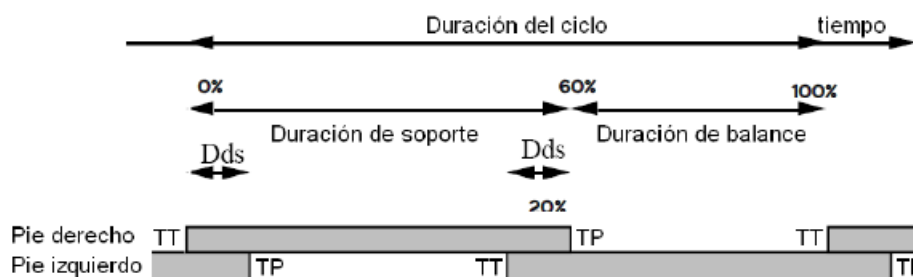


Figura 3.26: Estructura Temporal del Ciclo de Marcha. Donde *Dds* es duración de doble soporte, *TT* es toque de talón y *TP* es es toque de punta

Tanto la marcha como la gama de movimientos inicialmente mencionados, son frecuentemente analizados de forma cuantitativa por profesionales de la biomecánica del deporte, fisioterapeutas y médicos con la finalidad de evaluar las funciones de las extremidades del miembro inferior. En la actualidad existe una variedad de análisis de la marcha humana tanto cualitativa como cuantitativamente. [29]

Con respecto al análisis de la marcha humana en el marco clínico, los profesionales médicos miden y analizan los patrones de la marcha en pacientes con patologías de la locomoción humana

con finalidades tales como determinar protocolos de tratamiento (*e.g.* prescripciones ortopédicas o intervenciones quirúrgicas).[30] Algunos ejemplos de patologías clínicas en las que se implementa este tipo de análisis son:

- Amputación
- Parálisis cerebral
- Enfermedades degenerativas en articulaciones
- Poliomielitis
- Esclerosis múltiple
- Distrofia muscular
- Mielodisplasia
- Artritis reumatoide
- Lesión de columna vertebral

Componentes Análisis Clínico de la Marcha [30]

Un análisis clínico de la marcha comprende una variedad de componentes, cuya combinación depende de la patología o habilidad (relacionada con el movimiento a estudiar) individual del sujeto de estudio (paciente). Los datos que actualmente proveen una evaluación de la marcha en el marco clínico son:

- Grabación en video del andar del individuo (anterior a la instrumentación) con el propósito de realizar revisiones cualitativas y control de calidad.
- Medidas de revisión médica en reposo, tales como el rango de movimiento de articulaciones pasivas, fuerza muscular, habilidad de aislar el movimiento, tono muscular, y presencia y grado de deformación en huesos.
- Posiciones angulares de segmentos y articulaciones en postura erguida. Medido mediante instrumentos llamados *Goniómetros*.
- Parámetros de zancada, tales como longitud de paso y velocidad de caminata.
- Desplazamientos angulares de segmentos y articulaciones durante la marcha (en plano). Donde también se implementan los goniómetros.
- Las fuerzas reactivas del suelo al pie del paciente durante la marcha
- Par de fuerzas reactivas intersegmentales producidas alrededor de las articulaciones de miembros inferiores por fuerzas de tejido blando (activo y pasivo), así como la potencia mecánica asociada del par de fuerzas intersegmentales durante la marcha.
- Indicadores de actividad muscular, comúnmente potenciales de voltaje producidos por músculos en contracción durante la marcha, en posición erguida relajada y evaluación del tono muscular. Conocido como *Electromiografía* dinámica (EMG).
- La distribución dinámica de la presión en la superficie plantar del pie durante la marcha y postura erguida, comúnmente conocido como *Podobarografía*.

- La medición del gasto de energía metabólica durante el reposo (al estar sentado) y durante la marcha (*e.g.* consumo de oxígeno).

Como se puede distinguir de la lista anterior, existen una variedad considerable de tecnologías dedicadas a la medición de variables dinámicas de la marcha: Cámaras de video, goniómetros, plataformas de fuerzas, electromiógrafos. De las cuales unas no son sujetadas al sujeto de estudio y otras forman parte del ambiente donde se desarrolla la prueba. En la Fig. 3.27 se pueden distinguir los instrumentos mayormente ocupados en análisis de marcha.

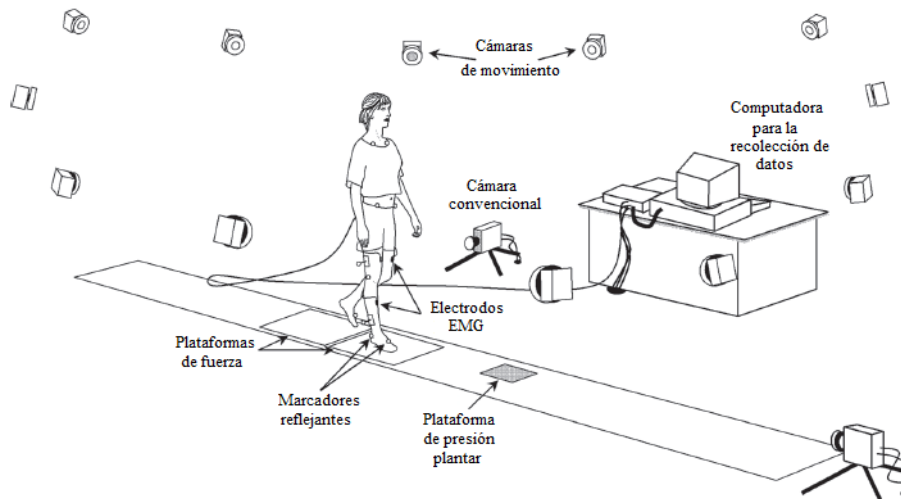


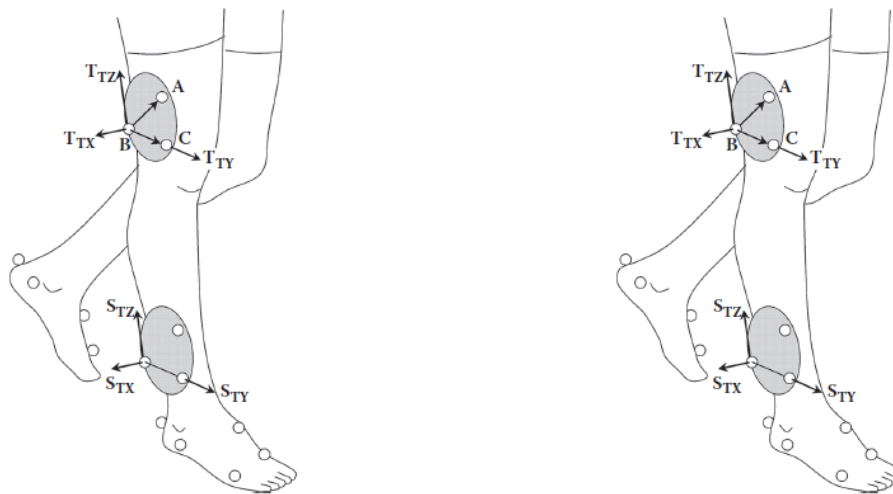
Figura 3.27: Una paciente «instrumentada» con marcadores y electrodos de EMG. Ella camina sobre un camino nivelado siendo monitoreada por 6-12 cámaras que captan el desplazamiento de los marcadores reflectantes. También camina por encima de plataformas de presión que captan la distribución de la carga en la zona plantar de cada pie. Su andar es capturado por 2 cámaras convencionales. Todas las señales captadas son enviadas a una computadora en el laboratorio, para posteriormente ser procesadas por el operador con la finalidad de producir información útil para el análisis clínico correspondiente [30]

El enfoque predominante para la recolección de datos de la marcha, involucra la colocación de marcadores externos en la superficie de los segmentos del cuerpo siendo alineados con puntos anatómicos particulares. Estos marcadores son colocados sobre el paciente ya sea como puntos discretos o segmentos de cuerpos rígidos conectados. [30]

Estos marcadores son colocados al paciente, de tal forma que el producto de los datos recolectados permita obtener un sistema de coordenadas para cada marcador (o conjunto de marcadores) que permitan asignarle a cada segmento del cuerpo su propio sistema de coordenadas y por consiguiente seis grados de libertad asociados a la traslación y orientación en el espacio.

Para el caso de la colocación de marcadores reflectantes en muslo y pierna, en la Fig. 3.28a se puede apreciar una distribución de tres marcadores en el muslo A , B y C que forman parte de un mismo plano. Realizando el producto cruz de los vectores \overrightarrow{BA} y \overrightarrow{BC} produce un vector perpendicular al plano de los marcadores. De estos vectores, los vectores unitarios T_{TX} y T_{TY} deben ser determinados para calcular el tercer vector ortogonal T_{TZ} para completar el sistema de coordenadas. De manera similar se obtiene el sistema de coordenadas para la pierna S_{TX} , S_{TY} y S_{TZ} . Una vez obtenidos ambos sistemas, es posible determinar la orientación de un segmento con respecto del otro o del sistema de referencia inercial del laboratorio. Como se observa en la Fig. 3.28b cada sistema de coordenadas de los conjuntos de marcadores se asocian con un sistema del segmento del cuerpo mismo. Para esto se asume que uno de los vectores

unitarios de este sistema de aproxima al eje anatómico del segmento del cuerpo (para el ejemplo de las Fig. 3.28 el vector T_{TZ} se aproxima al eje anatómico femoral). De hecho en enfoques mas rigurosos, los pacientes se someten a pruebas de calibración de tal manera que la aproximación del sistema de coordenadas con los ejes anatómicos correspondientes sea óptima. [30]



(a) Colocación de marcadores en muslo y pierna y obtención del sistema de coordenadas ortogonal para conjunto de tres marcadores [30]

(b) Asociación del sistema de coordenadas del conjunto de marcadores con un sistema propio de cada segmento (muslo y pierna) [30]

Figura 3.28: Colocación de marcadores reflejantes

Capítulo 4

Metodología

4.1. Planteamiento del problema

En distintas disciplinas y áreas del conocimiento se analiza la marcha humana con propósitos específicos diversos. Para la identificación de patologías de la marcha humana, tratamientos ortopédicos y de rehabilitación, mejorar el rendimiento de un atleta, etc. Dentro de este análisis biomecánico es necesario conocer con precisión los desplazamientos angulares de las articulaciones involucradas en la locomoción humana. Por tal motivo, se han desarrollado soluciones con diferentes tipos de tecnologías para realizar las mediciones antes mencionadas.

Para la medición específica de los cambios en posición angular de los segmentos del cuerpo, los dispositivos mayormente implementados son los goniómetros. Los sistemas de goniometría que se han desarrollado consisten en elementos rígidos o flexibles con un centro de rotación en común donde se mide mediante transductores el movimiento angular correspondiente a la deformación de la estructura del equipo. Una importante desventaja de este tipo de sistemas es la presencia de fuerzas reactivas que ejerce la estructura al sujeto de estudio, debido a la rigidez o las limitaciones de deformación del sistema en sí.

Existen sistemas alternativos que miden el movimiento de los segmentos del cuerpo mediante el procesamiento de señales proporcionadas por sensores inerciales (IMU o MARG). De dichas mediciones (dependiendo de los algoritmos de adquisición y procesamiento de las señales) es posible determinar la orientación y posición en el espacio del sensor (y por ende del cuerpo al cual se acopla). Como cada sensor realiza las mediciones correspondientes de forma aislada, no es necesario que existan elementos rígidos y/o flexibles que los conecten (salvo los sistemas que conectan en paralelo o en serie los sensores mediante cables de datos hacia un procesador central).

Por consiguiente, es necesario diseñar una solución alternativa capaz de medir los movimientos angulares de manera óptima, disminuyendo en cierta medida las desventajas que presentan los diseños comerciales actuales.

4.1.1. Pregunta de investigación

Considerando lo anteriormente mencionado; si los goniómetros convencionales llevan consigo desventajas inherentes a su propia estructura, mientras que los sensores inerciales pueden operar sin que existan eslabones que los unan, ¿es posible diseñar un sistema de goniometría que implemente la tecnología que ofrecen los sensores inerciales?

4.1.2. Justificación

El proyecto a desarrollar, inicialmente se consideró como el diseño de un producto solicitado por el Laboratorio de Biomecánica de la Universidad Nacional Autónoma de México (UNAM) en el Campus Juriquilla con la finalidad de poseer una alternativa a la instrumentación que se ocupa actualmente.

Incluso, una vez establecidos los alcances en dicho proyecto surge la iniciativa del diseño de un dispositivo ortopédico *biomecatrónico* innovador en el Centro de Diseño Mecánico e Innovación Tecnológica (CDMIT) de la División de Ingeniería Mecánica e Industrial (DIMEI) de la Facultad de Ingeniería de la UNAM. Formando parte de un conjunto de subsistemas que integraran a dicho dispositivo, el subsistema de «Sensado Inercial» tomaba en consideración el desarrollo de un prototipo que fuera capaz de sensar los ángulos de flexo-extensión en la rodilla de manera que se pudieran aprovechar los datos como las entradas de un subsistema de control.

4.2. Propuesta de solución: Sistema de Goniometría Propuesto

El objetivo del funcionamiento del Sistema de Goniometría Propuesto (en adelante, el SGP), tomando en cuenta el principio de *Interacción Segmentada*, es conocer los desplazamientos angulares relativos entre dos eslabones de una cadena cinemática que se mueve a través del espacio. Por lo cual, para el diseño de una solución que cuente con la tecnología que implementen los sensores MARG, **se asume que dichos desplazamientos angulares se toman a partir de conocer los cambios de las orientaciones del sistema de coordenadas que tiene como referencia cada sensor**. Si se conoce la orientación de un cuerpo, definido mediante una trama, y se compara con la trama que define un segundo eslabón unido al primero por una articulación, entonces conocemos los ángulos relativos entre ambos eslabones.

4.3. Diseño a Nivel Sistema

4.3.1. Sistemas funcionales

Para definir los sistemas funcionales de la solución propuesta, es necesario reconocer dos partes fundamentales que integran al SGP inalámbrico, en su totalidad: **los transmisores** y **el receptor**. Un arreglo de transmisores con los sensores tipo MARG, así como un módulo de transmisión inalámbrica integrados a cada uno podrían enviar los datos al receptor para su debido procesamiento; y una vez capturados los datos se podría obtener la orientación en el espacio de cada sensor (transmisor). En la Fig. 4.1 se representa de forma gráfica la relación entre los componentes del sistema (transmisor y receptor).

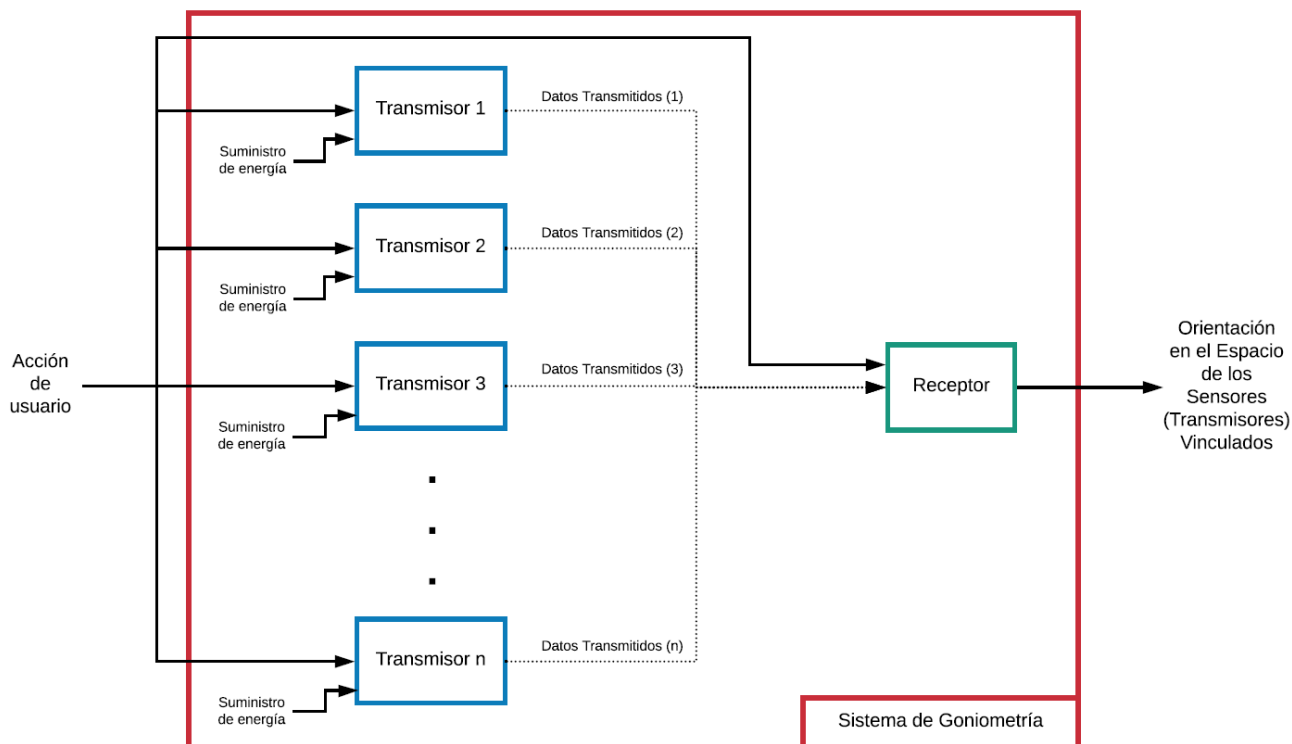


Figura 4.1: Diagrama del SGP

Se considera a la entrada del sistema como «Acción del usuario» a toda interacción de configuración, vinculación o encendido/apagado que ejerce el usuario al subsistema en particular. Como salida se observan las orientaciones en el espacio (Representada mediante cuaterniones al ser el método de representación en el espacio seleccionado) de cada sensor vinculado al mismo receptor. Cabe resaltar que al ser un sistema inalámbrico es necesario que cada transmisor se conecte al receptor mediante un «bus» para la transmisión de los datos del sensor inercial (representado mediante la línea punteada mostrada). Una vez considerados el transmisor y el receptor como dos elementos aislados, para cada uno se determinan los sistemas funcionales que lo integran.

Transmisor

En la Fig. 4.2 se muestran los sistemas funcionales que integran a cada transmisor, así como las entradas que permiten su correcto funcionamiento y la salida de datos (enviadas de manera inalámbrica al transmisor).

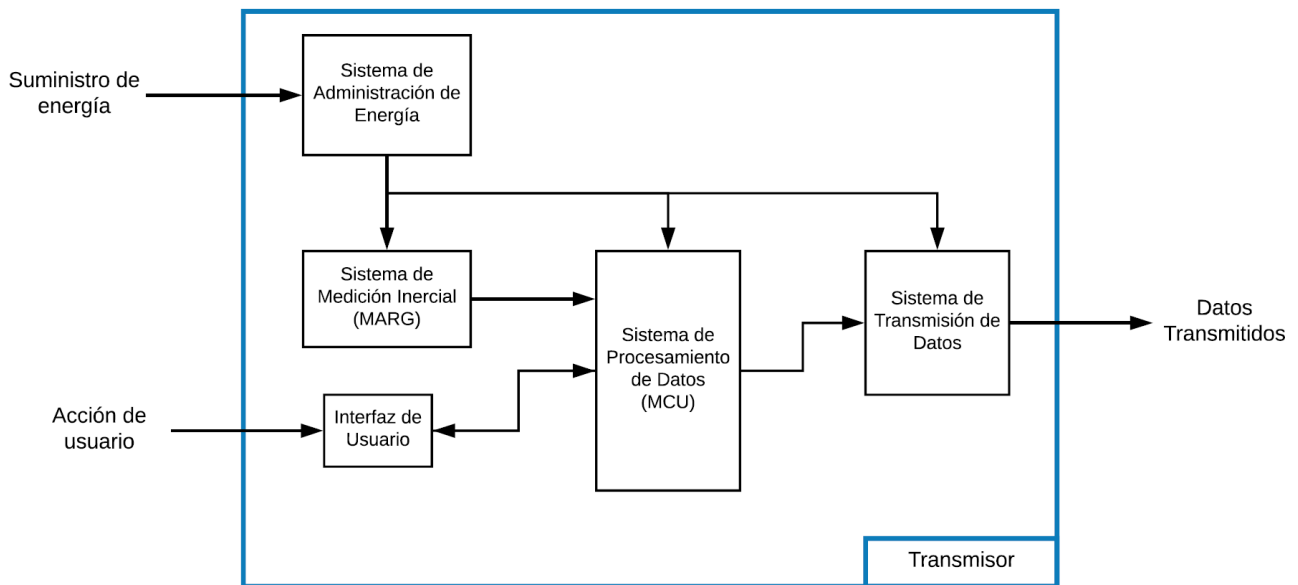


Figura 4.2: Diagrama de los sistemas funcionales en el transmisor

Receptor

En la Fig. 4.3 se muestran los sistemas funcionales que integran al receptor. Se observa como entradas los datos que recibe de cada transmisor vinculado así como las acciones del usuario para la configuración del sistema (arreglo de transmisores) y la recolección de datos.

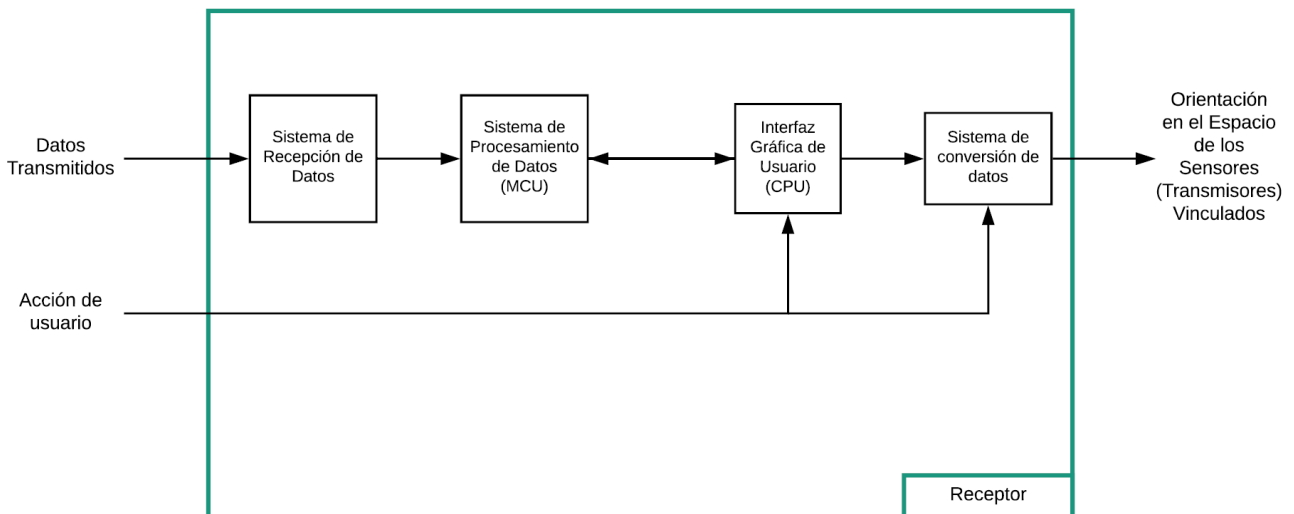


Figura 4.3: Diagrama de los sistemas funcionales en el receptor

En la Fig. 4.4 se ve representado de manera sencilla la disposición que tendrán tanto el receptor como los transmisores en una prueba de marcha.

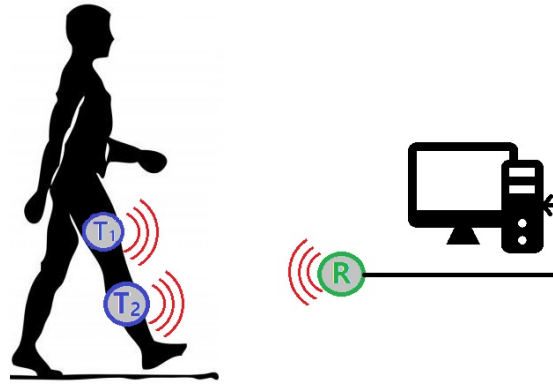


Figura 4.4: Representación gráfica de la solución propuesta

4.3.2. Generación y selección de conceptos

Para la selección de conceptos se realiza una carta morfológica para exponer las opciones de solución para cada sistema funcional. Posteriormente se discuten las ventajas y desventajas de la selección para acotar la carta y proponer una nueva. Cabe resaltar que **para la solución propuesta el sistema fundamental para el funcionamiento deseado es la Unidad de Medición Inercial**, por lo cual se establece en primera instancia la configuración ideal para el Transmisor.

Transmisor

La primer propuesta se muestra en la Tabla 4.1

Energía	Medición Inercial	Procesamiento de Datos	Transmisión
Baterías desechables (AAA o AA)	IMU	Microcontrolador Atmel	Bluetooth
Baterías recargables (AAA o AA)	MARG	Microcontrolador Texas	Transmisión por Radiofrecuencia
Batería tipo Botón		Microcontrolador Microchip	Wi-Fi
Batería LiPo			

Tabla 4.1: Carta morfológica para el transmisor

Una vez identificadas las posibles soluciones a cada sistema funcional, se analizan las siguientes consideraciones:

- Como se analizó en el Capítulo 1, es necesario la implantación de un sistema MARG para el funcionamiento óptimo del algoritmo deseado.
- La facilidad de uso y sincronización (siendo este uno de los objetivos particulares) se ve comprometida con el uso de baterías desechables o recargables debido a que eventualmente se habrán de extraer del transmisor para su cambio y/o recarga.

⁰Debido a que en año 2016 la empresa Atmel Corporation fue adquirida por Microchip Technology, se realiza la diferencia entre una y otra por los productos de mayor uso por estudiantes: *PIC* de Microchip Technology y *ATmega* (y *ATtiny*) de Atmel Corporation.

Por lo cual, las opciones se reducen a las mostradas en la Tabla 4.2:

Energía	Medición Inercial	Procesamiento de Datos	Transmisión
Batería LiPo	MARG	Microcontrolador Atmel	Bluetooth
Batería tipo Botón		Microcontrolador Texas	Transmisión por Radiofrecuencia
		Microcontrolador Microchip	Wi-Fi

Tabla 4.2: Carta morfológica modificada para el transmisor

Tras reducir la carta las opciones más viables acorde a la combinación de soluciones en la carta morfológica se tienen las siguientes configuraciones:

1. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Atmel (*Arduino*), transmitidos mediante Bluetooth. Circuito alimentado mediante batería LiPo.
2. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Texas (*Launchpad*), transmitidos mediante Bluetooth. Circuito alimentado mediante batería LiPo.
3. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Microchip (*PIC*), transmitidos mediante Bluetooth. Circuito alimentado mediante batería LiPo.
4. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Atmel (*Arduino*), transmitidos mediante un Transmisor de radiofrecuencia. Circuito alimentado mediante batería LiPo.
5. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Texas (*Launchpad*), transmitidos mediante Transmisor de radiofrecuencia. Circuito alimentado mediante batería LiPo.
6. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Microchip (*PIC*), transmitidos mediante Transmisor de radiofrecuencia. Circuito alimentado mediante batería LiPo.
7. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Atmel (*Arduino*), transmitidos mediante WI-FI. Circuito alimentado mediante batería LiPo.
8. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Texas (*Launchpad*), transmitidos mediante WI-FI. Circuito alimentado mediante batería LiPo.
9. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Microchip (*PIC*), transmitidos mediante WI-FI. Circuito alimentado mediante batería LiPo.
10. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Atmel (*Arduino*), transmitidos mediante Bluetooth. Circuito alimentado mediante batería tipo botón.
11. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Texas (*Launchpad*), transmitidos mediante Bluetooth. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.

12. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Microchip (*PIC*), transmitidos mediante Bluetooth. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.
13. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Atmel (*Arduino*), transmitidos mediante un Transmisor de radiofrecuencia. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.
14. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Texas (*Launchpad*), transmitidos mediante Transmisor de radiofrecuencia. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.
15. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Microchip (*PIC*), transmitidos mediante Transmisor de radiofrecuencia. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.
16. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Atmel (*Arduino*), transmitidos mediante WI-FI. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.
17. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Texas (*Launchpad*), transmitidos mediante WI-FI. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.
18. Procesamiento de datos adquiridos del sensor MARG mediante microcontrolador Microchip (*PIC*), transmitidos mediante WI-FI. Circuito alimentado mediante Circuito alimentado mediante batería tipo botón.

Analizando las combinaciones posibles se discuten las ventajas y desventajas de cada uno y a continuación se enlistan las consideraciones aplicables a las configuraciones de la solución:

- Las baterías tipo botón representan una ventaja considerable en cuanto al tamaño. Sin embargo, al igual que las baterías tipo AA o AAA, existen los modelos recargables y desechables; es decir traen consigo las desventajas de dichas opciones. En cuanto a la capacidad de las baterías tipo botón, las que se encuentran en el mercado existe una notable diversidad, no obstante las que son recargables tienen un costo considerablemente elevado.
- Las pruebas a realizar para el procesamiento de datos adquiridos del sensor MARG, fácilmente pueden realizarse con las tres opciones (especialmente con las tarjetas de desarrollo que ofrecen tanto Atmel como Texas). Sin embargo, se consideran ciertas ventajas en la programación por parte de Atmel ya que no sólo la mayoría de las librerías de muchos sensores se encuentran diseñadas para las tarjetas de desarrollo de Arduino, sino que este tipo de tarjetas, siendo de la familia de microcontroladores AVR, ofrecen la capacidad de programar los circuitos integrados mediante la comunicación SPI entre una tarjeta de desarrollo y un circuito integrado una vez integrados en el circuito final (*i.e.* una vez realizado el prototipo es posible realizar configuraciones del programa mismo mediante el uso del recurso de programación *Arduino ISP* del ambiente de desarrollo).
- A pesar de que los protocolos de comunicación Bluetooth y WI-FI ofrecen una amplia gama de ventajas para transmisión de datos, debido a sus estándares de comunicación representa un reto la modificación de dicho tipo de comunicación. Dichos protocolos operan en la banda reservada internacionalmente ISM (Industrial, Científico y Médica, por sus siglas en inglés) definida por la ITU (Unión Internacional de Telecomunicaciones),

las cuales son utilizadas para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. No obstante, si se reconoce el rango del espectro electromagnético en el cual se presenta un menor número de transmisión de datos, es posible elegir el canal que sea menos susceptible al ruido que pueda ocasionar otras redes inalámbricas.

En conclusión de las 12 configuraciones posibles y tomando en cuenta las consideraciones previas, la opción a desarrollar para el transmisor será el número 4.

Receptor

Una vez seleccionada la configuración del transmisor, siendo la solución para la transmisión de datos un *Transceptor* de radiofrecuencia, para el receptor, obligatoriamente debe ser el mismo para el Sistema de Recepción de datos. La primer propuesta se muestra en la Tabla 4.3

Recepción de datos	Procesamiento de Datos	Interfaz Gráfica	Sistema de Conversión de Datos
Receptor de Radiofrecuencia	Microcontrolador Atmel	Aplicación en Microsoft Visual Studio (C#)	Aplicación en Microsoft Visual Studio (C#)
	Microcontrolador Texas	Aplicación en MATLAB	Aplicación en MATLAB
	Microcontrolador Microchip	Aplicación en LabView	Aplicación en LabView

Tabla 4.3: Carta morfológica para el receptor

Una vez identificadas las posibles soluciones a cada sistema funcional, se analizan las siguientes consideraciones:

- Al igual que para el transmisor, conviene para el Sistema de procesamiento de datos (dadas las ventajas anteriormente presentadas) el microcontrolador Atmel (en una configuración adecuada para la recepción de datos).
- La opción de desarrollar una aplicación en LabView resulta muy atractiva al permitirle al desarrollador diseñar un ambiente gráfico agradable al usuario. Sin embargo la programación se considera que una aplicación de consola resulta suficiente para cumplir con los objetivos específicos del proyecto. Por tal motivo una aplicación de consola desarrollada en Visual Studio para la Interfaz gráfica resulta una opción más sencilla.
- Como se puede observar en la carta morfológica, tanto en Sistema de conversión de datos como la interfaz de usuario pueden ser desarrolladas en las mismas opciones descritas, convenientemente en el mismo programa dedicado a la interfaz de usuario, se puede implantar la conversión de datos cuya salida proporcione los cuaterniones que describen la orientación en el espacio correspondiente a cada transmisor. Por lo cual se puede asumir que también se implementa la conversión de datos en la interfaz de usuario (programa en C# desarrollado en Visual Studio).
- Se considera que la opción de Matlab como interfaz de usuario o conversión de datos resulta de mayor conveniencia para el análisis de las pruebas realizadas.

Analizando las consideraciones anteriores las combinaciones posibles se reducen a una sola aplicable: Recepción de datos mediante un Receptor de radiofrecuencia. Procesamiento de datos recibidos mediante microcontrolador Atmel (*Arduino*). Interfaz gráfica para el usuario y sistema de conversión de datos mediante un programa de consola desarrollado en Visual Studio (C#).

4.4. Diseño de detalle

4.4.1. Selección de componentes

Sensor inercial tipo MARG

En esta sección se describen brevemente los sensores inerciales a utilizar en el presente proyecto, así como una explicación del funcionamiento de cada una de sus partes (Acelerómetro, Giroscopio, Magnetómetro).

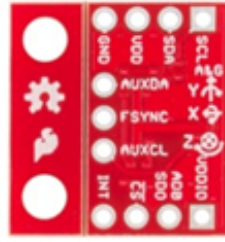
El sensor en cuestión consiste en una tarjeta de desarrollo diseñada por Sparkfun Electronics en la cual se aprovecha el módulo *multi-chip* (MCM) **MPU-9250 de *InvenSense***, para facilitar a los programadores la comunicación con un microcontrolador. La placa mide 18.4x11.6x2.7mm, además de una pestaña removible de 6mm en la cara más larga de la placa, diseñada para sujetar la tarjeta con tornillos M3. La MPU-9250 combina un **giroscopio de tres ejes, un acelerómetro de tres ejes y un magnetómetro de tres ejes (ortogonales)** con un Procesador Digital de Movimiento o DMP (para *Digital Motion Processor*) en un encapsulado de 3x3x1mm; es una actualización de la MPU-6515 trabajando en conjunto con un magnetómetro AK8963 de *Asahi Kasei Microdevices Corporation*. Cabe mencionar que incluye un sensor de temperatura embebido. El rango apropiado para el voltaje de alimentación del módulo es de **2.4 a 3.6 [V]**. [27] [33]

El sensor cuenta con un total de nueve **convertidores analógico-digital (ADCs)**, utilizados en tercias para la digitalización de las salidas en tres ejes tanto para el giroscopio, como para el acelerómetro y el magnetómetro. La MPU-9250 ofrece al programador mediante el bus de I²C, una salida de 9 lecturas *MotionFusion* así como un firmware para calibración en tiempo de ejecución, ahorrando tiempo de procesamiento para el acondicionamiento de señales discretas. En adición, contiene filtros digitales programables para las salidas, un reloj de precisión con un desvío de un 1% a una temperatura de 40°C a 85°C, así como interrupciones programables. [33]

Para la comunicación con todos los registros del dispositivo, es posible utilizar un protocolo ya sea **I²C a 400kHz, o SPI a 1MHz**. Además cuenta con pines específicos para la comunicación por I²C a dispositivos esclavos.



(a) Vista superior de la placa



(b) Vista inferior de la placa



(c) Comparación de tamaño de la placa con 25¢ USD

Figura 4.5: Placa de desarrollo MPU-9250 de SparkFun

Las características de cada sensor incorporado en la placa de desarrollo se enlistan a continuación:

■ Acelerómetro

- Salida digital en tres ejes por Convertidor Analógico Digital de 16 bits
- Rango configurable de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$
- Corriente nominal de $450\mu A$
- Interrupciones programables
- Auto-evaluación

■ Giroscopio

- Salida digital en tres ejes por Convertidor Analógico Digital de 16 bits
- Rango configurable de ± 250 , ± 500 , ± 1000 y $\pm 2000^\circ/\text{seg}$
- Filtro pasa-bajas configurable
- Corriente nominal de 3.2mA
- Auto-evaluación

■ Magnetómetro

- Salida digital en tres ejes por Convertidor Analógico Digital configurable
- Resolución de 14 bits ($15\mu\text{T}/\text{LSB}$) o 16 bits ($0.6\mu\text{T}/\text{LSB}$ ¹)
- Rango total de $\pm 4800\mu\text{T}$.
- Consumo de corriente promedio a mediciones de 8 Hz: $280\mu A$

Una ventaja considerable de este componente en particular es el acceso que ofrece el fabricante a las librerías desarrolladas por Kris Winer en la plataforma GitHub, para el uso del sensor en la tarjeta de desarrollo *Arduino*. En dicho código no sólo le ofrece al desarrollador el manejo de los registros ni la recolección de datos de la tarjeta, sino que incorpora un archivo en el cual permiten implementación de dos algoritmos de orientación espacial: El algoritmo de **Madwick** y el algoritmo de **Mahony**. Sin mencionar que también ofrece diferentes salidas de datos:

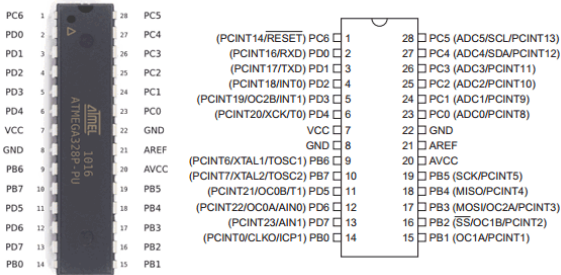
¹LSB, del inglés *Least Significant Bit* (bit menos significativo) indica el menor dígito de un número binario. Al mencionar cantidades como $15\mu\text{T}/\text{LSB}$ se indica que el ADC cambia la salida digital con un mínimo de $15\mu\text{T}$ detectados.

- Datos sin procesar de cada sensor
- Cuaterniones resultantes del procesamiento de las señales de los sensores incorporados
- Ángulos de Euler (resultado de la conversión de cuaterniones)

Microcontrolador

Al concluir que los microcontroladores Atmel son los más adecuados para la construcción de un prototipo funcional, se considera que las pruebas de adquisición y transmisión de datos es posible realizarlas mediante el uso de la tarjeta de desarrollo Arduino UNO (o Arduino MEGA). Por tal motivo, el microcontrolador seleccionado es el **microcontrolador ATmega328P** (Fig. 4.6a), el cual es el mismo microcontrolador que se encuentra en la tarjeta de desarrollo Arduino (Fig. 4.6b) (Lo cual también representa una ventaja, debido a que si las pruebas son exitosas en la tarjeta de desarrollo, también lo serán en la tarjeta con el componente soldado adecuadamente sobre una placa).

ATmega328P pin mapping



(a) Microcontrolador ATmega328P con mapeo de pines



(b) Placa de desarrollo Arduino UNO con el microcontrolador ATmega328P incorporado

Figura 4.6: Microcontrolador ATmega328P

El ATmega328P es un microcontrolador de arquitectura AVR de 8-bits, cuyas principales características se enlistan a continuación:

- Tipo de memoria: Flash
- Tamaño de memoria (propósito general): 32 KB
- Tamaño de SRAM: 2,048 bytes
- Tamaño de EEPROM: 1,024 bytes
- Periféricos de comunicación digital
 - 1-UART
 - 2-SPI
 - 1-I²C
- Periféricos de captura, comparación y PWM: 1 captura de entrada, 1 CCP y 6 PWM
- Timers
 - 2 x 8-bits

- 1 x 16-bits
- 1 Comparador
- Rango de temperatura: De -40 a 85 °C
- Rango de voltaje de operación: 1.8 a 5.5 V
- 32 Pines
- 6 Canales de CAD a 10-bits
- 32 Pines de I/O (Entrada/Salida)
- Oscilador interno RC a 8 MHz
- Corriente máxima absoluta en cualquier pin: 40 mA
- Corriente máxima absoluta total del dispositivo: 200 mA

Transceptor por Radiofrecuencia

En el mercado existe un módulo que incorpora tanto un transmisor como un receptor (De ahí el nombre *Transceptor*) de datos por radiofrecuencia. **Dicho módulo incorpora el circuito integrado nRF24L01 desarrollado por Nordic Semiconductor** (Mostrado en la Fig. 4.7). El fabricante lo describe como un circuito altamente integrado, de muy bajo consumo energético que opera a 2.4 GHz en la banda ISM. Las principales especificaciones del circuito se enlistan a continuación: [34]

- Bajo consumo energético
 - 900 nA en *Deep sleep mode*
 - 11.3 mA en modo transmisor a 0 dBm
 - 12.3 mA en modo receptor a 2 Mbps de velocidad de datos de transmisión
- Radio de 2.4 GHz
 - Banda de operación ISM de 2.4 Hz
 - 6 canales de comunicación (de 2400 a 2525 Hz) con un espaciamiento de 1 MHz de uno a otro.
 - Modulación GFSK, 1 a 2 MHz de ancho de banda
 - 0, -6, -12 y -18 dBm programables para el transmisor
 - Velocidad de datos programables de 1 a 2 Mbps
 - Sensibilidad de receptor de -82 dBm a 2 Mbps
 - Sensibilidad de receptor de -85 dBm a 1 Mbps
 - Compatibilidad con un cristal a 16 MHz ± 60 ppm
- Protocolo ShockBurst
 - Ensamble automático de paquete de datos
 - Detección y validación de paquete de datos automáticas
 - Longitud del paquete de datos dinámica (de 1 a 32 bytes)

- Retransmisión automática
 - Reconocimiento (*acknowledgement*) automático con paquete de datos como respuesta opcional
 - Conexión con 6 puertos (*pipes*) transmisores (MultiCeiver)
 - 3 conjuntos de 32 bytes FIFO's para transmisión y recepción de datos
- Interfaz de comunicación SPI de hasta 10 Mbps
 - Temperatura de operación de -40 a 80 °C
 - Energía
 - Regulador de voltaje interno
 - Voltaje de operación de 1.9 a 3.6V

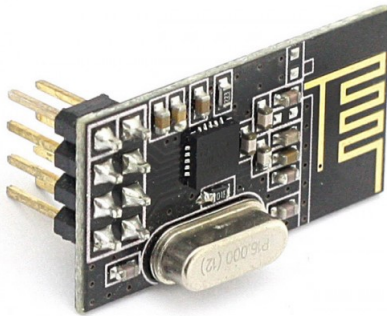


Figura 4.7: Módulo nRF24L01

Interfaz de usuario

Como se menciona anteriormente se tiene en consideración el diseño de dos interfaces mediante las cuales el usuario será capaz de vincular los transmisores con el receptor. Una destinada a los transmisores y otra más dedicada al receptor. Al tratarse de diferentes soluciones (microcontrolador y computadora) las características de cada uno difieren y se definen a continuación.

Transmisor

Se considera suficiente que para el transmisor se manifiesten los estados del dispositivo mediante **LED's** y que el usuario pueda ejercer acciones a éste mediante **interruptores**. Para ello se enlistan los estados posibles en los cuales el dispositivo puede operar:

1. Configuración de inicio
2. Vinculación y transmisión activa de datos
3. Calibración del magnetómetro
4. Error de conexión
5. Nivel de batería bajo

Dependiendo del estado en el que se encuentre el dispositivo le permite al usuario conocer mediante ciertas configuraciones de los LED's el estado en el que se encuentra el programa corriendo en el microcontrolador

Receptor

Como se aprecia en la Fig. 4.3, la función principal del sistema funcional de recepción de datos, es el de recibir los datos de manera aislada provenientes de cada transmisor. Por lo tanto en lo que refiere al microcontrolador y *transceptor* del receptor no se requiere una interfaz que le indique al usuario del estado del programa, así como las configuraciones necesarias. Debido a que la comunicación entre el microcontrolador del receptor y la computadora puede ser mediante comunicación serial, la configuración del microcontrolador puede ser realizada en la computadora y ser enviados los comandos necesarios mediante dicha comunicación. En resumen, la interfaz gráfica de usuario (**aplicación en consola**) le permite al usuario realizar las acciones necesarias para configurar la red de transmisión y recepción de datos, así como la recepción misma y la conversión de los datos recibidos.

Suministro de energía del transmisor

Considerando las corrientes máximas de los principales componentes en el transmisor (microcontrolador, MPU-9250, y nRF24L01), de las especificaciones anteriormente mencionadas se tiene un total de 203.93 mA (considerando un funcionamiento continuo y los componentes trabajando a los límites superiores de consumo). Si se considera que una prueba de marcha con los marcadores ya colocados en su sitio con las consideraciones previas ya establecidas en el estudio, tiene una duración mínima de 10 minutos [30], entonces se puede calcular la capacidad mínima en mAh de una batería. En la ecuación 4.1 se muestra dicho calculo. Es necesario resaltar que no se tienen en consideración las posibles pérdidas de componentes adicionales, que representen un aumento en la corriente de consumo del transmisor en su totalidad (*e.g.* reguladores de voltaje necesarios para una operación óptima).

$$x[mAh] = 203.93[mA] \times 10[min] \times \frac{1[h]}{60[min]} = 33.9883[mAh] \quad (4.1)$$

Por lo que para una prueba de marcha convencional, es necesario energizar el circuito con una batería cuya capacidad sea de mínimo 34mAh. En el mercado existen baterías de tipo LiPo de mucha mayor capacidad y de reducido tamaño que pueden ser utilizadas en el prototipo. En este caso se eligió una **batería 701725 de VVRC** mostrada en la Fig 4.8. Este tipo de baterías son altamente utilizadas en vehículos y juguetes de control remoto y para la alimentación energética de drones pequeños, por lo cual resulta conveniente tanto en capacidad como en tamaño. A continuación se enlistan las características de dicha batería:

- Voltaje nominal: 3.7 V
- Voltaje de carga: 4.25 V
- Corriente máxima de carga: 1c
- Corriente de descarga continua: 15c
- Corriente de descarga sin carga: <3 % por mes
- Capacidad nominal: 150 mAh
- Resistencia interna: 358 mΩ (obtenida mediante medición directa)

- Peso aproximado: 7 g
- Tamaño: 6×16×27 mm
- Incluye circuito de protección para la batería

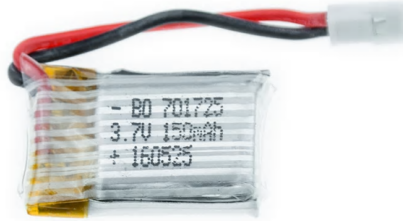


Figura 4.8: Batería 701725

Con la capacidad de la batería seleccionada se estima (aplicando las mismas consideraciones de los cálculos de la ecuación 4.1) como mínimo un tiempo de operación por encima de los 30 min.

4.4.2. Transmisión de Datos

Para el correcto funcionamiento del sistema planteado, el algoritmo a seguir consiste en una serie de pasos que involucran tanto al(los) transmisor(es) y al receptor. Para ello es importante definir que para la transmisión de datos de los transmisores al receptor, son necesarios dos procedimientos:

1. Vinculación
2. Transmisión-Recepción de datos

Sin embargo el algoritmo en su totalidad puede ser representado mediante el diagrama de flujo mostrado en la Fig. 4.9, en donde se simplifican ambos procedimientos

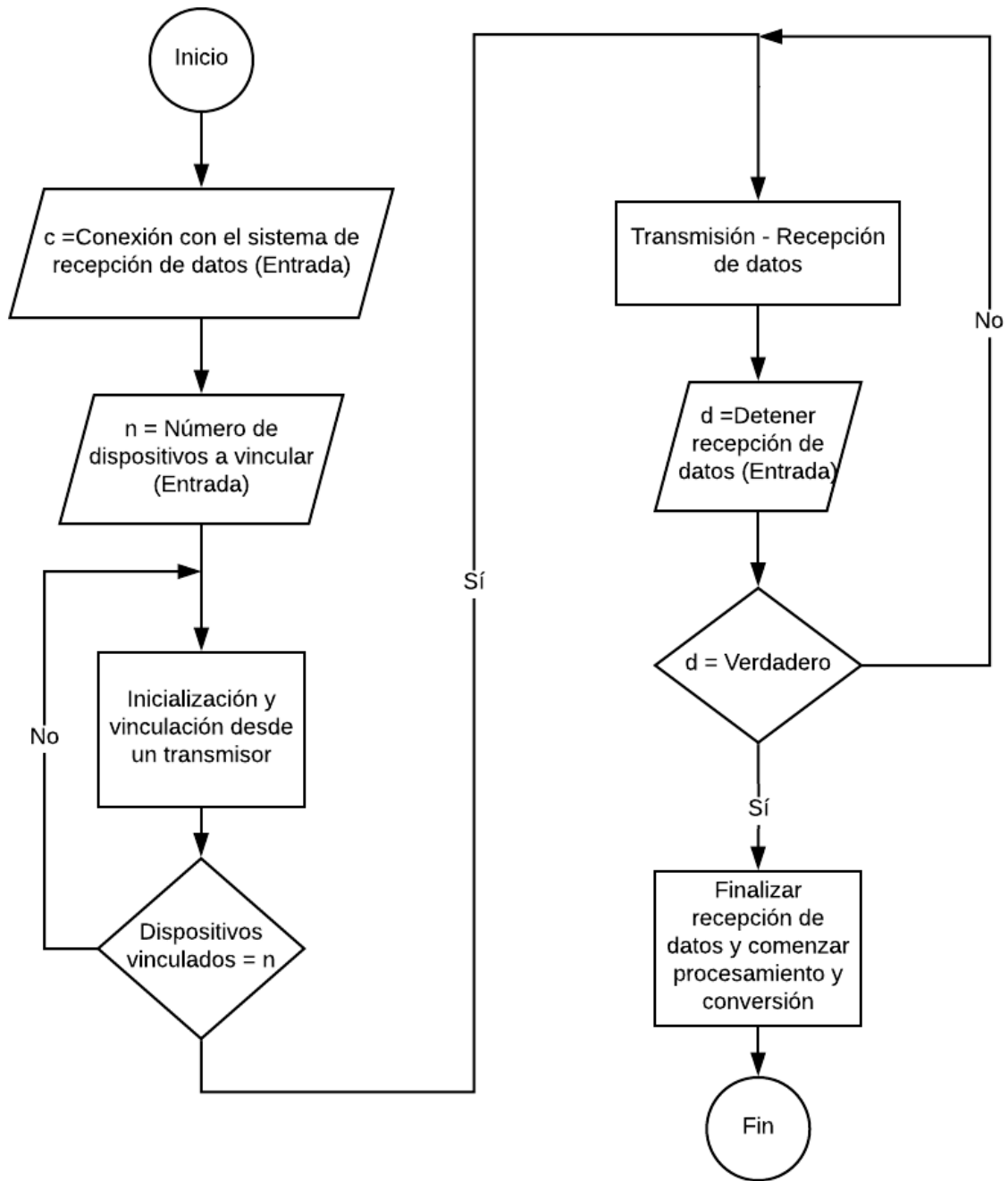


Figura 4.9: Diagrama de flujo simplificado de los procedimientos de vinculación y recepción de datos

Para realizar las pruebas pertinentes a la programación de ambos circuitos, se utilizan las tarjetas de desarrollo de Arduino UNO y MEGA; implementando en los programas las librerías correspondientes a cada prueba. Para su programación se hace uso del ambiente de desarrollo de Arduino en su versión 1.8.6. Para un manejo simplificado del módulo nRF24L01 se agrega a los *sketches*² del transmisor y del receptor la **librería RF24** desarrollada por **MRh20** en su versión 1.3.1. [35]

²Se le llama comúnmente *sketch* al programa que se quema en la memoria del microcontrolador de las tarjetas de desarrollo Arduino.

Vinculación

Como especifica el fabricante, el transceptor nRF24L01 puede operar en un rango de frecuencia entre 2.4 a 2.525 MHz, en la banda de operación ISM. Al operar en el mismo rango que múltiples dispositivos de comunicación inalámbrica, resulta conveniente una herramienta que ofrece la librería utilizada (en la sección de ejemplos) que le permite al desarrollador «escanear» los 125 canales de comunicación disponibles y así detectar aquellas frecuencias en los que se detectan señales así como su intensidad (representado en una escala de 0 a F en base hexadecimal). En este sentido uno puede ser capaz de seleccionar el canal que sea menor susceptible al ruido.

Para realizar la prueba, se conectan los pines del módulo nRF24L01 a la tarjeta Arduino MEGA con el programa ya cargado en memoria. Tras 20 escaneos se obtienen los resultados mostrados en las Fig. 4.10. Los datos se obtienen como texto mediante el monitor serial del ambiente de desarrollo de Arduino (Fig. 4.10a), estos datos son posteriormente graficados para un análisis más sencillo (Fig. 4.10b). De dicho escaneo, se puede observar la actividad inalámbrica en la zona de pruebas³ y se muestra que entre los canales 50 y 85, las señales captadas tienen una alta intensidad. Del lado izquierdo de este rango (canales 1 al 49) existen detecciones de menor intensidad, sin embargo la actividad aún sigue presente y puede causar interferencias en la comunicación del SGP. Por el otro lado, para los canales del 86 al 125 se nota una actividad inalámbrica prácticamente nula. Con la finalidad de que la comunicación no se vea afectada por otro tipo de transmisión inalámbrica (inclusive fuera de la banda ISM) se considera adecuado ocupar el canal 110.

es necesario que los programas en los microcontroladores de ambos dispositivos trabajen en conjunto para una correcta vinculación. Para que dos dispositivos nRF24L01 puedan transmitir y recibir datos de forma exitosa, es necesario asignarles una única dirección a ambos. Esta dirección consiste en un número de 64 bits sin signo y es comúnmente asociado a la palabra en inglés «*pipe*» o «conducto» por el cual dos dispositivos se comunican. Por lo tanto para el proyecto se propone elegir la dirección $0F0F0F0F00_{16}$ como base de las direcciones disponibles para la vinculación de los transmisores, admitiendo la variación incremental de los 2 bytes menos significativos. Con esta condición se propone un protocolo de vinculación en el cual las direcciones lleven el conteo mismo de los transmisores a emparejar únicamente al incrementar en uno la dirección base. Es decir, el primer transmisor tendría como dirección la base ($0F0F0F0F00_{16}$), el segundo tendría como dirección la base incrementada en uno ($0F0F0F0F01_{16}$), el tercero tendría como dirección la base incrementada en dos ($0F0F0F0F02_{16}$), y de la misma manera de forma sucesiva.

De esta manera, el receptor solamente necesitaría guardar en memoria el número de dispositivos a vincular y esperar a que el primer transmisor se vincule con la dirección base a través de un envío de paquete de datos arbitrario (*ping*); para vincular un segundo transmisor se espera la conexión con la dirección base incrementada en uno, y así sucesivamente. Por otro lado los transmisores tendrían como dirección la base, posteriormente enviarían el paquete de datos (*ping*) esperando respuesta del receptor, de no ser así se incrementa en uno el número de dirección y se envía nuevamente el paquete de datos. Para evitar una secuencia de incremento interminable, se limita el número de dispositivos a vincular y por ende el número de dirección máxima. Por consiguiente, se elabora un diagrama de flujo correspondiente a los algoritmos de vinculación tanto del transmisor (Fig. 4.12) como del receptor (Fig. 4.12).

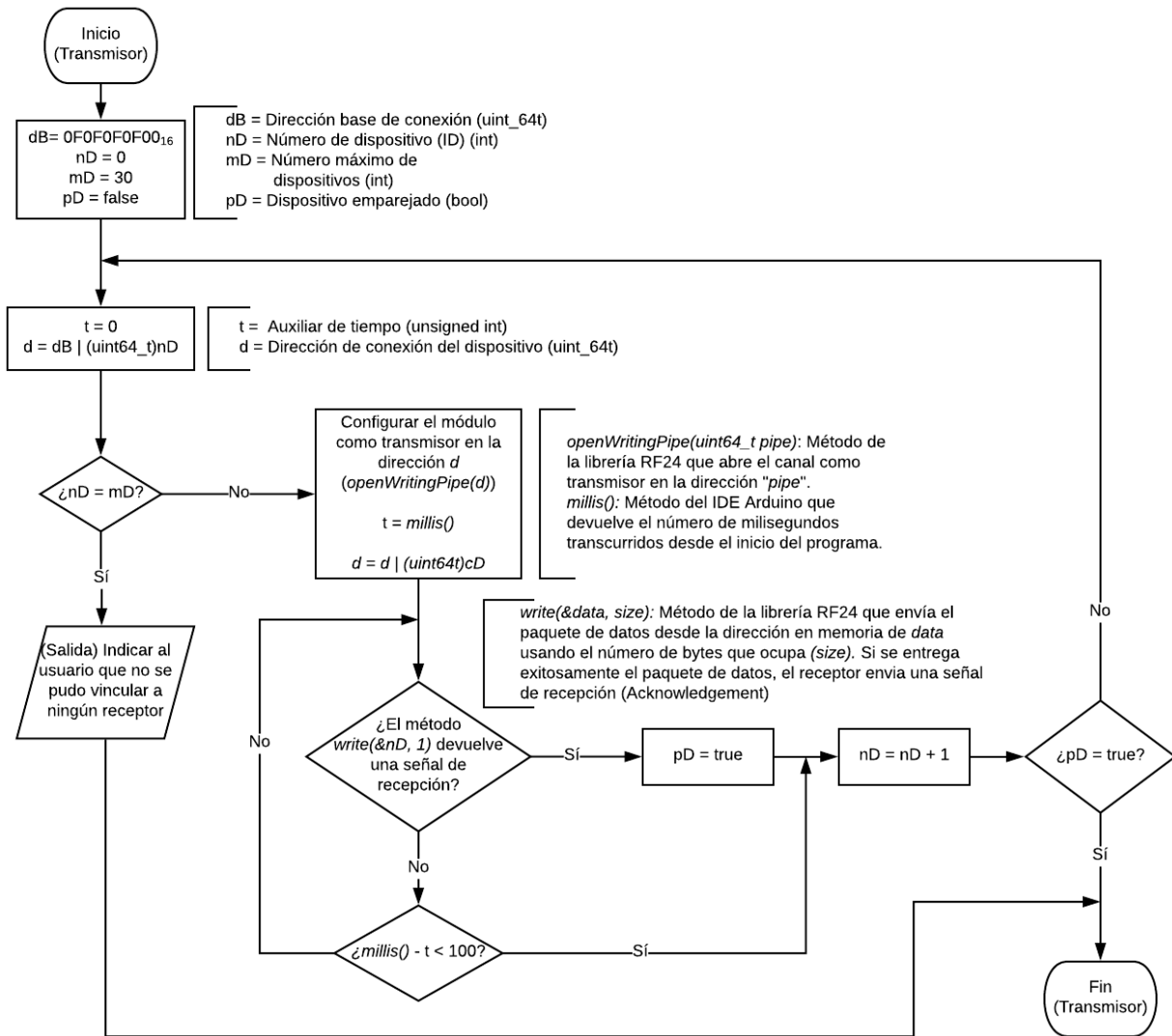


Figura 4.11: Diagrama de flujo del proceso de vinculación para el transmisor

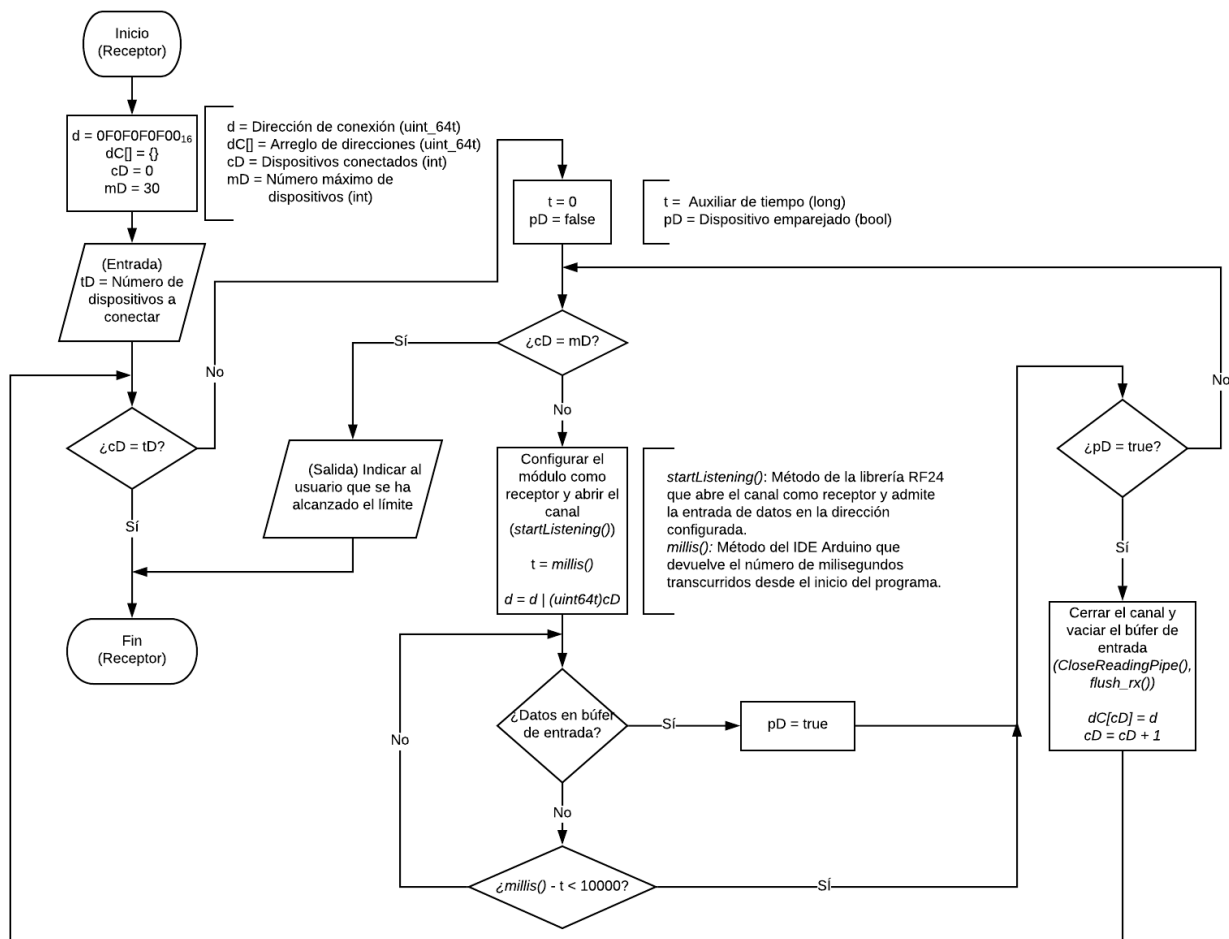


Figura 4.12: Diagrama de flujo del proceso de vinculación para el receptor

Transmisión - Recepción de datos

Una vez identificado el proceso de vinculación tanto para el receptor como para el transmisor se da por hecho que la vinculación ha sido exitosa para proceder al diseño de la transmisión y recepción de datos. Como se puede apreciar en las especificaciones del módulo nRFL01 un sólo receptor puede recibir señales de 6 transmisores diferentes en la misma dirección, sin embargo esto limita la posibilidad de recibir datos de más de 6 sensores utilizando un sólo protocolo de envío y recepción de datos. Si se desea analizar las señales de más de 6 transmisores a la vez es necesario diseñar un algoritmo que reciba los datos en "paquetes" de 6 intercambiando la dirección del receptor; lo cual es perfectamente viable, aunque sería necesario controlar cual es el sensor del cual se desean recibir los datos. Para lograr un muestreo de la orientación de todos los sensores distribuida de la mejor manera se tendrían que recibir los datos de manera secuencial, tal y como se recibirían los datos si se implementara un multiplexor: atendiendo las señales una a una.

En este sentido resulta más conveniente que los transmisores envíen los datos de manera continua sin importar que el receptor haya recibido los datos de manera satisfactoria (ignorando la señal de *acknowledgement*), mientras que el receptor cambia de manera secuencial la dirección de conexión hasta que que reciba un paquete de datos del transmisor con la misma dirección (*i.e.* El receptor irá cambiando la dirección de conexión uno a uno escuchando.^a cada transmisor uno a uno de manera aislada hasta que se alcance el límite de los dispositivos conectados volviendo empezar en la dirección base y por ende al primer transmisor). En la Fig. 4.13 se

muestra de manera gráfica el algoritmo antes explicado, con un arreglo de 4 transmisores enviando los datos a un solo receptor. Se observa que todos los transmisores envían los datos de manera arbitraria, manteniendo su dirección inicialmente configurada. En la Fig. 4.13a se aprecia que el receptor configura la dirección de conexión (acorde a la explicada en la sección de vinculación) a la dirección base $0F0F0F0F00_{16}$, y por consiguiente al Transmisor 1. Una vez que se reciba el paquete de datos el receptor cambia su dirección de conexión de manera incremental a $0F0F0F0F01_{16}$ como muestra la Fig. 4.13b, que coincide con la dirección del Transmisor 2 recibiendo así un paquete de datos correspondiente a dicho dispositivo. La secuencia de incremento a la dirección de conexión en el receptor se repite para recabar los datos de los Transmisores 3 y 4 (Fig. 4.13c y Fig. 4.13d respectivamente). En este punto, al tener en memoria la cantidad de dispositivos que se vincularon inicialmente, vuelve a la dirección base e inicia nuevamente la secuencia de transmisión y recepción de datos.

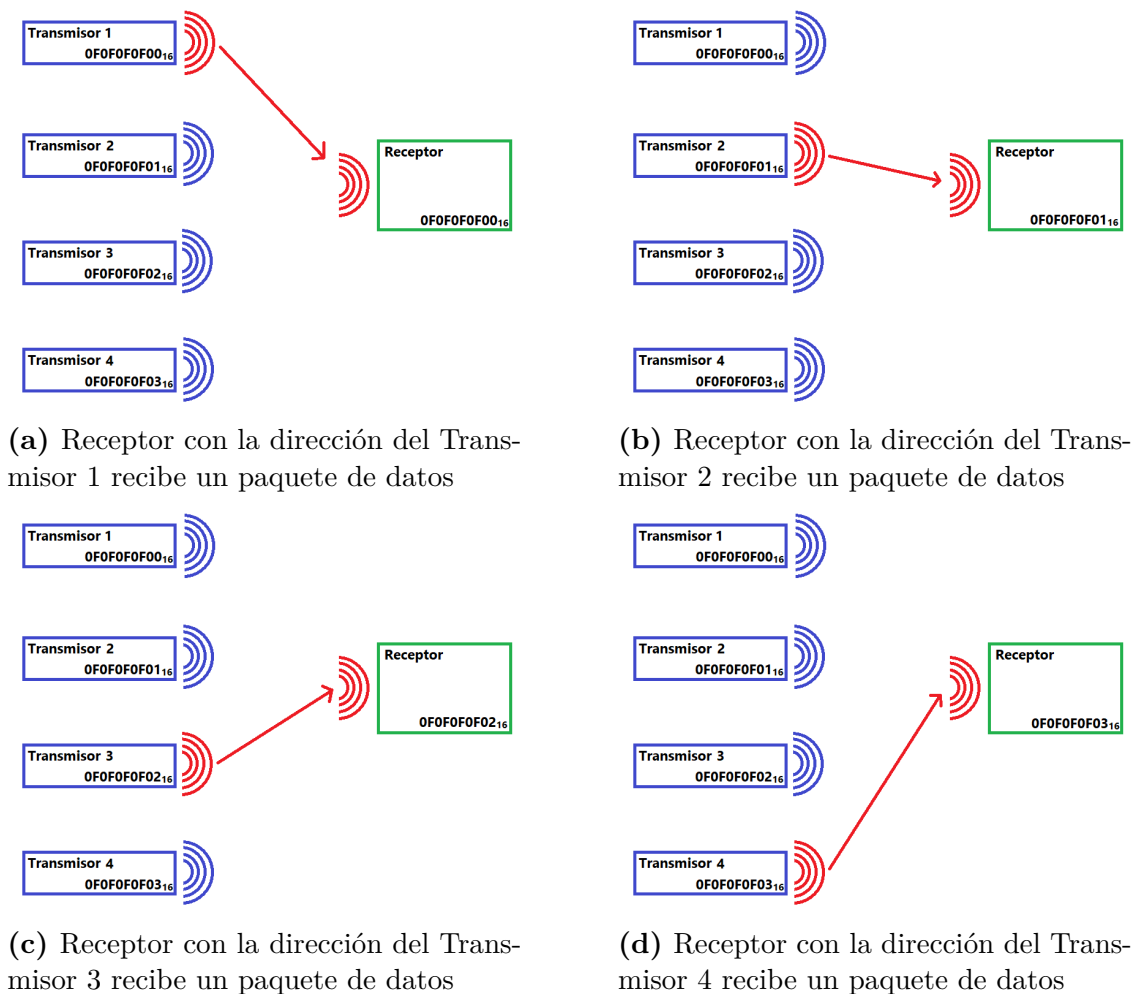


Figura 4.13: Diagrama secuencia de transmisión - recepción de datos de un arreglo de 4 transmisores y un receptor

Datos a transmitir

Acorde a la propuesta de solución, los datos a enviar contendrán la información necesaria para obtener la orientación en el espacio de los sensores MPU-9250 tras la aplicación del algoritmo de Madgwick.

Para la comunicación entre el microcontrolador y el sensor MPU-9250, se realiza una **configuración inicial** para el correcto funcionamiento del algoritmo en el programa desarrollado por Kris Winer. En dicha configuración se asignan valores como las velocidades de muestreo, resoluciones de los sensores, tipo de comunicación, etc. a través de la lectura y escritura de registros; además de la asignación de valores a las variables propias de la librería implementada como valores de resolución, velocidades de muestreo, factores de conversión de datos, etc además de valores de escala y factores de corrección para los sensores mediante algoritmos de medición y procesamiento de datos. Mientras que para el acelerómetro y el giroscopio basta que la unidad permanezca en reposo, para el magnetómetro se lleva a cabo una serie de pasos para obtener los datos antes mencionados. Para la obtención de los datos de calibración del magnetómetro⁴ Kris Winer desarrolló un método en el cual el usuario realiza movimientos en «8» por 15 segundos de la misma forma en la que se calibran actualmente los magnetómetros en los teléfonos celulares como se aprecia en la Fig. 4.14. Durante este proceso, se recaban datos que posteriormente son procesados y resultan en un total de 6 factores de corrección en las mediciones del magnetómetro: 3 para la escala de la medición y 3 para el error de la medición. En cuestión al programa del microcontrolador del transmisor, en este sentido deben reservarse en una estructura de datos un total de 24 bytes (Al ser datos de tipo *float*, los cuales utilizan 4 bytes en memoria y son 6 variables en total).

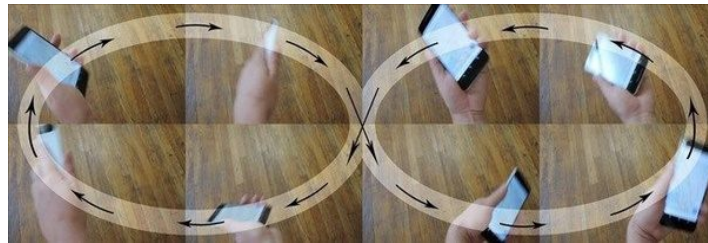


Figura 4.14: Calibración del magnetómetro en un teléfono celular

El inconveniente con el programa, es que cada vez que inicie los datos necesitan ser obtenidos mediante el protocolo de calibración de 15 segundos. Para dejar al usuario la posibilidad de elegir en qué momento se desea recalibrar el magnetómetro, se aprovecha la memoria EEPROM del microcontrolador ATmega328P para guardar el valor de los 6 valores antes mencionados y utilizarlos al iniciar el programa; en caso de que se requiera una nueva calibración, el microcontrolador entra en el estado de *Calibración del Magnetómetro* donde se efectúan las mediciones pertinentes y se reemplazan los nuevos valores obtenidos en la EEPROM.

Una vez finalizada la configuración inicial se leen de forma continua los valores de cada sensor integrado a la MPU-9250 y dependiendo de los datos que se deseen enviar, se transmiten haciendo uso del módulo nRF24L01.

Se pueden considerar dos opciones para la transmisión de datos:

1. Que los datos a enviar contenga la información sin procesar (*raw*) de los sensores que integran a la MPU-9250 (tres ejes para acelerómetro, giroscopio y magnetómetro) en cada iteración y que el procesamiento se realice tras recibir los datos en la computadora, aplicando el algoritmo de estimación de orientación espacial de Madgwick
2. Que el procesamiento de los datos de cada sensor en la MPU-9250 sean procesados por el mismo microcontrolador del transmisor y envíe al receptor el cuaternión resultante de la aplicación del algoritmo de Madgwick en cada iteración

⁴Los datos de calibración corresponden a la corrección de los errores resultado de los efectos de Acero-duro y Acero-blando del ambiente en donde se realicen las pruebas.

Los métodos que ofrece la librería de Kris Winer se pueden obtener tanto los datos sin procesar de los sensores, como los cuaterniones resultantes de la aplicación del algoritmo de Madwgick, por lo cual se realizan pruebas que determinen la cantidad de datos que pueden ser recibidos en 1 segundo para cada alternativa de solución. Para la prueba se utiliza una tarjeta de desarrollo Arduino como transmisor y un Arduino MEGA como receptor. Ver códigos en Anexo 10.4, en donde se muestra el programa a utilizar y se aprecia que mediante una variable de tipo booleana se define si se desea realizar una prueba o la recepción continua de datos; para éste caso se configura para la recepción de datos como prueba.

Se realizaron 3 pruebas para cada configuración y se realizó el conteo de paquetes de datos en cada caso. Se obtuvieron los resultados mostrados en la Tabla 4.4.

Datos sin procesar [Núm. de paquetes de datos]	Cuaterniones [Núm. de paquetes de datos]
130	164
132	164
131	164

Tabla 4.4: Resultados de la prueba de velocidad de transmisión en las configuraciones de envío de datos sin procesar y cuaterniones resultantes del algoritmo de Madgwick

Por lo cual, se concluye que los paquetes de datos a enviar deben incluir los valores de cada componente del cuaternión resultante, al ser más rápida la transmisión. Por lo tanto, al ser 4 componentes de tipo *float*, se reserva para la estructura de datos el tamaño correspondiente a la concatenación de los 4 valores, es decir 16 bytes (4 bytes para cada componente del cuaternión) y 1 byte más del número de dispositivo transmisor. **Resultando en 17 bytes como paquetes de información a enviar.**

En la Fig. 4.15 se aprecia el diagrama de flujo que el programa del transmisor para inicializar la comunicación con la MPU-9250, obtener los datos, procesarlos y posteriormente enviarlos de manera continua en paquetes de datos. A su vez se puede apreciar el ingreso al estado de Calibración del Magnetómetro a consideración del usuario.

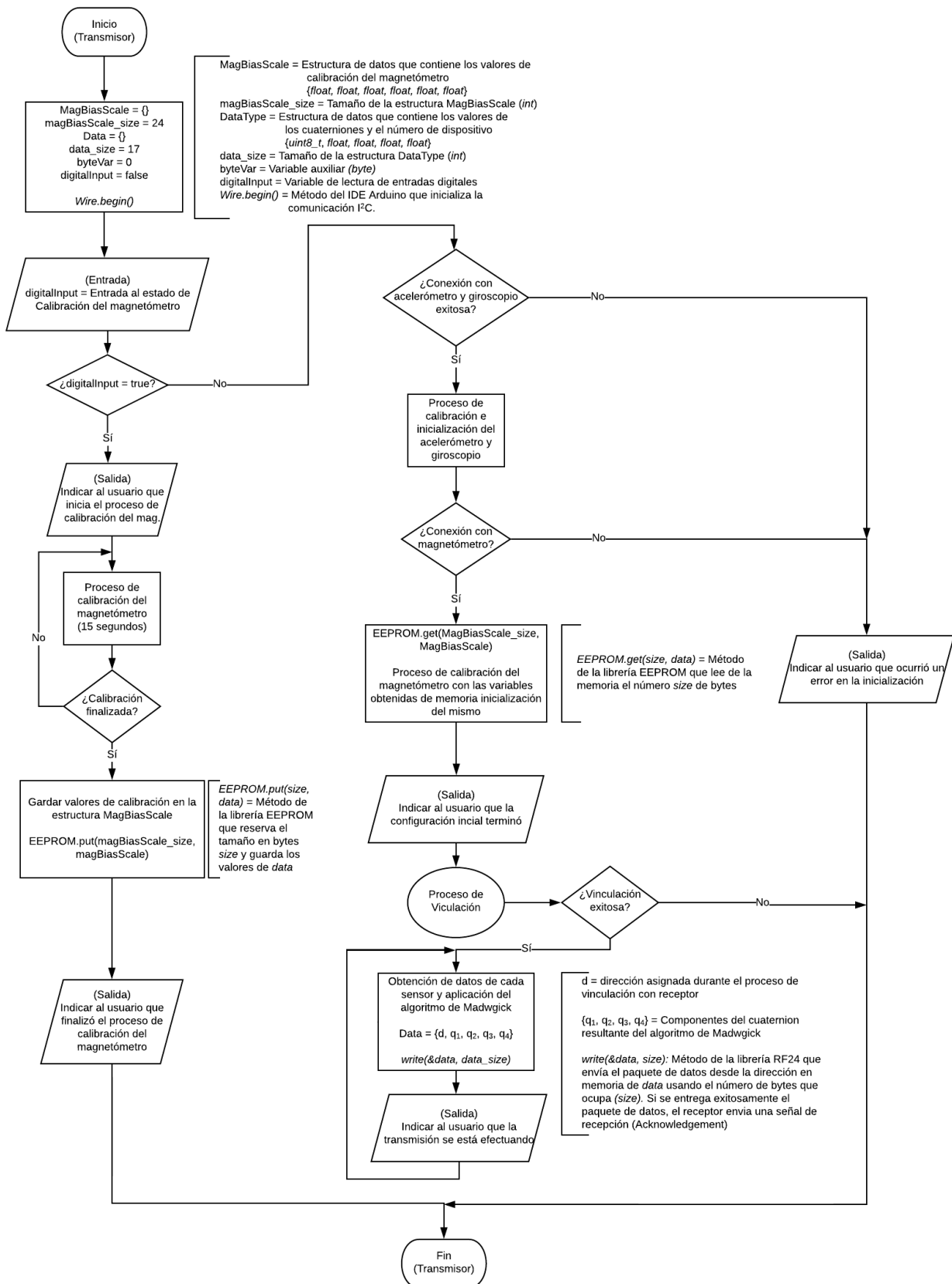


Figura 4.15: Diagrama de flujo de la obtención y procesamiento de datos en el transmisor

Por otro lado, el receptor necesita conmutar las direcciones a modo de multiplexor, como se muestra en la Fig. 4.13. Además de reenviar los datos al sistema de procesamiento de datos; que para este caso a la computadora vía comunicación serial (Puerto USB). En la Fig. 4.16 se

puede apreciar el diagrama de flujo del programa del sistema de recepción de datos.

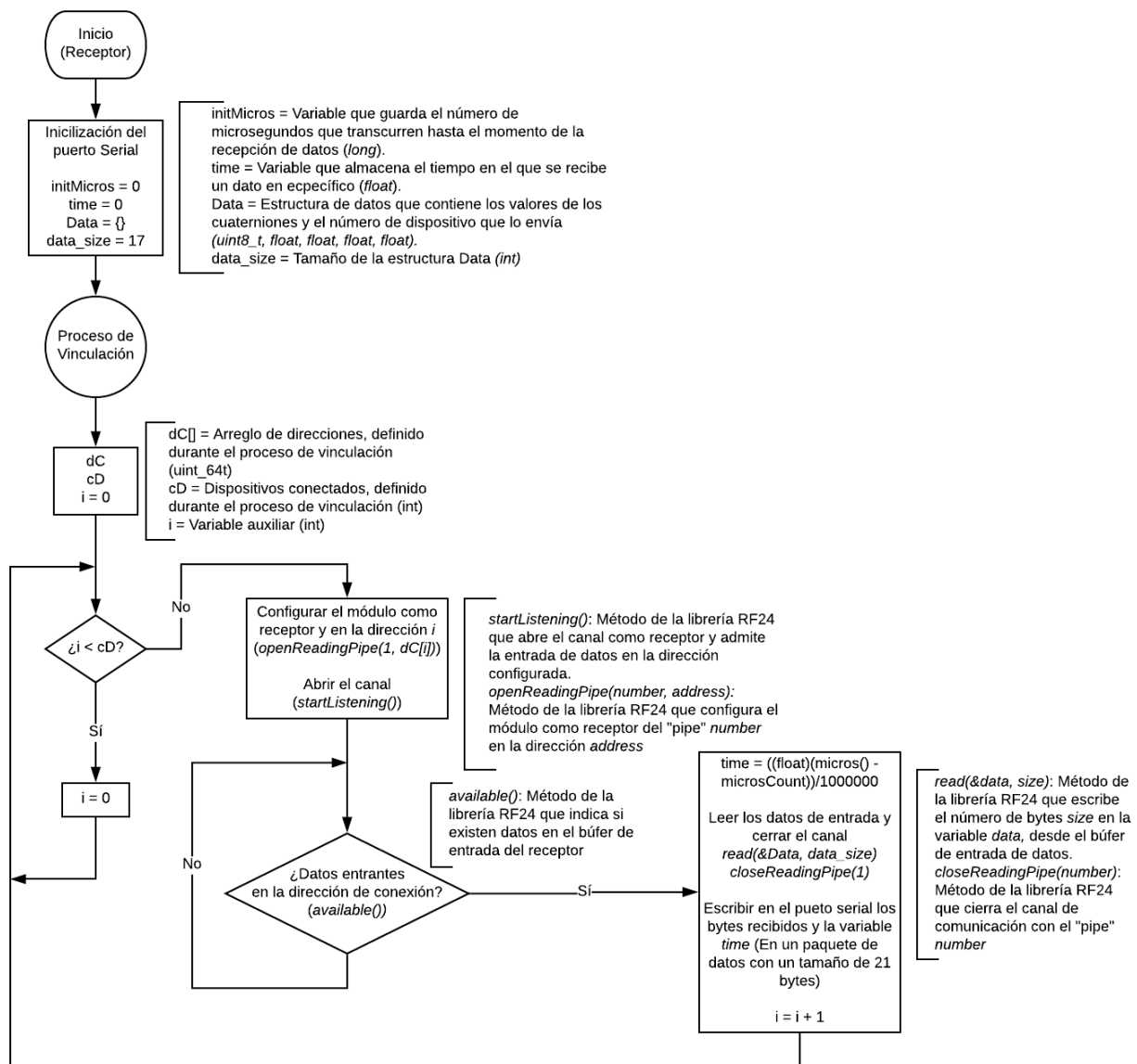


Figura 4.16: Diagrama de flujo de la obtención y reenvío de datos al puerto serial en el receptor

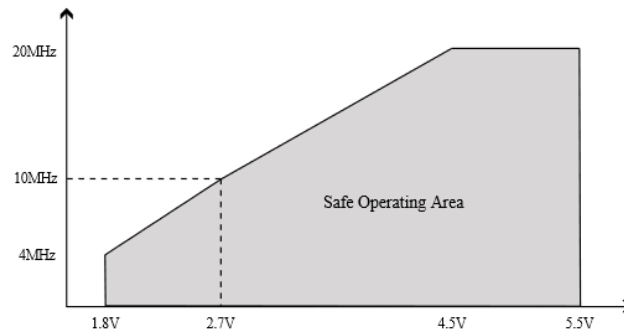
4.4.3. Administración de Energía

Al seleccionar una batería LiPo como solución al suministro de energía de los transmisores, es de suma importancia reconocer los voltajes de operación tanto de la batería misma como de los componentes. De esta manera, se procede a seleccionar los componentes necesarios para la administración de la energía a los diferentes sistemas del circuito; partiendo del hecho de que la batería tiene un voltaje nominal de 3.7 [V].

Microcontrolador

La documentación indica que el ATmega328P tiene un rango de voltaje de operación que le permite en teoría funcionar bajo la alimentación de la batería de forma directa. Sin embargo, al considerar la frecuencia de operación de 16 [MHz], es importante recalcar que la frecuencia máxima de operación depende del voltaje de alimentación. [36] Tanto de la Fig. 4.17a como de

la Tabla 4.17b se puede apreciar que para el funcionamiento del microcontrolador a la frecuencia deseada, es necesario que el voltaje de alimentación sea de mínimo 4.5 [V].



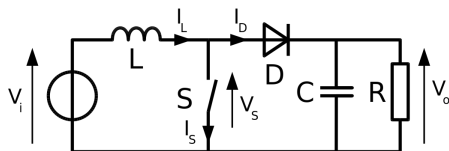
(a) Gráfica de voltaje de alimentación contra la frecuencia máxima de operación del ATmega328P [36]

Symbol	Parameter	$V_{CC}= 1.8 - 5.5V$		$V_{CC}= 2.7 - 5.5V$		$V_{CC}= 4.5 - 5.5V$		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
$1/t_{CLCL}$	Oscillator Frequency	0	4	0	10	0	20	MHz
t_{CLCL}	Clock Period	250	-	100	-	50	-	ns
t_{CHCX}	High Time	100	-	40	-	20	-	ns

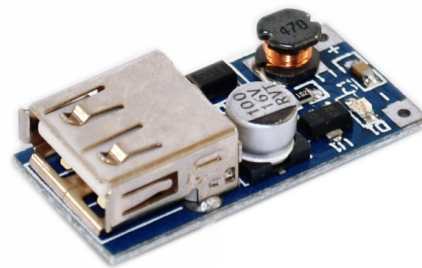
(b) Parámetros de la unidad de reloj externo del ATmega328P [36]

Figura 4.17: Especificaciones de frecuencia y voltaje de alimentación del microcontrolador ATmega328P

En este sentido, alimentar al microcontrolador con la batería de forma directa no es viable. Para ello es necesario añadir un circuito capaz de aumentar el voltaje de salida de la batería a 5 [V]; un **circuito convertidor elevador de voltaje (*Booster*)**, cuyo esquema simplificado se puede apreciar en la Fig. 4.18a. En el mercado existen actualmente una gran variedad de circuitos de este tipo ya incorporados en una placa de desarrollo. Uno de estos circuitos es el mostrado en la Fig. 4.18b, una placa de desarrollo genérica (EAN: 0738283293320) diseñada para obtener desde una batería LiPo (de una celda) un voltaje de 5 [V] directo a una terminal USB.



(a) Diagrama simplificado de un circuito elevador de voltaje



(b) Modulo convertidor elevador de 0.9 [V] a 5 [V] genérico con salida USB

Figura 4.18: Circuito convertidor elevador de voltaje

Por consiguiente, un circuito de estas características se considera adecuado para polarizar

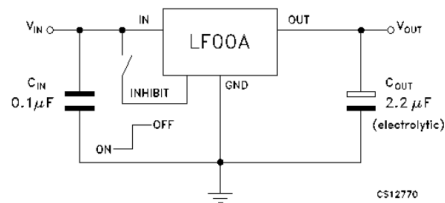
al microcontrolador con 5 [V].

MPU-9250 (Sensor MARG) y nRF42L01 (*Transceptor*)

Para ambos circuitos el suministro de voltaje máximo es de 3.6 [V], debido a que el voltaje nominal de la batería es de 3.7 [V] sería en teoría inviable el suministro directo de energía debido a que en ocasiones se excede el voltaje disponible de la batería. A diferencia del microcontrolador, tanto para el sensor inercial como el *transceptor* de radiofrecuencia seleccionados es necesario que se añada un circuito regulador de voltaje, preferentemente a 3.3 [V]. Circuitos en el mercado como el **LF33CV** (Fig. 4.19a) en conjunto con un arreglo de capacitores, pueden proporcionar y regular el voltaje anteriormente mencionado (Fig. 4.19b).



(a) Regulador de voltaje a 3.3 [V] LF33CV



(b) Circuito regulador de voltaje a 3.3 [V] con arreglo de capacitores

Figura 4.19: Circuito regulador de voltaje a 3.3 [V]

Carga de la batería

Cada transmisor funciona como un módulo del SGP en su totalidad. Por tal motivo, al funcionar de manera independiente uno de otros transmisores y del receptor mismo, es importante hacer saber al usuario que la batería está por agotarse y proporcionar la solución de manera sencilla para la recarga de la misma. En este sentido, es importante conocer que en el mercado actualmente existen módulos que ofrecen la posibilidad de cargar baterías, como es el caso del integrado **TP4056** que en tarjetas de desarrollo ya se incorporan los componentes necesarios para lograr la carga mediante la conexión de un cable *mini-USB* o *micro-USB*, como se muestra en la Fig. 4.20.

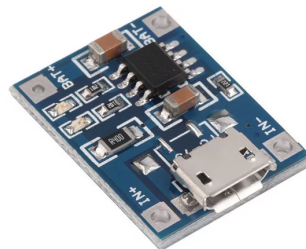
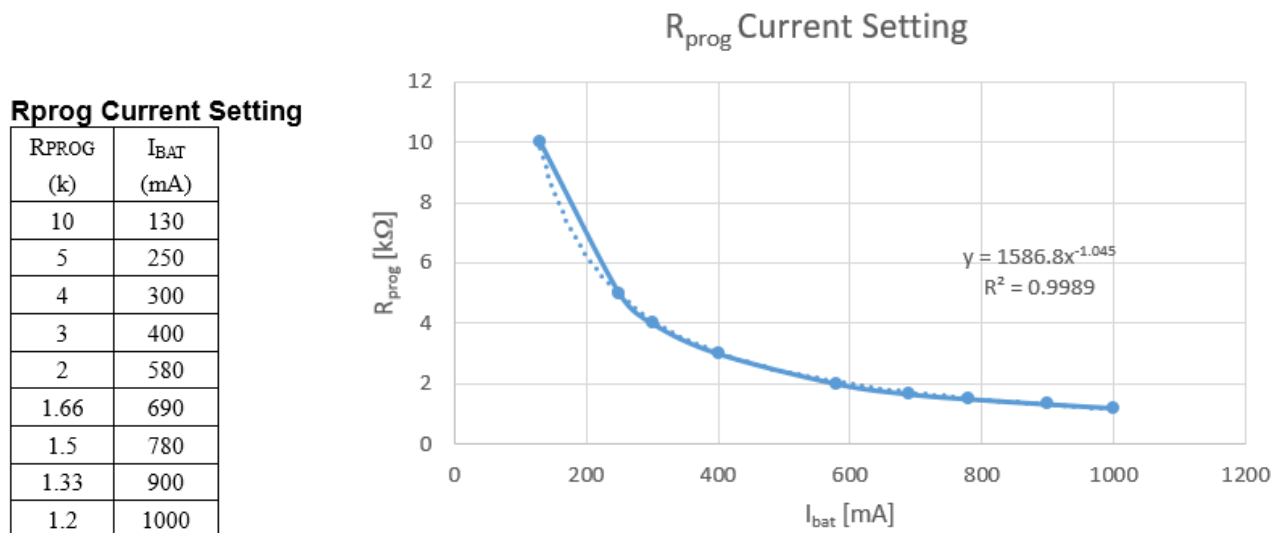


Figura 4.20: Módulo cargador de baterías con el circuito TP4056 incluido

A pesar de ser un cargador para baterías tipo Li-Ion, al revisar la hoja de especificaciones se menciona que se trata de un cargador a corriente constante, por lo que el voltaje de salida a la batería oscila entre los 4.137 [V] y 4.263 [V] con un valor nominal de 4.2 [V]. Mas la

corriente a la salida es de 1 [A] para el módulo de la Fig. 4.20. Dentro de las especificaciones de la batería, se menciona que la corriente de carga debe cumplir con 1 [c]. Esto significa que para que la batería se cargue en 1 [hr] debe ser alimentada con 150 [mA] por el cargador. Afortunadamente en la hoja de especificaciones del módulo ofrece una tabla que relaciona el valor de una resistencia nombrada R_{prog} para que se deje a consideración del desarrollador la corriente máxima de suministro. En la Tabla mostrada en la Fig. 4.21a se muestran los valores de la resistencia para cada valor de corriente en la batería.



(a) Relación entre el valor del resistor R_{prog} y la corriente de suministro a la batería

(b) Gráfica y regresión de tipo exponencial de la tabla en 4.21

Figura 4.21: Acondicionamiento de la corriente de suministro en el módulo cargador TP4056

Tras graficar los valores de dicha tabla se realiza una regresión de tipo exponencial para obtener una fórmula que nos proporcione un valor cercano a la resistencia que requerimos. Tanto la gráfica como la ecuación obtenida se muestran en la Fig. 4.21b. Partiendo del hecho que se requiere un suministro de 150 [mA] para cumplir con el suministro máximo que se especifica para la carga de la batería, se calcula el valor del resistor R_{prog} ; el cual será reemplazado en la tarjeta de desarrollo.

$$R_{prog} = 1586.8(150)^{-1.045} = 8.4431[k\Omega]$$

Por lo tanto se reemplaza por un resistor en empaquetado SMD con un valor comercial de 8.2 [kΩ].

4.4.4. Interfaces de Usuario

Receptor

La interfaz de usuario para el receptor en conjunto con el sistema de conversión de datos, recibirán los datos por parte del microprocesador de los sistemas de recepción y procesamiento de datos, mediante comunicación serial. Por lo tanto es de suma importancia elegir la velocidad de transmisión serial tanto en el microprocesador como en la aplicación de consola (elegida como solución a la interfaz de usuario y el sistema de conversión de datos), *i.e.* los Baudios.

Acorde a la hoja de especificaciones del ATmega328P, a una frecuencia de 16 [MHz] la velocidad máxima es de 2 [Mbps]; como se muestra en la Fig. 4.22 el error de transmisión correspondiente a esta configuración es del 0.0 %.

Baud Rate [bps]	f _{osc} = 16.0000MHz				f _{osc} = 18.4320MHz				f _{osc} = 20.0000MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	-	-	4	-7.8%	-	-	4	0.0%
1M	0	0.0%	1	0.0%	-	-	-	-	-	-	-	-
Max.(1)	1Mbps		2Mbps		1.152Mbps		2.304Mbps		1.25Mbps		2.5Mbps	

(1) UBRRn = 0, Error = 0.0%

Figura 4.22: Ejemplo de configuraciones de baudios para frecuencias de oscilación comunes

Por tal motivo se diseña el diagrama de flujo correspondiente al programa de consola que se despliegue al usuario para selección del número de transmisores, establezca comunicación con el microcontrolador receptor de datos, y obtenga los mismos; dicho diagrama se muestra en la Fig. 4.23. De la misma manera, se obtiene el diagrama de flujo correspondiente al programa del sistema de conversión y almacenamiento de datos, el cual convierta y salve los datos posterior a la recepción de los mismos; dicho diagrama se muestra en la Fig. 4.24. En éste último, se puede apreciar que los datos recibidos terminan siendo salvados en un formato **.csv* mediante el método *WriteCSV()* guardado en el escritorio del usuario. Todo el programa en conjunto se encuentra detallado en el anexo 10.6.

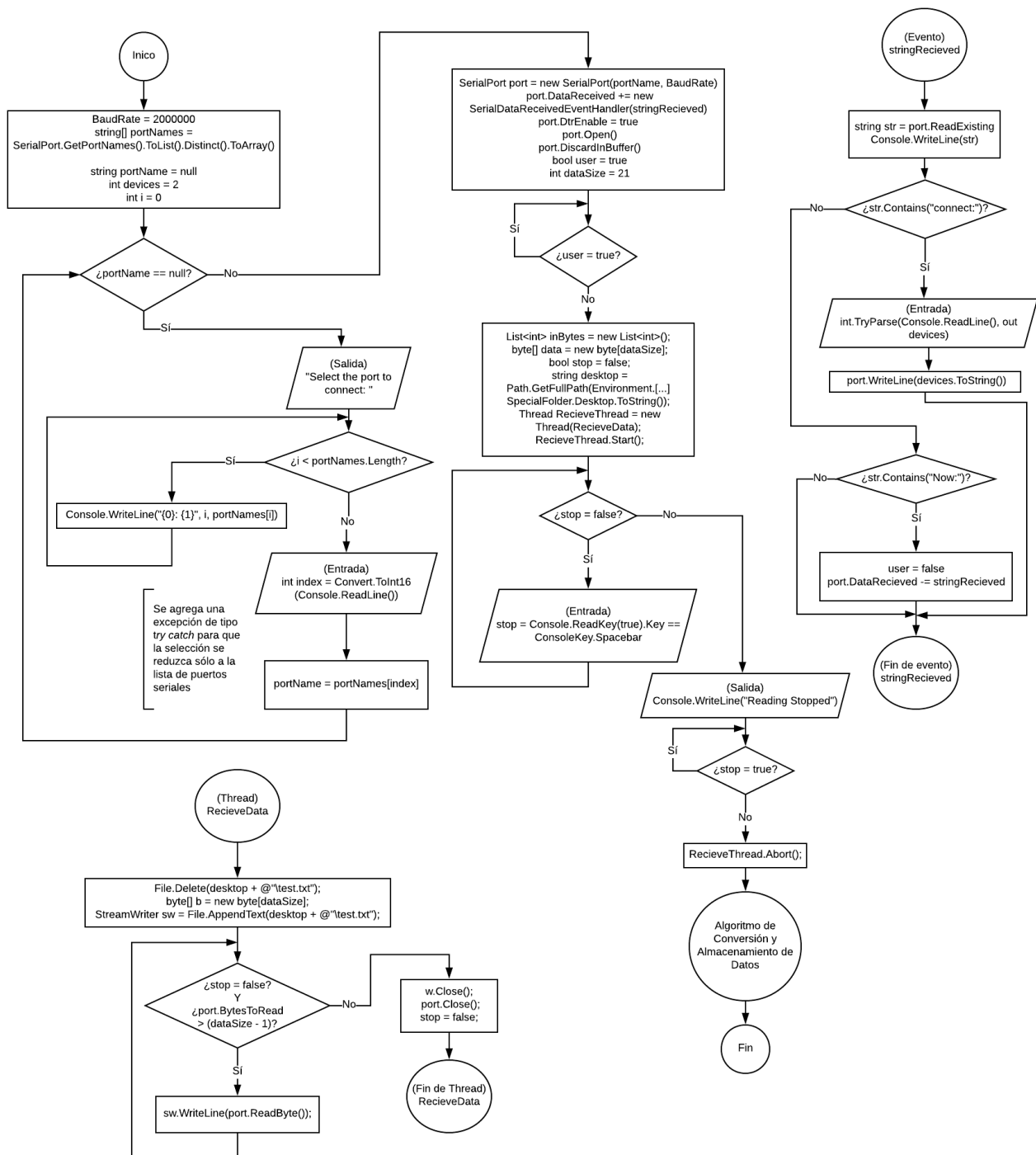


Figura 4.23: Diagrama de flujo de la interfaz de usuario y sistema de obtención de datos

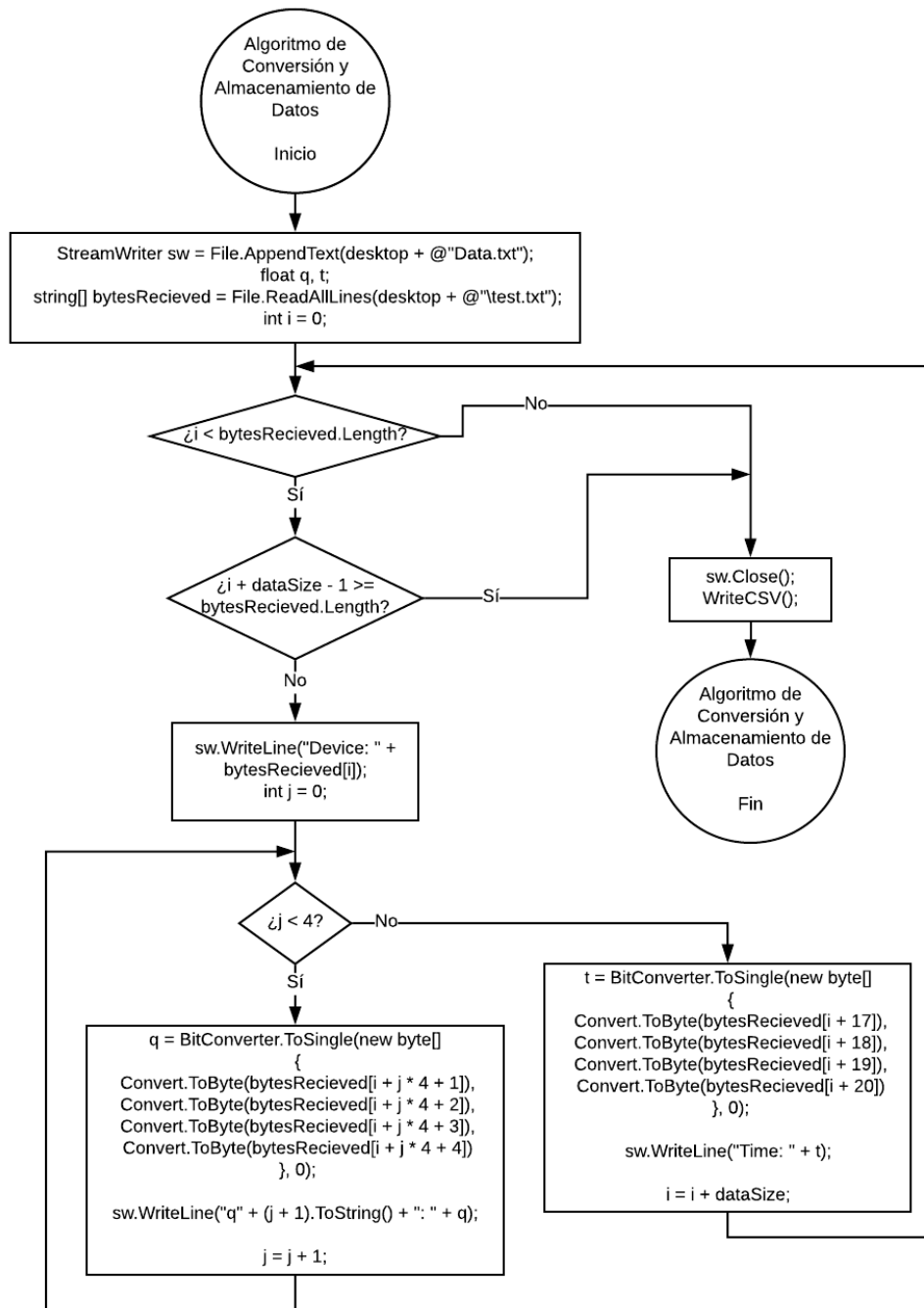


Figura 4.24: Diagrama de flujo del algoritmo de conversión y almacenamiento de datos

Transmisor

Como se mencionó anteriormente, la interfaz de usuario para cada transmisor será reducido a interruptores como entradas y LEDs indicadores de estado como salida.

Para las entradas se utilizan interruptores tipo *switch* deslizable de 1 polo, 2 tiros (dos posiciones con una terminal común):

- **Interruptor de encendido:** Posicionado entre la batería y el sistema de administración de energía. Al cerrar el interruptor se energiza el circuito en su totalidad.
- **Interruptor de calibración del magnetómetro:** Se conecta el común en contacto con

un pin de entrada digital del microcontrolador, mientras que las terminales restantes del interruptor se conectan a 5 [V] y tierra. De esta manera el usuario determina, anterior al encendido del transmisor, el funcionamiento del dispositivo; en caso de que se encuentre conectado a 5 [V] significa que se desea calibrar el magnetómetro al ambiente en el que se encuentra, por el contrario si está conectado a tierra significa que se desea transmitir los datos con los parámetros de calibración guardados en memoria.

El comportamiento de cada LED indicador estará directamente relacionado con el estado en el que se encuentra el programa. Se utilizarán 4 LEDs de colores verde, amarillo, azul, rojo:

- **Configuración de inicio:** LED verde encendido, los demás apagados.
- **Vinculación y transmisión activa de datos:** LED verde encendido, LED azul enciende con cada transmisión de datos (al ser una frecuencia alta, pareciera permanecer encendido), los demás apagados.
- **Calibración del magnetómetro:** Mientras se realice la rutina de calibración del magnetómetro se enciende únicamente el LED amarillo. Una vez que concluya se apaga el LED amarillo y se enciende el LED verde indicando que la calibración finalizó y los parámetros se guardaron en memoria.
- **Error de conexión:** En caso de que el microcontrolador no pueda establecer comunicación con los sensores, la antena o el receptor de datos se enciende el LED rojo mientras los demás permanecen apagados.
- **Nivel de batería bajo:** Desde el inicio del programa y de forma continua se monitorea el nivel de la batería mediante un arreglo de resistores divisores de voltaje y una referencia interna del microcontrolador de 1.1 [V]. Al llegar a un voltaje menor a 3 [V] se detiene el programa y entra en una rutina de parpadeo constante del LED rojo a intervalos de 1 [s] entre encendido y apagado; los demás permanecen apagados.

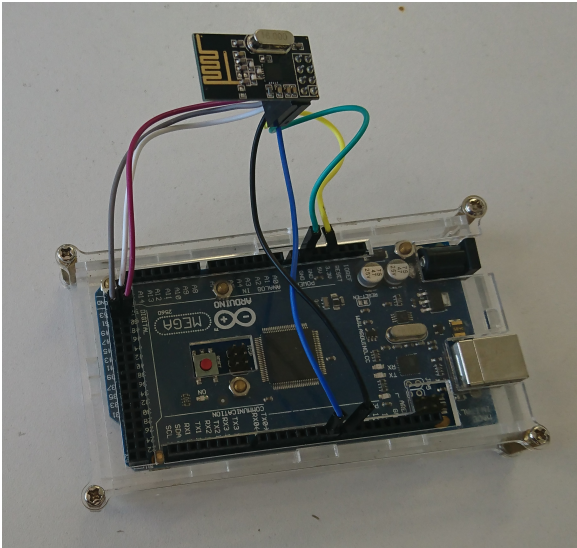
4.4.5. Programación de transmisores

Como se mencionó en la propuesta de solución, este tipo de microcontroladores se pueden programar mediante comunicación SPI en donde otro microcontrolador maestro con el programa *ArduinoISP* (obtenido de los programas ejemplos instalados en el IDE de Arduino) cargado puede programar los microcontroladores esclavos mediante 6 líneas:

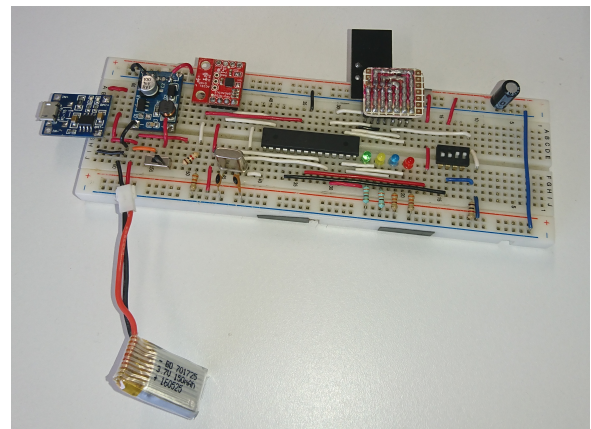
- V_{cc} (5 [V])
- Tierra
- Reset (pin de reset de microcontrolador esclavo)
- Tres líneas de comunicación SPI:
 - MOSI
 - MISO
 - CLK

4.5. Prototipo

Tomando en cuenta lo anterior se procede a construir un modelo funcional del transmisor en una *protoboard* implementando los componentes seleccionados y programando el microcontrolador mediante una tarjeta de desarrollo Arduino UNO; tomando en cuenta el circuito propuesto. Este modelo se muestra en la Fig. 4.25b. Para el receptor se implementa únicamente una tarjeta de desarrollo Arduino MEGA con un módulo nRF24l01 conectado como modelo funcional; éste se muestra en la Fig. 4.25a.



(a) Receptor implementado mediante la tarjeta de desarrollo Arduino MEGA



(b) Modelo funcional del Transmisor en *protoboard*

Figura 4.25: Modelo funcional del SGP

Tras verificar el correcto funcionamiento del SGP con un solo transmisor se procede a realizar un primer prototipo por duplicado, utilizando placas perforadas. A estos prototipos se le añaden 6 terminales correspondientes a los requeridos para la programación por medio de ISP. Se muestran en la Fig. 4.26.

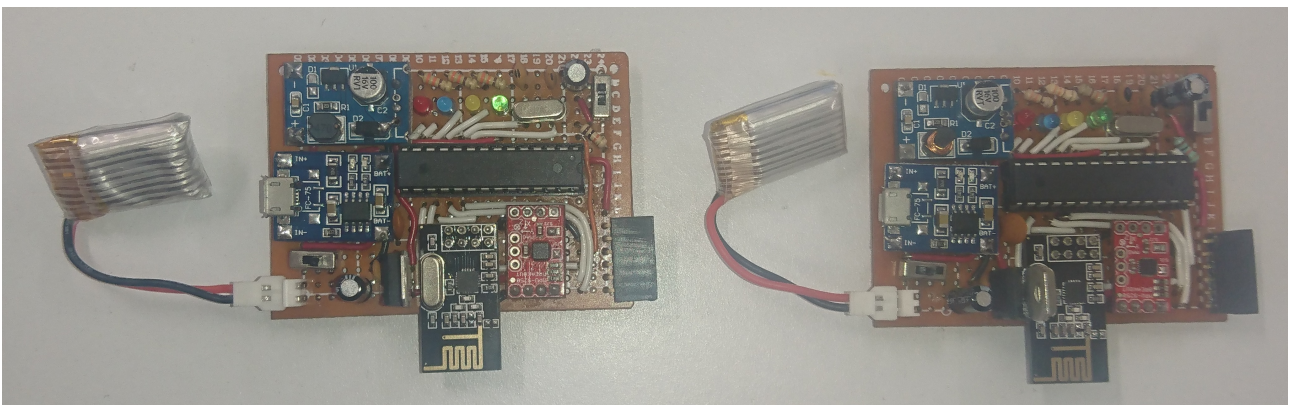


Figura 4.26: Prototipos de transmisores

Para el encapsulamiento de los transmisores, se utilizan láminas de etilvinilacetato (conocido como "*foamy*") que en su conjunto forman una caja en cuyo interior se encuentran fijados con adhesivo el circuito y la batería. Para asegurar la sujeción a la pierna del sujeto de estudio se utilizan correas elásticas ajustables mediante hebilla y velcro.

Los materiales fueron elegidos principalmente para asegurar la comodidad del usuario comprometiendo al mínimo la fijación del dispositivo en el punto de interés. Se tomaron como referencia los componentes del Sistema Shadow, para la construcción de los encapsulados del SGP. En la Fig. 4.27 se muestran los sensores encapsulados con las correas elásticas instaladas.

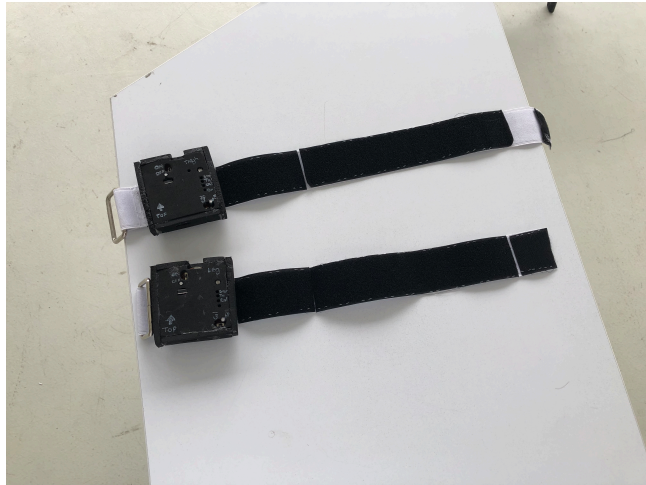


Figura 4.27: Prototipos de transmisores encapsulados

Capítulo 5

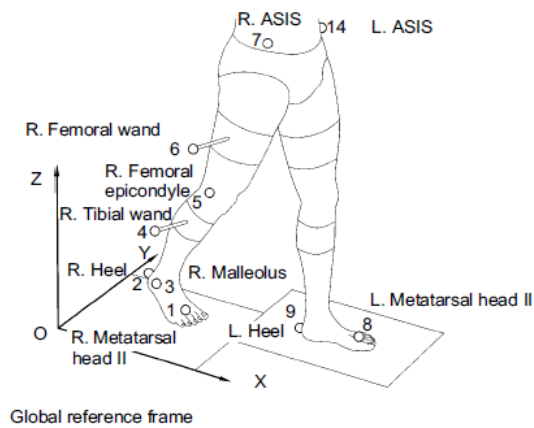
Protocolo de Validación

Para verificar el correcto funcionamiento del SGP en el presente proyecto, se pretende realizar una prueba de marcha implementando dicho sistema. Para la prueba se analizan únicamente los desplazamientos angulares de flexo-extensión de rodilla. Tras obtener los resultados correspondientes se comparan los datos recabados por el SGP con respecto a los datos obtenidos mediante un sistema comercial de captura de movimiento a base de sensores tipo MARG: "**Lower Body Motion Shadow System**" (Para este proyecto, referido simplemente como el Sistema Shadow), aplicando la misma prueba. Éste sistema consiste de 7 sensores interconectados de forma alámbrica a un módulo central, al cual un usuario se conecta mediante conexión Wi-Fi y el uso de un navegador web conectado a la dirección IP que proporciona el fabricante.

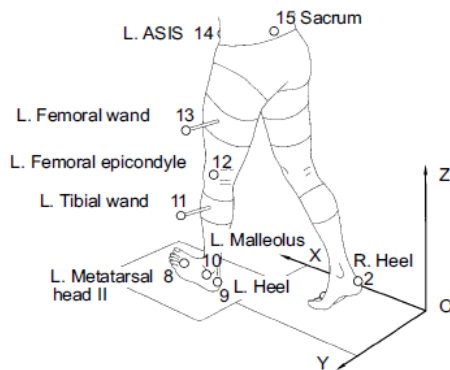
Los sensores que integran al sistema propuesto serán posicionados siguiendo el arreglo Helen Hayes como se muestra en la Fig. 5.1a (en donde se colocan los sensores en lugar de las varas con marcadores), siendo una configuración diferente a la que se ocupa en el Sistema Shadow, dado que el sensor del Sistema Shadow en el centro de masa aproximado del segmento tibia-peroné se coloca en dirección anterior de la pierna, como se puede apreciar en la Fig. 5.1b (a diferencia del arreglo Helen Hayes, en donde el marcador se coloca en dirección lateral).

Se pretende realizar la prueba a un grupo reducido de personas debido a que la comparación entre sistemas se realiza con respecto al ciclo de marcha del sujeto de pruebas y no al de un grupo control. Por consiguiente se informa al sujeto de prueba de las actividades a realizar y los riesgos vinculados con las pruebas con la finalidad de recibir el consentimiento del grupo en cuestión, mediante una Carta de Consentimiento Informado (Ver Anexo 10.8). Finalmente se procede al Protocolo de Pruebas mostrado a continuación.

Para el presente protocolo, se toma como referencia el protocolo utilizado en las pruebas de marcha del trabajo de investigación del M.I. Carlos Canchola. [3]



Global reference frame



(a) Arreglo Helen Hayes de marcadores para tren inferior



(b) Persona con los sensores del Sistema Shadow posicionados

Figura 5.1: Posicionamiento de los sensores

5.1. Protocolo de Pruebas

5.1.1. Objetivo

Adquirir los desplazamientos angulares de flexo-extensión de rodilla derecha de un grupo de mínimo 4 personas con distintas características antropométricas, implementando tanto el sistema propuesto (SGP) como el sistema comercial (Sistema Shadow).

5.1.2. Lugar de pruebas

Sótano del Centro de Ingeniería Avanzada en el Anexo de la Facultad de Ingeniería, de la Universidad Nacional Autónoma de México. Ciudad Universitaria, Ciudad de México.

5.1.3. Fecha

Jueves, 17 de enero de 2019.

5.1.4. Instrumentos de medición

Lower Body Motion Shadow System (7 sensores, 2 plantillas y 1 concentrador)
Sistema de Goniometría Propuesto (2 sensores transmisores, 1 receptor, 1 equipo de cómputo)



Figura 5.2: Sensores del Sistema Shadow (derecha) y sensores del sistema propuesto (izquierda)

5.1.5. Módulos

Antropometría

Para éste módulo se desea obtener la posición idónea para la colocación de los sensores del sistema propuesto y los del sistema Shadow. Por tal motivo se divide en dos etapas:

1. Obtención y registro de las medidas mostradas en la Tabla 5.1
2. Determinación de los centros de masa y colocación de etiquetas indicadoras

Dato Antropométrico	Valor [cm]
Longitud de muslo derecho	
Longitud de pierna derecha	

Tabla 5.1: Datos antropométricos

Para los datos antropométricos se considera que la longitud del muslo es la distancia entre el punto superior del trocánter mayor del fémur y el margen superior del epicóndilo lateral de la tibia, mientras que la longitud de pierna es la distancia entre el margen superior del epicóndilo lateral de la tibia y el maléolo lateral en el tobillo. [37] En ambos casos se realiza únicamente para la pierna derecha, debido a que se implementan únicamente dos sensores transmisores del SGP.

Para la determinación del centro de masa de cada segmento, se toman como referencia los parámetros reportados por Vaughan *et al.* en la sección "*Prediction of Segment Moments of Inertia*" del artículo "*Dynamics of Human Gait*". En dicha sección se muestra una tabla que presenta diversos parámetros de cada segmento de las extremidades inferiores; para conocer el centro de gravedad"de cada segmento ocupa una proporción entre la longitud de la posición del centro de gravedad y la longitud distal (Ver Fig. 5.3). Para las medidas necesarias, las proporciones son las siguientes:

- Longitud de muslo derecho: 0.39
- Longitud de pierna derecha: 0.42

Una vez identificados los centros de gravedad de cada segmento, se coloca una etiqueta adhesiva a modo de marcador visual en el punto coincidente con el centro de gravedad y el eje anatómico en dirección lateral. Dicha etiqueta se toma como referencia para la colocación de los sensores durante los siguientes módulos.

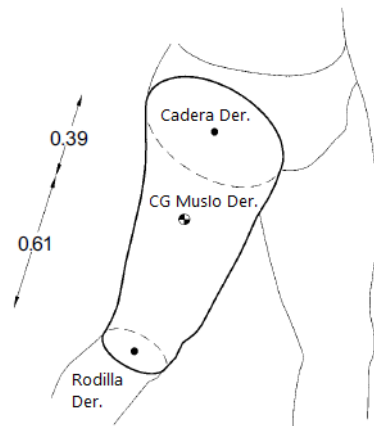


Figura 5.3: Localización del centro de gravedad del muslo derecho

Número mínimo de colaboradores en el módulo: 1: Persona del área clínica que tome lectura y registro de los datos

Colaborador(es):

Sofía Atenea Pimentel López (estudiante de la Facultad de Medicina de la UNAM)

Instrumentos de medición:

- Cinta métrica

Tiempo estimado de prueba: 2 minutos



(a) Medición de longitud de muslo



(b) Colocación de marcador en centro de masa calculado

Figura 5.4: Módulo 1

Prueba de marcha

Con la finalidad de recabar los datos de una misma prueba de marcha, se utilizan de manera simultanea tanto el SGP como el Sistema Shadow.

Se inicia con la calibración de los transmisores del SGP previo a las pruebas en el lugar designado. Posteriormente se procede a la colocación de los sensores (transmisores) del SGP sobre los marcadores visuales colocados en el módulo anterior. A continuación, se colocan los sensores del Sistema Shadow en la parte posterior de la pelvis y en los centros de masa aproximados de pierna, muslo y pie tanto derechos como izquierdos; para el muslo y pierna se colocan las cintas de sujeción de los sensores por debajo de los sensores del sistema de goniometría, en la dirección correspondiente.

La prueba de marcha consta de dos etapas: una **Prueba estática** en la cual se recaban los datos iniciales de los sensores, y una **Prueba dinámica** en la que se obtienen los datos propios de la marcha humana. Con ambos sistemas colocados, se le solicita a la persona que permanezca quieto en su sitio y en posición de reposo durante 10 segundos. Una vez finalizado este tiempo, se le pide a la persona que camine sobre una pista recta de 4 metros trazada en el suelo de ida y vuelta; siendo un evento de marcha.^{el} andar de la persona en una dirección. Finalmente, se capturan un mínimo de 4 eventos de forma continua.

Número mínimo de colaboradores en el módulo: Tres: Dos personas para la colocación y desmontaje de los sensores (uno por lado para el caso del Sistema Shadow) así como el encendido y apagado de los sensores y una persona para la captura de datos vía remota.

Colaborador(es): (Del Centro de Diseño Mecánico e Innovación Tecnológica (CD-MIT) de la Facultad de Ingeniería de la UNAM)

Ing. Luis Darío Reyes Delgado

Ing. Anahí Velázquez Silva

Ing. David Negrete Rojas

Instrumentos de medición:

- Lower Body Motion Shadow System
- Sistema de goniometría propuesto
- Equipo de cómputo para adquisición de datos de ambos sistemas



(a) Colocación de ambos sistemas al sujeto de pruebas



(b) Ambos sistemas colocados



(c) Captura del ciclo de marcha

Figura 5.5: Módulo 2

Capítulo 6

Resultados

Las pruebas se realizaron en el tiempo y lugar establecidos en el protocolo de pruebas. Se invitaron a participar a 6 personas como sujetos de prueba, de los cuales todos se presentaron. Se cumple con el objetivo del protocolo que pretende seleccionar personas de distintos datos antropométricos, como se puede observar en el Anexo 10.9. A su vez, se realizaron las pruebas correspondientes al protocolo mismo y se obtuvieron los archivos de ambos sistemas para cada uno de los participantes. Sin embargo, al analizar los archivos dos de ellos resultaron estar dañados por lo que se trabaja con los datos de únicamente cuatro sujetos de estudio para el análisis de los resultados.

En cuestión a la sincronización de los sensores al inicio de las pruebas, resultó ser bastante rápida y de mayor comodidad para el SGP en comparación con el Sistema Shadow ya que la vinculación se realizaba de manera automática al encenderlos. Sin embargo, en varias ocasiones ambos sistemas presentaron fallas de comunicación con la computadora. En dichos momentos la reconexión con el Sistema Shadow resultaba ser más ágil en comparación con el SGP. En ocasiones el usuario en el computador no se percataba de la desconexión de alguno de los dos transmisores sino hasta el final de la prueba. Las dos principales causas de la desconexión de los sensores del SGP fueron:

- El encapsulado de foamy que, tras realizar un número determinado de pruebas, resultó ser poco fiable. Durante en las pruebas realizadas al último sujeto de estudios se despegó la cubierta superior. De tal manera que se tuvo que finalizar la prueba.
- Un falso contacto en el interruptor de encendido, detectado en el circuito de uno de los transmisores, provocando así una interrupción en el envío de datos.

Una vez finalizadas las pruebas correspondientes, se obtienen archivos tipo (*.CSV) tanto del Sistema Shadow como del SGP. El primero contiene las orientaciones en el espacio representado mediante ángulos de Euler de cada uno de los sensores, mientras que el segundo almacena las orientaciones representadas mediante cuaterniones. Con estos datos, se pueden obtener las orientaciones de cada uno de los sensores y por ende es posible asumir las orientaciones del segmento del cuerpo a la que dichos sensores se encuentra fijo. Al representar mediante un sistema de referencia tanto la pierna como el muslo, se pretende visualizar al desplazamiento angular de flexo-extensión de rodilla como el desplazamiento angular mismo de los sistemas de referencia dentro de un mismo plano. Por ejemplo, el ángulo que se forma entre los ejes anatómicos de cada segmento.

Para el caso del Sistema Shadow, la calibración inicial de los sensores previa a la captura del movimiento posiciona a los ejes «Z» paralelos al eje anatómico tanto de pierna como de muslo, mientras que a los ejes «Y» en dirección frontal al cuerpo. Por lo tanto, la variación del ángulo

de flexo-extensión en rodilla derecha se puede monitorizar mediante la medición del desplazamiento angular entre los ejes «Z» de los sensores de pierna y muslo. La disposición de de los ejes para este caso se muestra en la Fig. 6.1a

Para el SGP, (tomando en cuenta la previa calibración de los magnetómetros) posiciona a los ejes «Z» en dirección lateral del cuerpo, mientras que los ejes «X» y «Y» en las direcciones distal y frontal respectivamente. Por lo cual, el análisis de la dinámica del ángulo de flexo-extensión en rodilla derecha sería posible al monitorizar los desplazamientos angulares entre los ejes «X» y los ejes «Y». La disposición de los ejes en este otro caso se muestra en la Fig. 6.1b

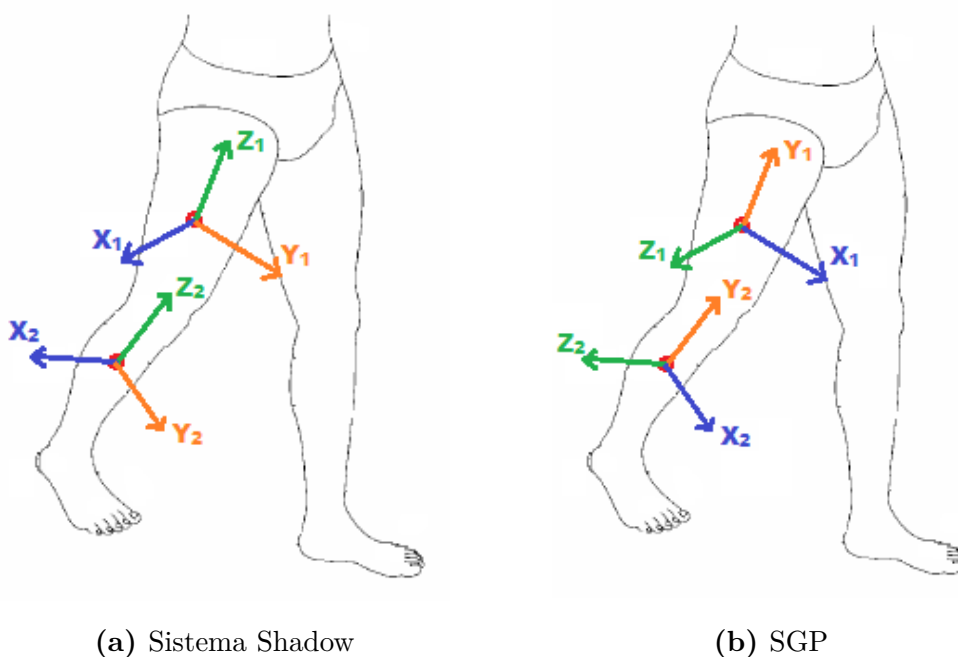


Figura 6.1: Disposición de los ejes en ambos sistemas implementados

Como se menciona en el protocolo de pruebas, éstas se inician con una **prueba estática**. Dicha prueba tiene como la finalidad de visualizar la orientación inicial de los sensores en ambos sistemas a analizar. Debido a que los sensores del sistema propuesto se calibran de manera independiente, se espera que las orientaciones iniciales no coincidan entre sí. Por tal motivo, para un mejor análisis de la medición del fenómeno de estudio, se considera necesario aproximar las condiciones iniciales tanto del SGP como las del Shadow trasladando su valor inicial al origen (a cero). Por consiguiente, previo al el post-procesamiento de los datos de ambos sistemas, se toman en cuenta dichas orientaciones iniciales y se alteran los datos de tal manera que coincidan los sistemas de referencia para pierna y muslo en la posición de reposo (Ángulo de flexo-extensión igual a cero). A éste proceso se le identifica en adelante como: Normalización.

Normalización de las señales del SGP[12 pt] Se inicia con el análisis de la variación en el tiempo de los cuaterniones obtenidos:

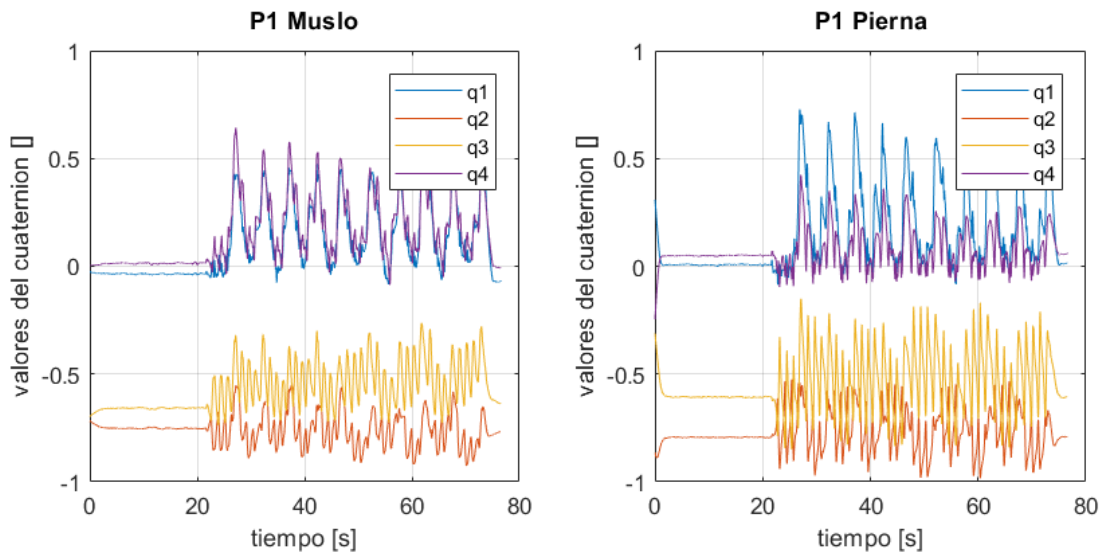


Figura 6.2: Variación de los cuaterniones en sensores de pierna y muslo en el tiempo para cada sujeto de pruebas - Persona 1

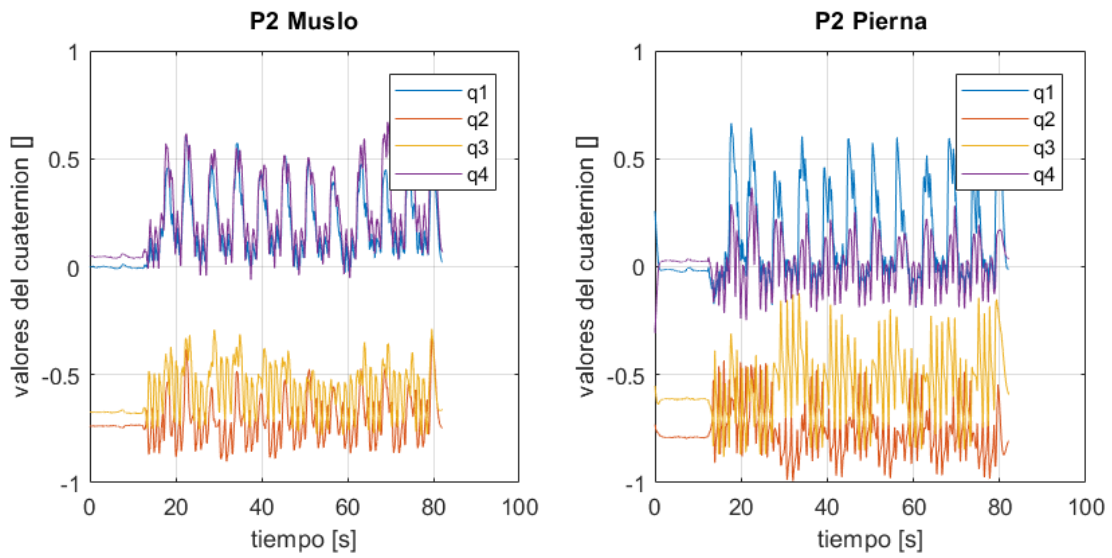


Figura 6.3: Variación de los cuaterniones en sensores de pierna y muslo en el tiempo para cada sujeto de pruebas - Persona 2

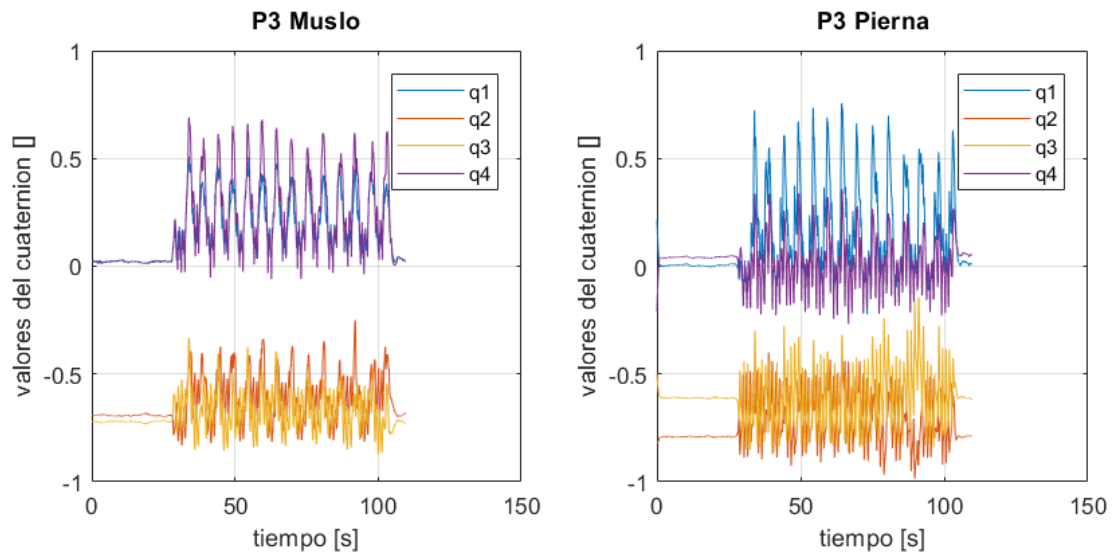


Figura 6.4: Variación de los cuaterniones en sensores de pierna y muslo en el tiempo para cada sujeto de pruebas - Persona 3

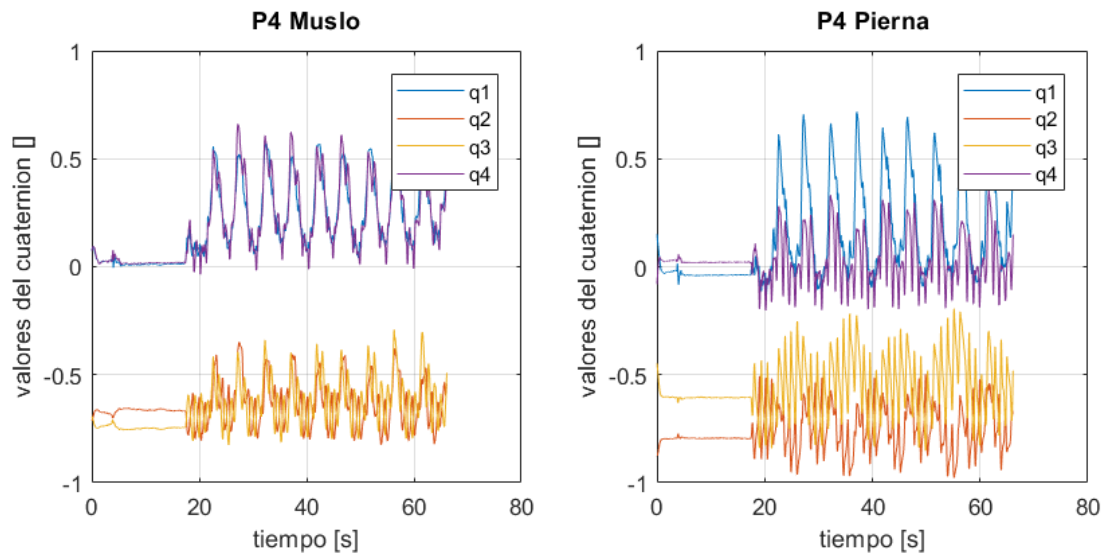


Figura 6.5: Variación de los cuaterniones en sensores de pierna y muslo en el tiempo para cada sujeto de pruebas - Persona 4

Como se observa en las Fig. 6.2 a 6.5, los cuaterniones obtenidos tienen componentes con valores distintos a $[1 \ 0 \ 0 \ 0]$, por lo cual existe un cierto valor inicial durante la etapa de la prueba estática. Dichos valores se promedian para obtener un valor de «offset» de los cuaterniones, el cual se pretende eliminar previo al post-procesamiento de datos. Una vez calculado el valor de offset, se postmultiplica su conjugado por cada uno de los cuaterniones del vector de datos. Por lo tanto, se «fuerza una orientación inicial nula» durante la prueba estática: $q_{inicial} = [1 \ 0 \ 0 \ 0]$, es decir se elimina el offset inicial para todos los valores de la señal.

Tomando en cuenta el cuaternión inicial, se aplica la rotación a los ejes unitarios \hat{i} , \hat{j} y \hat{k} de un sistema de referencia propuesto que represente la orientación en el espacio de un sensor. Se aplican las rotaciones utilizando cada uno de los cuaterniones resultantes y guardando los grupos de vectores obtenidos como la dinámica del sistema de referencia propuesto.

Tras realizar el mismo procedimiento descrito anteriormente tanto para el sensor colocado en muslo como para el sensor en pierna, se analizan los desplazamientos angulares entre los ejes coincidentes de ambos sistemas de referencia. Dichos desplazamientos se obtienen mediante el producto punto entre el par de vectores en «X» y el par en «Y» (« X_{muslo} » con « X_{pierna} » y « Y_{muslo} » con « Y_{pierna} », respectivamente). En las Fig. 6.6 - 6.9 se muestran los desplazamientos angulares de flexo-extensión de los cuatro sujetos de estudio.

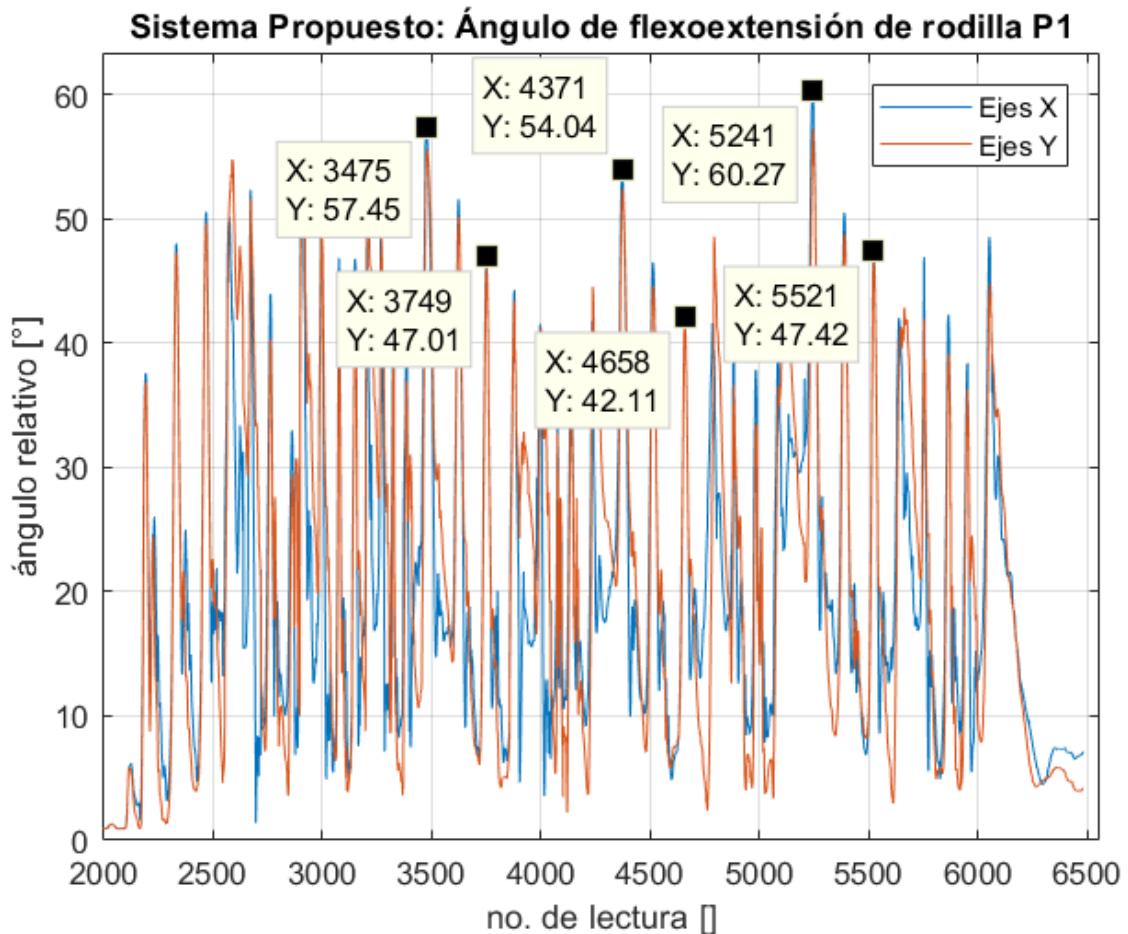


Figura 6.6: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del SGP) - Persona 1

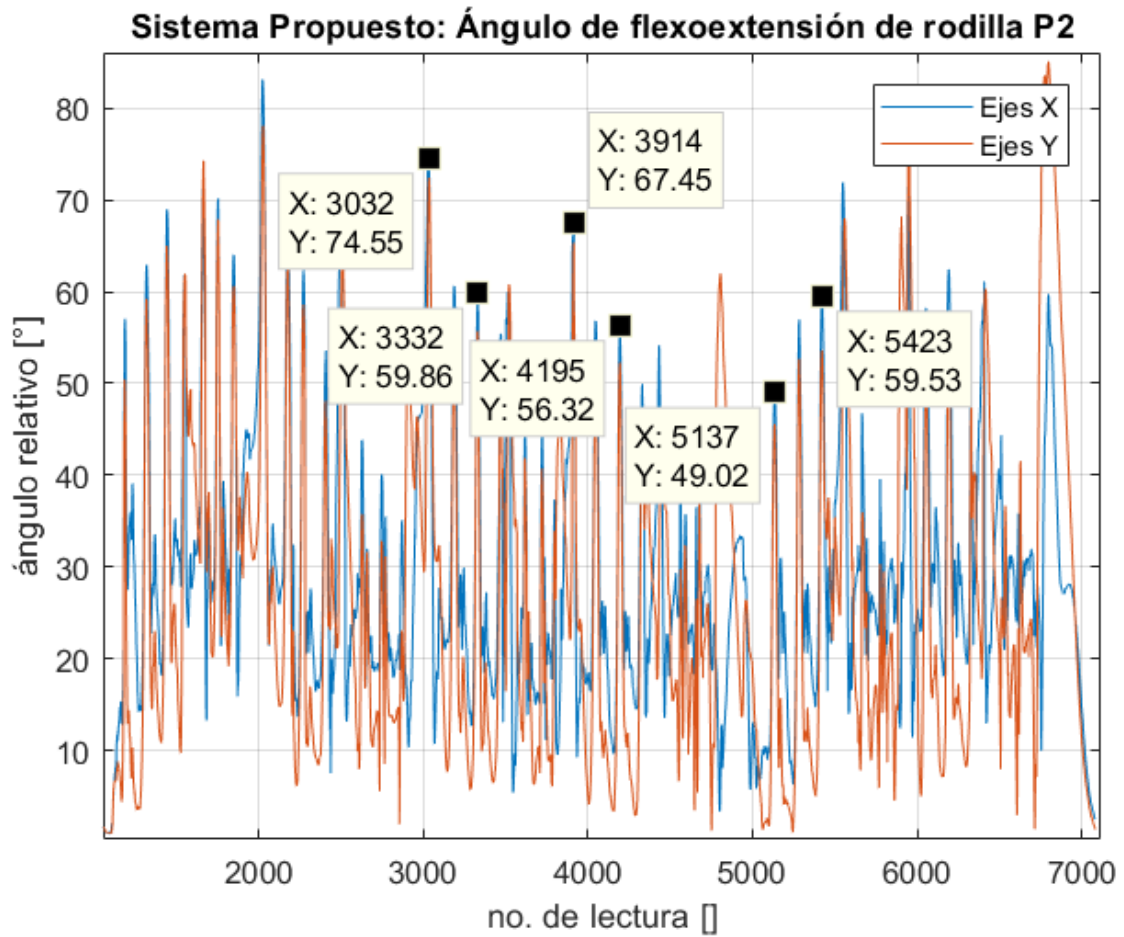


Figura 6.7: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del SGP) - Persona 2

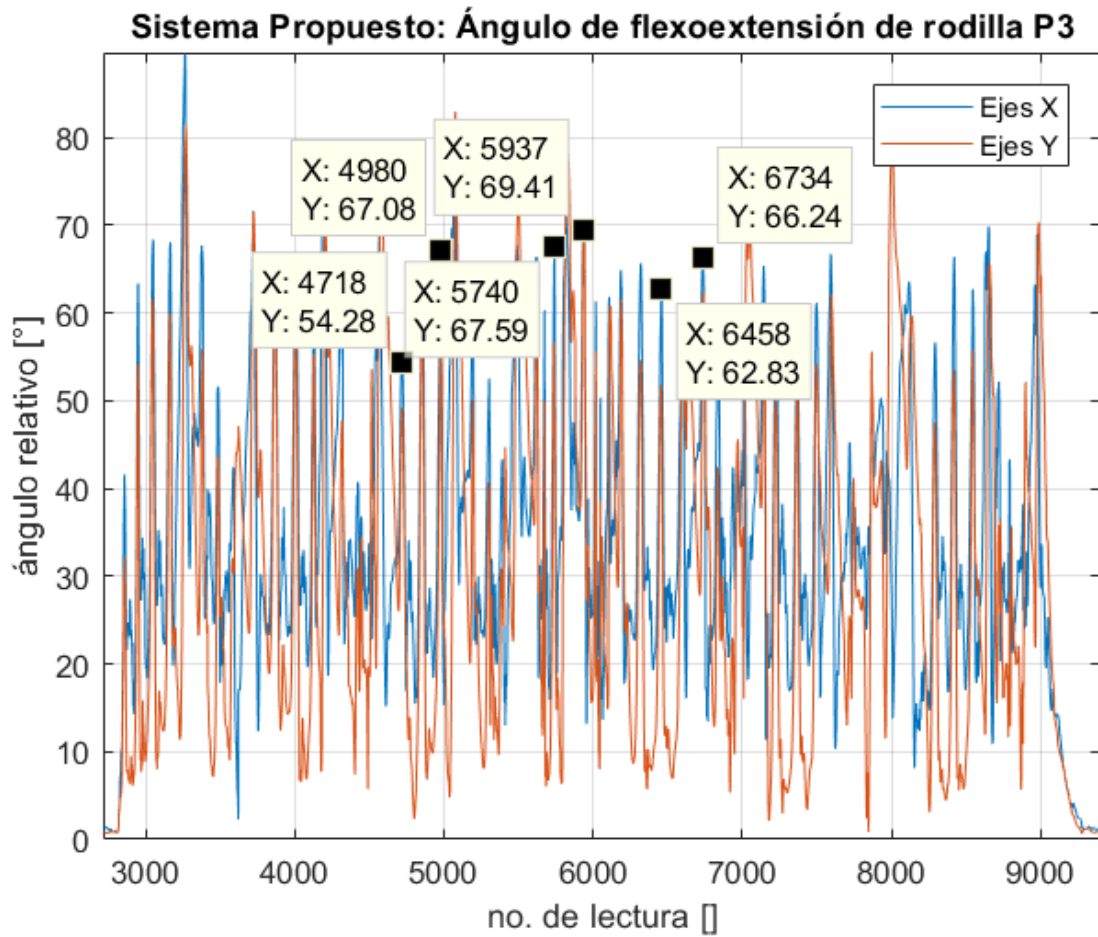


Figura 6.8: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del SGP) - Persona 3

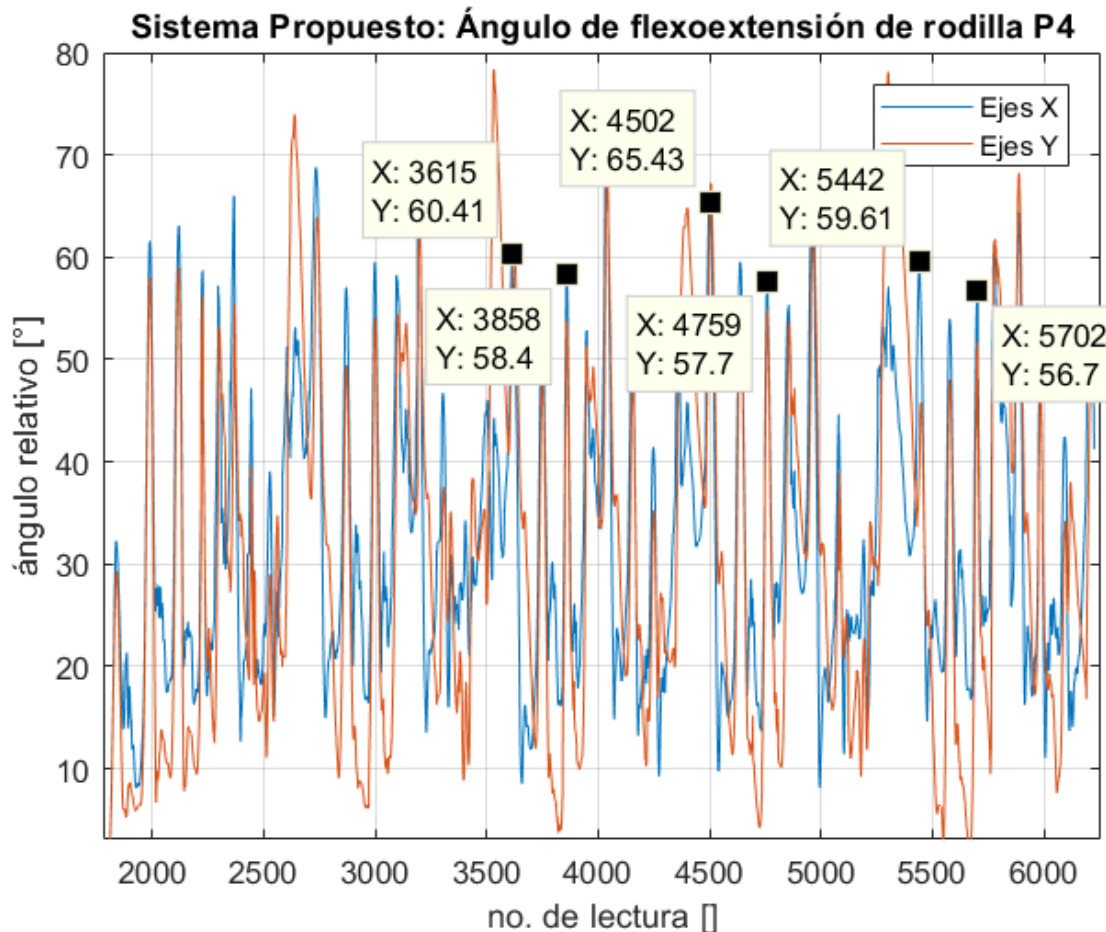


Figura 6.9: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del SGP) - Persona 4

Normalización de las señales del Sistema Shadow[12 pt] Posteriormente, del Sistema Shadow, se calculan los ángulos de flexo-extensión capturados por los sensores. Para ello se leen los datos del archivo *.CSV obtenidos tras las pruebas correspondientes y se toman los ángulos de Euler de muslo y pierna derechas.

Al igual que se realizó con los cuaterniones para el SGP, en este caso se aplican las rotaciones a los ejes unitarios \hat{i} , \hat{j} y \hat{k} de un sistema de referencia propuesto que represente la orientación en el espacio de uno de los sensores del Sistema Shadow, mediante Rotaciones de Euler X - Y - Z. Tras tener las orientaciones de cada sensor, se procesa el ángulo de flexo-extensión de rodilla capturado por el Sistema Shadow mediante la variación del ángulo existente entre el par de vectores en «Z» (« Z_{muslo} » con « Z_{pierna} »). Posteriormente, se selecciona la región en la que la prueba estática se realiza, con la finalidad de calcular el valor inicial del ángulo de flexo-extensión y normalizar la señal. Este cálculo se realiza tomando la media de los valores del ángulo en la fase de prueba estática y restando dicho valor a cada elemento del vector de datos.

En las Fig. 6.10 - 6.13 se muestran los desplazamientos angulares de flexo-extensión resultantes de los cuatro sujetos de estudio.

Selección de muestras de análisis

Para realizar un adecuado análisis estadístico de los datos obtenidos, se considera suficiente tomar tres muestras de cada conjunto de datos para obtener un total de 12 muestras. Las

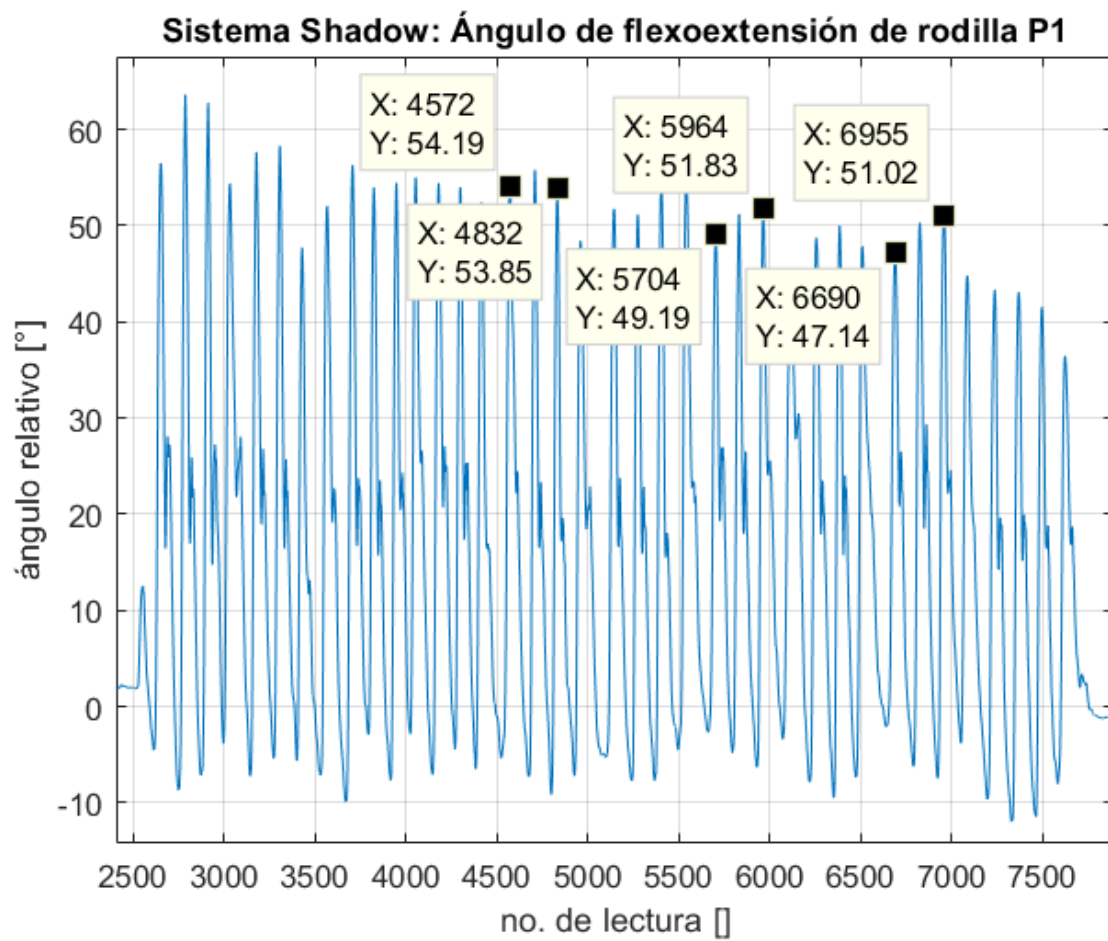


Figura 6.10: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del Sistema Shadow) - Persona 1

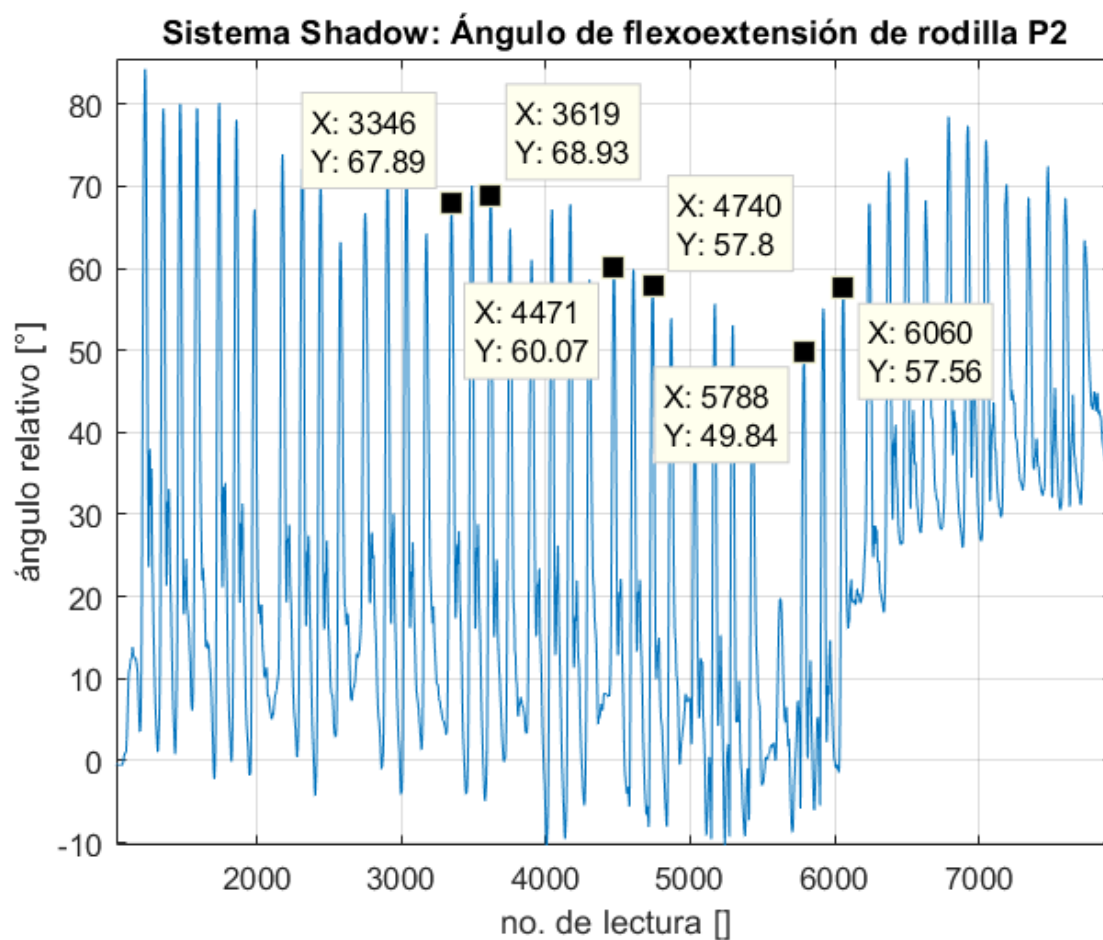


Figura 6.11: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del Sistema Shadow) - Persona 2

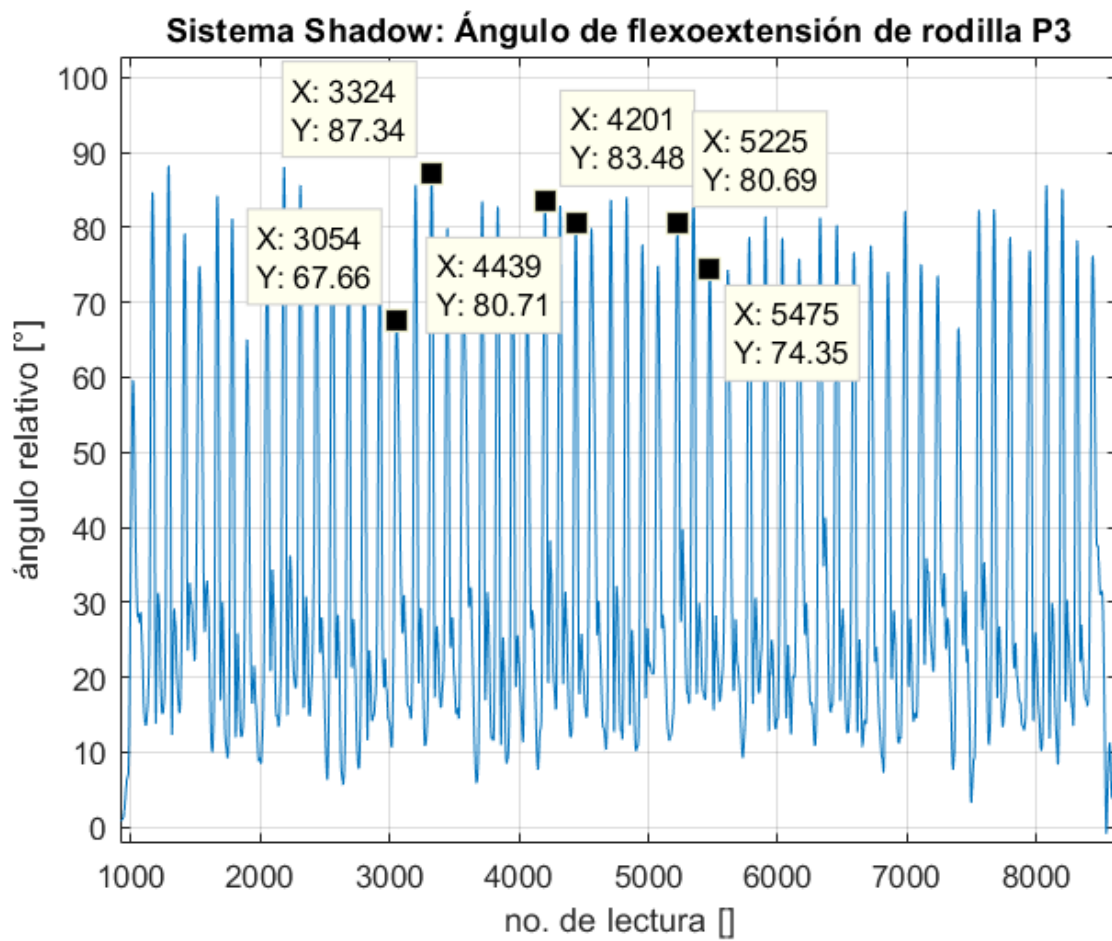


Figura 6.12: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del Sistema Shadow) - Persona 3

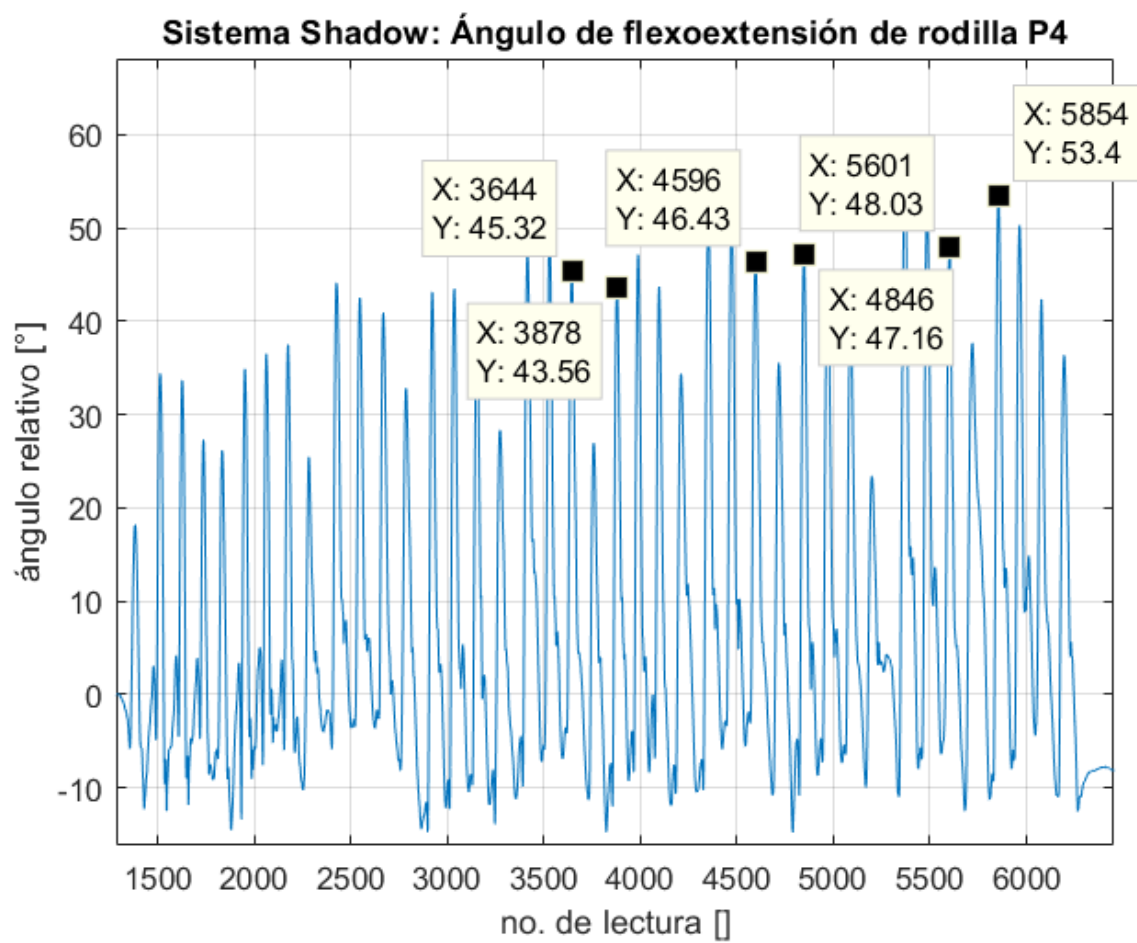


Figura 6.13: Ángulo de flexo-extensión de rodilla (procesado de las señales de los sensores del Sistema Shadow) - Persona 4

muestras se toman a partir del tercer evento (siendo un «evento» el andar de un sujeto de estudio en una misma dirección) y solo se analizan los datos contenidos dentro de los tres puntos máximos a la mitad de un evento; tomando en cuenta que, cada máximo local o pico representa el ángulo máximo de flexo-extensión en rodilla. Esta selección tiene como propósito analizar los ciclos de marcha que realice el sujeto de estudio con mayor naturalidad. Se omiten los primeros dos eventos para que la marcha en general se regularice; por otro lado, para los eventos analizados, se omiten los primeros y últimos ciclos para que se normalice la marcha previo y posterior a una media vuelta (Para cada sujeto de estudio el número de ciclos a omitir varía dentro de un rango de 2 a 4). De las Fig. 6.6 a 6.9 se toman los datos mostrados en la Tabla 6.1.

Sujeto de Estudio	Análisis	Primer Paso	Último Paso	Primer Dato	Último Dato
1	1	16	18	4572	4832
	2	24	26	5704	5964
	3	31	33	6690	6955
2	4	16	18	3346	3619
	5	24	26	4471	4740
	6	32	34	5788	6060
3	7	17	19	3054	3324
	8	26	28	4201	4439
	9	34	36	5225	5475
4	10	20	22	3644	3878
	11	28	30	4596	4846
	12	36	38	5601	5854

Tabla 6.1: Datos de análisis del SGP

Del mismo modo que se seleccionan los datos de análisis para el SGP se seleccionan los datos para el Sistema Shadow. De las Fig. 6.10 a 6.13 se toman los datos mostrados en la Tabla 6.2.

Sujeto de Estudio	Análisis	Primer Paso	Último Paso	Primer Dato	Último Dato
1	1	16	18	4572	4832
	2	24	26	5704	5964
	3	31	33	6690	6955
2	4	16	18	3346	3619
	5	24	26	4471	4740
	6	32	34	5788	6060
3	7	17	19	3054	3324
	8	26	28	4201	4439
	9	34	36	5225	5475
4	10	20	22	3644	3878
	11	28	30	4596	4846
	12	36	38	5601	5854

Tabla 6.2: Datos de análisis del Sistema Shadow

Para analizar correctamente los datos, es necesario que ambos vectores contengan el mismo número de elementos. De las muestras a analizar, es posible apreciar que existe una diferencia en la cantidad de datos por sistema. Esto se debe a que el Sistema Shadow demuestra tener una velocidad de muestreo mayor que la del sistema propuesto, por lo cual es necesario ajustar uno de los vectores de datos para que ambas señales contengan el mismo número de datos. Esto se realiza mediante un «redimensionamiento» del vector de datos del Sistema Shadow a un vector más pequeño, mediante el método del valor más cercano, el cual proporciona un vector de datos de un tamaño proporcional a una escala establecida.

Tras sesgar los datos del ángulo de flexo-extensión tanto en el caso del Sistema Shadow como del SGP, se realiza el «redimensionamiento» antes descrito para lograr la coincidencia en el número de datos. Por consiguiente, se obtienen las gráficas de las señales de las muestras de ambos sistemas en un mismo sistema de referencia y con el mismo número de elementos. De las Fig. 6.14 a 6.25, se muestran las gráficas correspondientes a cada análisis.

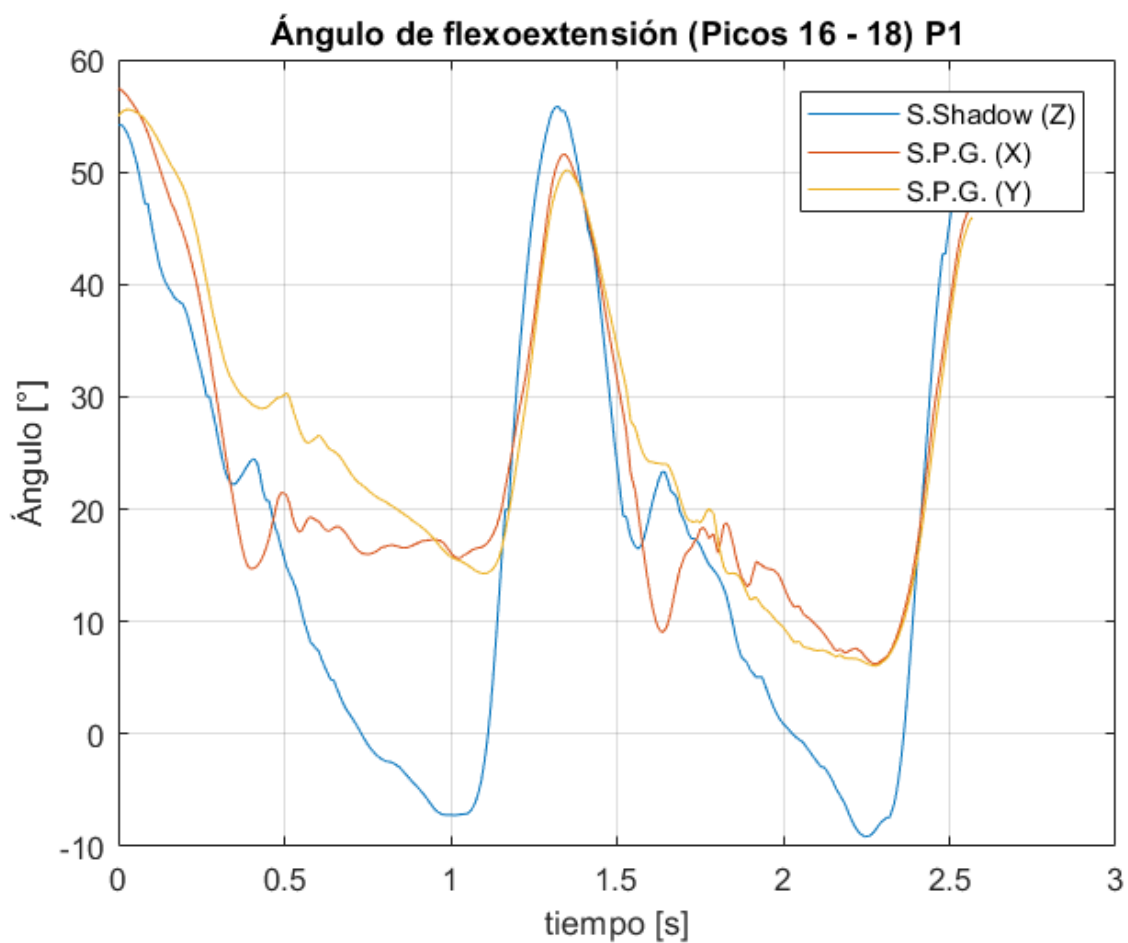


Figura 6.14: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 1

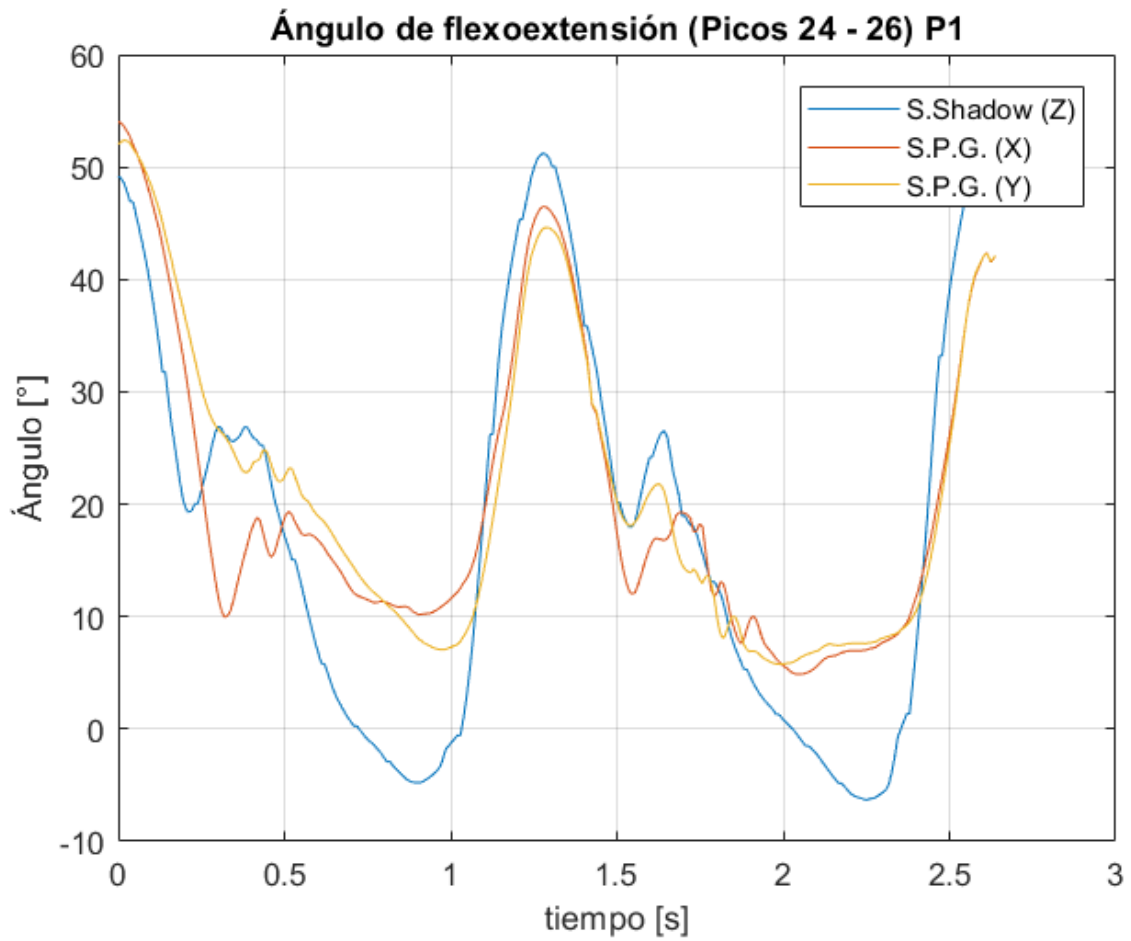


Figura 6.15: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 2

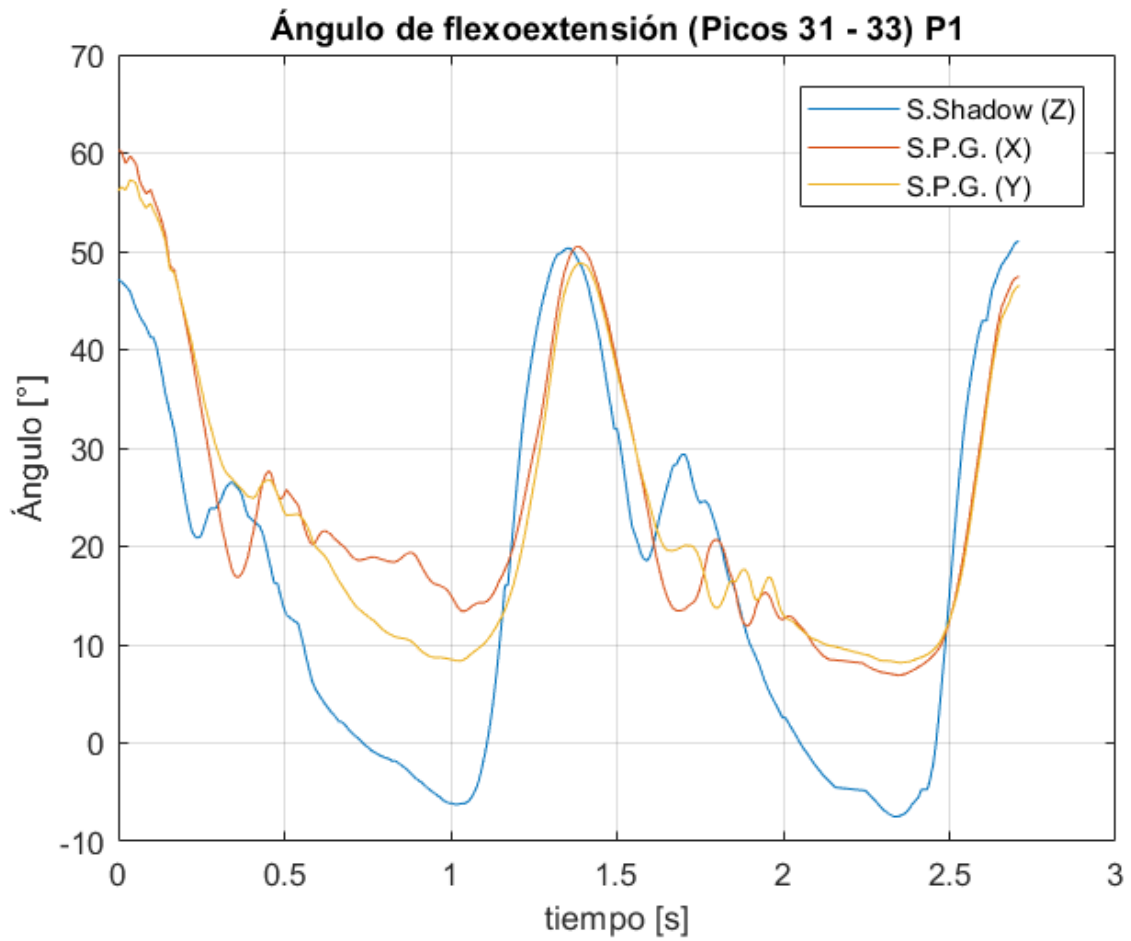


Figura 6.16: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 3

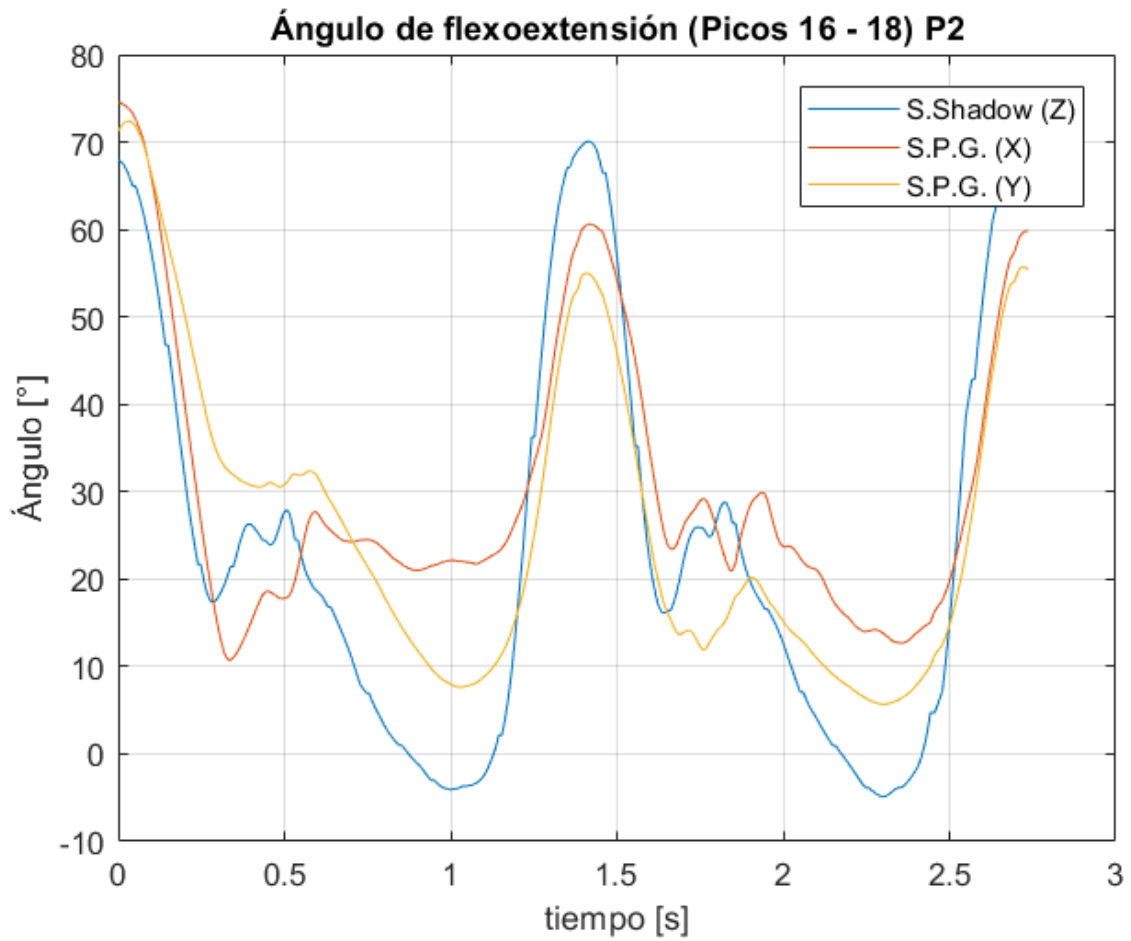


Figura 6.17: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 4

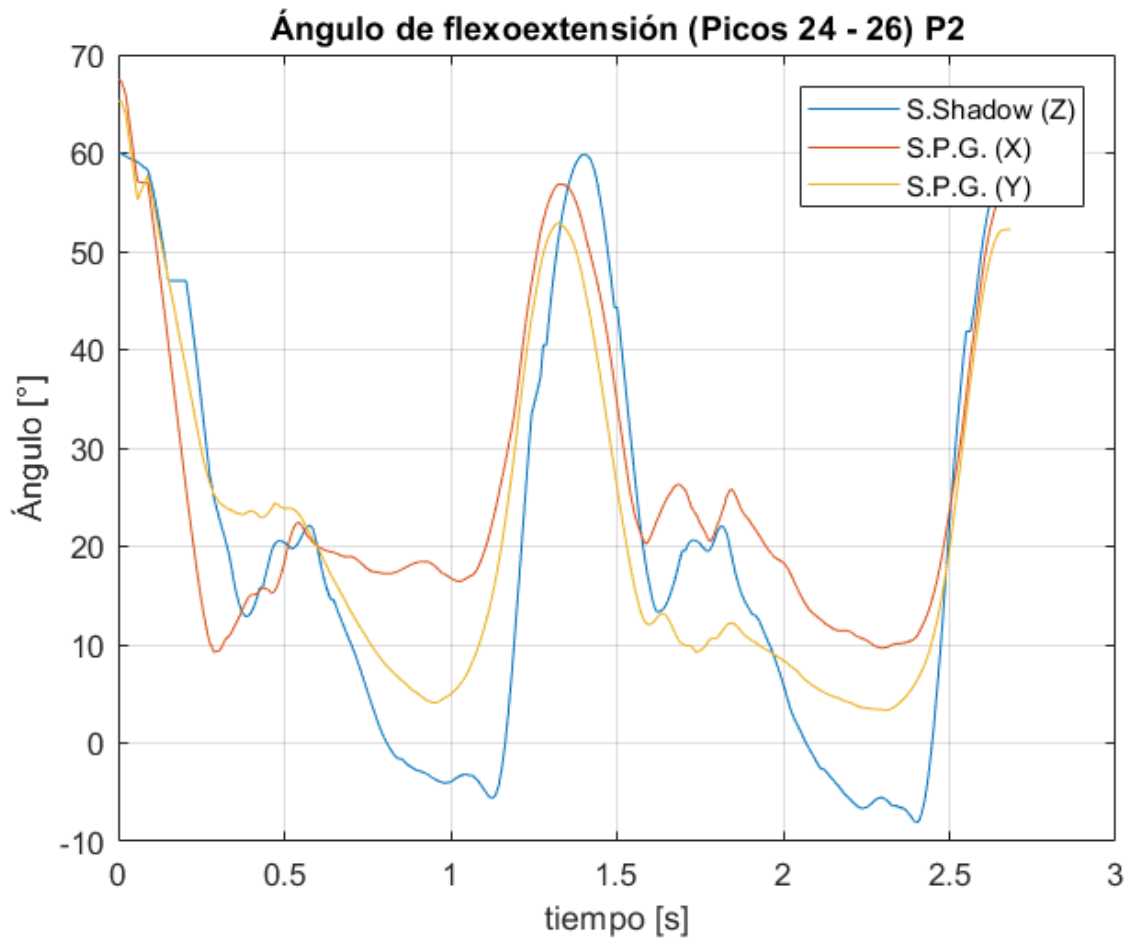


Figura 6.18: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 5

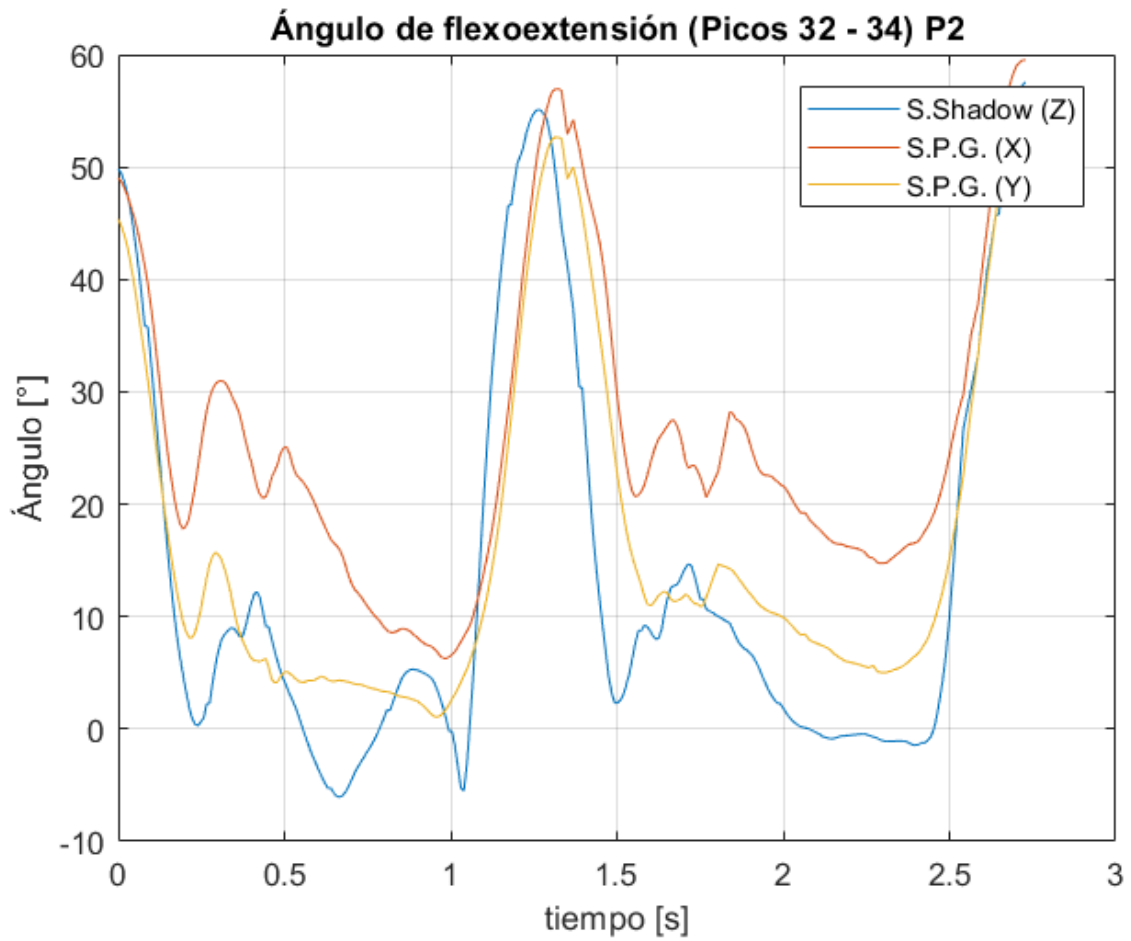


Figura 6.19: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 6

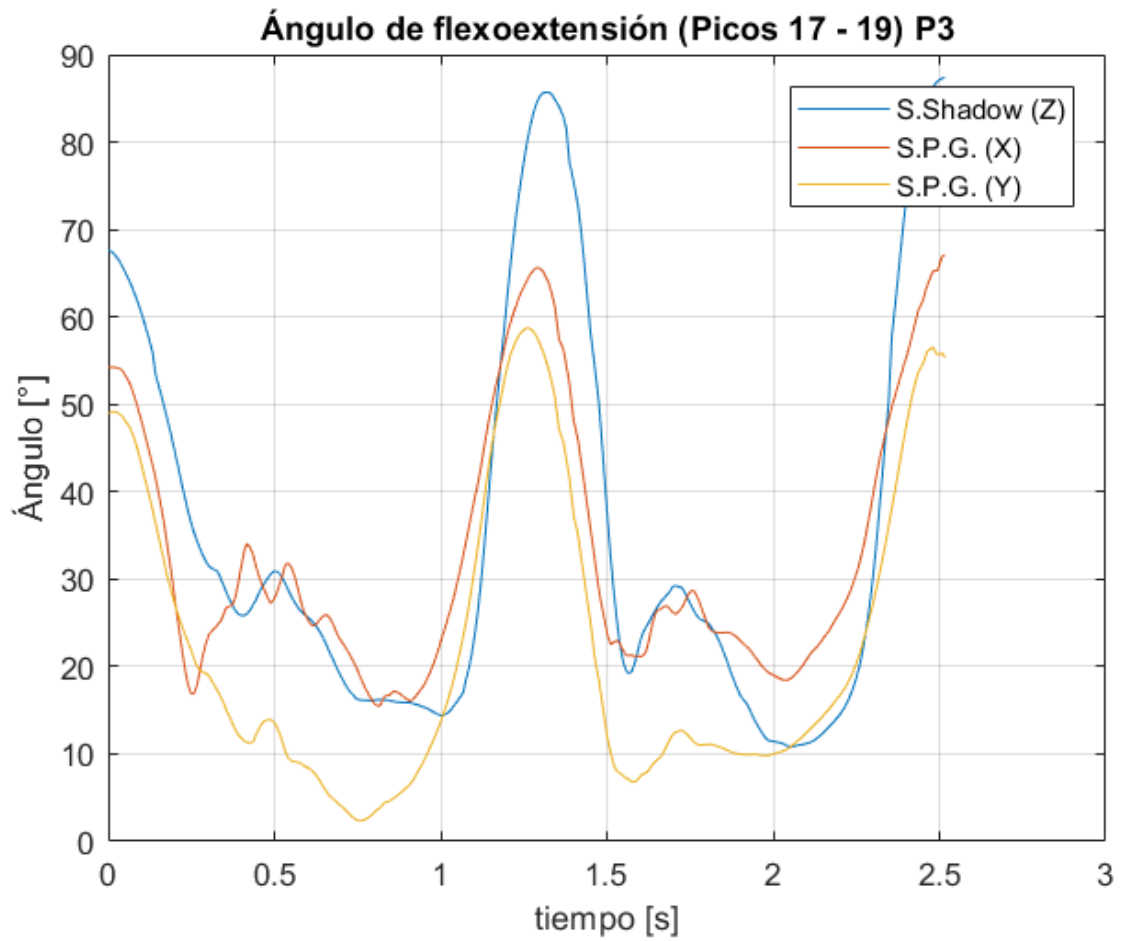


Figura 6.20: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 7

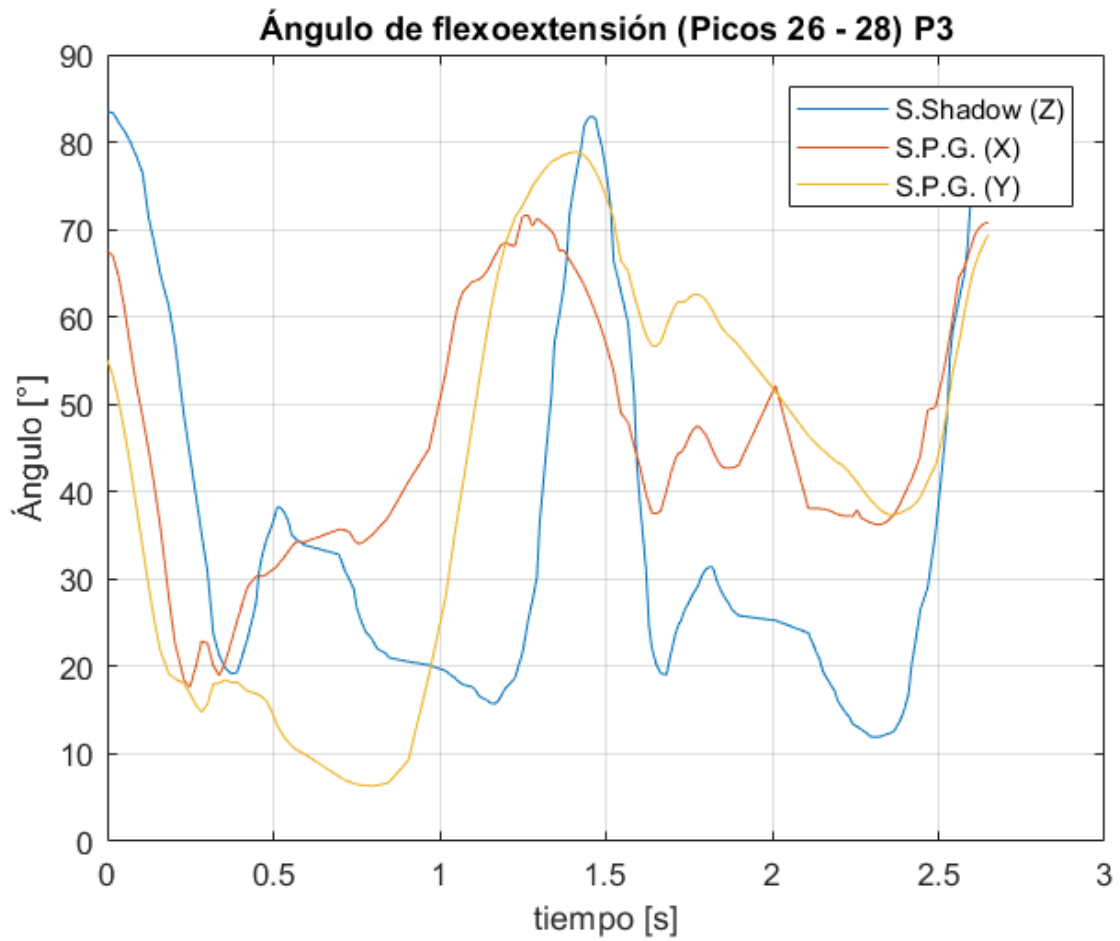


Figura 6.21: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 8

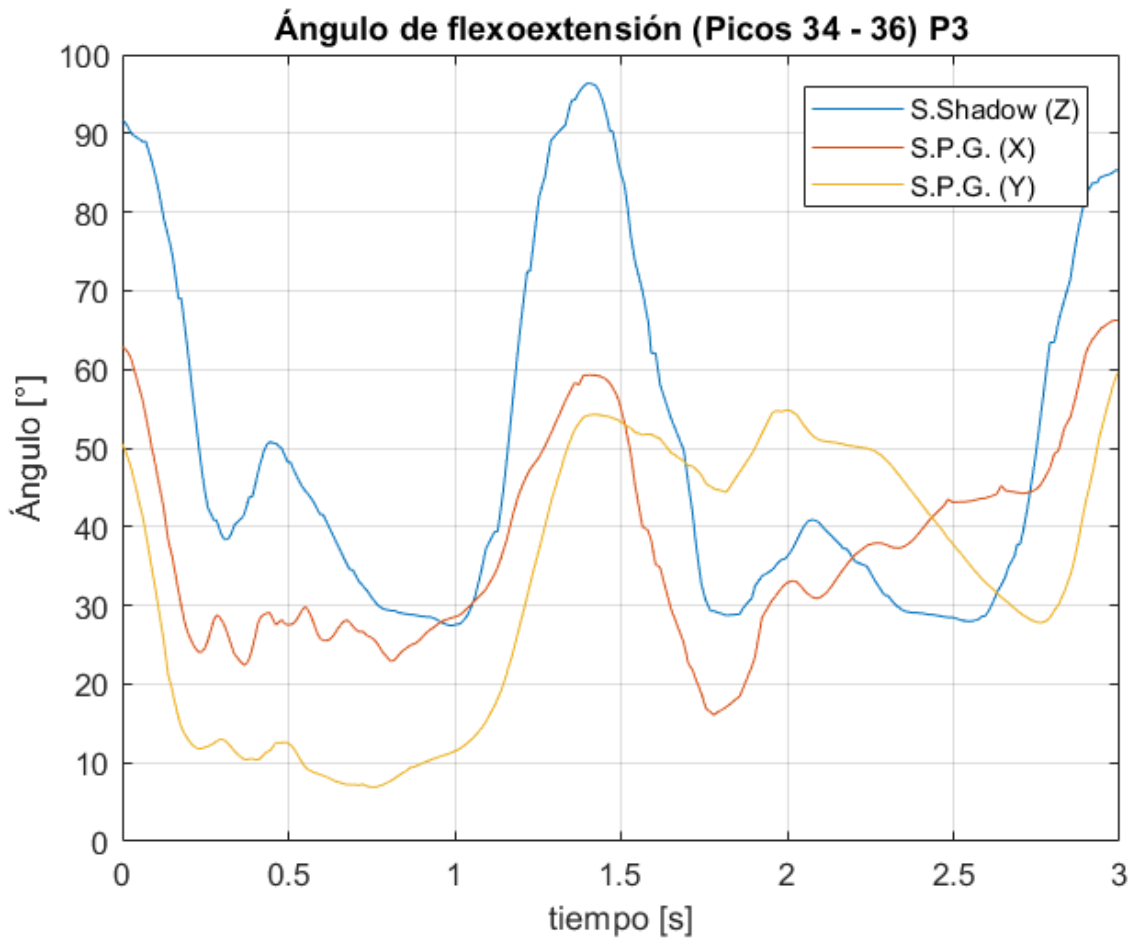


Figura 6.22: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 9

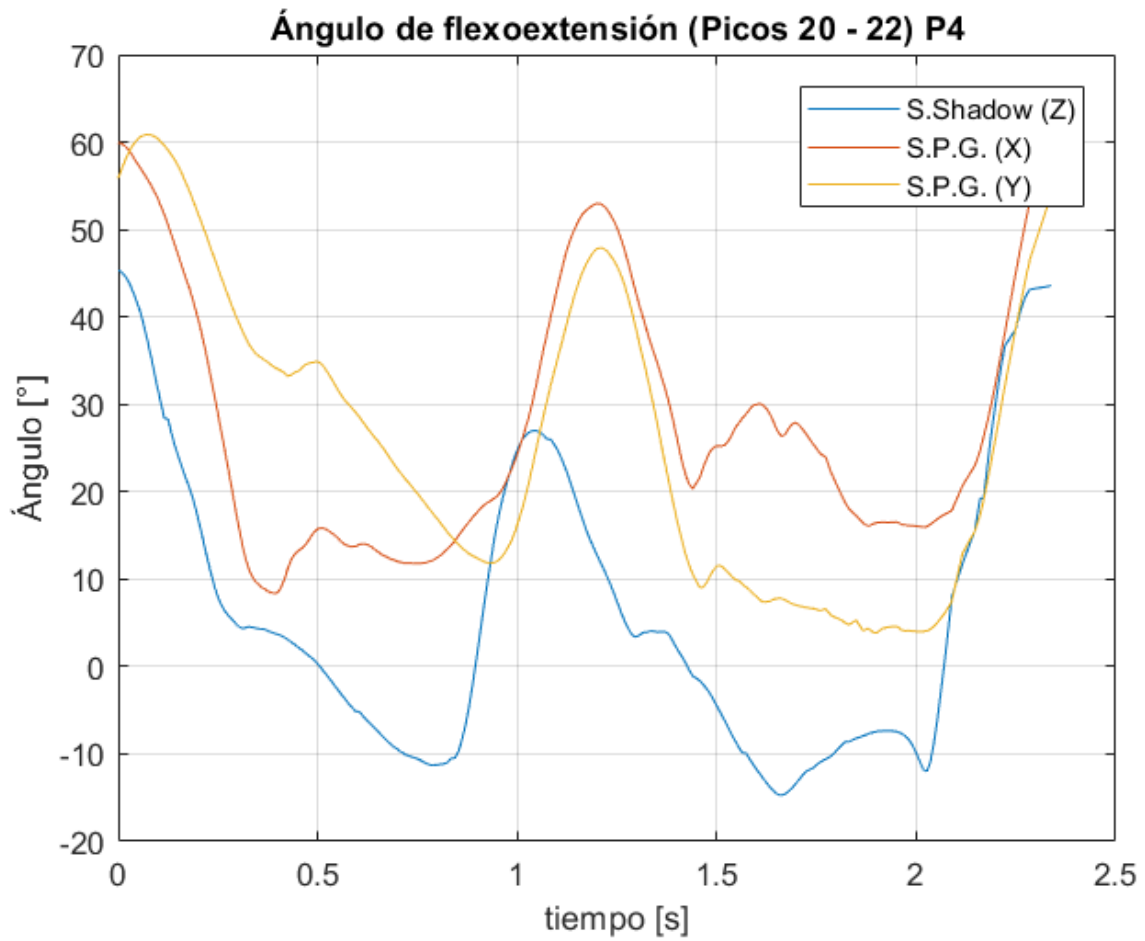


Figura 6.23: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 10

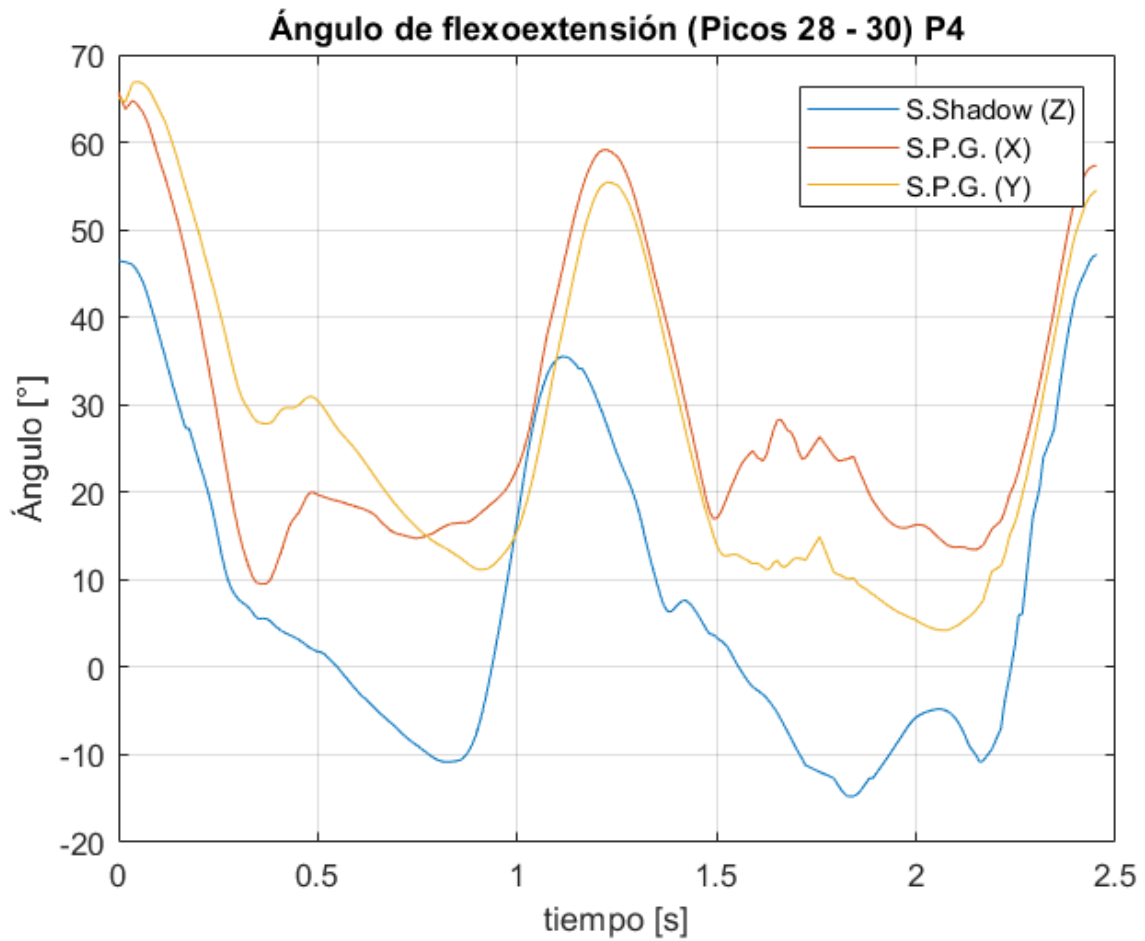


Figura 6.24: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 11

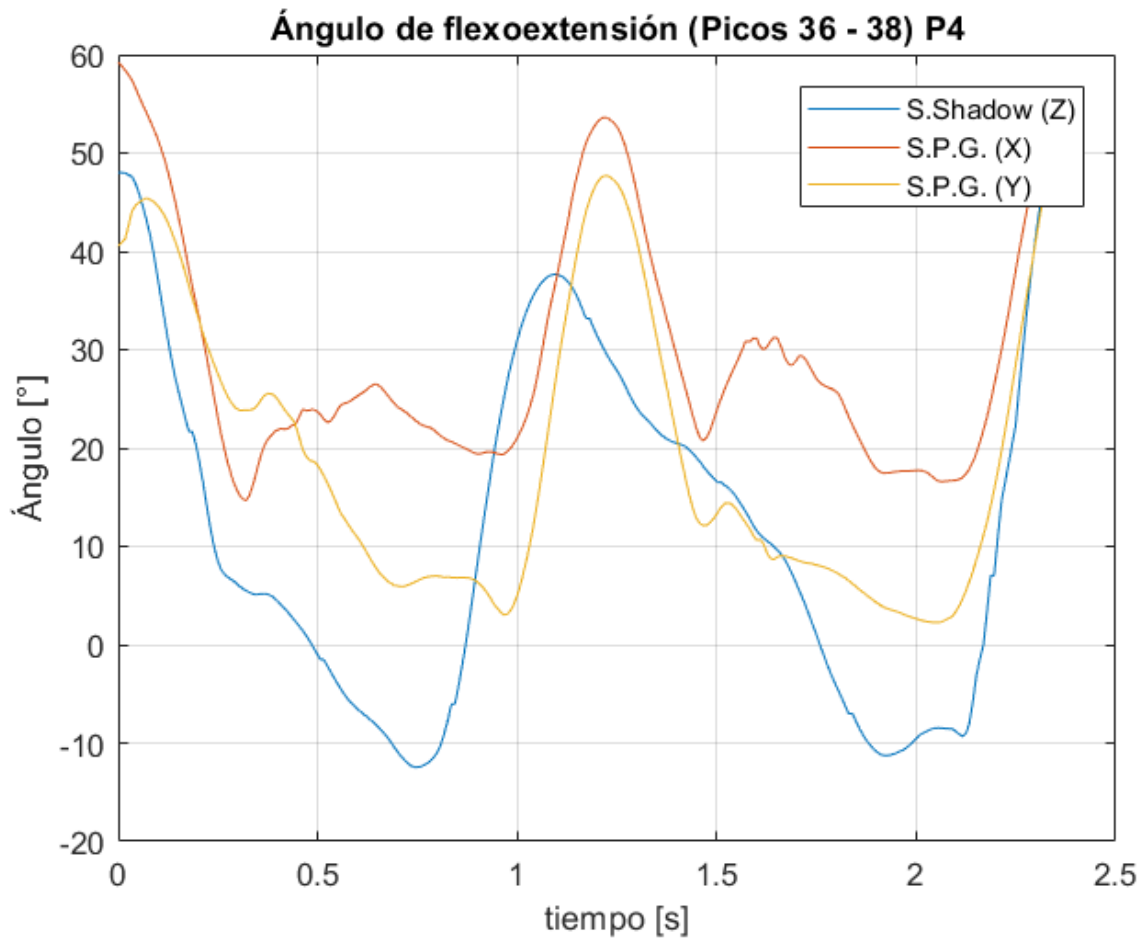


Figura 6.25: Ángulo de flexo-extensión de rodilla procesado de las señales de ambos sistemas - Análisis 12

Capítulo 7

Análisis de Resultados

Para las muestras seleccionadas en la sección de Resultados (Tablas 6.1 y 6.2), se calcularon tres valores necesarios para su análisis y una adecuada comparación cuantitativa de las señales obtenidas por ambos sistemas, tomando en consideración que ambas señales contienen el mismo número N de elementos. Estos valores son: la **media** (μ), la **desviación estándar** (σ) y el **coeficiente de correlación cruzada** ($\rho(A, B)$), **dadas dos poblaciones de variable aleatoria A y B** . Se hace uso de una herramienta informática que calcula mediante una función, a σ , a μ y a $\rho(A, B)$ para cada muestra seleccionada.

Media

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (7.0)$$

En la Tabla 7.1 se muestran los resultados del cálculo de la media para cada muestra. Este parámetro resulta de utilidad para establecer un nivel de similitud entre dos señales en cuestión a su magnitud. Por tal motivo, en conjunto con el valor promedio de cada población de datos, se calcula además el valor del **porcentaje de error experimental relativo** para cuantificar la desviación de las mediciones realizadas con ambos sistemas. Se consideran a los datos del Sistema Shadow como los valores «esperados» (al tratarse de un sistema comercial, previamente validado), mientras que a los datos del SGP como los valores «experimentales». En la eq. 7 se muestra el cálculo realizado para el porcentaje de error experimental.

$$\%Error = \frac{|v_{experimental} - v_{esperado}|}{v_{esperado}} \quad (7.0)$$

Análisis	Sistema Shadow [ř]	SGP (Ejes X) [ř]	SGP (Ejes Y) [ř]	Error (Ejes X) [%]	Error (Ejes Y) [%]
1	17.3481	23.8663	25.9498	37.57	49.58
2	17.6476	19.9464	20.8743	13.03	18.28
3	16.6474	23.6688	22.6774	42.18	36.22
4	23.4883	30.1324	26.4927	28.29	12.79
5	17.7401	24.8512	19.8969	40.08	12.16
6	13.1251	25.5449	16.1801	94.63	23.28
7	35.4005	33.0359	22.2719	6.68	37.09
8	36.3944	46.9341	45.6417	28.96	25.41
9	37.3028	37.1690	31.3648	0.36	15.92
10	5.6981	27.4928	26.4137	382.49	363.55
11	7.9967	28.6121	26.9225	257.8	236.67
12	11.7195	29.6412	19.3603	152.92	65.20

Tabla 7.1: Cálculo de la media y porcentajes de error relativo experimental de los valores de ambos sistemas

Al analizar los porcentajes de error calculados, se observa que la mayoría de los valores son muy altos. Si se calcula un promedio de los porcentajes de error se obtienen, para los ejes analizados del SGP, los siguientes valores:

- Ejes X: 90.42 %
- Ejes Y: 74.68 %

Por un lado, existen lecturas en las cuales su semejanza es alta (es decir, tienen un porcentaje de error bajo), como los análisis 7 y 9 de los Ejes Y; por otro lado, existen deferencias muy grandes en otras lecturas (es decir, porcentajes de error muy altos) como los análisis 10, 11 y 12 para ambos ejes. Sin embargo, los promedios calculados reflejan una clara diferencia en cuestión a la magnitud de las señales entre ambos sistemas. Para las 12 muestras de análisis seleccionadas, el valor medio del Sistema Shadow es menor que para el SGP. Adicionalmente, se observa que la similitud de las magnitudes de las señales, es mayor para la señal de los ejes Y que para los ejes X.

Desviación estándar

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2} \quad (7.0)$$

En la Tabla 7.2 se muestran los resultados del cálculo de la desviación estándar para cada muestra. Ésta cuantifica la dispersión del valor de los elementos de cierta población de datos con respecto a su media (independiente al valor de ésta); lo cual significa que al comparar la desviación estándar de los datos obtenidos por ambos sistemas, se puede cuantificar la «volatilidad» de la señal en torno a su valor promedio. Mientras mayor sea el valor de σ , existe una mayor cantidad de elementos alejados del valor medio. Gráficamente, las crestas y los valles de las señales tienen una magnitud más grande. Al igual que para la media, se calcula el porcentaje de error relativo experimental para cuantificar el nivel de coincidencia de las desviaciones estándar calculadas.

Análisis	Sistema Shadow [ř]	SGP (Ejes X) [ř]	SGP (Ejes Y) [ř]	Error (Ejes X) [%]	Error (Ejes Y) [%]
1	20.0175	14.0760	14.0994	29.68	29.56
2	17.8743	13.1531	13.2124	26.41	26.08
3	18.2286	13.8126	13.9301	24.23	23.58
4	22.9193	15.8336	17.7396	30.92	22.60
5	20.5428	14.0335	15.8312	31.69	22.94
6	17.8733	13.5203	15.0675	24.35	15.70
7	22.6098	14.5116	16.8677	35.82	25.40
8	22.2038	14.4648	22.3171	34.85	0.51
9	21.4065	11.9237	17.4778	44.30	18.35
10	16.1878	14.5566	17.1026	10.08	5.65
11	17.3069	15.7523	17.6675	8.98	2.08
12	17.9829	11.9944	14.7503	33.30	17.98

Tabla 7.2: Cálculo de la desviación estándar y porcentajes de error relativo experimental de los valores de ambos sistemas

Como se observa de la Tabla 7.2, la consistencia de los valores de la desviación estándar entre ambos sistemas es mayor que en el caso de la media. Lo cual significa que las señales (tanto del Sistema Shadow, como para el SGP), a pesar de tener un valor distinto de la media para la misma muestra, la dinámica de las magnitudes es la misma. Es decir, que las señales se comportan en el tiempo de manera similar entre sí, independientemente de la media.

Al existir una mayor consistencia en los datos, el promedio de los valores de la desviación estándar es:

- Para el Sistema Shadow: 19.5961 [°]
- Para el SGP (Ejes X): 13.9694 [°]
- Para el SGP (Ejes Y): 16.3386 [°]

Si se calcula un promedio de los porcentajes de error se obtienen, para los ejes analizados del SGP, los siguientes valores:

- Ejes X: 27.88 %
- Ejes Y: 17.54 %

De estos valores, se demuestra la similitud de la dispersión de datos anteriormente mencionada, siendo para las señales de los ejes Y mayor que para los de los ejes X. Por otro lado, al analizar los valores promedio de las desviaciones estándar, es notorio que para el Sistema Shadow la dispersión de sus datos con respecto a su valor medio es mayor que para las dos señales del SGP. Esto implica que los valores de los ángulos de flexo-extensión, medidos por el SGP tiene una mayor cantidad de valores alejados de su valor medio en comparación con lo medido por el Sistema Shadow; las variaciones de las señales son mayores para el SGP.

Coeficiente de correlación cruzada

$$\rho(A, B) = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{A_i - \mu_A}{\sigma_A} \right) \left(\frac{B_i - \mu_B}{\sigma_B} \right) = \frac{cov(A, B)}{\sigma_A \sigma_B} \quad (7.0)$$

El coeficiente de correlación (también es conocido como el coeficiente de correlación de Pearson) de dos variables aleatorias es una medida de su dependencia lineal. [38] Es decir, cuantifica la similitud entre dos señales con base en la dirección de su trayectoria completa. El coeficiente se define como:

Sean dos variables aleatorias A y B con N observaciones, si la variable A cambia su valor de manera proporcional al cambio de la variable B , se supone la existencia de una dependencia lineal directa entre ambas variables, en cuyo caso el máximo valor del coeficiente de correlación es 1. En caso contrario, si el cambio de la variable A resulta en un cambio inverso en la variable B , se supone una dependencia lineal inversamente proporcional entre ambas variables, en cuyo caso el valor máximo del coeficiente de correlación es -1. Mientras más cercano se encuentre el valor del coeficiente de correlación a cero (desde 0^+ o desde 0^-), significa una menor dependencia lineal y por ende menor relación entre las variables.

Se calcula el coeficiente de correlación de las señales de los ejes «X» como los ejes «Y» para el SGP, con respecto de las señales del Sistema Shadow. Los coeficientes de correlación obtenidos en cada muestra de análisis se muestran en la Tabla 7.3:

Análisis	Sistema Shadow - SGP (Ejes X) []	Sistema Shadow - SGP (Ejes Y) []
1	0.8952	0.8960
2	0.8854	0.8956
3	0.8135	0.8669
4	0.8784	0.8878
5	0.8591	0.9104
6	0.8350	0.8979
7	0.9143	0.9029
8	0.4594	0.3872
9	0.7344	0.3357
10	0.7212	0.7034
11	0.8619	0.8691
12	0.7994	0.7760

Tabla 7.3: Cálculo del coeficiente de correlación de las señales del SGP con respecto a la señal del Sistema Shadow

Se observa de la Tabla 7.3 que los valores de correlación para ambas señales del SGP alcanzan valores muy cercanos a 1, en la mayoría de las muestras de análisis. Aún a pesar de la existencia de valores bajos de este coeficiente (como, por ejemplo, para los análisis 8 y 9 de los ejes Y), si se calcularan los valores promedio para ambos ejes, resulta:

- Para el SGP (Ejes X): 0.8047 []

- Para el SGP (Ejes Y): 0.7774 []

Esto implica que las señales del SGP (al tener en promedio un coeficiente de correlación muy cercano a 1 para ambos conjuntos de ejes) resultan ser muy parecidas a la señal del Sistema Shadow en cuestión a su trayectoria, independientemente al valor de su media y a la magnitud de sus crestas y valles.

Diferencias entre las señales de los Ejes X y Y del SGP

Tras observar las gráficas del ángulo de flexo-extensión de rodilla (Fig. 6.14 - 6.25), procesadas de los datos recabados por el SGP, es posible detectar cierta diferencia entre los ángulos relativos obtenidos mediante el análisis de los ejes X y los ejes Y. En teoría, la diferencia entre ambos ejes debería ser cercana a cero y mantener un valor constante u oscilar dentro de un rango pequeño ya que se asume de la calibración inicial que los desplazamientos angulares resultarían ser al rededor de ejes paralelos (en este caso al rededor de los ejes Z); por lo que los desplazamientos entre los ejes X y los ejes Y serían idealmente coplanares (específicamente en el plano «XY»).

Sin embargo, como se muestra en las Fig. 7.1 - 7.12, las diferencias entre ambos procesamientos alcanzan valores relativamente altos con respecto a lo esperado.

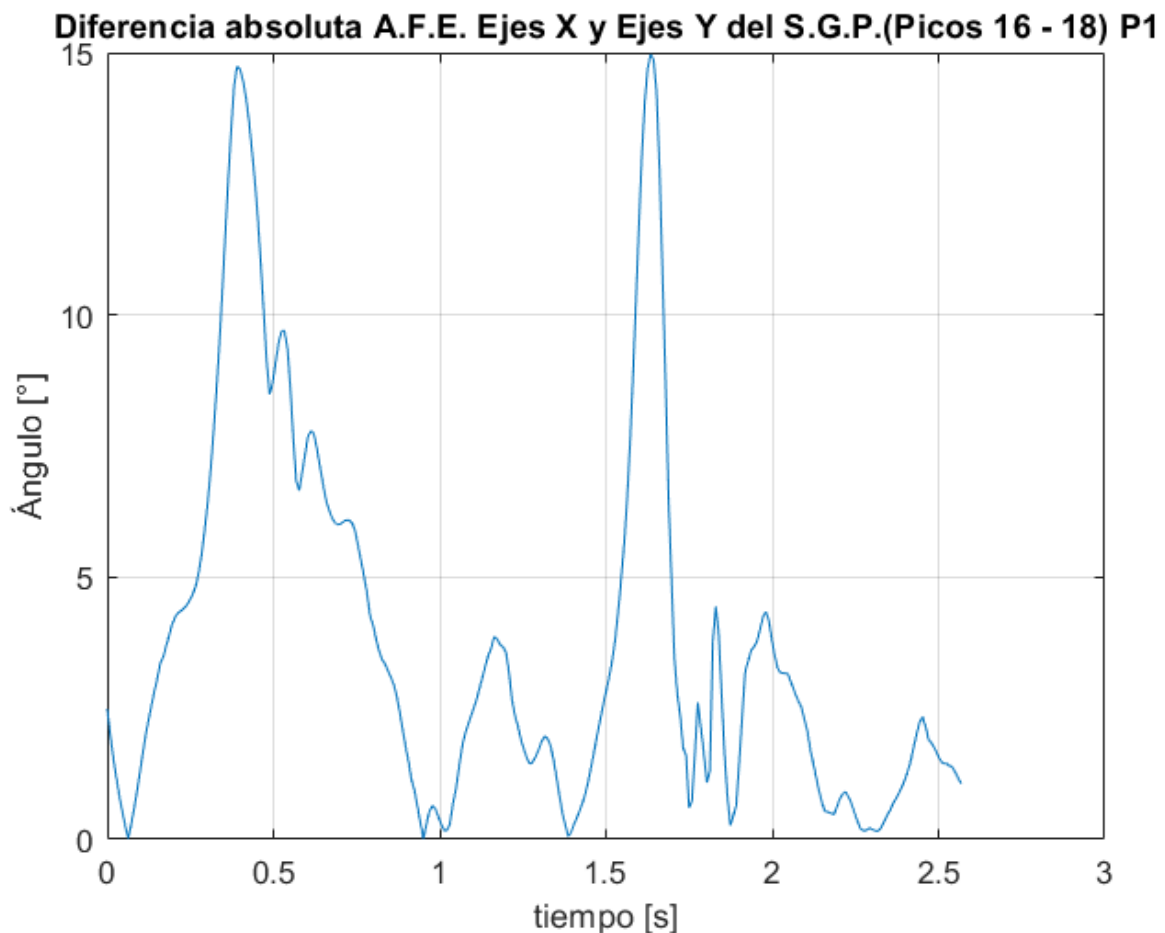


Figura 7.1: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 1

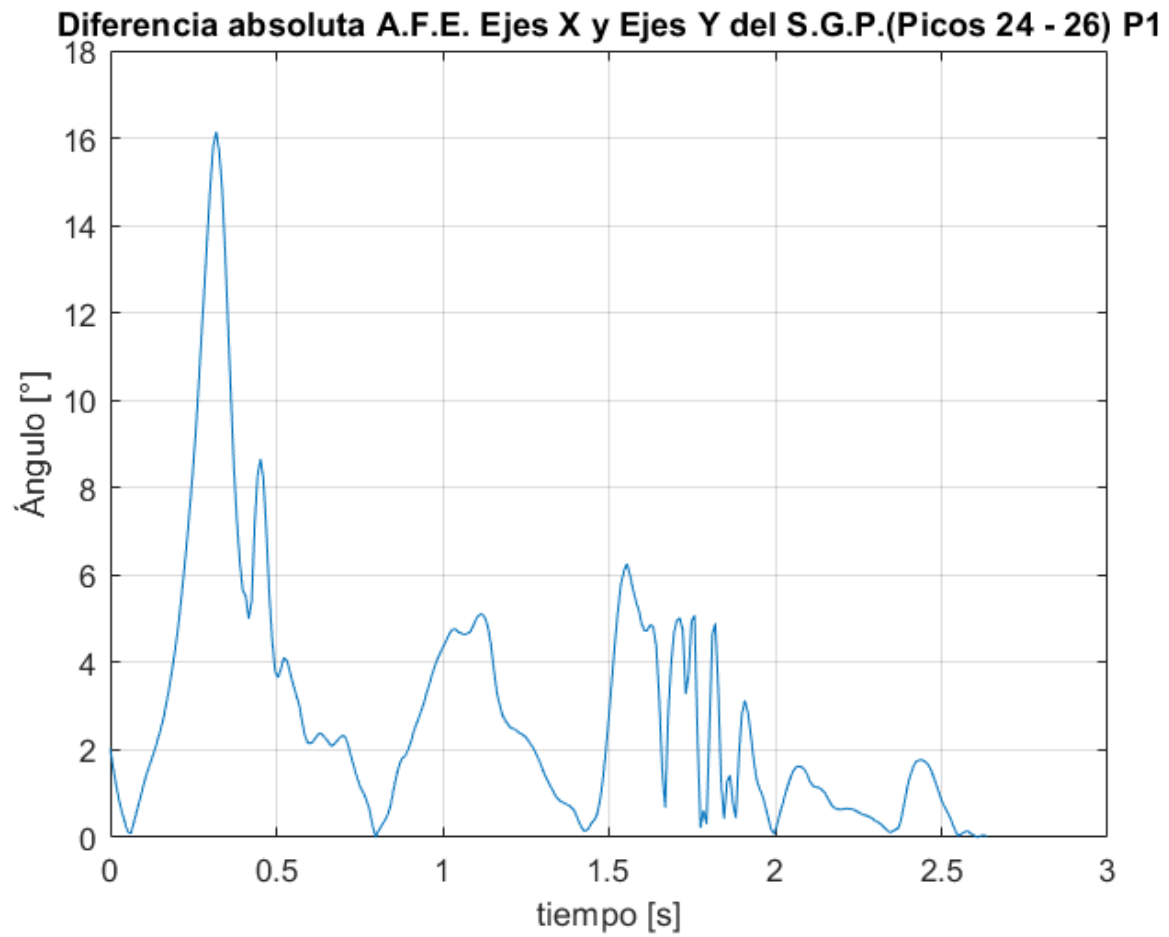


Figura 7.2: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 2

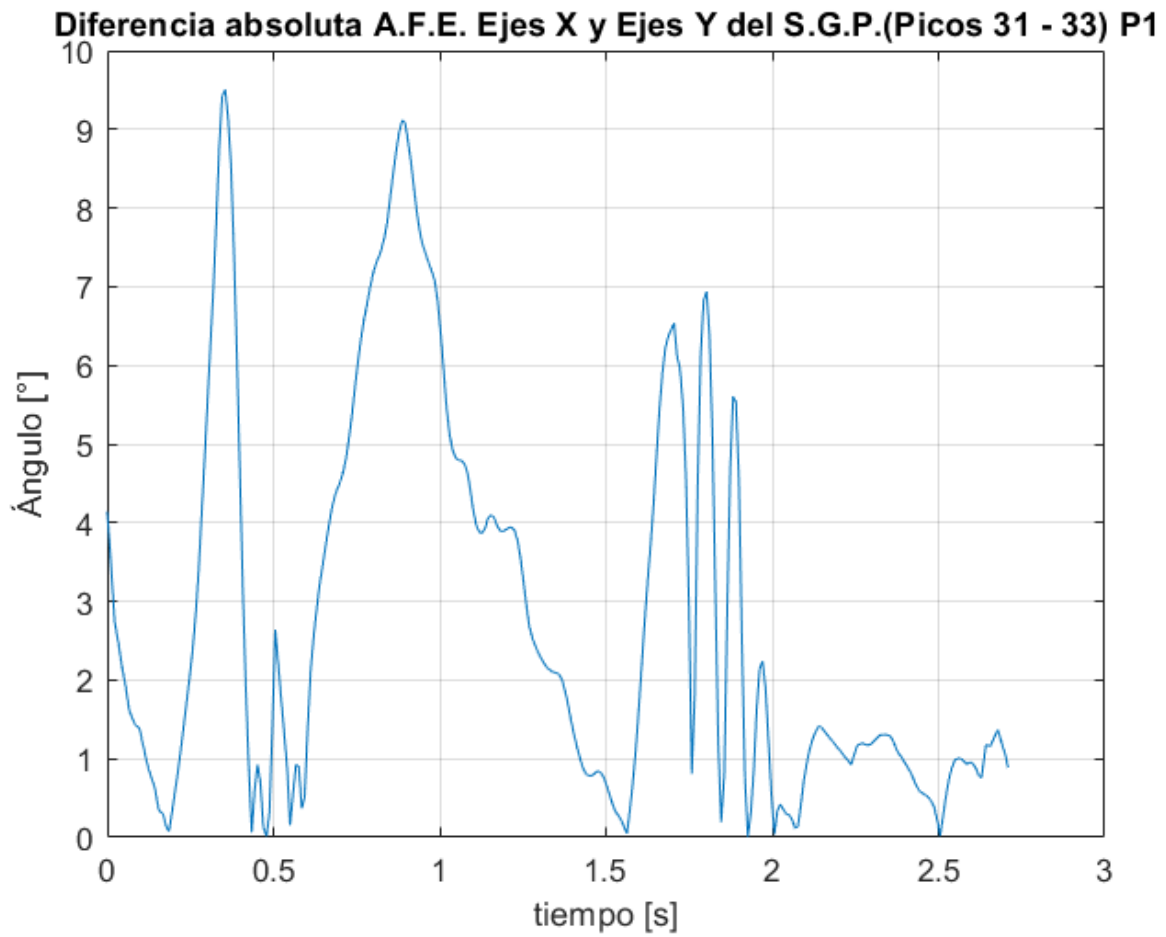


Figura 7.3: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 3

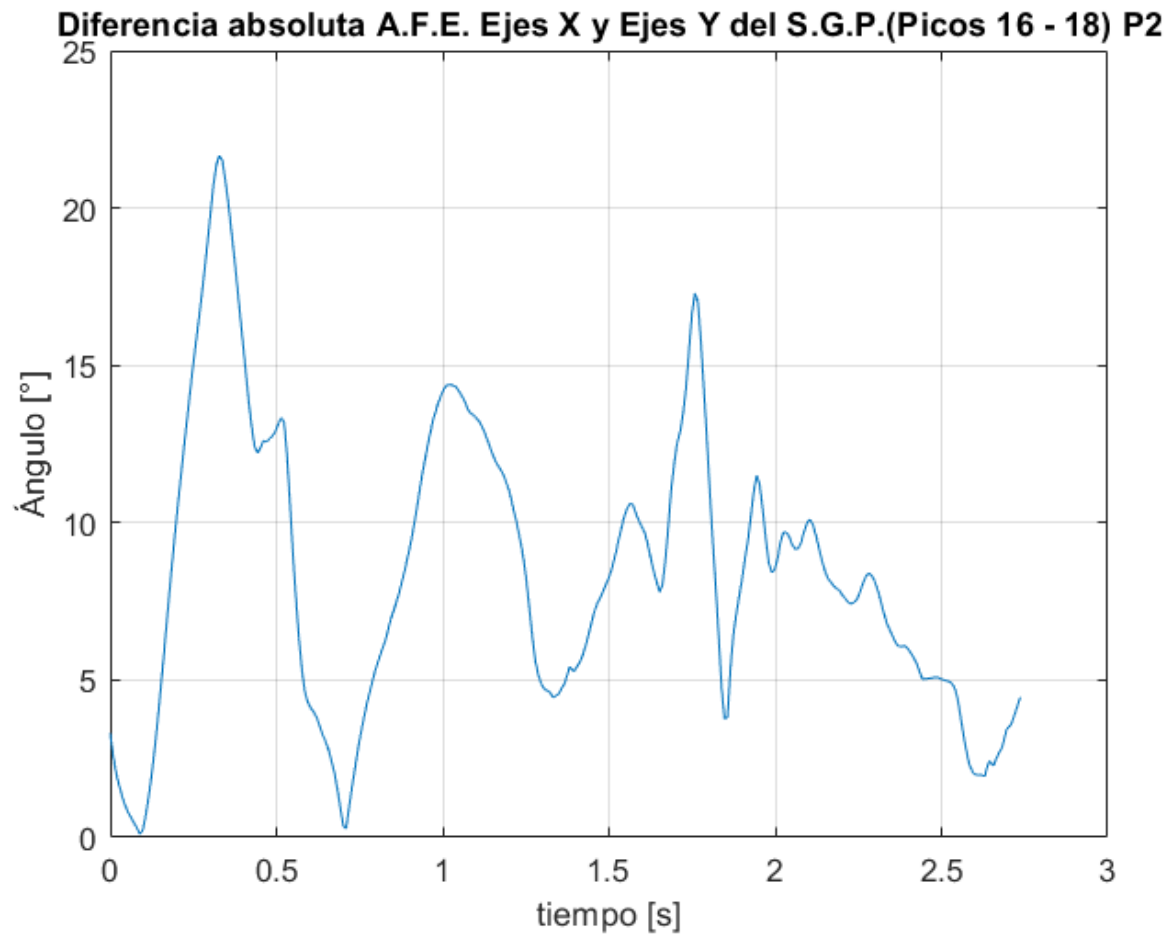


Figura 7.4: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 4

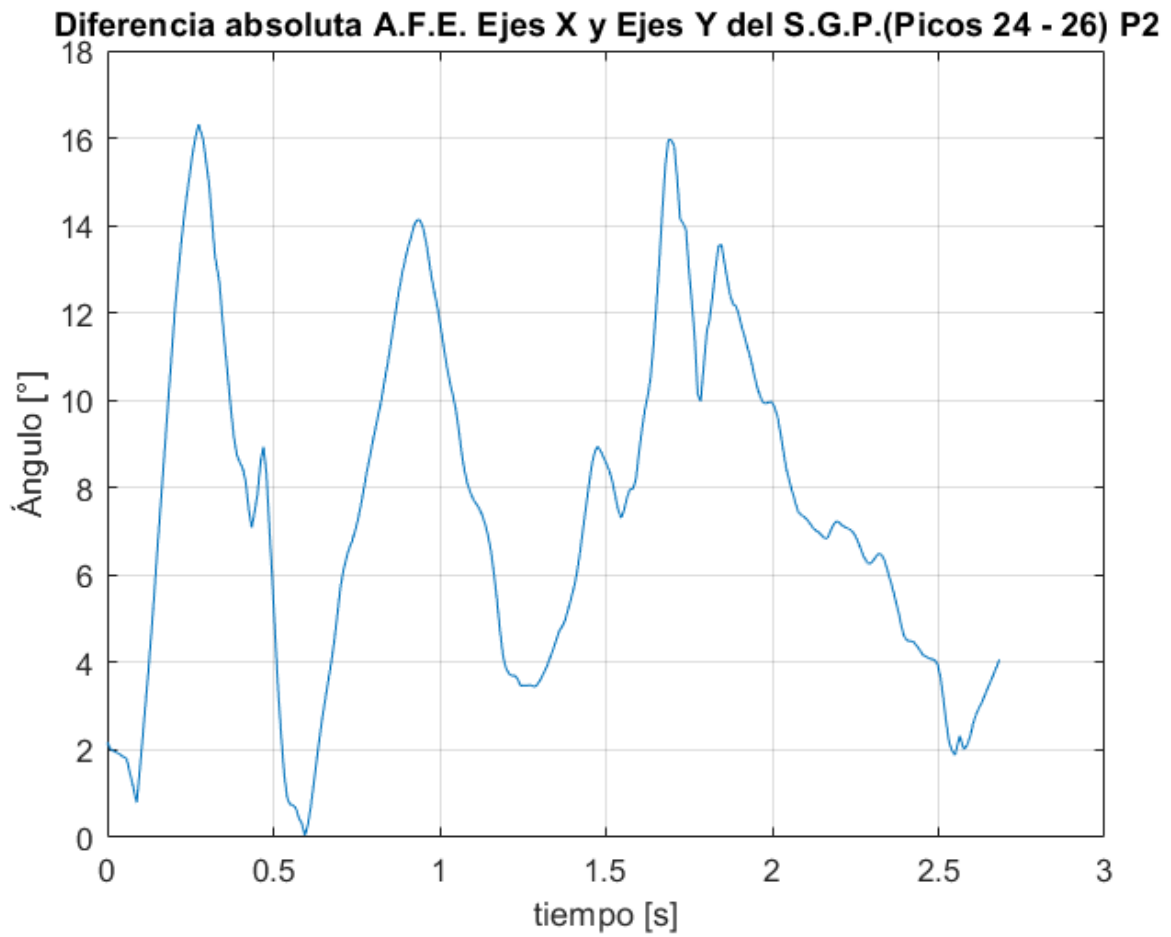


Figura 7.5: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 5

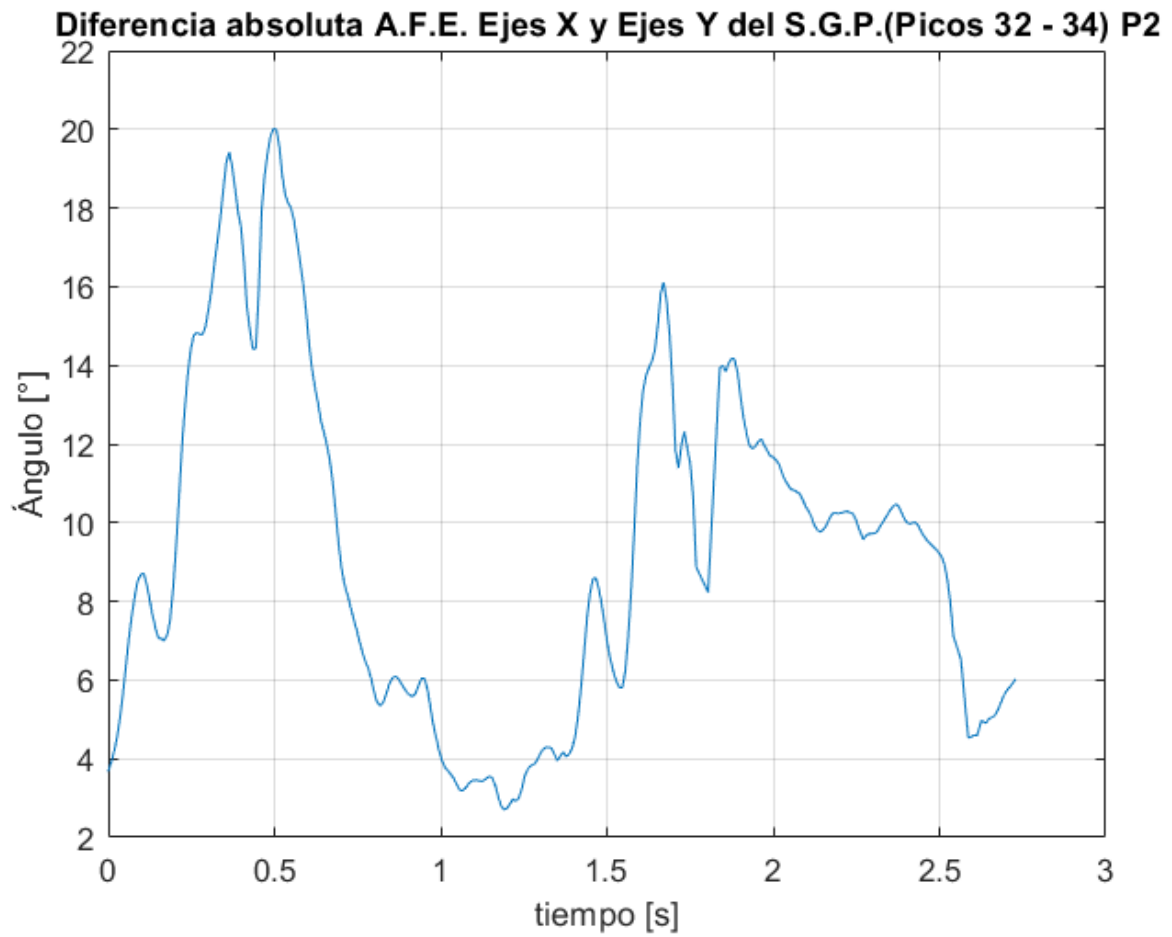


Figura 7.6: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 6

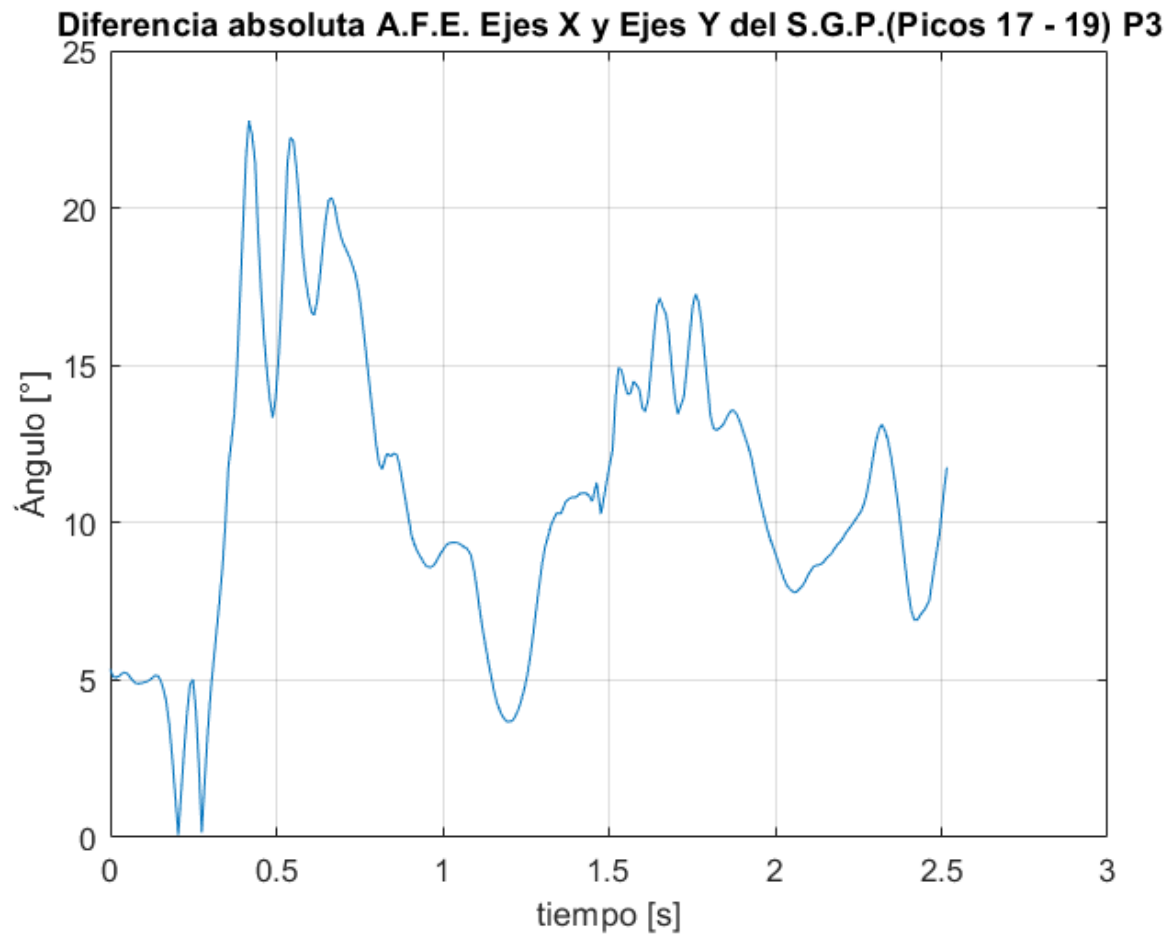


Figura 7.7: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 7

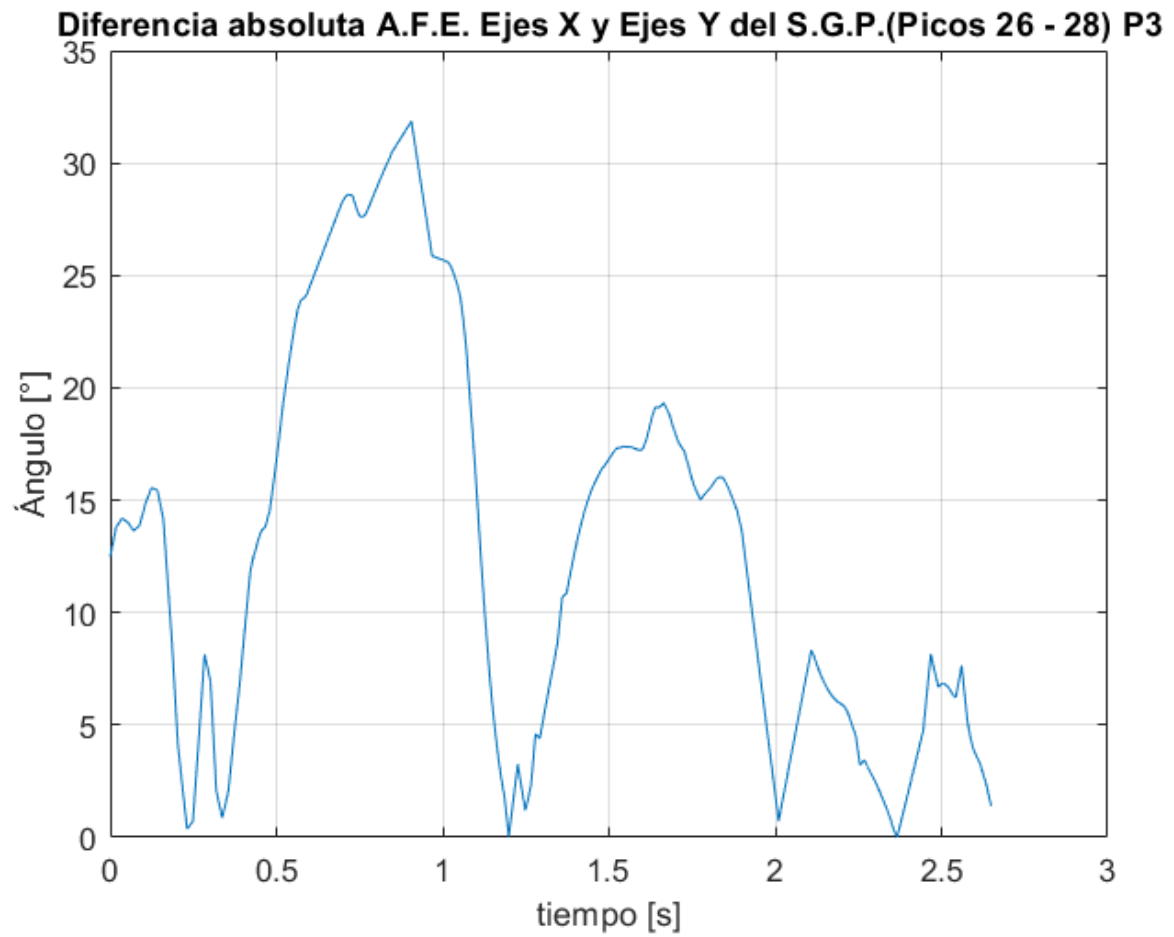


Figura 7.8: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 8

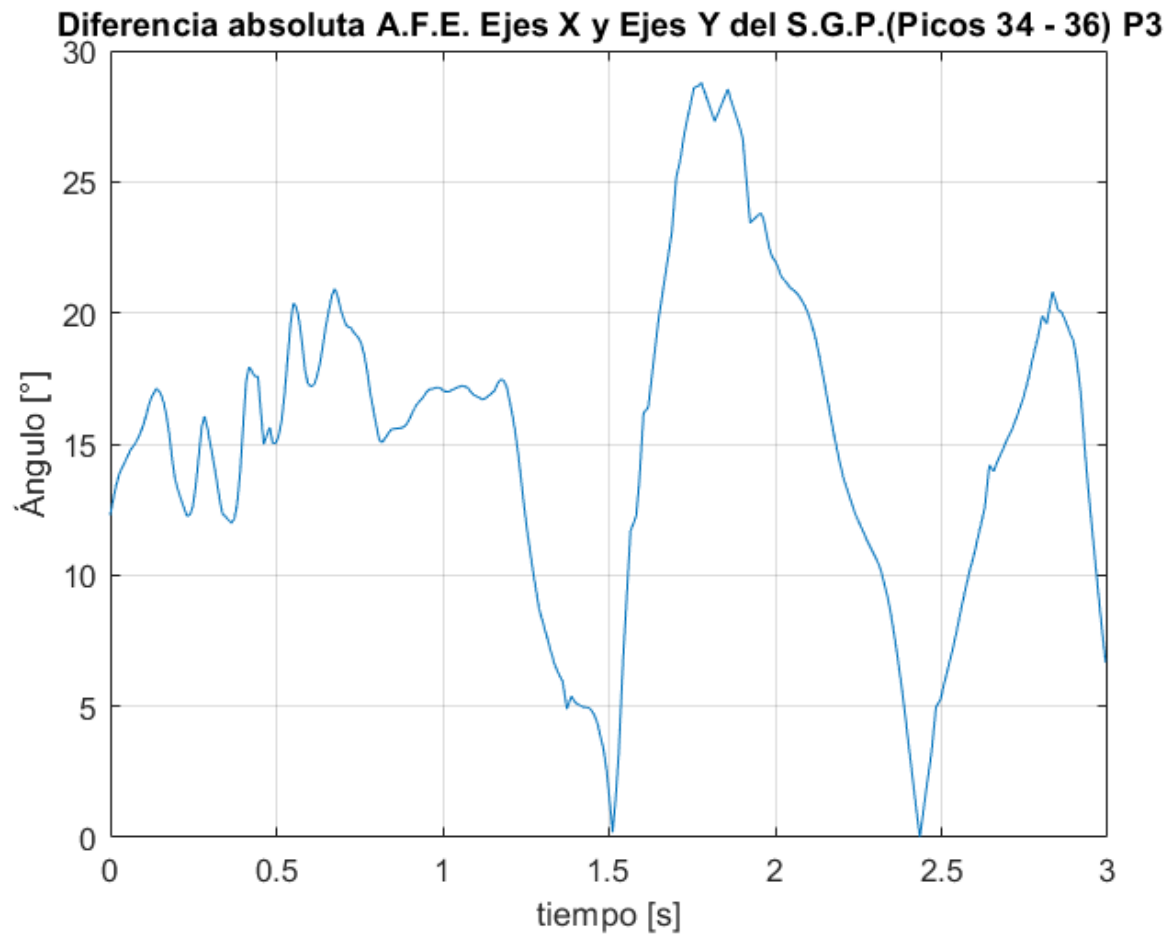


Figura 7.9: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 9

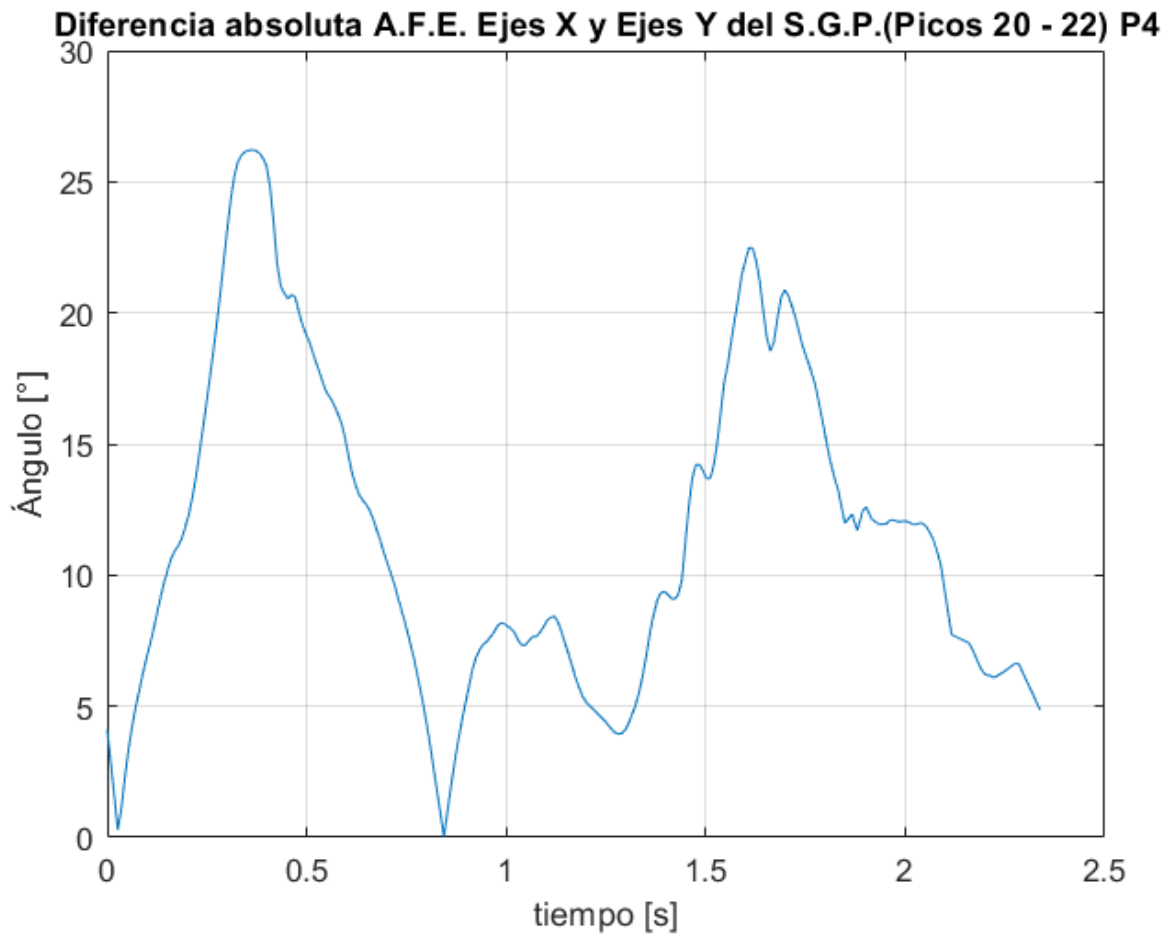


Figura 7.10: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 10

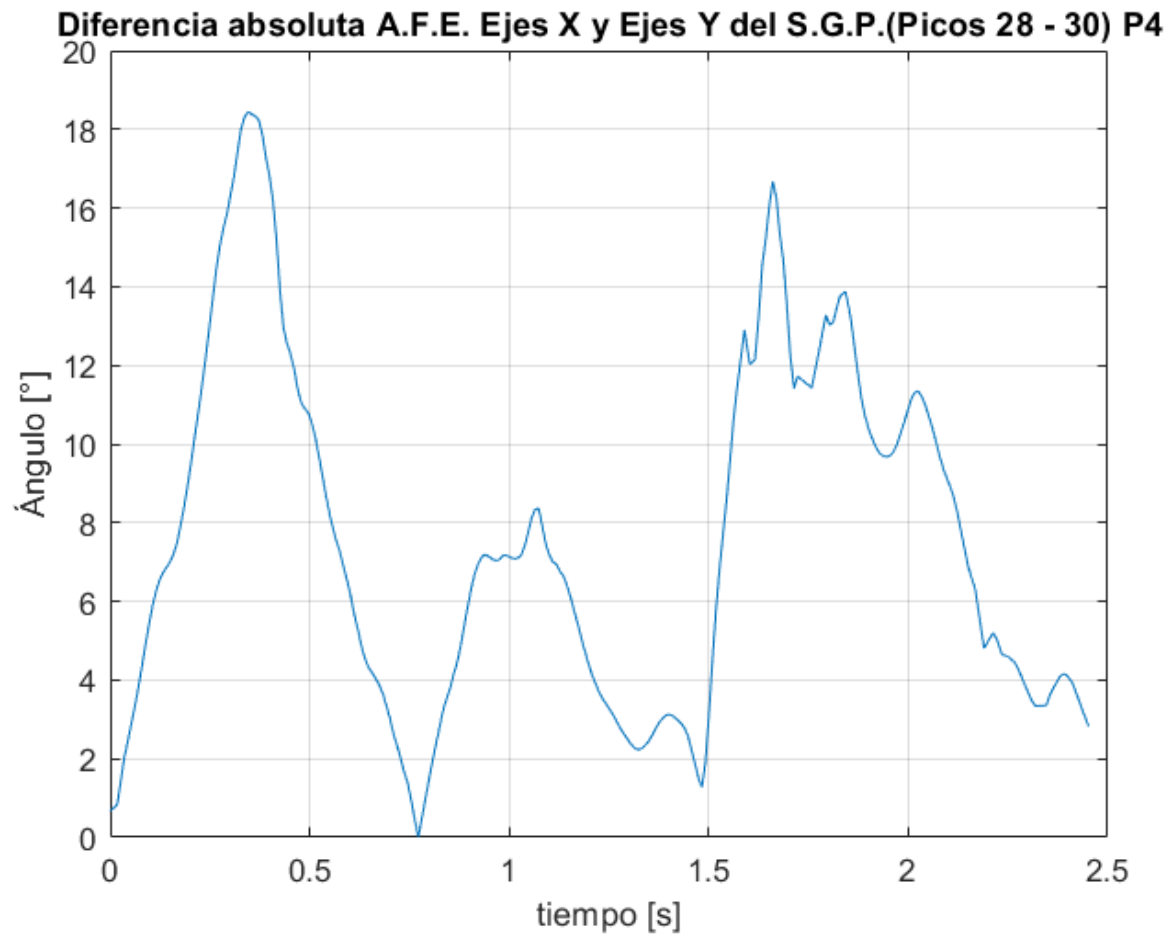


Figura 7.11: Análisis 11

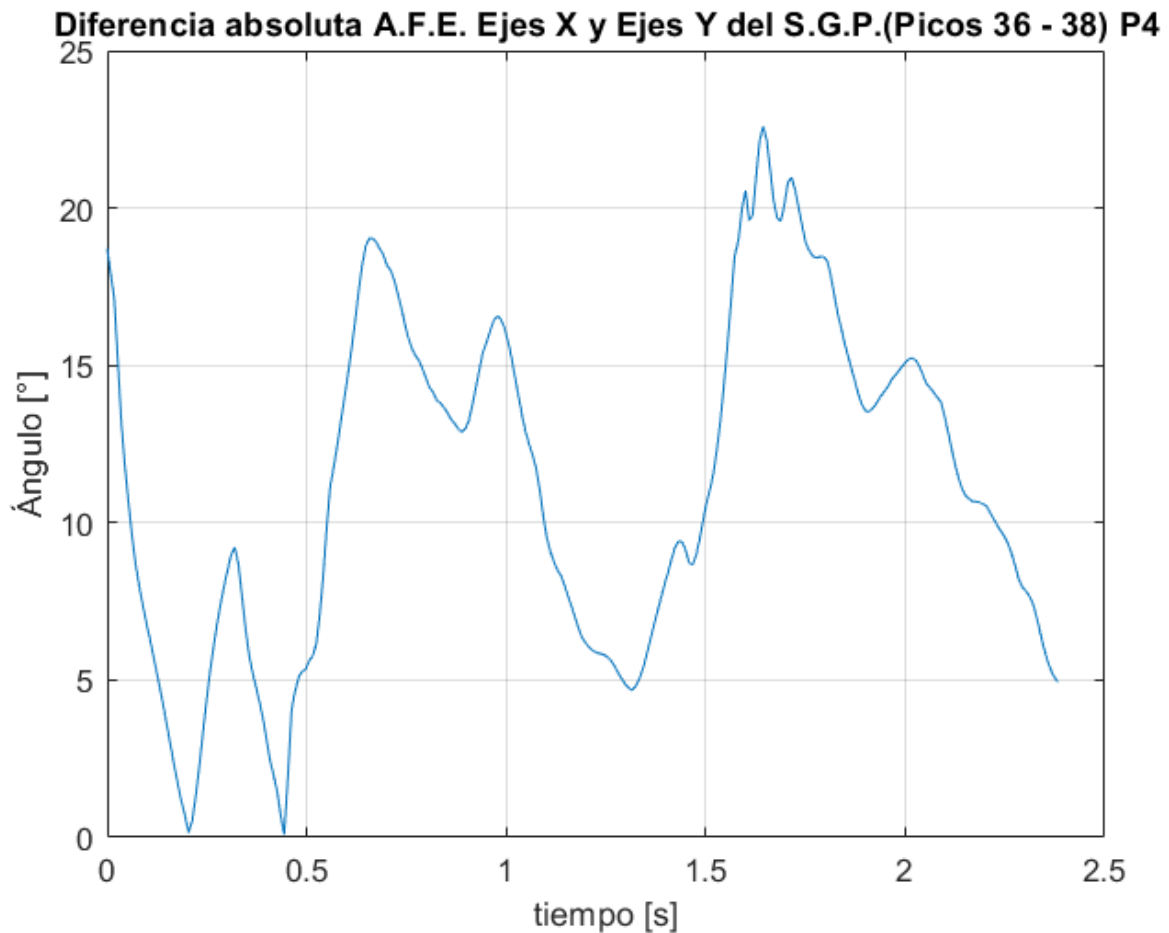


Figura 7.12: Diferencia en el ángulo de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP - Análisis 12

Análisis global

Del cálculo de los valores μ , σ y ρ y de las diferencias entre los ángulos de flexo-extensión de rodilla procesado de las señales de de los ejes X y Y del SGP, es posible deducir que para el SGP existe un plano en el cual no están incluidos ambos ejes de los sistemas de referencia y en cuyo caso, las proyecciones de estos ejes en dicho plano. Por lo tanto, analizando la dinámica de los desplazamientos angulares resultaría en una señal de mayor parentesco con la señal del Sistema Shadow. Es por tal motivo que los ángulos procesados por el SGP son mayores que aquellos procesados por el Sistema Shadow.

Por otro lado, en el caso del ángulo de flexo-extensión procesado mediante la señal del Sistema Shadow (Fig. 6.10 - 6.13), resultan en gráficas mucho más suaves que para el caso del SGP; dado que es un sistema finamente calibrado y caracterizado para su incursión al mercado. Para el caso del sistema propuesto, los datos pasan únicamente por la etapa de filtrado de Madgwick, por lo cual, resulta probable que los valores de los parámetros β y γ no sean los más adecuados para un filtrado de mayor efectividad.

Finalmente, a pesar de las diferencias existentes entre ambas señales en cuestión a su magnitud, el coeficiente de correlación en ambas señales del SGP con respecto al Sistema Shadow son considerablemente altas.

Capítulo 8

Conclusiones

Tras realizar las pruebas y analizar los resultados obtenidos, se concluye que el SGP es un primer prototipo que cumple parcialmente con las funcionalidades de un sistema de goniometría para análisis de marcha humana constituido por sensores inerciales de tipo MARG. Debido a que, de los resultados, se observa un claro seguimiento de los desplazamientos angulares de la articulación de rodilla, obtenido mediante el SGP y su semejanza con las señales recabadas por el Sistema Shadow, sin embargo, las comparaciones realizadas revelan ajustes necesarios al diseño del sistema para garantizar su fiabilidad y repetibilidad.

Las señales que se obtuvieron del SGP construido fueron validadas realizando pruebas comparativas de carácter cuantitativo contra un sistema comercial probado previamente. Por lo cual, al analizar los datos procesados del Sistema Shadow se puede concluir que un sistema comercial caracterizado resulta de gran utilidad para validar un prototipo y a través de la comparación entre ambos sistemas; mediante dicha comparación, se detectan áreas de oportunidad para la optimización en el diseño y funcionalidad del sistema a validar.

Derivado de la validación, los altos valores de los factores de correlación obtenidos son prueba de la viabilidad de la fabricación de un prototipo que cumpla con el objetivo central de este proyecto. No obstante, en este primer prototipo también se concluye que las tareas de filtrado y de calibración previa de los sensores deben ser aun más rigurosas debido a que en estricto sentido, las mediciones y el post-procesamiento realizados no permiten conocer de manera certera los desplazamientos angulares de flexo-extensión de rodilla. Prueba de ello son los altos porcentajes de error relativo experimental obtenidos de los promedios de las señales (μ) y los valores de la desviación estándar (σ) de cada una de las muestras analizadas. A pesar de la alta correlación de los datos, para fines de diagnóstico o control, las señales necesitan tener una similitud mayor en relación a la magnitud del ángulo de flexo-extensión.

Así, si bien la implementación del SGP en el análisis de la marcha humana no es del todo viable, el sistema podría ser utilizado para el desarrollo de un producto cuyas aplicaciones requieran identificar o cuantificar los ciclos de marcha; por ejemplo, implementándolo en la activación en tiempo real de un sistema de tipo ON-OFF.

Por otro lado, se consiguió una frecuencia de muestreo por encima de la esperada para dos sensores transmisores conectados al receptor central. Sin embargo, si se desea diseñar una red más amplia la capacidad de procesamiento se ve reducida mientras más sean los dispositivos que se conecten. Por consiguiente, en tal caso rediseñar la arquitectura electrónica (reestructurando los circuitos o seleccionando componentes de mayor velocidad de muestreo) de tal manera que los procesadores operen a una mayor velocidad. En el caso del proyecto al que inicialmente se pretende vincular este sistema (léase la justificación del proyecto), cumple de manera ideal con

la velocidad de muestreo necesaria para el análisis previsto.

Aun a pesar de las fallas presentadas relacionadas con el diseño y construcción del prototipo expuestas en la sección de Resultados, la simplicidad de conexión del SGP cumple con el objetivo particular de conseguir que el diseño del sistema permita una conexión sencilla y modular. Para el desarrollo del presente proyecto, la configuración de la cantidad de sensores estaba pre-determinada, mas es posible modificar el programa para que el usuario sea capaz de construir su propia red inalámbrica con un número de sensores configurable.

Capítulo 9

Trabajo Futuro

A continuación se enlista una serie de recomendaciones que en un futuro podrían ayudar a perfeccionar el diseño, operación y desempeño del SGP en su totalidad:

- Los parámetros del Filtro de Madgwick deberán someterse a un procedimiento de caracterización y optimización para lograr una mejor respuesta. A pesar de que es posible aplicar un filtro digital durante el post-procesamiento, si se utiliza el sistema para ser incorporado a uno mayor como entradas de una etapa de control, las señales deberán verse reflejadas gráficamente como curvas suaves.
- Aun cuando las funciones de calibración implantadas resultan ser de gran utilidad y si bien se tomó en cuenta la calibración como un algoritmo independiente del sensado inercial (aprovechando los recursos de memoria que ofrece el microcontrolador) es necesario optimizar el código en conjunto con los parámetros del filtro. Tomando en cuenta que se requirió de una prueba estática para minimizar errores debido al desfase en los ejes de los sensores provocado por la calibración independiente, y considerando que se utiliza un filtro de carácter iterativo, resulta de gran conveniencia en un futuro diseñar e implantar un algoritmo que calcule el «offset» existente entre los sensores, previa a la medición.
- Al ser un prototipo rápido se decidió utilizar tarjeta perforada para fijar e interconectar los componentes necesarios para el proyecto (haciendo referencia únicamente a los transmisores). Sin embargo, se deberá diseñar un circuito en tarjeta impresa con la finalidad de obtener una mejor distribución de los componentes, reduciendo el espacio a la par de minimizar los errores de contacto, efecto capacitivo entre otros. El diseño deberá considerar no sólo prescindir de los módulos, implantando directo los componentes que lo conforman, sino la arquitectura del microcontrolador para reducir en mayor medida el espacio y el consumo de energía.
- Se deberán analizar alternativas de microcontroladores que ofrezcan una mayor velocidad de procesamiento tanto para los transmisores, como el receptor. De esta manera se amplía la gama de transmisores vinculados a un solo receptor sin comprometer la velocidad de muestreo que se desee obtener.
- Es necesario que se diseñe un encapsulamiento de mayor robustez para los transmisores. Es fundamental que los circuitos se encuentren protegidos a toda costa manteniendo a la par la comodidad del usuario al máximo.
- Para el receptor, en este caso se utilizó una tarjeta de desarrollo Arduino, debido a la rapidez del prototipo. No obstante, resulta de mayor conveniencia para el usuario en el computador minimizar el cuidado en el dispositivo. Es necesario que se diseñe un circuito (con su encapsulado correspondiente) que le ofrezca a dicho usuario la oportunidad de

concentrar su atención en el programa en pantalla después de la conexión al computador. De igual manera, resultaría de gran ayuda, que dicho circuito tenga los indicadores (ya sea en pantalla o en el dispositivo mismo) que le permitan al usuario conocer el estado de la conexión con los transmisores.

- Se deberá diseñar y programar una interfaz de usuario en el computador mucho más intuitiva y gráfica; que le permita al usuario en el computador conocer más a detalle las características del sistema, el estado en el que se encuentra la conexión, realizar configuraciones, etc. con la finalidad de facilitar el manejo del sistema.

Capítulo 10

Anexos

10.1. Representación de la orientación de un cuerpo en el espacio mediante matrices de rotación

Las rotaciones que usualmente se implementan, se realizan al rededor de los ejes propios del sistema de referencia. Tomando en cuenta que el producto punto de los vectores unitarios es el coseno del ángulo entre ellos, las rotaciones que representan al sistema {B} visto desde {A}, girando un ángulo θ al rededor de los ejes, son:

$${}^A_B R_Z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10.1)$$

$${}^A_B R_Y(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (10.2)$$

$${}^A_B R_X(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (10.3)$$

$$(10.4)$$

En la Figura 10.1 se muestra de forma gráfica el principio de las ecuaciones anteriores, aplicado a una rotación al rededor del eje \hat{Z}_A para definir la orientación del sistema {B} (Ecuación 10.1).

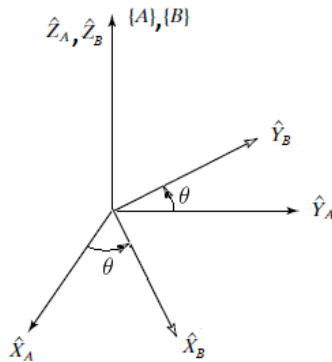


Figura 10.1: Rotación al rededor del eje \hat{Z}_A [20]

Las matrices de rotación, así como pueden definir una sola orientación en el espacio, si se multiplican varias matrices, representan una sucesión de rotaciones dependiendo del orden de la multiplicación.

- Si el sistema B está rotando con respecto al sistema A, entonces se **premultiplica**. Es decir si se tienen dos rotaciones sucesivas del sistema {B} en el eje ${}^A\hat{X}$ seguido por el eje ${}^A\hat{Y}$, con respecto al sistema {A}, se representaría como:

$${}^A R = R_Y \cdot R_X$$

- Si el sistema {B} rota sobre su propio eje, entonces se **postmultiplica**. Tomando el ejemplo anterior, si se quiere rotar al sistema {B} sobre sus propios ejes ${}^B\hat{X}$ seguido por su eje ${}^B\hat{Y}$. se representaría como:

$${}^A R = {}^A_{B'} R_X \cdot {}^{B'}_B R_Y$$

Como las rotaciones pueden considerarse ya sea como operadores o como descripciones de orientación, no es sorprendente que distintas representaciones sean preferidas para cada uno de estos usos. sin embargo, resulta un tanto impráctico lidiar con los nueve elementos de una matriz de rotación en un giro compuesto por varias rotaciones coordinadas. Por tal motivo, resulta más simple una representación que requiriera solamente tres números. Tanto en Robótica como en Navegación, las transformaciones que se realizan a los sistemas de coordenadas, haciendo uso de matrices de rotación, se definen acorde a diferentes convenciones. Dos ejemplos de ello son los giros por **Ángulos fijos X - Y - Z** y otro es por medio de los llamados **Ángulos de Euler Z - Y - X** ó **Z - Y - Z**.

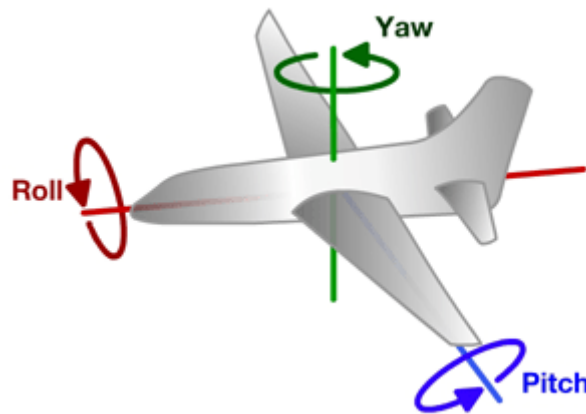


Figura 10.2: Movimientos *Roll*, *Pitch*, *Yaw* (Balanceo, Inclinación, Orientación) en una aeronave

Ángulos fijos X - Y - Z

El método para orientar {B} con respecto a {A} por medio de ángulos fijos es rotar a {B} en el orden que se especifica ($X_A \rightarrow Y_A \rightarrow Z_A$). Cada una de las tres rotaciones se lleva a cabo sobre un eje en el sistema de referencia fija A. Se empieza con la trama {B} coincidente con {A}, y posteriormente se gira {B} primero sobre \hat{X}_A usando un ángulo γ , luego sobre \hat{Y}_B usando un ángulo β y finalmente sobre \hat{Z}_A usando un ángulo α . Algunas veces esta convención se conoce como **ángulos de balanceo, inclinación y orientación** (en inglés, *roll*, *pitch*, *yaw*), pero hay que tener cuidado ya que a menudo este nombre se da a otras convenciones, relacionadas pero distintas. [20]

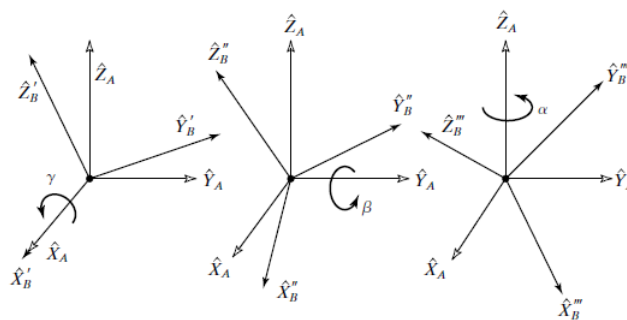


Figura 10.3: Rotación por ángulos fijos X - Y - Z

En las Ecuaciones de 10.3 se aprecia la función de tres argumentos que devuelve la matriz de rotación de ángulos fijos para X-Y-Z al ingresar los ángulos de roll, pitch y yaw.

$$\begin{aligned}
 {}^A_B R_{XYZ}(\gamma, \beta, \alpha) &= R_Z(\alpha)R_Y(\beta)R_X(\gamma) \\
 &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} \\
 &= \begin{bmatrix} cac\beta & cas\beta c\gamma - sac\gamma & cas\beta c\gamma + sas\gamma \\ sac\beta & sas\beta c\gamma + cac\gamma & sas\beta c\gamma - cas\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad (10.3)
 \end{aligned}$$

Si se considera cada elemento de la matriz como r_{ij} , se pueden obtener los ángulos al resolver un sistema de ecuaciones propias de la expresión en 10.3:

$$\begin{aligned}
 \beta &= \text{atan2}(r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\
 \alpha &= \text{atan2}\left(\frac{r_{21}}{c\beta}, \frac{r_{11}}{c\beta}\right) \\
 \gamma &= \text{atan2}\left(\frac{r_{32}}{c\beta}, \frac{r_{33}}{c\beta}\right)
 \end{aligned} \quad (10.3)$$

en donde atan2 es la función de arco tangente de dos argumentos, el cual calcula el arco en los cuatro cuadrantes usando los signos de ambos argumentos.

De las ecuaciones en 10.3, es posible notar que son válidas para $c\beta \neq 0$, ya que de lo contrario la solución se indetermina. por lo que si se llega a dar el caso, una posible convención es asignar el valor $\alpha = 0$ Generando los siguientes valores solución[20]:

$$\begin{aligned}
 \beta &= (90.0^\circ, -90.0^\circ) \\
 \alpha &= (0, 0)
 \end{aligned} \quad (10.2)$$

$$\gamma = (\text{atan2}(r_{12}, r_{22}), -\text{atan2}(r_{12}, r_{22})) \quad (10.3)$$

Ángulos d Euler Z - Y - X

Para describir al sistema {B} por medio de ángulos de Euler es girar a \hat{Z}_B usando el ángulo α , después sobre \hat{Y}_B usando un ángulo β , y finalmente sobre \hat{X}_B usando un ángulo γ . Nótese que cada rotación se lleva a cabo sobre un eje el cual depende de las rotaciones anteriores. [20]

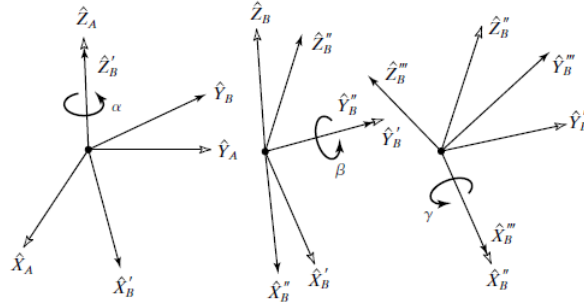


Figura 10.4: Rotación por ángulos de Euler Z - Y - X

En la Figura 10.4, se muestran las rotaciones antes descritas. Se agrega una prima adicional a cada eje con cada rotación. por lo tanto la matriz de rotación resultante resulta:

$$\begin{aligned}
 {}^A_B R_{Z'Y'X'} &= R_Z(\alpha)R_Y(\beta)R_X(\gamma) \\
 &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} \\
 &= \begin{bmatrix} cac\beta & cas\beta c\gamma - sac\gamma & cas\beta c\gamma + sas\gamma \\ sac\beta & sas\beta c\gamma + cac\gamma & sas\beta c\gamma - cas\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad (10.2)
 \end{aligned}$$

Se puede apreciar que las ecuaciones 10.3 y 10.2 son las mismas. Esto se debe a que tres rotaciones realizadas sobre tres ejes fijos producen igual orientación final que las mismas tres rotaciones realizadas en orden opuesto sobre los ejes de la trama móvil. Por consiguiente, las ecuaciones 10.3 también son válidas para resolver ángulos de Euler Z-Y-X a partir de una matriz de rotación dada, así como las limitaciones ya mencionadas.

Ángulos de Euler Z - Y - Z

Otra posible descripción para la trama {B} es girarla de la misma manera en la que se rota mediante la configuración de los ángulos de Euler Z - Y - Z. Siguiendo el mismo desarrollo que en la sección anterior, se llega a la matriz:

$${}^A_B R_{Z'Y'X'} = \begin{bmatrix} cac\beta c\gamma - sas\gamma & -cac\beta s\gamma - sac\gamma & cas\beta \\ sac\beta c\gamma + cas\gamma & -sac\beta s\gamma + cac\gamma & sas\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix} \quad (10.2)$$

En cuyo caso, los ángulos solución para $\beta \neq 0$ son:

$$\begin{aligned}
 \beta &= \text{atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}) \\
 \alpha &= \text{atan2}\left(\frac{r_{23}}{s\beta}, \frac{r_{13}}{s\beta}\right) \\
 \gamma &= \text{atan2}\left(\frac{r_{32}}{s\beta}, -\frac{r_{31}}{s\beta}\right)
 \end{aligned} \quad (10.2)$$

De la misma manera que en las matrices de rotación de ángulos fijos, dado el caso $\sin(\beta) = 0$, la solución se indetermina. Por tal motivo, una convención válida es asumir que $\alpha = 0$. Generando así las soluciones [20]:

$$\begin{aligned}
\beta &= (0, 180) \\
\alpha &= \text{atan2}\left(\frac{r_{23}}{s\beta}, \frac{r_{13}}{s\beta}\right) \\
\gamma &= \text{atan2}\left(\frac{r_{32}}{s\beta}, -\frac{r_{31}}{s\beta}\right)
\end{aligned} \tag{10.1}$$

Ángulo - Eje Equivalente

Sea una trama {A} que es girada a {B} sobre un vector ${}^A\hat{K}$ usando un ángulo θ , se define una rotación finita y a ésta representación se le conoce como representación de ángulo-eje equivalente. Se escribe como: ${}^A_B R(\hat{K}, \theta)$ o $R_K(\theta)$. La especificación del vector ${}^A\hat{K}$ sólo requiere de dos parámetros debido a que se trata de un vector unitario. Por otro lado, el ángulo θ representa un tercer parámetro. [20]

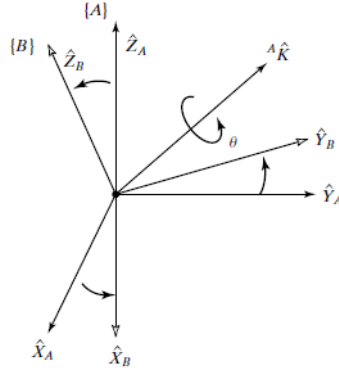


Figura 10.5: Representación de Ángulo - Eje Equivalente [20]

La matriz de rotación equivalente a esta representación es:

$$R_K(\theta) = \begin{bmatrix} k_x^2 v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y^2 v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z^2 v\theta + c\theta \end{bmatrix} \tag{10.1}$$

$$\tag{10.2}$$

en donde $c\theta = \cos \theta$, $s\theta = \sin \theta$, $v\theta = 1 - \cos \theta$ y ${}^A\hat{K} = [k_x \quad k_y \quad k_z]^T$.

Cuando se selecciona el eje de rotación de entre los ejes principales de {A} se obtienen las matrices de rotación equivalentes.

$$R_X(\theta) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \tag{10.3}$$

$$R_Y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \tag{10.4}$$

$$R_Z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10.5}$$

$$\tag{10.6}$$

El problema inverso de calcular \hat{K} y θ a partir de una matriz de rotación conocida se obtienen sus parámetros mediante:

$$\theta = \operatorname{angcos} \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right) \quad (10.7)$$

$$\hat{K} = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \quad (10.8)$$

$$(10.9)$$

Sin embargo, ésta solución es válida para $\theta \in (0, \pi)$ y para todo par de ángulo-eje $(^A K, \theta)$ hay otro par $(-K, -\theta)$ para el cual resulta la misma orientación y lo describe la misma matriz. En adición, se puede apreciar que para rotaciones pequeñas lo puede definir de manera incorrecta; y si es que es nula o igual a π , se indetermina.[20]

10.2. Propiedades algebraicas de los cuaterniones

Las propiedades de operaciones binarias de los Cuaterniones son las siguientes[23]:

1. Adición:

$$\begin{aligned} p + q &= (p_1 + ip_2 + jp_3 + kp_4) + (q_1 + iq_2 + jq_3 + kq_4) \\ &= (p_1 + q_1) + i(p_2 + q_2) + j(p_3 + q_3) + k(p_4 + q_4) \end{aligned}$$

Esta regla preserva asociatividad y conmutatividad para la adición.

2. Producto:

La regla de multiplicación o el producto aplica de la misma manera que se aplica a los polinomios teniendo en cuenta las propiedades de las entidades i, j, k :

$$\begin{aligned} p \otimes q &= (p_1 + ip_2 + jp_3 + kp_4)(q_1 + iq_2 + jq_3 + kq_4) \\ &= (p_1q_1 - p_2q_2 - p_3q_3 - p_4q_4) + i(p_1q_2 + p_2q_1 + p_3q_4 - p_4q_3) \\ &\quad + j(p_1q_3 + p_3q_1 + p_4q_2 - p_2q_4) + k(p_1q_4 + p_4q_1 + p_2q_3 - p_3q_2) \end{aligned} \quad (10.5)$$

Para el producto entre cuaterniones no se preserva la conmutatividad, analizando minuciosamente los términos resultantes.

3. Conjugados:

Manteniendo la consistencia de los números complejos, un conjugado de un cuaternión se define como:

$$q^* = (q_1 + iq_2 + jq_3 + kq_4)^* \equiv q_1 - iq_2 - jq_3 - kq_4$$

Al igual que los números complejos, cabe resaltar que tanto $q \otimes q^*$ como $(q + q^*)$ resultan en números reales. Y encima de ello, si se define la norma de un cuaternión como:

$$|q| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}$$

Asumiendo que la raíz resultante es positiva entonces se cumple que:

$$q \otimes q^* = q^* \otimes q = |q|^2$$

La operación del conjugado es distributiva para la suma $(p + q)^* = (q + p)^*$. Por otro lado, para la multiplicación resulta ser que $(p \otimes q)^* = q^* \otimes p^*$.

4. Inverso multiplicativo («División»)

Ya que **matemáticamente no existe un cuaternión como cociente de una división**. Para el conjunto de todos los cuaterniones con las operaciones descritas (adición y producto) define a un **anillo**, o de forma mas específica a un **anillo no conmutativo para la división**. Lo cual quiere enfatizar que para todos aquellos cuaterniones no nulos existe un cuaternión inverso y dichos productos de cuaterniones no son conmutativos. Por lo tanto, para definir un multiplicativo inverso de un cuaternión se parte de definir a dos ecuaciones. Debido a que la multiplicación no es conmutativa, se toman en consideración a dos elementos: q_L^{-1} y q_R^{-1} como las propuestas simbólicas de la solución de p/q :

$$q \otimes q_L^{-1} = p, \quad q_R^{-1} \otimes q = p$$

Multiplicando en ambas ecuaciones a por la izquierda y por la derecha respectivamente por $\frac{q^*}{|q|^2}$, obtenemos:

$$q_L^{-1} = \frac{q^* \otimes p}{|q|^2}, \quad q_R^{-1} = \frac{p \otimes q^*}{|q|^2}$$

Una vez obtenido las expresiones antes descritas, se considera a $p = 1$ para obtener el inverso multiplicativo del cuaternión q :

$$q_L^{-1} = q_R^{-1} = q^{-1} = \frac{q^*}{|q|^2} \quad (10.4)$$

10.3. Filtro de Madgwick[19]

Orientación a partir de la velocidad angular

Se obtiene un vector ${}^S\omega = [0 \ \omega_x \ \omega_y \ \omega_z]$ a partir de las lecturas de los giroscopios en $[rad^{-1}]$. Donde cada término ω representa dichas lecturas en los respectivos ejes del sensor. Este vector describe la razón de cambio en la orientación respecto al sensor mismo.

Por lo tanto se puede determinar la orientación del marco terrestre con respecto del sensor en un tiempo t asumiendo que existe una derivada de cuaternión que defina la razón de cambio de la orientación del marco terrestre con respecto al sensor [39] de la forma ${}^S_E\dot{q}$, identificada como la derivada del cuaternión S_Eq .

Para la obtención de la orientación del marco terrestre relativo al sensor en el tiempo t , ${}^S_Eq_{\omega,t}$, se obtiene integrando numéricamente su primer derivada descrita anteriormente, conociendo las condiciones iniciales del mismo. Por lo que primero se obtiene dicha derivada y posteriormente se «integra» de forma numérica multiplicando por el tiempo de muestreo Δt . Para las condiciones iniciales se conoce un cuaternión estimado de la iteración pasada (en tiempo $t - 1$) ${}^S_E\hat{q}_{est,t-1}$. Finalmente la orientación obtenida a partir de la lecturas de los giroscopios se denota con el subíndice ω . Las ecuaciones 10.4 describen lo anterior explicado.

$${}^S_E \dot{q}_{\omega,t} = \frac{1}{2} {}^S_E \hat{q}_{est,t-1} \otimes {}^S \omega_t$$

$${}^S_E q_{\omega,t} = {}^S_E \hat{q}_{est,t-1} + {}^S_E \dot{q}_{\omega,t} \Delta t \quad (10.4)$$

Orientación a partir de la observación de vectores

En la estimación de la orientación de un cuerpo con respecto a un marco de referencia, es necesario conocer por lo menos dos vectores en el marco de referencia del cuerpo (asumiendo ambas referencias son ortogonales), de manera que exista una solución única para la orientación. Para ello, se utilizan dos referencias que encontramos de manera natural en el marco terrestre: La dirección del campo gravitatorio y la dirección del flujo magnético terrestres. Un acelerómetro de tres ejes, medirá la magnitud y dirección del campo gravitatorio en el marco del sensor, tomando en cuenta las componentes de aceleración provocadas por el movimiento del sensor. De manera similar, un magnetómetro de tres ejes mide la dirección y la magnitud del campo magnético terrestre además de los campos magnéticos locales y demás distorsiones. Dentro de este contexto, para estimar la orientación del sensor por medio de las referencias antes mencionadas, se asume que el acelerómetro medirá únicamente el campo gravitatorio terrestre, así como el magnetómetro sólo medirá el campo magnético terrestre. [19]

Al hacer uso de cuaterniones, Madgwick propone una solución a un problema de optimización para obtener una solución única a partir de las mediciones del acelerómetro y del magnetómetro. La orientación del sensor ${}^S_E \hat{q}$ será aquella que alinee una dirección predeterminada en el marco de referencia terrestre ${}^E \hat{d}$ con la dirección resultante de las mediciones del campo en el marco de referencia del sensor ${}^S \hat{s}$. Expresado mediante ecuaciones, es necesario encontrar a ${}^S_E \hat{q}$ como solución de la ecuación 10.5, en donde la función objetivo se expresa en la ecuación 10.6. [19]

$$\min_{{}^S_E \hat{q} \in \mathbb{R}^4} f \left({}^S_E \hat{q}, {}^E \hat{d}, {}^S \hat{s} \right) \quad (10.5)$$

$$f \left({}^S_E \hat{q}, {}^E \hat{d}, {}^S \hat{s} \right) = {}^S_E \hat{q}^* \otimes {}^E \hat{d} \otimes {}^S_E \hat{q} - {}^S \hat{s} \quad (10.6)$$

Las componentes de cada término de la función objetivo serían descritas como:

$${}^S_E \hat{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]$$

$${}^E \hat{d} = [0 \quad d_x \quad d_y \quad d_z]$$

$${}^S \hat{s} = [0 \quad s_x \quad s_y \quad s_z]$$

El método de optimización elegido por Madgwick es el **Algoritmo de Descenso de Gradiente** debido a que menciona que es el más sencillo de implementar y el que menor recursos de procesamiento requiere. Para este algoritmo se requieren un número n de iteraciones que resulten en una estimación de la orientación ${}^S_E \hat{q}_{n+1}$ basados en una orientación inicial arbitraria ${}^S_E \hat{q}_0$ y un coeficiente μ . La ecuación 10.3 expresa el algoritmo en sí. En la ecuación 10.3 se calcula el gradiente e la superficie solución definida como la función objetivo premultiplicada por su Jacobiano transpuesto. De manera que para la obtención de la multiplicación, se definen la función objetivo y su Jacobiano como matrices de tres filas en las ecuaciones 10.3 y 10.3 [19]

$${}^S_E \hat{q}_{k+1} = {}^S_E \hat{q}_k - \mu \frac{\nabla f \left({}^S_E \hat{q}, {}^E \hat{d}, {}^S \hat{s} \right)}{\|\nabla f \left({}^S_E \hat{q}, {}^E \hat{d}, {}^S \hat{s} \right)\|} \quad (10.3)$$

$$\nabla f \left({}^S_E \hat{q}, {}^E \hat{d}, {}^S \hat{s} \right) = J^T \left({}^S_E \hat{q}_k, {}^E \hat{d} \right) f \left({}^S_E \hat{q}, {}^E \hat{d}, {}^S \hat{s} \right) \quad (10.3)$$

$$f \left({}^S_E \hat{q}, {}^E \hat{d}, {}^S \hat{s} \right) = \begin{bmatrix} 2d_x \left(\frac{1}{2} q_3^2 q_4^2 \right) + 2d_y (q_1 q_4 + q_2 q_3) + 2d_z (q_2 q_4 q_1 q_3) s_x \\ 2d_x (q_2 q_3 q_1 q_4) + 2d_y \left(\frac{1}{2} q_2^2 q_4^2 \right) + 2d_z (q_1 q_2 q_1 q_3) s_y \\ 2d_x (q_1 q_3 + q_2 q_4) + 2d_y (q_3 q_4 q_1 q_2) + 2d_z \left(\frac{1}{2} q_2^2 q_3^2 \right) \end{bmatrix} \quad (10.3)$$

$$J \left({}^S_E \hat{q}_k, {}^E \hat{d} \right) =$$

$$\begin{bmatrix} 2d_y q_4 2d_z q_3 & 2d_y q_3 + 2d_z q_4 & -4d_x q_3 + 2d_y q_2 2d_z q_1 & -4d_x q_4 + 2d_y q_1 + 2d_z q_2 \\ -2d_x q_4 + 2d_z q_2 & 2d_x q_3 4d_y q_2 + 2d_z q_1 & 2d_x q_2 + 2d_z q_4 & -2d_x q_1 4d_y q_4 + 2d_z q_3 \\ 2d_z q_3 2d_y q_2 & 2d_x q_4 2d_y q_1 4d_z q_2 & 2d_x q_1 + 2d_y q_4 4d_z q_3 & 2d_x q_2 + 2d_y q_3 \end{bmatrix} \quad (10.3)$$

Las ecuaciones anteriores describen el algoritmo para un caso general en el que el campo que se desea referenciar en la estimación de la orientación, contenga componentes en los tres ejes del marco de referencia terrestre. Sin embargo, recordemos que se analizan a dos campos en particular: el gravitatorio y el magnético terrestre. Por lo tanto, si asumimos que la dirección del campo gravitatorio solo tiene componente en el eje z y que la dirección de campo magnético terrestre tiene componentes en los ejes x y z , las expresiones para ${}^E \hat{d}$ y ${}^S \hat{s}$ cambian para cada sensor. En específico, para el acelerómetro corresponden ${}^E \hat{g}$ y ${}^S \hat{a}$ asumiendo que las mediciones del acelerómetro están normalizadas. Para el magnetómetro corresponden ${}^E \hat{b}$ y ${}^S \hat{m}$ respectivamente. En dichos casos, la función objetivo será diferente para cada caso, por lo que para el acelerómetro se tiene una expresión $f_g \left({}^S_E \hat{q}, {}^S \hat{a} \right)$ para la función objetivo y una expresión $J_g \left({}^S_E \hat{q} \right)$ para su Jacobiano; mientras que para el magnetómetro se tiene una expresión $f_b \left({}^S_E \hat{q}, {}^E \hat{b}, {}^S \hat{m} \right)$ para la función objetivo y una expresión $J_b \left({}^S_E \hat{q}, {}^E \hat{b} \right)$ para su Jacobiano.

En las ecuaciones 10.1 se definen las expresiones correspondientes al acelerómetro.

$${}^E \hat{g} = [0 \quad 0 \quad 0 \quad 1]$$

$${}^S \hat{a} = [0 \quad a_x \quad a_y \quad a_z]$$

$$f_g \left({}^S_E \hat{q}, {}^S \hat{a} \right) = \begin{bmatrix} 2(q_2 q_4 q_1 q_3) a_x \\ 2(q_1 q_2 + q_3 q_4) a_y \\ 2 \left(\frac{1}{2} q_2^2 q_3^2 \right) a_z \end{bmatrix}$$

$$J_g \left({}^S_E \hat{q} \right) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \quad (10.1)$$

En las ecuaciones 10.3 definen las expresiones correspondientes al magnetómetro.

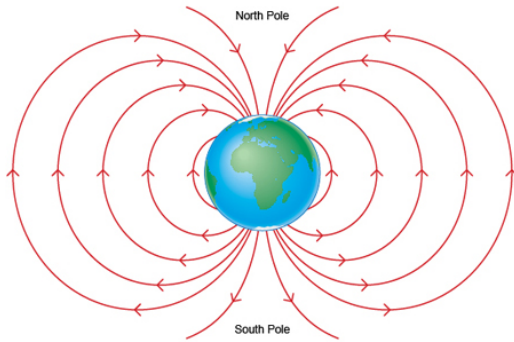
$${}^E\hat{b} = [0 \quad b_x \quad 0 \quad b_z]$$

$${}^S\hat{m} = [0 \quad m_x \quad m_y \quad m_z]$$

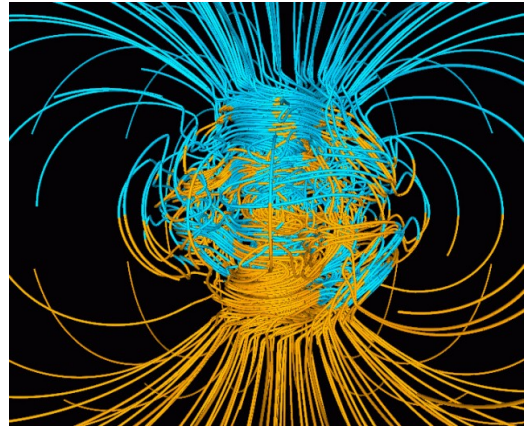
$$f_b \left({}^S\hat{q}, {}^E\hat{b}, {}^S\hat{m} \right) = \begin{bmatrix} 2b_x(0.5q_3^2q_4^2) + 2b_z(q_2q_4q_1q_3)m_x \\ 2b_x(q_2q_3q_1q_4) + 2b_z(q_1q_2 + q_3q_4)m_y \\ 2b_x(q_1q_3 + q_2q_4) + 2b_z(0.5q_2^2q_3^2)m_z \end{bmatrix}$$

$$J_b \left({}^S\hat{q}, {}^E\hat{b} \right) = \begin{bmatrix} b_zq_3 & 2b_zq_4 & -4b_xq_32b_zq_1 & -4b_xq_4 + 2b_zq_2 \\ -2b_xq_4 + 2b_zq_2 & 2b_xq_3 + 2b_zq_1 & 2b_xq_2 + 2b_zq_4 & -2b_xq_1 + 2b_zq_3 \\ 2b_xq_3 & 2b_xq_44b_zq_2 & 2b_xq_14b_zq_3 & 2b_xq_2 \end{bmatrix} \quad (10.-3)$$

Es importante resaltar que el número de parámetros en ambos casos son diferentes debido que la dirección del campo gravitatorio en cualquier parte del planeta se considera la misma, por lo tanto, el vector se vuelve una constante y ya no es un parámetro que deba considerarse; simplemente se calcula y simplifica. Por el contrario, el campo magnético terrestre es diferente según la zona geográfica. Las líneas del campo magnético son irregulares como se muestra en la Fig. 10.6b además que a medida que uno se aleja del ecuador hacia los polos, la componente vertical del campo magnético aumenta. A los fenómenos de cambio de dirección en el campo magnético terrestre real contra el ideal, se les conoce como *declinación e inclinación* magnéticas. La declinación es la corrección del ángulo que describe la dirección del norte magnético terrestre en el plano perpendicular al suelo (o *yaw*). La inclinación es el ángulo existente entre el plano paralelo al suelo y el norte verdadero (o *pitch*).



(a) Líneas de Campo Magnético Terrestre de Forma Ideal



(b) Líneas de Campo Magnético Terrestre de Forma Real

Figura 10.6: Campo Magnético Terrestre

Como se mencionó en un inicio, la obtención de ambos vectores en el marco de referencia del sensor es necesaria para que la solución sea única. Al aplicar el algoritmo a las funciones objetivo en las ecuaciones 10.1 y 10.3, se obtienen de las superficie de soluciones, una curva para cada una. Al aplicar el algoritmo en conjunto, la solución del mínimo deseado se define como un punto (intersectado por ambas curvas) volviéndola así una solución única. Por consiguiente, se agrupan las expresiones de ambos sensores de manera matricial (como se muestra en las ecuaciones 10.-2) para el cálculo de la solución. [19]

$$f_{g,b} \left({}^S_E \hat{q}, {}^S \hat{a}, {}^E \hat{b}, {}^S \hat{m} \right) = \begin{bmatrix} f_g \left({}^S_E \hat{q}, {}^S \hat{a} \right) \\ f_b \left({}^S_E \hat{q}, {}^E \hat{b}, {}^S \hat{m} \right) \end{bmatrix} \quad (10.-2)$$

$$J_{g,b} \left({}^S_E \hat{q}, {}^E \hat{b} \right) = \begin{bmatrix} J_g \left({}^S_E \hat{q} \right) \\ J_b \left({}^S_E \hat{q}, {}^E \hat{b} \right) \end{bmatrix}$$

Un algoritmo de optimización de esta naturaleza requiere un número de iteraciones considerables e incluso una adaptación del valor de coeficiente μ en cada una de éstas. Sin embargo, el costo en cuestión del procesamiento requerido se eleva considerablemente. Por tal motivo, se considera suficiente una iteración por cada medición que se realice, garantizando la convergencia en los valores asumiendo que el valor de μ sea mayor o igual que la razón de cambio física en la orientación del sensor. Finalmente se calcula el cuaternión ${}^S_E \hat{q}_{\nabla,t}$ (Donde el subíndice ∇ indica que se obtiene la estimación mediante el algoritmo de descenso de gradiente) que estima la orientación del sensor en el tiempo t basado en una condición inicial en el tiempo $t - 1$ y el gradiente de la función objetivo ∇f definido por las mediciones de los sensores ${}^S \hat{a}_t$ y ${}^S \hat{m}_t$ medidos en el tiempo t .

$${}^S_E q_{\nabla,t} = {}^S_E \hat{q}_{est,t-1} - \mu_t \frac{\nabla f}{\|\nabla f\|}$$

$$\nabla f = J_{g,b}^T \left({}^S_E \hat{q}, {}^E \hat{b} \right) f_{g,b} \left({}^S_E \hat{q}, {}^S \hat{a}, {}^E \hat{b}, {}^S \hat{m} \right)$$

Para el coeficiente μ , se considera que al ser proporcional a la razón de cambio de la orientación del sensor, se obtiene su valor de ${}^S_E \dot{q}_{\omega,t}$ obtenido desde las mediciones del giroscopio. Para μ en el tiempo t :

$$\mu_t = \alpha \| {}^S_E \dot{q}_{\omega,t} \| \Delta t, \quad \alpha > 1 \quad (10.-5)$$

En donde α es un coeficiente proporcional al ruido de las mediciones de los acelerómetros y magnetómetro.

Algoritmo de fusión de filtros

Para obtener una orientación estimada del sensor en un tiempo t , se define un cuaternión ${}^S_E q_{est,t}$. Dicho cuaternión se obtiene mediante la fusión de los cuaterniones resultantes de los cálculos previos: ${}^S_E q_{\omega,t}$ y ${}^S_E q_{\nabla,t}$. La manera en la que se combinan ambos resultados es mediante el concepto utilizado en un filtro complementario, asignando un peso a cada término como se muestra en la ecuación 10.3.

$${}^S_E q_{est,t} = \gamma_t {}^S_E q_{\nabla,t} + (1 - \gamma_t) {}^S_E q_{\omega,t}, \quad 0 \leq \gamma_t \leq 1 \quad (10.-5)$$

En donde el valor de γ_t debe ser tal que asegure que la divergencia en la orientación estimada en comparación con la orientación real de ${}^S_E q_{\omega,t}$ sea igual a la convergencia de ${}^S_E q_{\nabla,t}$. Como se expresa en la ecuación 10.-4, $\frac{\mu_t}{\Delta t}$ representa la razón de convergencia de ${}^S_E q_{\nabla,t}$, mientras que β representa la razón de convergencia de ${}^S_E q_{\omega,t}$ expresado como la magnitud de la derivada de un cuaternión correspondiente con el error de medición del giroscopio.

$$(1 - \gamma_t) \beta = \gamma_t \frac{\mu_t}{\Delta t} \quad (10.-4)$$

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta} \quad (10.-3)$$

Para el funcionamiento óptimo del algoritmo, es necesario que la razón de convergencia de ${}^S_E q_{\nabla,t}$ sea mayor que la razón de divergencia de ${}^S_E q_{\omega,t}$. Por lo tanto de la ecuación 10.3 se puede asumir que α no tiene un límite superior. Si se define el valor de α como un número muy grande, entonces μ_t se vuelve muy grande a su vez. Lo que significa que de la ecuación 10.-2, el término ${}^S_E \hat{q}_{est,t-1}$ se vuelve despreciable y la ecuación se simplifica a la ecuación 10.3.

$${}^S_E q_{\nabla,t} \approx -\mu_t \frac{\nabla f}{\|\nabla f\|} \quad (10.-3)$$

De la ecuación 10.-3 donde se define γ_t , también se simplifica al hacer despreciable el valor de β en el denominador, dando como resultado la ecuación 10.3. E incluso se puede llegar a asumir que $\gamma_t \approx 0$.

$$\gamma_t \approx \frac{\beta \Delta t}{\mu_t} \quad (10.-3)$$

Substituyendo las ecuaciones 10.4, 10.3 y 10.3 en la ecuación 10.3, se obtiene la ecuación 10.3. Cabe resaltar que para γ_t se utilizaron los valores de la expresión 10.3 y 0.

$${}^S_E q_{est,t} = \frac{\beta \Delta t}{\mu_t} \left(-\mu_t \frac{\nabla f}{\|\nabla f\|} \right) + (1 - 0) ({}^S_E \hat{q}_{est,t-1} + {}^S_E \dot{q}_{\omega,t} \Delta t) \quad (10.-3)$$

De la ecuación 10.3 se puede obtener un arreglo de ecuaciones simplificadas. Donde ${}^S_E \dot{q}_{est,t}$ es la razón de cambio de la orientación estimada y ${}^S_E \dot{q}_{\epsilon,t}$ es la dirección del error de ${}^S_E \dot{q}_{est,t}$.

$${}^S_E q_{est,t} = {}^S_E \hat{q}_{est,t-1} + {}^S_E \dot{q}_{est,t} \Delta t \quad (10.-2)$$

$${}^S_E \dot{q}_{est,t} = {}^S_E \dot{q}_{\omega,t} - \beta {}^S_E \dot{q}_{\epsilon,t} \quad (10.-1)$$

$${}^S_E \dot{q}_{\epsilon,t} = \frac{\nabla f}{\|\nabla f\|} \quad (10.0)$$

Lo que se establece mediante las ecuaciones anteriores es que:

- Se calcula la orientación ${}^S_E q_{est,t}$ mediante la integración numérica de ${}^S_E \dot{q}_{est,t}$ (ecuación 10.-2)
- El filtro calcula ${}^S_E \dot{q}_{est,t}$ como el cambio de la orientación obtenida mediante los giroscopios ${}^S_E \dot{q}_{\omega}$ restando a éste la magnitud del error de medición del giroscopio (β) en la dirección del cambio de la orientación estimada a partir de las mediciones de acelerómetros y magnetómetros (ecuaciones 10.-1 y 10.0)

Compensación de distorsión magnética

Como se abordó anteriormente, las mediciones del campo magnético terrestre pueden verse distorsionadas ante la presencia de elementos ferromagnéticos en las cercanías del sensor.[19]

Madgwick, junto con su equipo de colaboradores, reportan que los errores por acero-duro pueden eliminarse mediante una calibración previa al sensado. Para la corrección de la declinación magnética indica que es necesario una referencia adicional para ser compensada; mientras que la inclinación puede ser compensada mediante la orientación estimada mediante las lecturas del acelerómetro.

Para una lectura de la dirección del campo magnético terrestre ${}^E \hat{h}_t$ en un tiempo t , puede ser

procesada como la medición normalizada del magnetómetro ${}^S\hat{m}_t$ rotado mediante la orientación estimada de la fusión de filtros ${}^S\hat{q}_{t-1}$ como se describe en la ecuación 10.1. Los efectos de una medición errónea del campo magnético terrestre ${}^E\hat{h}_t$ atribuida a la inclinación puede ser corregida si la inclinación de la dirección de referencia del filtro ${}^E\hat{b}_t$ tiene la misma inclinación. En este caso, es necesario procesar tanto a ${}^E\hat{b}_t$ como ${}^E\hat{h}_t$ (normalizadas) con componentes exclusivas a los ejes x y z del marco de referencia terrestre, como se muestra en la ecuación 10.2.[19]

$${}^E\hat{h}_t = [0 \quad h_x \quad h_y \quad h_z] = {}^S\hat{q}_{t-1} \otimes {}^S\hat{m}_t \otimes {}^S\hat{q}_{t-1}^* \quad (10.1)$$

$${}^E\hat{b}_t = [0 \quad \sqrt{h_x^2 + h_y^2} \quad 0 \quad h_z] \quad (10.2)$$

De esta manera se asegura que las distorsiones magnéticas se limiten a afectar únicamente la dirección en el plano tangente al suelo. Además de que elimina la necesidad de predefinir una referencia magnética.[19] Sin embargo, si se requiere dicha referencia basta con realizar una compensación al ángulo en dicho plano («yaw»).[27]

Compensación del *drift* del Giroscopio

Toda aplicación de los giroscopio, debe tener en cuenta la divergencia de la medición del desplazamiento angular o *drift*. Una ventaja que ofrece la implementación del filtro de Kalman, es la estimación del error del giroscopio como un estado que define al modelo del sistema. Sin embargo, Mahony *et al.* demuestran que otra solución viable es la compensación del error del giroscopio mediante filtros de orientación que aprovechan el error en el cambio de la orientación como retroalimentación en la integración de las mediciones del giroscopio.[40]

Por lo tanto, el error en el cambio de la orientación ${}^S\dot{\hat{q}}_\epsilon$ puede expresarse como el error angular en cada eje del giroscopio como se muestra en la ecuación 10.3; derivado como el inverso de la relación expresada en la ecuación 10.4, en donde ${}^S\omega_b$ es representado por la componente constante de ${}^S\omega_\epsilon$ y debe ser eliminada de la integral de ${}^S\omega_\epsilon$ mediante un respectivo peso ζ . Por lo que se obtiene el la medición compensada del giroscopio ${}^S\omega_c$ como se muestra en la ecuación 10.5, cuyo valor inicial será ${}^S\omega_{c,0} = 0$. Este valor habrá de utilizarse en lugar de la medición directa del giroscopio (sustituyendo a ${}^S\omega_t$ en las ecuaciones 10.4). [19]

$${}^S\omega_{\epsilon,t} = 2 {}^S\hat{q}_{est,t-1}^* \otimes {}^S\dot{\hat{q}}_{\epsilon,t} \quad (10.3)$$

$${}^S\omega_{b,t} = \zeta \sum_t {}^S\omega_{\epsilon,t} \Delta t \quad (10.4)$$

$${}^S\omega_{c,t} = {}^S\omega_t - {}^S\omega_{b,t} \quad (10.5)$$

Ganancias del Filtro [19]

Al completar el modelo del algoritmo de Madgwick, es necesario establecer los valores de las ganancias β y ζ para una estimación óptima de la orientación del sensor.

La ganancia β representa el promedio de los errores en las mediciones. Los errores involucrados pueden ser: ruido, muestreo insuficiente («*aliasing*»), errores de cuantización, errores de calibración, desalineamiento de los ejes del sensor, entre otras.

La ganancia ζ representa la razón de convergencia a eliminar los errores de medición del giroscopio.

Es conveniente definir a β y ζ usando las cantidades angulares $\tilde{\omega}_\beta$ y $\dot{\omega}_\zeta$ respectivamente. En donde, $\tilde{\omega}_\beta$ representa el promedio del error estimado de las mediciones del giroscopio en cada uno de sus ejes y $\dot{\omega}_\zeta$ representa la divergencia estimada («*drift*») en las mediciones del giroscopio en cada uno de sus ejes. Utilizando la relación descrita en las ecuaciones 10.4, se pueden expresar matemáticamente como 10.3 y 10.3.

$$\beta = \left\| \frac{1}{2} \hat{q} \otimes [0 \quad \tilde{\omega}_\beta \quad \tilde{\omega}_\beta \quad \tilde{\omega}_\beta] \right\| = \sqrt{\frac{3}{4}} \tilde{\omega}_\beta \quad (10.5)$$

$$\zeta = \sqrt{\frac{3}{4}} \dot{\omega}_\zeta \quad (10.5)$$

10.4. Código del Receptor

10.4.1. Receptor de datos sin procesar

En el siguiente código el receptor está configurado para recibir los datos sin procesar enviados desde el transmisor:

```

1 #include <SPI.h>
2 #include "RF24.h"
3
4 #define CE 53
5 #define CSN 48
6
7 // Control
8 bool test = true;
9
10 /***** USER Configuration *****/
11 // Hardware configuration
12 RF24 radio(CE, CSN); // Set up nRF24L01 radio on SPI bus plus pins 7 & 8
13
14 /*****/
15 const uint8_t maxDevices = 30;
16 uint8_t devicesConnected = 0;
17 uint8_t devicesToConnect = 0;
18
19 const uint64_t basePipe = 0x0F0F0F0F0LL; // Radio pipe addresses for the 2
    ↔ nodes to communicate.
20 uint64_t pipes[maxDevices];
21 byte b[37];
22 float timeCount = 0;
23 uint8_t timeData_size = sizeof(unsigned long);
24
25 struct DataType
26 {
27     uint8_t address;
28     float ax;
29     float ay;

```

```

30 float az;
31 float gx;
32 float gy;
33 float gz;
34 float mx;
35 float my;
36 float mz;
37 }data;
38 uint8_t data_size = sizeof(DataType);
39
40 void pairNewDevice();
41 void readAllPipes();
42
43 long microsCount;
44 int dataCount, lostCount;
45
46 void setup()
47 {
48     Serial.begin(2000000);
49
50     radio.begin(); // Setup and configure rf radio
51     radio.setPALevel(RF24_PA_MAX);
52     radio.setDataRate(RF24_2MBPS);
53     radio.setChannel(110);
54
55     while (devicesToConnect == 0)
56     {
57         Serial.print("Enter number of devices to connect: ");
58         while (!Serial.available()){}
59         devicesToConnect = Serial.parseInt();
60         if (devicesToConnect == 0)
61         {
62             Serial.println("Invalid entry");
63             continue;
64         }
65         Serial.println(devicesToConnect);
66     }
67
68     while (devicesConnected < devicesToConnect)
69         pairNewDevice();
70     radio.setAutoAck(false);
71     radio.flush_rx();
72     Serial.print("3...");
73     delay(1000);
74     Serial.print("2...");
75     delay(1000);
76     Serial.print("1...");
77     delay(1000);
78     Serial.println("Now!");
79     delay(10);

```

```

80
81  if (test)
82  {
83      dataCount = lostCount = 0;
84      microsCount = micros();
85      while (micros() - microsCount < 1000000)
86          readAllPipes();
87
88
89      Serial.println("");
90      Serial.print("Packages sent: "); Serial.println(dataCount);
91      Serial.print("Packages lost: "); Serial.println(lostCount);
92  }
93 }
94
95 void loop()
96 {
97     if (!test) readAllPipes();
98 }
99
100 void pairNewDevice()
101 {
102     if (devicesConnected == maxDevices)
103     {
104         Serial.println("No more devices can be paired.");
105         return;
106     }
107     radio.setAutoAck(true);
108     bool pairedDevice = false;
109
110     pipes[devicesConnected] = basePipe | (uint64_t)(devicesConnected); // Generate
        ↪ 64bits pipe address
111     radio.openReadingPipe(1, pipes[devicesConnected]); // Open new reading pipe
112     radio.flush_rx();
113
114     Serial.print("Waiting for device with ID: ");
115     Serial.print(devicesConnected);
116     Serial.println("'...'");
117     radio.startListening();
118
119     unsigned long millisCount = millis();
120     do
121     {
122         if (radio.available())
123         {
124             pairedDevice = true;
125             break;
126         }
127     } while (millis() - millisCount < 10000);
128

```



```

129  if (pairedDevice)
130  {
131      Serial.print("Connected device with ID: "); Serial.print(devicesConnected);
132      Serial.println("!!!");
133      devicesConnected++;
134      Serial.print("Connected devices: "); Serial.println(devicesConnected);
135
136      radio.closeReadingPipe(1);
137      radio.flush_rx();
138  }
139  else
140  {
141      Serial.println("No response.");
142      pairNewDevice();
143  }
144 }
145
146 void readAllPipes()
147 {
148     for (int i = 0; i < devicesConnected; i++)
149     {
150         radio.openReadingPipe(1, pipes[i]);
151
152         radio.startListening();
153         while (!radio.available()) {}
154         if (radio.available())
155         {
156             timeCount = ((float)(micros() - microsCount))/1000000;
157             radio.read(&data, data_size);
158             radio.closeReadingPipe(1);
159             dataCount++;
160             Serial.write((byte*)&data, data_size);
161             Serial.write((byte*)&timeCount, 4);
162         }
163         else
164         {
165             lostCount++;
166         }
167     }
168 }

```

10.4.2. Receptor de cuaterniones

En el siguiente código el receptor está configurado para recibir los cuaterniones enviados desde el transmisor:

```

1 #include <SPI.h>
2 #include "RF24.h"
3
4 #define CE 7

```

```

5 #define CSN 8
6
7 // Control
8 bool test = false;
9
10 /***** USER Configuration *****/
11 // Hardware configuration
12 RF24 radio(CE, CSN); // Set up nRF24L01 radio on SPI bus plus pins 7 & 8
13
14 /*****/
15 const uint8_t maxDevices = 30;
16 uint8_t devicesConnected = 0;
17 uint8_t devicesToConnect = 0;
18
19 const uint64_t basePipe = 0xF0F0F0F0LL; // Radio pipe addresses for the 2
    ↔ nodes to communicate.
20 uint64_t pipes[maxDevices];
21 byte b[17];
22 float timeCount = 0;
23 uint8_t timeData_size = sizeof(unsigned long);
24
25 struct DataType
26 {
27     uint8_t address;
28     float q0;
29     float q1;
30     float q2;
31     float q3;
32 }data;
33 uint8_t data_size = sizeof(DataType);
34
35 void pairNewDevice();
36 void readAllPipes();
37
38 long microsCount;
39 int dataCount, lostCount;
40
41 void setup()
42 {
43     Serial.begin(2000000);
44
45     radio.begin(); // Setup and configure rf radio
46     radio.setPALevel(RF24_PA_MAX);
47     radio.setDataRate(RF24_2MBPS);
48     radio.setChannel(110);
49
50     while (devicesToConnect == 0)
51     {
52         Serial.print("Enter number of devices to connect: ");
53         while (!Serial.available()){

```

```

54     devicesToConnect = Serial.parseInt();
55     if (devicesToConnect == 0)
56     {
57         Serial.println("Invalid entry");
58         continue;
59     }
60     Serial.println(devicesToConnect);
61 }
62
63 while (devicesConnected < devicesToConnect)
64     pairNewDevice();
65 radio.setAutoAck(false);
66 radio.flush_rx();
67 Serial.print("3...");
68 delay(1000);
69 Serial.print("2...");
70 delay(1000);
71 Serial.print("1...");
72 delay(1000);
73 Serial.println("Now!");
74 delay(10);
75
76 if (test)
77 {
78     dataCount = lostCount = 0;
79     microsCount = micros();
80     while (micros() - microsCount < 1000000)
81         readAllPipes();
82
83
84     Serial.println("");
85     Serial.print("Packages sent: "); Serial.println(dataCount);
86     Serial.print("Packages lost: "); Serial.println(lostCount);
87     delay(500);
88     abort();
89 }
90 else microsCount = micros();
91 }
92
93
94 void loop()
95 {
96     readAllPipes();
97 }
98
99 void pairNewDevice()
100 {
101     if (devicesConnected == maxDevices)
102     {
103         Serial.println("No more devices can be paired.");

```

```

104     return;
105 }
106 radio.setAutoAck(true);
107 bool pairedDevice = false;
108
109 pipes[devicesConnected] = basePipe | (uint64_t)(devicesConnected); // Generate
    ↪ 64bits pipe address
110 radio.openReadingPipe(1, pipes[devicesConnected]); // Open new reading pipe
111 radio.flush_rx();
112
113 Serial.print("Waiting for device with ID: ");
114 Serial.print(devicesConnected);
115 Serial.println("...");
116 radio.startListening();
117
118 unsigned long millisCount = millis();
119 do
120 {
121     if (radio.available())
122     {
123         pairedDevice = true;
124         break;
125     }
126 } while (millis() - millisCount < 10000);
127
128 if (pairedDevice)
129 {
130     Serial.print("Connected device with ID: "); Serial.print(devicesConnected);
131     Serial.println("!!!");
132     devicesConnected++;
133     Serial.print("Connected devices: "); Serial.println(devicesConnected);
134
135     radio.closeReadingPipe(1);
136     radio.flush_rx();
137 }
138 else
139 {
140     Serial.println("No response.");
141     pairNewDevice();
142 }
143 }
144
145 void readAllPipes()
146 {
147     for (int i = 0; i < devicesConnected; i++)
148     {
149         radio.openReadingPipe(1, pipes[i]);
150
151         radio.startListening();
152         while (!radio.available()) {}

```

```

153   if (radio.available())
154   {
155       timeCount = ((float)(micros() - microsCount))/1000000;
156       radio.read(&data, data_size);
157       radio.closeReadingPipe(1);
158       dataCount++;
159       Serial.write((byte*)&data, data_size);
160       Serial.write((byte*)&timeCount, 4);
161   }
162   else
163   {
164       lostCount++;
165   }
166 }
167 }

```

10.5. Códigos del Transmisor

10.5.1. Transmisor de datos sin procesar

En el siguiente código el transmisor se configura para enviar los datos provenientes de la MPU-9250 únicamente aplicando las operaciones necesarias para enviarlos en las unidades necesarias (*i.e. Para la aceleración [g], para el giroscopio [°/s] y para el magnetómetro [T]*):

```

1 /* MPU9250 Basic Example Code
2  by: Kris Winer
3  date: April 1, 2014
4  license: Beerware - Use this code however you'd like. If you
5  find it useful you can buy me a beer some time.
6  Modified by Arturo Ruiz 2018
7
8  Hardware setup:
9  MPU9250 Breakout ----- ATMEGA328P
10 VDD ----- 3.3V
11 SDA ----- 27
12 SCL ----- 28
13 GND ----- GND
14 */
15 // ----- VARIABLES -----
16 #include "quaternionFilters.h"
17 #include "MPU9250.h"
18 #include <SPI.h>
19 #include <nRF24L01.h>
20 #include <RF24.h>
21 #include <EEPROM.h>
22
23 // Control
24 bool transmit = true,
25 debug = true;
26

```

```

27 // Digital Pins
28 #define ONLED 5
29 #define MagCalLED 6
30 #define TxLED 7
31 #define ErrorLED 8
32 #define MagCalButton 2
33 #define CE 9
34 #define CSN 10
35 #define BattVoltPin A0
36
37 // Data Structures
38 struct MagBiasScaleType
39 {
40     float bx;
41     float by;
42     float bz;
43     float sx;
44     float sy;
45     float sz;
46 }magBiasScale;
47 int magBiasScale_size = sizeof(magBiasScale);
48
49 struct DataType
50 {
51     uint8_t address;
52     float ax;
53     float ay;
54     float az;
55     float gx;
56     float gy;
57     float gz;
58     float mx;
59     float my;
60     float mz;
61 }data;
62 uint8_t data_size = sizeof(DataType);
63
64
65 // Auxiliaries
66 byte byteVariable;
67 float batteryLevel;
68
69 const uint8_t maxDevices = 30;
70 uint8_t deviceID = 0;
71 const uint64_t basePipe = 0x0F0F0F0F0F0F0F0FLL; // Radio pipe addresses for the 2
    ↔ nodes to communicate.
72 uint64_t pipe;
73
74 // Object declaration
75 MPU9250 myIMU;

```

```

76 RF24 radio(CE, CSN);
77
78 // Methods
79 void pairDevice();
80 void checkBattery();
81
82 // ----- SETUP -----
83
84 void setup()
85 {
86   Serial.begin(2000000);
87   pinMode(ONLED, OUTPUT);
88   pinMode(TxLED, OUTPUT);
89   pinMode(MagCalLED, OUTPUT);
90   pinMode(ErrorLED, OUTPUT);
91
92   pinMode(MagCalButton, INPUT);
93   pinMode(BattVoltPin, INPUT);
94
95   analogReference(INTERNAL);
96
97   digitalWrite(ONLED, HIGH);
98   digitalWrite(TxLED, LOW);
99   digitalWrite(MagCalLED, LOW);
100  digitalWrite(ErrorLED, LOW);
101
102  delay(1000);
103  checkBattery();
104
105  // I2C Initialization
106  Wire.begin();
107
108  if (digitalRead(MagCalButton) == HIGH)
109  {
110    // -----<Magnetometer Calibration> -----
111    digitalWrite(ONLED, LOW);
112    digitalWrite(MagCalLED, HIGH);
113    if (debug)
114      Serial.println("Calibrating magnetometer...");
115
116    myIMU.magCalMPU9250(myIMU.magBias, myIMU.magScale);
117
118    magBiasScale.bx = myIMU.magBias[0];
119    magBiasScale.by = myIMU.magBias[1];
120    magBiasScale.bz = myIMU.magBias[2];
121
122    magBiasScale.sx = myIMU.magScale[0];
123    magBiasScale.sy = myIMU.magScale[1];
124    magBiasScale.sz = myIMU.magScale[2];
125

```

```

126 EEPROM.put(magBiasScale_size, magBiasScale);
127
128 if (debug)
129 {
130     Serial.print("Adress: "); Serial.println(byteVariable);
131     Serial.println("AK8963 mag Biases (mG)");
132     Serial.println(magBiasScale.bx);
133     Serial.println(magBiasScale.by);
134     Serial.println(magBiasScale.bz);
135
136     Serial.println("AK8963 mag Scales (mG)");
137     Serial.println(magBiasScale.sx);
138     Serial.println(magBiasScale.sy);
139     Serial.println(magBiasScale.sz);
140 }
141 digitalWrite(MagCALLEd, LOW);
142 digitalWrite(ONLED, HIGH);
143 delay(100);
144 abort();
145 // -----</Magnetometer Calibration> -----
146 }
147
148 else
149 {
150 // ----- <MPU9250 SETUP> -----
151 // Check MPU9250 connectivity
152 byteVariable = myIMU.readByte(MPU9250_ADDRESS, WHO_AM_I_MPU9250);
153 if (byteVariable != 0x71)
154 {
155     if (debug)
156     {
157         Serial.print("MPU9250-ERROR! 0x"); Serial.println(byteVariable, HEX);
158         Serial.flush();
159     }
160     digitalWrite(ErrorLED, HIGH);
161     digitalWrite(ONLED, LOW);
162     abort();
163 }
164
165 // Start by performing self test and reporting values
166 myIMU.MPU9250SelfTest(myIMU.selfTest);
167
168 // Calibrate gyro and accelerometers, load biases in bias registers
169 myIMU.calibrateMPU9250(myIMU.gyroBias, myIMU.accelBias);
170
171 myIMU.initMPU9250();
172 // Initialize device for active mode read of acclerometer, gyroscope, and
173 // temperature
174
175     if (debug) Serial.println("MPU9250 initialized...");

```



```

176
177     delay(100);
178
179 // Check AK8963(Magnetometer) connectivity
180 byteVariable = myIMU.readByte(AK8963_ADDRESS, WHO_AM_I_AK8963);
181 if (byteVariable != 0x48)
182 {
183     if (debug)
184     {
185         Serial.print("AK8963-ERROR! 0x"); Serial.println(byteVariable, HEX);
186         Serial.flush();
187     }
188     digitalWrite(ErrorLED, HIGH);
189     digitalWrite(ONLED, LOW);
190     abort();
191 }
192 // Get magnetometer calibration from AK8963 ROM
193 myIMU.initAK8963(myIMU.factoryMagCalibration);
194 // Initialize device for active mode read of magnetometer
195
196 // Get sensor resolutions, only need to do this once
197 myIMU.getAres();
198 myIMU.getGres();
199 myIMU.getMres();
200
201 EEPROM.get(magBiasScale_size, magBiasScale);
202 myIMU.magBias[0] = magBiasScale.bx;
203 myIMU.magBias[1] = magBiasScale.by;
204 myIMU.magBias[2] = magBiasScale.bz;
205
206 myIMU.magScale[0] = magBiasScale.sx;
207 myIMU.magScale[1] = magBiasScale.sy;
208 myIMU.magScale[2] = magBiasScale.sz;
209
210 if(debug)
211 {
212     Serial.print("Address: "); Serial.println(byteVariable);
213     Serial.println("AK8963 mag Biases (mG)");
214     Serial.println(magBiasScale.bx);
215     Serial.println(magBiasScale.by);
216     Serial.println(magBiasScale.bz);
217
218     Serial.println("AK8963 mag Scales (mG)");
219     Serial.println(magBiasScale.sx);
220     Serial.println(magBiasScale.sy);
221     Serial.println(magBiasScale.sz);
222
223     Serial.println("AK8963 initialized...");
224 }
225 // -----</MPU9250 SETUP> -----

```

```

226
227 // -----<NRF24L01 SETUP> -----
228 // Initialize RF
229     radio.begin();
230
231 // Set Data Rate
232     radio.setDataRate(RF24_2MBPS);
233
234 // Set amplification power level
235     radio.setPALevel(RF24_PA_MAX);
236
237 // Set channel with low noise
238     radio.setChannel(110);
239
240 // Disable retries in writing data
241     radio.setRetries(0,0);
242     pairDevice();
243     radio.setAutoAck(false);
244
245     delay(500);
246 // -----</NRF24L01 SETUP> -----
247 }
248 digitalWrite(ONLED, LOW);
249 delay(1000);
250 digitalWrite(ONLED, HIGH);
251 }
252
253 // ----- LOOP -----
254 void loop()
255 {
256     checkBattery();
257 // -----<Data adquisition> -----
258 // If intPin goes high, all qOut registers have new qOut
259 if (myIMU.readByte(MPU9250_ADDRESS, INT_STATUS) & 0x01)
260 {
261     myIMU.readAccelData(myIMU.accelCount); // Read the x/y/z adc values
262
263     // Now we'll calculate the acceleration value into actual g's
264     // This depends on scale being set
265     myIMU.ax = (float)myIMU.accelCount[0] * myIMU.aRes;
266     myIMU.ay = (float)myIMU.accelCount[1] * myIMU.aRes;
267     myIMU.az = (float)myIMU.accelCount[2] * myIMU.aRes;
268
269     myIMU.readGyroData(myIMU.gyroCount); // Read the x/y/z adc values
270
271     // Calculate the gyro value into actual degrees per second
272     // This depends on scale being set
273     myIMU.gx = (float)myIMU.gyroCount[0] * myIMU.gRes;
274     myIMU.gy = (float)myIMU.gyroCount[1] * myIMU.gRes;
275     myIMU.gz = (float)myIMU.gyroCount[2] * myIMU.gRes;

```

```

276
277 myIMU.readMagData(myIMU.magCount); // Read the x/y/z adc values
278
279 // Calculate the magnetometer values in milliGauss
280 // Include factory calibration per qOut sheet and user environmental
281 // corrections
282 // get actual magnetometer value, this depends on scale being set
283 myIMU.mx = (float)myIMU.magCount[0] * myIMU.mRes
284           * myIMU.factoryMagCalibration[0] - myIMU.magBias[0];
285 myIMU.my = (float)myIMU.magCount[1] * myIMU.mRes
286           * myIMU.factoryMagCalibration[1] - myIMU.magBias[1];
287 myIMU.mz = (float)myIMU.magCount[2] * myIMU.mRes
288           * myIMU.factoryMagCalibration[2] - myIMU.magBias[2];
289 }
290
291 // -----</Data adquisition> -----
292
293 // -----<RF qOut sending> -----
294 if(transmit)
295 {
296   data.ax = myIMU.ax;
297   data.ay = myIMU.ay;
298   data.az = myIMU.az;
299   data.gx = myIMU.gx;
300   data.gy = myIMU.gy;
301   data.gz = myIMU.gz;
302   data.mx = myIMU.mx;
303   data.my = myIMU.my;
304   data.mz = myIMU.mz;
305
306   digitalWrite(TxLED,HIGH);
307   radio.write(&data, data_size);
308
309   if (debug)
310   {
311     Serial.println("Sent: ");
312     Serial.println(data.address);
313     Serial.println(data.ax, 4);
314     Serial.println(data.ay, 4);
315     Serial.println(data.az, 4);
316     Serial.println(data.gx, 4);
317     Serial.println(data.gy, 4);
318     Serial.println(data.gz, 4);
319     Serial.println(data.mx, 4);
320     Serial.println(data.my, 4);
321     Serial.println(data.mz, 4);
322   }
323 }
324 // -----</RF qOut sending> -----
325

```

```

326 digitalWrite(TxLED,LOW);
327 }
328
329
330 // ----- Methods Definitions -----
331 void pairDevice()
332 {
333     if (deviceID == maxDevices)
334     {
335         if (debug) Serial.println("Reciever not found...");
336         digitalWrite(ONLED, LOW);
337         digitalWrite(ErrorLED, HIGH);
338         delay(100);
339         abort();
340     }
341     radio.setAutoAck(true);
342     bool pairedDevice = false;
343
344     pipe = basePipe | (uint64_t)(deviceID); // Generate 64bits pipe address
345     radio.openWritingPipe(pipe); // Open new writing pipe
346     radio.flush_tx();
347     data.address = deviceID;
348     if (debug)
349     {
350         Serial.print("Trying pipe address: "); Serial.print(deviceID, HEX);
351         Serial.println("...");
352     }
353
354     unsigned int millisCount = millis();
355     do
356     {
357         if (radio.write(&data.address, 1))
358         {
359             pairedDevice = true;
360             break;
361         }
362     } while (millis() - millisCount < 50);
363
364     deviceID++;
365     if (!pairedDevice) pairDevice();
366     else if (debug)
367     {
368         Serial.print("Connected, device ID: ");
369         Serial.print(deviceID - 1, HEX);
370         Serial.println("!!!");
371     }
372 }
373
374 void checkBattery()
375 {

```

```

376 batteryLevel = (float)analogRead(BattVoltPin)*(1.1/1024);
377 if (debug)
378 {
379     Serial.print("Battery level: ");
380     Serial.println(batteryLevel);
381 }
382 if(batteryLevel < 0.79)
383 {
384     digitalWrite(ONLED, LOW);
385     digitalWrite(TxLED, LOW);
386     digitalWrite(MagCalLED, LOW);
387     while(true)
388     {
389         digitalWrite(ErrorLED, LOW);
390         delay(1000);
391         digitalWrite(ErrorLED, HIGH);
392         delay(1000);
393     }
394 }
395 }

```

10.5.2. Transmisor de cuaterniones

En el siguiente código el transmisor está configurado para procesar los datos provenientes de la MPU-9250 aplicando el algoritmo de Madgwick y enviar los datos hacia el receptor:

```

1 /* MPU9250 Basic Example Code
2  by: Kris Winer
3  date: April 1, 2014
4  license: Beerware - Use this code however you'd like. If you
5  find it useful you can buy me a beer some time.
6  Modified by Arturo Ruiz 2018
7
8  Hardware setup:
9  MPU9250 Breakout ----- ATMEGA328P
10 VDD ----- 3.3V
11 SDA ----- 27
12 SCL ----- 28
13 GND ----- GND
14 */
15 // ----- VARIABLES -----
16 #include "quaternionFilters.h"
17 #include "MPU9250.h"
18 #include <SPI.h>
19 #include <nRF24L01.h>
20 #include <RF24.h>
21 #include <EEPROM.h>
22
23 // Control
24 bool transmit = true,

```

```

25 debug = true,
26 quaternionsNotEuler = true,
27 mahonyNotMadgwick = false;
28
29 // Digital Pins
30 #define ONLED 5
31 #define MagCalLED 6
32 #define TxLED 7
33 #define ErrorLED 8
34 #define MagCalButton 2
35 #define CE 9
36 #define CSN 10
37 #define BattVoltPin A0
38
39 // Data Structures
40 struct MagBiasScaleType
41 {
42     float bx;
43     float by;
44     float bz;
45     float sx;
46     float sy;
47     float sz;
48 }magBiasScale;
49 int magBiasScale_size = sizeof(magBiasScale);
50
51 struct DataType
52 {
53     uint8_t address;
54     float q0;
55     float q1;
56     float q2;
57     float q3;
58 }data;
59 uint8_t data_size = sizeof(DataType);
60
61
62 // Auxiliaries
63 byte byteVariable;
64 float batteryLevel;
65
66 const uint8_t maxDevices = 30;
67 uint8_t deviceID = 0;
68 const uint64_t basePipe = 0x0F0F0F0F0F0F0F0FLL; // Radio pipe addresses for the 2
        ↔ nodes to communicate.
69 uint64_t pipe;
70
71 // Object declaration
72 MPU9250 myIMU;
73 RF24 radio(CE, CSN);

```

```

74
75 // Methods
76 void pairDevice();
77 void checkBattery();
78
79 // ----- SETUP -----
80
81 void setup()
82 {
83   Serial.begin(2000000);
84   pinMode(ONLED, OUTPUT);
85   pinMode(TxLED, OUTPUT);
86   pinMode(MagCalLED, OUTPUT);
87   pinMode(ErrorLED, OUTPUT);
88
89   pinMode(MagCalButton, INPUT);
90   pinMode(BattVoltPin, INPUT);
91
92   analogReference(INTERNAL);
93
94   digitalWrite(ONLED, HIGH);
95   digitalWrite(TxLED, LOW);
96   digitalWrite(MagCalLED, LOW);
97   digitalWrite(ErrorLED, LOW);
98
99   delay(1000);
100  checkBattery();
101
102 // I2C Initialization
103  Wire.begin();
104
105  if (digitalRead(MagCalButton) == HIGH)
106  {
107 // -----<Magnetometer Calibration> -----
108     digitalWrite(ONLED, LOW);
109     digitalWrite(MagCalLED, HIGH);
110     if (debug)
111         Serial.println("Calibrating magnetometer...");
112
113     myIMU.magCalMPU9250(myIMU.magBias, myIMU.magScale);
114
115     magBiasScale.bx = myIMU.magBias[0];
116     magBiasScale.by = myIMU.magBias[1];
117     magBiasScale.bz = myIMU.magBias[2];
118
119     magBiasScale.sx = myIMU.magScale[0];
120     magBiasScale.sy = myIMU.magScale[1];
121     magBiasScale.sz = myIMU.magScale[2];
122
123     EEPROM.put(magBiasScale_size, magBiasScale);

```

```

124
125   if (debug)
126   {
127       Serial.print("Adress: "); Serial.println(byteVariable);
128       Serial.println("AK8963 mag Biases (mG)");
129       Serial.println(magBiasScale.bx);
130       Serial.println(magBiasScale.by);
131       Serial.println(magBiasScale.bz);
132
133       Serial.println("AK8963 mag Scales (mG)");
134       Serial.println(magBiasScale.sx);
135       Serial.println(magBiasScale.sy);
136       Serial.println(magBiasScale.sz);
137   }
138   digitalWrite(MagCALLEd, LOW);
139   digitalWrite(ONLED, HIGH);
140   delay(100);
141   abort();
142 // -----</Magnetometer Calibration> -----
143 }
144
145 else
146 {
147 // ----- <MPU9250 SETUP> -----
148 // Check MPU9250 connectivity
149   byteVariable = myIMU.readByte(MPU9250_ADDRESS, WHO_AM_I_MPU9250);
150   if (byteVariable != 0x71)
151   {
152       if (debug)
153       {
154           Serial.print("MPU9250-ERROR! 0x"); Serial.println(byteVariable, HEX);
155           Serial.flush();
156       }
157       digitalWrite(ErrorLED, HIGH);
158       digitalWrite(ONLED, LOW);
159       abort();
160   }
161
162 // Start by performing self test and reporting values
163   myIMU.MPU9250SelfTest(myIMU.selfTest);
164
165 // Calibrate gyro and accelerometers, load biases in bias registers
166   myIMU.calibrateMPU9250(myIMU.gyroBias, myIMU.accelBias);
167
168   myIMU.initMPU9250();
169 // Initialize device for active mode read of acclerometer, gyroscope, and
170 // temperature
171
172   if (debug) Serial.println("MPU9250 initialized...");
173

```



```

174     delay(100);
175
176 // Check AK8963(Magnetometer) connectivity
177 byteVariable = myIMU.readByte(AK8963_ADDRESS, WHO_AM_I_AK8963);
178 if (byteVariable != 0x48)
179 {
180     if (debug)
181     {
182         Serial.print("AK8963-ERROR! 0x"); Serial.println(byteVariable, HEX);
183         Serial.flush();
184     }
185     digitalWrite(ErrorLED, HIGH);
186     digitalWrite(ONLED, LOW);
187     abort();
188 }
189 // Get magnetometer calibration from AK8963 ROM
190 myIMU.initAK8963(myIMU.factoryMagCalibration);
191 // Initialize device for active mode read of magnetometer
192
193 // Get sensor resolutions, only need to do this once
194 myIMU.getAres();
195 myIMU.getGres();
196 myIMU.getMres();
197
198 EEPROM.get(magBiasScale_size, magBiasScale);
199 myIMU.magBias[0] = magBiasScale.bx;
200 myIMU.magBias[1] = magBiasScale.by;
201 myIMU.magBias[2] = magBiasScale.bz;
202
203 myIMU.magScale[0] = magBiasScale.sx;
204 myIMU.magScale[1] = magBiasScale.sy;
205 myIMU.magScale[2] = magBiasScale.sz;
206
207 if(debug)
208 {
209     Serial.print("Address: "); Serial.println(byteVariable);
210     Serial.println("AK8963 mag Biases (mG)");
211     Serial.println(magBiasScale.bx);
212     Serial.println(magBiasScale.by);
213     Serial.println(magBiasScale.bz);
214
215     Serial.println("AK8963 mag Scales (mG)");
216     Serial.println(magBiasScale.sx);
217     Serial.println(magBiasScale.sy);
218     Serial.println(magBiasScale.sz);
219
220     Serial.println("AK8963 initialized...");
221 }
222 // -----</MPU9250 SETUP> -----
223

```

```

224 // -----<NRF24L01 SETUP> -----
225 // Initialize RF
226     radio.begin();
227
228 // Set Data Rate
229     radio.setDataRate(RF24_2MBPS);
230
231 // Set amplification power level
232     radio.setPALevel(RF24_PA_MAX);
233
234 // Set channel with low noise
235     radio.setChannel(110);
236
237 // Disable retries in writing data
238     radio.setRetries(0,0);
239     pairDevice();
240     radio.setAutoAck(false);
241
242     delay(500);
243 // -----</NRF24L01 SETUP> -----
244 }
245 digitalWrite(ONLED, LOW);
246 delay(1000);
247 digitalWrite(ONLED, HIGH);
248 }
249
250 // ----- LOOP -----
251 void loop()
252 {
253     checkBattery();
254 // -----<Data acquisition> -----
255 // If intPin goes high, all qOut registers have new qOut
256 if (myIMU.readByte(MPU9250_ADDRESS, INT_STATUS) & 0x01)
257 {
258     myIMU.readAccelData(myIMU.accelCount); // Read the x/y/z adc values
259
260     // Now we'll calculate the accleration value into actual g's
261     // This depends on scale being set
262     myIMU.ax = (float)myIMU.accelCount[0] * myIMU.aRes;
263     myIMU.ay = (float)myIMU.accelCount[1] * myIMU.aRes;
264     myIMU.az = (float)myIMU.accelCount[2] * myIMU.aRes;
265
266     myIMU.readGyroData(myIMU.gyroCount); // Read the x/y/z adc values
267
268     // Calculate the gyro value into actual degrees per second
269     // This depends on scale being set
270     myIMU.gx = (float)myIMU.gyroCount[0] * myIMU.gRes;
271     myIMU.gy = (float)myIMU.gyroCount[1] * myIMU.gRes;
272     myIMU.gz = (float)myIMU.gyroCount[2] * myIMU.gRes;
273

```

```

274 myIMU.readMagData(myIMU.magCount); // Read the x/y/z adc values
275
276 // Calculate the magnetometer values in milliGauss
277 // Include factory calibration per qOut sheet and user environmental
278 // corrections
279 // get actual magnetometer value, this depends on scale being set
280 myIMU.mx = (float)myIMU.magCount[0] * myIMU.mRes
281           * myIMU.factoryMagCalibration[0] - myIMU.magBias[0];
282 myIMU.my = (float)myIMU.magCount[1] * myIMU.mRes
283           * myIMU.factoryMagCalibration[1] - myIMU.magBias[1];
284 myIMU.mz = (float)myIMU.magCount[2] * myIMU.mRes
285           * myIMU.factoryMagCalibration[2] - myIMU.magBias[2];
286 }
287
288 // -----</Data acquisition> -----
289
290 // Must be called before updating quaternions!
291 myIMU.updateTime();
292
293 // -----<AHRS algorithm> -----
294 /* Sensors x (y)-axis of the accelerometer is aligned with the y (x)-axis of
295 the magnetometer; the magnetometer z-axis (+ down) is opposite to z-axis
296 (+ up) of accelerometer and gyro! We have to make some allowance for this
297 orientation mismatch in feeding the output to the quaternion filter. For the
298 MPU-9250, we have chosen a magnetic rotation that keeps the sensor forward
299 along the x-axis just like in the LSM9DS0 sensor. This rotation can be
300 modified to allow any convenient orientation convention. This is ok by
301 aircraft orientation standards! Pass gyro rate as rad/s */
302
303 if (mahonyNotMadgwick)MahonyQuaternionUpdate(myIMU.ax, myIMU.ay, myIMU.az,
↪ myIMU.gx * DEG_TO_RAD,
304           myIMU.gy * DEG_TO_RAD, myIMU.gz * DEG_TO_RAD, myIMU.my,
305           myIMU.mx, myIMU.mz, myIMU.deltat);
306 else MadgwickQuaternionUpdate(myIMU.ax, myIMU.ay, myIMU.az, myIMU.gx *
↪ DEG_TO_RAD,
307           myIMU.gy * DEG_TO_RAD, myIMU.gz * DEG_TO_RAD, myIMU.my,
308           myIMU.mx, myIMU.mz, myIMU.deltat);
309
310 // -----<AHRS algorithm> -----
311
312 myIMU.delt_t = millis() - myIMU.count;
313
314
315 /* Define output variables from updated quaternion---these are Tait-Bryan
316 angles, commonly used in aircraft orientation. In this coordinate system,
317 the positive z-axis is down toward Earth. Yaw is the angle between Sensor
318 x-axis and Earth magnetic North (or true North if corrected for local
319 declination, looking down on the sensor positive yaw is counterclockwise.
320 Pitch is angle between sensor x-axis and Earth ground plane, toward the
321 Earth is positive, up toward the sky is negative. Roll is angle between

```

```

322 sensor y-axis and Earth ground plane, y-axis up is positive roll. These
323 arise from the definition of the homogeneous rotation matrix constructed
324 from quaternions. Tait-Bryan angles as well as Euler angles are
325 non-commutative; that is, to get the correct orientation the rotations
326 must be applied in the correct order which for this configuration is yaw,
327 pitch, and then roll.
328 For more see
329 http://en.wikipedia.org/wiki/Conversion\_between\_quaternions\_and\_Euler\_angles
330 which has additional links.*/
331
332 if (!quaternionsNotEuler)
333 {
334   myIMU.yaw = atan2(2.0f * (*(getQ()+1) * *(getQ()+2) + *getQ()
335                       *(getQ()+3)), *getQ() * *getQ() + *(getQ()+1)
336                       * *(getQ()+1) - *(getQ()+2) * *(getQ()+2) - *(getQ()+3)
337                       * *(getQ()+3));
338   myIMU.pitch = -asin(2.0f * (*(getQ()+1) * *(getQ()+3) - *getQ()
339                              * *(getQ()+2)));
340   myIMU.roll = atan2(2.0f * (*getQ() * *(getQ()+1) + *(getQ()+2)
341                             * *(getQ()+3)), *getQ() * *getQ() - *(getQ()+1)
342                             * *(getQ()+1) - *(getQ()+2) * *(getQ()+2) + *(getQ()+3)
343                             * *(getQ()+3));
344
345   // Declination and Inclination components
346   // of CIA, Anexo de Ingeniería, Ciudad Universitaria, Ciudad de México
347   // ( 19.3263°N, -99.1821°W) on 2018-05-08
348   // is Declination: 4.629°
349   // - http://geomag.nrcan.gc.ca/calc/mfcal-r-en.php?date=2018-05-08&latitude
350     ↪ =19.326335&latitude_direction=1&longitude=99.182103&
351     ↪ longitude_direction=-1
352   myIMU.yaw -= 4.629*PI/180;
353
354   if (debug)
355   {
356     Serial.print(myIMU.yaw, 2); Serial.print("/");
357     Serial.print(myIMU.pitch, 2); Serial.print("/");
358     Serial.println(myIMU.roll, 2);
359   }
360 }
361
362 /* With these settings the filter is updating at a ~145 Hz rate using the
363 Madgwick scheme and >200 Hz using the Mahony scheme even though the
364 display refreshes at only 2 Hz. The filter update rate is determined
365 mostly by the mathematical steps in the respective algorithms, the
366 processor speed (8 MHz for the 3.3V Pro Mini), and the magnetometer ODR:
367 an ODR of 10 Hz for the magnetometer produce the above rates, maximum
368 magnetometer ODR of 100 Hz produces filter update rates of 36 - 145 and
369 ~38 Hz for the Madgwick and Mahony schemes, respectively. This is
370 presumably because the magnetometer read takes longer than the gyro or
371 accelerometer reads. This filter update rate should be fast enough to
372 maintain accurate platform orientation for stabilization control of a

```

```

370     fast-moving robot or quadcopter. Compare to the update rate of 200 Hz
371     produced by the on-board Digital Motion Processor of Invensense's MPU6050
372     6 DoF and MPU9150 9DoF sensors. The 3.3 V 8 MHz Pro Mini is doing pretty
373     well! */
374
375 // -----<RF qOut sending> -----
376 if(transmit)
377 {
378     if (quaternionsNotEuler)
379     {
380         data.q0 = *getQ();
381         data.q1 = *(getQ()+1);
382         data.q2 = *(getQ()+2);
383         data.q3 = *(getQ()+3);
384
385         digitalWrite(TxLED,HIGH);
386         radio.write(&data, data_size);
387
388         if (debug)
389         {
390             Serial.println("Sent: ");
391             Serial.println(data.address);
392             Serial.println(data.q0, 4);
393             Serial.println(data.q1, 4);
394             Serial.println(data.q2, 4);
395             Serial.println(data.q3, 4);
396         }
397     }
398 }
399 // -----</RF qOut sending> -----
400
401 digitalWrite(TxLED,LOW);
402 myIMU.count = millis();
403 myIMU.sumCount = 0;
404 myIMU.sum = 0;
405 }
406
407 // ----- Methods Definitions -----
408 void pairDevice()
409 {
410     if (deviceID == maxDevices)
411     {
412         if (debug) Serial.println("Reciever not found...");
413         digitalWrite(ONLED, LOW);
414         digitalWrite(ErrorLED, HIGH);
415         delay(100);
416         abort();
417     }
418     radio.setAutoAck(true);
419     bool pairedDevice = false;

```

```

420
421 pipe = basePipe | (uint64_t)(deviceID); // Generate 64bits pipe address
422 radio.openWritingPipe(pipe); // Open new writing pipe
423 radio.flush_tx();
424 data.address = deviceID;
425 if (debug)
426 {
427     Serial.print("Trying pipe address: "); Serial.print(deviceID, HEX);
428     Serial.println("...");
429 }
430
431 unsigned int millisCount = millis();
432 do
433 {
434     if (radio.write(&data.address, 1))
435     {
436         pairedDevice = true;
437         break;
438     }
439 } while (millis() - millisCount < 100);
440
441 deviceID++;
442 if (!pairedDevice) pairDevice();
443 else if (debug)
444 {
445     Serial.print("Connected, device ID: ");
446     Serial.print(deviceID - 1, HEX);
447     Serial.println("!!!");
448 }
449 }
450
451 void checkBattery()
452 {
453     batteryLevel = (float)analogRead(BattVoltPin)*(1.1/1024);
454     if (debug)
455     {
456         Serial.print("Battery level: ");
457         Serial.println(batteryLevel);
458     }
459     if(batteryLevel < 0.79)
460     {
461         digitalWrite(ONLED, LOW);
462         digitalWrite(TxLED, LOW);
463         digitalWrite(MagCalLED, LOW);
464         while(true)
465         {
466             digitalWrite(ErrorLED, LOW);
467             delay(1000);
468             digitalWrite(ErrorLED, HIGH);
469             delay(1000);

```

```
470 }
471 }
472 }
```

10.6. Código de Interfaz de Usuario y Sistema de Conversión de Datos del Receptor

El siguiente código corresponde a la aplicación de consola desarrollada en Visual Studio 2017, que le brinda al usuario la capacidad de seleccionar el número de dispositivos a vincular con el receptor así como iniciar y detener la recepción de datos; de igual manera funciona como el sistema de conversión y almacenamiento de datos

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.IO.Ports;
5 using System.Linq;
6 using System.Text;
7 using System.Threading;
8 using System.Threading.Tasks;
9
10 namespace RawSerial_MPU9250
11 {
12     class Program
13     {
14         private static object cw;
15
16         static void Main(string[] args)
17         {
18             bool restart = true;
19
20             while (restart)
21             {
22                 Console.Clear();
23                 bool fixedValues = true;
24                 string fixedPortName = "COM4";
25                 int fixedDevices = 2;
26
27                 const int BaudRate = 2000000;
28                 string desktop = Environment.GetFolderPath(Environment.
29                     ↪ SpecialFolder.Desktop);
30                 string[] portNames;
31                 string portName = null;
32                 int devices = 2;
33                 if (!fixedValues)
34                 {
35                     while (portName == null)
36                     {
37                         portNames = SerialPort.GetPortNames().ToList().Distinct().
```

```

37         ↪ toArray();
38     Console.WriteLine("Select the port to connect: ");
39     for (int i = 0; i < portNames.Length; i++)
40         Console.WriteLine("{0}: {1}", i, portNames[i]);
41     try
42     {
43         int index = Convert.ToInt16(Console.ReadLine());
44         portName = portNames[index];
45     }
46     catch
47     {
48         Console.WriteLine("Invalid selection");
49     }
50 }
51 else
52 {
53     portName = fixedPortName;
54     Console.WriteLine("Port: " + portName);
55 }
56 SerialPort port = new SerialPort(portName, BaudRate);
57 port.DataReceived += new SerialDataReceivedEventHandler(
58     ↪ stringRecieved);
59 port.DtrEnable = true;
60 port.Open();
61 port.DiscardInBuffer();
62 bool user = true;
63 int dataSize = 21;
64 //Console.WriteLine("Buffersize: {0}", port.ReadBufferSize);
65 while (user) { }
66 List<int> inBytes = new List<int>();
67 byte[] data = new byte[dataSize];
68 bool stop = false;
69 Thread RecieveThread = new Thread(RecieveData);
70 RecieveThread.Start();
71 Console.WriteLine("Tap spacebar to stop reading...");
72 while (!stop) stop = Console.ReadKey(true).Key == ConsoleKey.
73     ↪ Spacebar;
74 Console.WriteLine("Reading Stopped. ");
75 while (stop) { }
76
77 Console.WriteLine("File's Name: ");
78 var fileName = Console.ReadLine();
79 File.Delete(desktop + @"\Data.txt");
80 using (StreamWriter sw = File.AppendText(desktop + @"\Data.txt"))
81 {
82     float q, t;
83     string[] bytesRecieved = File.ReadAllLines(desktop + @"\test.
84         ↪ txt");
85     for (int i = 0; i < bytesRecieved.Length; i += dataSize)

```



```

83     {
84         if (i + dataSize - 1 >= bytesRecieved.Length) break;
85
86         // Device ID
87         sw.WriteLine("Device: " + bytesRecieved[i]);
88         // Q1, Q2, Q3, Q4
89         for (int j = 0; j < 4; j++)
90         {
91             q = BitConverter.ToSingle(new byte[]
92             {
93                 Convert.ToByte(bytesRecieved[i + j * 4 + 1]),
94                 Convert.ToByte(bytesRecieved[i + j * 4 + 2]),
95                 Convert.ToByte(bytesRecieved[i + j * 4 + 3]),
96                 Convert.ToByte(bytesRecieved[i + j * 4 + 4])
97             }, 0);
98             sw.WriteLine("q" + (j + 1).ToString() + ": " + q);
99         }
100        t = BitConverter.ToSingle(new byte[]
101        {
102            Convert.ToByte(bytesRecieved[i + 17]),
103            Convert.ToByte(bytesRecieved[i + 18]),
104            Convert.ToByte(bytesRecieved[i + 19]),
105            Convert.ToByte(bytesRecieved[i + 20])
106        }, 0);
107        sw.WriteLine("Time: " + t);
108    }
109    sw.Close();
110 }
111
112 Console.WriteLine("Writing CSV File...");
113 WriteCSV();
114 Console.WriteLine("Done. Restart? Y/N");
115 restart = (Console.ReadKey(true).Key == ConsoleKey.Y);
116
117
118
119 //Methods
120 void stringRecieved(object s, SerialDataReceivedEventArgs e)
121 {
122     var str = port.ReadExisting();
123     Console.WriteLine(str);
124     if (str.Contains("connect:"))
125     {
126         if (!fixedValues) int.TryParse(Console.ReadLine(), out
127             ↪ devices);
128         else devices = fixedDevices;
129
130         port.WriteLine(devices.ToString());
131     }

```

```

132
133         else if (str.Contains("Now"))
134         {
135             user = false;
136             port.DataReceived -= stringRecieved;
137         }
138     }
139     void RecieveData()
140     {
141         byte[] b = new byte[dataSize];
142         File.Delete(desktop + @"\test.txt");
143         using (StreamWriter sw = File.AppendText(desktop + @"\test.txt
144             ↪ "))
145         {
146             while (!stop) sw.WriteLine(port.ReadByte());
147             //for (int i = port.BytesToRead - 1; i >= dataSize; i--)
148             // sw.WriteLine(port.ReadByte());
149
150             sw.Close();
151             port.Close();
152         }
153         stop = false;
154     }
155     void WriteCSV()
156     {
157         File.Delete(desktop + @"\ " + fileName + ".csv");
158         StringBuilder content = new StringBuilder();
159         string columns = "";
160         for (int i = 1; i <= devices; i++)
161             columns += "t_" + i + ",q1_" + i + ",q2_" + i + ",q3_" + i
162             ↪ + ",q4_" + i + ", ";
163
164         content.AppendLine(columns.Substring(0, columns.Length - 1));
165         string[] lines = File.ReadAllLines(desktop + @"\Data.txt");
166         for (int i = 0; i < lines.Count(); i += (6 * devices))
167         {
168             columns = "";
169             try
170             {
171                 for (int j = 0; j < devices; j++)
172                 {
173                     columns += lines[i + 5 + 6 * j].Substring(6, lines[
174                         ↪ i + 5 + 6 * j].Length - 6) + ", ";
175                     for (int k = 1; k <= 4; k++)
176                         columns += lines[i + k + 6 * j].Substring(4,
177                             ↪ lines[i + k + 6 * j].Length - 4) + ", ";
178                 }
179
180                 content.AppendLine(columns.Substring(0, columns.Length

```

```

178         ↪ - 1));
179     }
180     catch { }
181     }
182     File.AppendAllText(desktop + @"\\" + fileName + ".csv", content
183         ↪ .ToString());
184
185     File.Delete(desktop + @"\Data.txt");
186     File.Delete(desktop + @"\test.txt");
187 }
188 }
189 }
190 }

```

10.7. Código de Post-procesamiento

A continuación se muestra el código desarrollado en MATLAB, utilizado para procesar los datos recabados de las pruebas realizadas.

```

1 %% Proposed System
2 % Get the file
3 close all; clear all; clc;
4 file = input('File number: ', 's');
5
6 M = csvread(strcat(file, '_A.csv'), 1, 0);
7 measures = size(M,1);
8
9 % Get time vector
10 t_A = (M(:,1) + M(:,6))/2;
11
12 % Axes' calibration during static period
13 p = input('Person number: ');
14 f1 = figure;
15 subplot(1,2,1);
16 plot(M(:,1), M(:,2), M(:,1), M(:,3), M(:,1), M(:,4), M(:,1), M(:,5)); grid on;
17 legend('q1', 'q2', 'q3', 'q4');
18 xlabel('tiempo [s]'); ylabel('valores del cuaternion []');
19 title(strcat(strcat('P', num2str(p)), '_ Muslo'));
20 subplot(1,2,2);
21 plot(M(:,6), M(:,7), M(:,6), M(:,8), M(:,6), M(:,9), M(:,6), M(:,10)); grid on;
22 legend('q1', 'q2', 'q3', 'q4');
23 xlabel('tiempo [s]'); ylabel('valores del cuaternion []');
24 title(strcat(strcat('P', num2str(p)), '_ Pierna'));
25 t1 = input('Time 1 for static q offset: '); t2 = input('Time 2 for static q
26     ↪ offset: ');
27
28 for i = 1:2
29     n = 0; q4_off = n; q3_off = q4_off; q2_off = q3_off; q1_off = q2_off;

```

```

29     for j = 1:measures/4
30         if M(j,(i-1)*5 + 1) >= t1 && M(j,(i-1)*5 + 1) <= t2
31             q1_off = q1_off + M(j,(i-1)*5 + 2);
32             q2_off = q2_off + M(j,(i-1)*5 + 3);
33             q3_off = q3_off + M(j,(i-1)*5 + 4);
34             q4_off = q4_off + M(j,(i-1)*5 + 5);
35             n = n + 1;
36         end
37     end
38     q_off(i,:) = [q1_off/n, q2_off/n, q3_off/n, q4_off/n];
39 end
40 clear n q1_off q2_off q3_off q4_off t1 t2
41
42 % Get quaternions from CSV file with offset postmultiplied
43 for i=1:2
44     for j=1:measures
45         q = [M(j,(i-1)*5 + 2), M(j,(i-1)*5 + 3), M(j,(i-1)*5 + 4), M(j,(i-1)*5 +
46             ↪ 5)];
47         Q(j, :, i) = qcross(q, qconj(q_off(i,:)));
48     end
49 end
50 % Get axis' dynamics
51 for j=1:measures
52     q = Q(j, :, 1);
53     % X1
54     var = qcross(q,qcross([0, 1, 0, 0], qconj(q)));
55     X(j,:,1) = [var(2:4)];
56     % Y1
57     var = qcross(q,qcross([0, 0, 1, 0], qconj(q)));
58     Y(j,:,1) = [var(2:4)];
59     % Z1
60     var = qcross(q,qcross([0, 0, 0, 1], qconj(q)));
61     Z(j,:,1) = [var(2:4)];
62
63     q = Q(j, :, 2);
64     % X2
65     var = qcross(q,qcross([0, 1, 0, 0], qconj(q)));
66     X(j,:,2) = [var(2:4)];
67     % Y2
68     var = qcross(q,qcross([0, 0, 1, 0], qconj(q)));
69     Y(j,:,2) = [var(2:4)];
70     % Z2
71     var = qcross(q,qcross([0, 0, 0, 1], qconj(q)));
72     Z(j,:,2) = [var(2:4)];
73 end
74
75 clear i j var q Eu11 Eu12 RotM
76
77 % Compute relative angles

```

```

78 for i = 1:measures
79     A_A(i, :) = [rad2deg(acos(dot([X(i, 1, 1), X(i, 2, 1), X(i, 3, 1)], [X(i, 1,
    ↪ 2), X(i, 2, 2), X(i, 3, 2)]))),...
80         rad2deg(acos(dot([Y(i, 1, 1), Y(i, 2, 1), Y(i, 3, 1)], [Y(i, 1, 2),
    ↪ Y(i, 2, 2), Y(i, 3, 2)]))),...
81         rad2deg(acos(dot([Z(i, 1, 1), Z(i, 2, 1), Z(i, 3, 1)], [Z(i, 1, 2),
    ↪ Z(i, 2, 2), Z(i, 3, 2)])))]);
82 end
83
84 % Plot relative angles' dynamics
85 f2 = figure;
86 plot (A_A(:, 1)); hold on;
87 plot (A_A(:, 2));
88 grid on;
89 legend('Ejes X', 'Ejes Y');
90 xlabel('no. de lectura []'); ylabel('ángulo relativo [ř]');
91 title(strcat('Sistema Propuesto: Ángulo de flexoextensión de rodilla' + " ",
    ↪ strcat('P', num2str(p))));
92
93
94 % Ask for corresponding timing
95 peak_f_A = input('Element in which first peak ocurred: ');
96 peak_l_A = input('Element in which last peak ocurred: ');
97 Apeaks_A = A_A([peak_f_A:peak_l_A], :);
98 tpeaks_A = t_A([peak_f_A:peak_l_A]);
99 tpeaks_A = tpeaks_A - tpeaks_A(1);
100
101
102 %% Shadow System
103
104 M = csvread(strcat(file, '.csv'), 1, 0);
105 measures = size(M,1);
106 seconds = input('Measurement time (s): ');
107 t_S = linspace(0, seconds, measures)';
108
109 % Compute Z-Axis' orientation dynamics
110
111 % R Thight RotXYZ columns: 46, 47, 48
112 % R Leg RotXYZ columns: 52, 53, 54
113 Eu1 = M(:, [46, 47, 48]).*(pi/180); Eu2 = M(:, [52, 53, 54]).*(pi/180);
114 RotM = [eul2rotm(Eu1, 'XYZ'), eul2rotm(Eu2, 'XYZ')];
115
116 for i = 1:2
117     c = i - 1;
118     for j = 1:measures
119         % Z
120         var = RotM(:, [3*c + 1, 3*c + 2, 3*c + 3], j)*[0;0;1];
121         Z(j, :, i) = var;
122     end
123 end

```

```

124
125 clear l i j c
126
127 % Compute relative angles
128 for i = 1:measures
129     A_S(i,1) = rad2deg(acos(dot([Z(i,1,1) Z(i,2,1) Z(i,3,1)], [Z(i,1,2) Z(i,2,2)
130         ↪ Z(i,3,2)])));
131 end
132 % Plot relative angles' dynamics
133 f3 = figure;
134 plot(A_S);
135 grid on;
136 xlabel('no. de lectura []'); ylabel('ángulo relativo [ř]');
137 title(strcat('Sistema Shadow: Ángulo de flexoextensión de rodilla' + " ",strcat(
138     ↪ 'P', num2str(p))));
139 % Static offset computation
140 t1 = input('Time 1 element for static offset: '); t2 = input('Time 2 element for
141     ↪ static offset: ');
142 n = 0; a_off = n;
143 for j = t1:t2
144     a_off = a_off + A_S(j);
145     n = n + 1;
146 end
147 a_off = a_off/n;
148 A_S = A_S - a_off;
149 plot(A_S);
150 hold on
151 grid on;
152 xlabel('no. de lectura []'); ylabel('ángulo relativo [ř]');
153 title(strcat('Sistema Shadow: Ángulo de flexoextensión de rodilla' + " ",strcat(
154     ↪ 'P', num2str(p))));
155 clear a_off var j
156
157 % Ask for corresponding timing
158
159 peak_f_S = input('Element in which first peak occurred: ');
160 peak_l_S = input('Element in which last peak occurred: ');
161
162
163 %% Resize Shadow System vector's length to Proposed System vector's length
164 scale = (peak_l_A - peak_f_A + 1)/(peak_l_S - peak_f_S);
165 Apeaks_S = resizem(diag(A_S([peak_f_S:peak_l_S], :)), scale, 'nearest');
166 Apeaks_S = diag(Apeaks_S);
167 if (length(Apeaks_A) ~= length(Apeaks_S))
168     scale = (peak_l_A - peak_f_A)/(peak_l_S - peak_f_S);
169     Apeaks_S = resizem(diag(A_S([peak_f_S:peak_l_S], :)), scale, 'nearest');

```

```

170     Apeaks_S = diag(Apeaks_S);
171 end
172
173 % Compute correlation coefficients
174 Xcorr = corrcoef(Apeaks_A(:,1), Apeaks_S);
175 Xcorr = Xcorr(1,2)
176 Ycorr = corrcoef(Apeaks_A(:,2), Apeaks_S);
177 Ycorr = Ycorr(1,2)
178
179 % Show both graphs
180 f4 = figure;
181 plot(tpeaks_A, Apeaks_S, tpeaks_A,Apeaks_A(:,1), tpeaks_A,Apeaks_A(:,2));
182 legend('S.Shadow (Z)', 'S.P.G. (X)', 'S.P.G. (Y)');
183 xlabel('tiempo [s]');
184 ylabel('Ángulo [ř]');
185 title(strcat('Ángulo de flexoextensión (Picos ) P', num2str(p)));
186 grid on;
187
188 %% Statistics (Mean & Std Deviation)
189 Mean_S = mean(Apeaks_S)
190 Mean_AX = mean(Apeaks_A(:,1))
191 Mean_AY = mean(Apeaks_A(:,2))
192 StD_S = std(Apeaks_S)
193 StD_AX = std(Apeaks_A(:,1))
194 StD_AY = std(Apeaks_A(:,2))
195
196 %% Difference between X and Y signals of Proposed System
197 DiffPeaks_A = abs(Apeaks_A(:,1) - Apeaks_A(:,2));
198 f5 = figure;
199 plot(tpeaks_A, DiffPeaks_A);
200 xlabel('tiempo [s]');
201 ylabel('Ángulo [ř]');
202 title(strcat('Diferencia absoluta A.F.E. Ejes X y Ejes Y del S.G.P.(Picos ) P',
    ↪ num2str(p)));
203 grid on;

```

10.8. Carta de Consentimiento Informado

A continuación se muestra el formato utilizado en la elaboración de la carta de consentimiento informado entregada a los sujetos de prueba con la finalidad de hacerles saber los procedimientos a realizar, beneficios y riesgos potenciales: [41]

Universidad Nacional Autónoma de México

Facultad de Ingeniería

CARTA DE CONSENTIMIENTO INFORMADO

<p>Título del Proyecto: Diseño de un Sistema Inalámbrico de Goniometría para Análisis de Marcha Humana</p>

Estimado(a) Señor/Señora:

Introducción/Objetivo:

La Facultad de Ingeniería de la Universidad Nacional Autónoma de México realiza un proyecto de investigación cuyo objetivo es el diseño y construcción de un sistema de goniometría inalámbrico cuyo enfoque es el análisis de la marcha humana; es decir un sistema que brinda la capacidad al usuario de analizar a detalle los desplazamientos angulares en las articulaciones de la parte inferior del cuerpo humano durante el caminar de una persona. El estudio se realiza dentro de las instalaciones de la Facultad de Ingeniería

Procedimientos:

Si Usted acepta participar en el estudio, ocurrirá lo siguiente:

Se requerirá que se presente con pantalones cortos y ajustados con la finalidad de lograr una óptima sujeción de los sensores a su cadera, piernas y pies.

Las pruebas se dividen en dos módulos: *Antropometría* y *Prueba de Marcha*.

Antropometría: Para este módulo se le realizarán mediciones con cinta métrica en muslo y pierna y se tomará registro de dichas medidas. Posteriormente se le colocarán etiquetas adhesivas a modo de marcadores visuales.

Prueba de marcha: Para este módulo se le colocarán sensores alámbricos e inalámbricos en cadera, muslos, piernas y pies. Posteriormente, se le solicitará que camine de manera natural sobre un camino trazado en el piso de ida y vuelta. Durante estas pruebas se recibirá de forma continua los datos arrojados por los sensores.

El finalizar ambos módulos se le retirarán los sensores, finalizando en su totalidad las pruebas programadas.

En caso de ser requerido nuevamente, los investigadores se pondrán en contacto con usted.

Beneficios:

Usted no recibirá un beneficio directo por su participación en el estudio, sin embargo, si acepta participar, estará colaborando con la Universidad Nacional Autónoma de México para el registro y obtención de datos para la validación de un sistema biomecatrónico innovador, posible desarrollo del mismo y demás investigaciones futuras.

Confidencialidad:

Toda la información que nos proporcione será de carácter estrictamente confidencial, y utilizada únicamente por el equipo de investigación del proyecto y no estará disponible para ningún otro propósito. Usted será identificado(a) mediante un número y no con su nombre. Los resultados de este estudio serán publicados con fines científicos, pero se presentarán de tal manera, que no podrá ser identificado(a).

De igual manera, al aceptar participar en el estudio, toda información y procedimiento realizado durante la investigación no deberá ser divulgada.

Riesgos potenciales/Compensación:

Los riesgos potenciales que implican su participación en este estudio son mínimos. Sin embargo, podría llegar sentir incomodidad por ajuste excesivo de las correas en sus piernas y/o cadera al momento de la colocación de los sensores. De igual manera, si se llegaran a colocar de manera incorrecta los sensores que se sujetan mediante broches, usted podría ser accidentalmente pellizcado por éstos.

Usted no recibirá ningún pago por participar en el estudio, y tampoco implicará algún costo para usted.

Participación Voluntaria/ Retiro:

La participación en este estudio es absolutamente voluntaria. Usted está en plena libertad de negarse a participar o de retirar su participación del mismo en cualquier momento. Su decisión de participar o de no participar no afectara de ninguna manera la forma en como le traten dentro de la universidad.

Contacto:

Si tiene alguna pregunta, comentario o preocupación con respecto al proyecto, hágasela saber al investigador a cargo del estudio.

Si acepta participar en el estudio le entregaremos una copia de este documento que le pedimos sea tan amable de firmar.

Consentimiento para su participación en el estudio

Su firma indica su aceptación para participar voluntariamente en el presente estudio.

Nombre del participante:

Fecha:

(Día / Mes / Año)

Firma: _____

Nombre del Testigo:

Fecha:

(Día / Mes / Año)

Firma: _____

Relación con el participante _____

Nombre de quien obtiene el consentimiento:

Fecha:

(Día / Mes / Año)

Firma: _____

10.9. Datos antropométricos recolectados durante las pruebas

En el siguiente documento se muestran los datos antropométricos de muslo y pierna derechas de los sujetos de estudio que participaron en las pruebas realizadas en el presente proyecto.

Datos Antropométricos

Sujeto #1	Longitud Pierna Derecha [cm]
Mulso	39
Pierna	37

Sujeto #2	Longitud Pierna Derecha [cm]
Mulso	40
Pierna	40

Sujeto #3	Longitud Pierna Derecha [cm]
Mulso	42
Pierna	41

Sujeto #4	Longitud Pierna Derecha [cm]
Mulso	38
Pierna	37

Sujeto #5	Longitud Pierna Derecha [cm]
Mulso	46
Pierna	42

Sujeto #6	Longitud Pierna Derecha [cm]
Mulso	48
Pierna	44

Bibliografía

- [1] H. Ortega Carrillo y J. Zavage Carmona. «Desarrollo de un sistema de captura de movimiento para el cuerpo humano». TESIUNAM: Universidad Nacional Autónoma de México, 2001.
- [2] R. Queirós. *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming*. Advances in higher education and professional development (AHEPD) book series. IGI Global, 2014. ISBN: 9781466673052. URL: <https://books.google.com.mx/books?id=A6WXBgAAQBAJ>.
- [3] C. Canchola Hernández. «Diseño de una órtesis de rodilla compensatoria para el desgaste de cartílago desde un enfoque biomecatrónico». TESIUNAM: Universidad Nacional Autónoma de México, 2017.
- [4] G.D. Mura. «Piezoresistive Goniometer Network for Sensing Gloves». En: *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013* (2014).
- [5] K. Kalgaonkar y B. Raj. «One-handed gesture recognition using ultrasonic Doppler sonar». En: *IEEE International Conference on Acoustics, Speech and Signal Processing* (2009).
- [6] Bogert van den, A. J. Read y B. M. L. Nigg. «A method for inverse dynamic analysis using accelerometry». En: *Journal of Biomechanics* (1996).
- [7] R. E. Mayagoitia, A. V. Nene y P. H. Veltink. «Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems». En: *Journal of Biomechanics* (2002), págs. 537-542.
- [8] H. J. Luinge y P. H. Veltink. «Measuring orientation of human body segments using miniature gyroscopes and accelerometers». En: *Medical & Biological Engineering & Computing* (2005), págs. 273-282.
- [9] S. J. Morris, A. Y. Bnebasat, Scarborough D., D. E. Krebs y J. A. Paradiso. «Gait Analysis Using a Shoe-Integrated Wireless Sensor System». En: *IEEE Transactions on Information Technology in Biomedicine* (2008), págs. 413-423.
- [10] M. Brodie, A Walmsley y W. Page. «Fusion motion capture: a prototype system using inertial measurement units and GPS for biomechanical analysis of ski racing». En: *Sports Technology* (2008).
- [11] S. et al Takeda R; Tadano. «Gait posture estimation using wearable acceleration and gyro sensors». En: *Sports Technology* (2008).
- [12] H. M. Herr, J. A. Weber, S. K. Au, B. Wayne, L. H. Magnusson, A. G. Hoffmann y B. B. Alsen. «Powered ankle-foot prosthesis». US8512415B2. Ago. de 2013.
- [13] G. Campiglio, J. Mazzeo y S. Rodriguez. «A wireless goniometry system». En: *Journal of Physics* (2013).
- [14] R. Bailey G. P.; Harle. «Assessment of Foot Kinematics During Steady State Running Using a Foot-Mounted IMU». En: *Elsevier. Procedia Engineering* (2014).

- [15] H. Roisin. «Wireless Sensor Devices in Sport Performance». En: *IEEE Potentials* (2016), págs. 40-42.
- [16] Ruoyu Zhi. «A Drift Eliminated Attitude & Position Estimation Algorithm In 3D». University of Vermont, 2016.
- [17] D. Nedelkovski. *MEMS Accelerometer Gyroscope Magnetometer & Arduino*. 2016. URL: <https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>.
- [18] D. P. Vigouroux Cavolina. «Implementación de Unidad de Mediciones Inerciales (IMU) para Robótica Utilizando Filtro Kalman». Universidad Simón Bolívar, 2010.
- [19] A. J. L. Harrison y R. Vaidyanathan S. O. H. Madgwick. «Estimation of IMU and MARG orientation using a gradient descent algorithm». En: *IEEE International Conference on Rehabilitation Robotics* (2011).
- [20] J.J. Craig. *Robótica*. Ed. por Pearson. 3a. 2006.
- [21] Adrián J. Elías. «Cinemática del Renate de Bádminton». Universidad Nacional Autónoma de México, 2014.
- [22] *Euler (gimbal lock) Explained*. The Guerrilla CG Project. Ene. de 2009. URL: <https://www.youtube.com/watch?v=zc8b2Jo7mno>.
- [23] Vernon Chi. «Quaternions and Rotations in 3-Space: How it Works». En: (1998).
- [24] J.B. Kuipers. «Quaternions and Rotation Sequences». En: (1999).
- [25] L. Tran. «Data Fusion with 9 Degrees of Freedom Inertial Measurement Unit To Determine Object's Orientation». California Polytechnic State University, Electrical Engineering Department, 2017.
- [26] Christopher Konvalin. *Compensating for Tilt, Hard-Iron, and Soft-Iron Effects*. Dic. de 2009. URL: <https://www.sensorsmag.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects>.
- [27] Sparkfun Electronics. *SPARKFUN IMU BREAKOUT - MPU-9250*. URL: <https://www.sparkfun.com/products/13762>.
- [28] Mike Jacobs. *Building an Arduino-based self-balancing robot Part 1*. Abr. de 2015. URL: <https://roboticdreams.wordpress.com/2015/04/24/building-an-arduino-based-self-balancing-robot-part-1/>.
- [29] Duane Knudson. *Fundamentals of Biomechanics*. Ed. por Springer. 2a. 2007.
- [30] Peterson D. y Bronzino J. *Biomechanics Principles and Practices*. Ed. por CRC Press. 2015.
- [31] Tatora G. y Derrickson B. *Principios de anatomía y fisiología*. Ed. por Panamericana. 13a. 2015.
- [32] Mihalko W. *et al.* «Controversies and Techniques in the Surgical Management of Patellofemoral Arthritis». En: *Journal of Bone & Joint Surgery* (jul. de 2010), pág. 2799.
- [33] *MPU-9250 Product Specification*. PS-MPU-9250A-01. Revision 1.0. Invensense Inc. 2014.
- [34] Nordic Semiconductor. *nRF24L01*. URL: <https://www.nordicsemi.com/eng/Products/2.4GHZ-RF/nRF24L01>.
- [35] TMRh20. *Optimized High Speed NRF24L01+ Driver Class Documentation*. URL: <http://tmrh20.github.io/RF24/>.
- [36] *8-bit AVR Microcontrollers ATmega328/P DATASHEET COMPLETE*. Atmel. 2016.

- [37] Christopher L Vaughan, Brian Davis y Jeremy C O' Connor. «Dynamics of Human Gait». En: (dic. de 2018).
- [38] MathWorks. *Documentación: corrcoef*. URL: <https://la.mathworks.com/help/matlab/ref/corrcoef.html#bunkanr>.
- [39] Joseph M. Cooke, Michael J. Zyda, David R. Pratt y Robert B. McGhee. «NPSNET: Flight Simulation Dynamic Modeling Using Quaternions». En: *Presence: Teleoperators and Virtual Environments* 1.4 (1992), págs. 404-420. DOI: 10.1162/pres.1992.1.4.404.
- [40] R. Mahony, T Hamel y M. Pflimlin. «Nonlinear complementary filters on the special orthogonal group». En: *Automatic Control, IEEE Transactions* (2008), págs. 1203-1218.
- [41] INSTITUTO NACIONAL DE SALUD PÚBLICA. *CARTA DE CONSENTIMIENTO ADULTOS*. URL: https://www.insp.mx/images/stories/comitesEvaluacion/comitEtica/docs/Anexo%5C%20A_Carta%5C%20Escrito%5C%20adultos.pdf.