



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
ELÉCTRICA – INSTRUMENTACIÓN

**SISTEMA DE ESTIMACIÓN DE ORIENTACIÓN BASADO EN SENSORES
INERCIALES PARA APLICACIONES EN EVALUACIÓN DE MOVIMIENTOS DEL
CUERPO HUMANO**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRA EN INGENIERÍA

PRESENTA:
ADRIANA MARTÍNEZ HERNÁNDEZ

TUTORES PRINCIPALES
DR. ASUR GUADARRAMA SANTANA, CCADET
DR. MIGUEL ÁNGEL PADILLA CASTAÑEDA, CCADET

CIUDAD UNIVERSITARIA, CD. MX.

ENERO 2017

JURADO ASIGNADO:

Presidente: Dr. Prado Molina Jorge

Secretario: Dra. Velasco Herrera Graciela

Vocal: Dr. Guadarrama Santana Asur

1 er. Suplente: Dr. Padilla Castañeda Miguel Ángel

2 do. Suplente: Dr. Gutiérrez Herrera Enoch

Ciudad Universitaria, Centro de Ciencias Aplicadas y Desarrollo Tecnológico, Programa de Maestría y Doctorado en Ingeniería Eléctrica, UNAM

TUTOR DE TESIS:

Dr. Guadarrama Santana Asur



FIRMA

*El presente trabajo de tesis fue desarrollado
en el Centro de Ciencias Aplicadas y
Desarrollo Tecnológico
de la Universidad Nacional Autónoma
de México, bajo la tutoría de los
Dres. Asur Guadarrama Santana y
Miguel Ángel Padilla Castañeda
con beca CONACYT 632254*

DEDICATORIA

*A mis padres: Araceli Hernández y Gabriel Martínez,
por todo su amor y apoyo... gracias, porque sin
ustedes nada de esto sería posible. Los amo.*

AGRADECIMIENTOS

Agradezco sinceramente a:

La Universidad Nacional Autónoma de México por brindarme la posibilidad de realizar mis estudios de posgrado.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico que me brindó durante mis estudios de maestría.

Al Dr. Miguel Ángel Padilla por invitarme a participar en este proyecto, guiándome y apoyándome durante todo el proceso.

Al Dr. Asur Guadarrama por el apoyo brindado para que este proyecto pudiera desarrollarse y por todos sus consejos.

A los sinodales por su tiempo para revisar este trabajo.

Al Ing. Carlos Sánchez Corona por su asesoría en el diseño del sistema de calibración.

A mis padres por su paciencia, apoyo y amor en todo momento.

A Montse, mi hermana, por acompañarme en mis noches de desvelo haciéndolas más livianas, te quiero.

A todos mis amigos que estuvieron siempre al pendiente y alentándome.

RESUMEN

Para realizar el análisis del movimiento del cuerpo humano, se debe hacer el seguimiento de cada una de las partes de este; las cuales pueden ser modeladas como un cuerpo rígido en un espacio 3D. En el seguimiento de cada parte del cuerpo, se tienen en cuenta dos indicadores: la orientación y la posición. La orientación puede ser deducida de las rotaciones, tales como cabeceo, rotación y guiñada, mientras que la posición se obtiene del desplazamiento desde un punto base.

La estimación de la orientación tiene aplicaciones clínicas en reumatismo, medicina física, medicina del deporte y rehabilitación; en donde es necesario realizar estudios de las capacidades de movilidad de los pacientes. Otra de las aplicaciones, es el área de monitoreo y cuidados a distancia; por ejemplo en geriatría, en donde un seguimiento constante de la postura puede ayudar a detectar riesgos de caída.

En este trabajo de tesis se hace la propuesta de un sistema que estima la orientación de dos sensores inerciales, cada sensor contiene tres acelerómetros y tres giróscopos, integrados en una unidad de medición inercial IMU (del inglés *Inertial Measurement Unit*). El giróscopo es un sensor que mide velocidad angular y por medio de la integración matemática de los datos que entrega es posible obtener los ángulos de giro; en consecuencia la orientación. Mientras que el acelerómetro es un sensor que mide aceleración lineal y nos ayuda a obtener una referencia de la horizontal.

Los datos arrojados por la IMU se envían a una computadora para su procesamiento, esta comunicación se hace de forma inalámbrica con dos módulos basados en un protocolo de radiofrecuencia. El algoritmo implementado para hacer la estimación de la orientación está basado en un filtro de Kalman que hace las correcciones de la deriva (errores acumulados) que presentan los giróscopos.

Para validar el sistema, se montaron los sensores en el brazo y en el antebrazo de una persona, a la cual se le pidió realizar movimientos de extensión y flexión hasta los 90°. Teniendo la orientación de cada IMU, se puede hacer el seguimiento del ángulo que forma la articulación del codo al hacer estos movimientos. Se hizo la comparación de los resultados con los datos arrojados por un sistema de seguimiento óptico Kinect, de la marca *Microsoft*, disponible en el laboratorio de la Unidad de Investigación y Desarrollo Tecnológico (UIDT) del CCADET. Obteniendo resultados consistentes respecto al sistema óptico de referencia.

ÍNDICE GENERAL

1	INTRODUCCIÓN	1
1.1	ANÁLISIS DEL MOVIMIENTO HUMANO	1
1.2	SISTEMAS DE SEGUIMIENTO DE MOVIMIENTO HUMANO	2
1.3	COMPARACIÓN ENTRE SISTEMAS	5
1.4	JUSTIFICACIÓN DE LA TESIS	5
1.5	OBJETIVOS	6
1.5.1	OBJETIVO PRINCIPAL	6
1.5.2	OBJETIVOS ESPECÍFICOS	6
1.6	METODOLOGÍA	7
1.7	INFRAESTRUCTURA	8
1.8	ESTRUCTURA DE LA TESIS	8
2	SEGUIMIENTO DEL MOVIMIENTO HUMANO CON SENSORES INERCIALES	10
2.1	SENSORES INERCIALES	10
2.2	ANTECEDENTES.....	10
2.3	ESTADO DEL ARTE	12
3	INTRODUCCIÓN A LOS FILTROS DE KALMAN	14
4	DESARROLLO	15
4.1	SELECCIÓN DE COMPONENTES.....	15
4.2	ALGORITMO DE ESTIMACIÓN DE ORIENTACIÓN.....	17
4.3	ARQUITECTURA DEL SISTEMA	25
4.3.1	COMUNICACIÓN ENTRE MICROCONTROLADOR Y SENSORES	25
4.3.2	COMUNICACIÓN ENTRE MICROCONTROLADOR Y COMPUTADORA.....	27
4.4	PRUEBAS DE COMUNICACIÓN	29
4.5	CALIBRACIÓN DE LOS SENSORES	30
4.6	PROCESAMIENTO DE LOS DATOS	33
4.7	DISEÑO DE LA TARJETA DE CONTROL	34
5	RESULTADOS EXPERIMENTALES Y EVALUACIÓN DEL SISTEMA	37
5.1	IMPLEMENTACIÓN DE LA ELECTRÓNICA	37
5.2	TARJETA DE CONTROL	41
5.3	CALIBRACIÓN DE LOS SENSORES	43
5.3.1	ESTABILIZACIÓN DE LOS SENSORES	43
5.3.2	CALIBRACIÓN DE LOS ACELERÓMETROS	44
5.3.3	CALIBRACIÓN DE LOS GIRÓSCOPOS.	45
5.4	MINIATURIZACIÓN DEL SENSOR	49

5.5	PROCESAMIENTO DE LOS DATOS	51
5.6	VALIDACIÓN DEL SISTEMA	55
6	CONCLUSIONES Y TRABAJO A FUTURO	60
	BIBLIOGRAFÍA	62
	ANEXO A	64

1 INTRODUCCIÓN

1.1 ANÁLISIS DEL MOVIMIENTO HUMANO

La ciencia del movimiento humano engloba a todas las ciencias aplicadas que se encargan del estudio del origen, mantenimiento y entrenamiento de cualquier tipo de movimiento que pueda ejecutar el ser humano en un contexto específico. Dichos movimientos pueden ser aquellos realizados en los deportes, en los ejercicios dirigidos hacia la prevención de enfermedades o mejoramiento de la salud, y en las actividades físicas de la vida cotidiana u ocupacional (Eston, Hawes, and Reilly, n.d.).

Algunas de las aplicaciones del estudio del movimiento humano son:

1. Mejorar la técnica deportiva en las ciencias del deporte y prevenir lesiones.
2. Reproducir los movimientos en animaciones para cine, videojuegos o arte digital.
3. Valorar progreso en terapias de rehabilitación, mediante una mayor comprensión del movimiento humano, objetivar resultados y valorar previamente a algunas terapias.
4. En ergonomía, valorar posiciones y movimientos de riesgo para el diseño de productos.
5. En seguridad laboral.

Además en el campo de la medicina, el seguimiento del movimiento humano proporciona instrumentación y técnicas que apoyan al médico y/o al investigador para obtener una valoración cuantitativa de parámetros del movimiento de los pacientes, para identificar el control neuromuscular anormal o desórdenes biomecánicos.

De acuerdo con la biomecánica, el cuerpo humano está dividido en al menos 15 partes independientes (Gowitzke and Milner 1999) como cabeza, región de la clavícula, región de la cadera, parte superior de las piernas, parte inferior de las piernas, pies, parte superior de los

brazos, parte inferior de los brazos y manos. Cada una de estas puede ser modelada como un cuerpo rígido en un espacio 3D; así, el movimiento humano puede ser capturado mediante el seguimiento de cada una de estas partes rígidas (Roetenberg, Luinge, and Slycke 2015). Para realizar el seguimiento de cada parte del cuerpo, se tienen en cuenta dos indicadores: la orientación y la posición. La orientación puede ser deducida de las rotaciones, tales como cabeceo, inclinación y vuelco, mientras que la posición se obtiene del desplazamiento desde un punto base (Zhu and Zhou 2004).

1.2 SISTEMAS DE SEGUIMIENTO DE MOVIMIENTO HUMANO

Para registrar el movimiento humano existen diferentes mecanismos que pueden ser divididos en dos (Zhou and Hu 2008): los basados en sistemas visuales y los basados en sistemas no visuales. Entre los primeros se encuentran el seguimiento por video y el óptico; mientras que en los no visuales tenemos el seguimiento mecánico, magnético, por ultrasonido y los sistemas inerciales.

A continuación se dará una pequeña explicación de los sistemas más utilizados:

Sistemas electromecánicos

Son trajes adaptables al cuerpo humano, son estructuras rígidas compuestas de barras metálicas o plásticas unidas mediante potenciómetros o goniómetros colocados en las principales articulaciones. Mientras el portador se mueve, el traje se adapta a sus movimientos y los potenciómetros o goniómetros recogen datos sobre el grado de apertura de las articulaciones (**figura 1**).



Figura 1. Traje electromecánico para seguimiento humano.

Este tipo de sistemas no tienen la capacidad de medir translaciones globales. Es decir, son capaces de medir las posiciones relativas de los miembros, pero no el desplazamiento del portador en el espacio. Además, estos sistemas asumen que las articulaciones son de tipo "bisagra", por lo que no tienen en cuenta rotaciones complejas como las que se producen en los hombros, los tobillos, las muñecas o la cabeza.

Otra desventaja de este tipo de sistemas, es que son pesados y suelen restringir el movimiento del portador. Sin embargo, con respecto a otros sistemas, tienen la ventaja, de tener un bajo costo, capacidad de registrar en tiempo real el movimiento a frecuencias altas, tener un alcance casi ilimitado y no tiene problemas de oclusión típicos de los sistemas ópticos (Menache and Sturza 2003).

Sistemas ópticos

Los sistemas ópticos utilizan cámaras para inferir la posición de un elemento en el espacio, es común que se utilice más de una cámara para proporcionar proyecciones simultáneas. Lo habitual es que los datos se recojan utilizando indicadores (*markers*) pegados al usuario (**figura 2**). La orientación de una superficie se infiere utilizando la posición relativa de al menos 3 indicadores.

El problema que surge con estos sistemas, es que, como todos los indicadores tienen el mismo aspecto el sistema puede confundir unos indicadores con otros (*marker swap*). Además, pueden necesitarse más cámaras para cubrir totalmente todas las superficies del sujeto (Menache and Sturza 2003).



Figura 2. Sistema óptico de seguimiento humano utilizando indicadores.

En general, los sistemas ópticos son muy fiables para capturar movimientos. Además, permiten la grabación en tiempo real con ciertas limitaciones como el número de indicadores, el número de actores y de cámaras.

Las desventajas de estos sistemas son el alto costo, debido al tipo y número de cámaras que se utilizan. Por otra parte, las cámaras se calibran respecto a un punto conocido, si la cámara se mueve mínimamente, se deberá recalibrar todo el sistema.

Sistemas inerciales

Con el desarrollo de los sistemas microelectromecánicos (MEMS, por sus siglas en inglés), fue posible crear dispositivos pequeños que se pueden adherir al cuerpo humano. Estos dispositivos son sensores inerciales, integrados por acelerómetros y giróscopos. Con los acelerómetros es posible medir la aceleración lineal y con los giróscopos la velocidad angular (**figura 3**); si se realiza una integración matemática de los datos arrojados por estos sensores se puede obtener la posición y la inclinación de los mismos. Para conocer la posición del usuario en el espacio es necesaria la integración de estos dos sensores y además se debe añadir un tercer sensor, que puede ser un magnetómetro, para tener una referencia adicional.

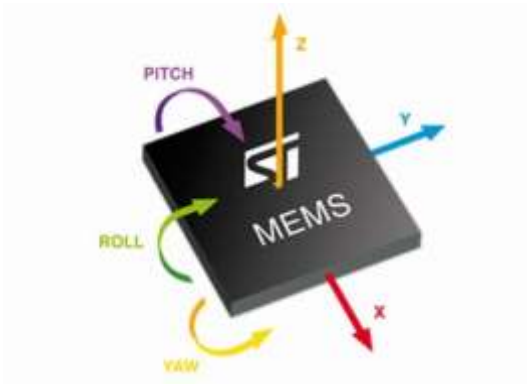


Figura 3. Sensor inercial.

Los sistemas inerciales presentan algunas ventajas como:

- Operan en tiempo real y tienen poca pérdida de información.
- No ocurren problemas de oclusión.
- No hay interferencias electromagnéticas.
- Son sistemas muy compactos y portables.
- No se requieren alta capacidad de procesamiento de los datos.
- Una sola persona puede operar el dispositivo.
- Se puede utilizar el sistema en condiciones de actividades cotidianas.

A pesar, que los sistemas inerciales presentan grandes ventajas, es necesario que los algoritmos de procesamiento de datos sean capaces de corregir con buena precisión los errores de deriva (acumulación de errores a lo largo del tiempo) inherentes al sensor de velocidad angular, ya que el desempeño del sistema depende totalmente de esto (Elguera Garcia 2015).

1.3 COMPARACIÓN ENTRE SISTEMAS

Cada uno de los sistemas antes mencionados, tienen sus ventajas y desventajas; el seguimiento mecánico es preciso pero tiene problemas con articulaciones de varios grados de libertad (Gowitzke and Milner 1999) y limita en gran medida la movilidad del usuario. El magnético es más flexible, pero en él interfieren fácilmente los objetos metálicos próximos. El seguimiento por vídeo u óptico es flexible, pero requiere que haya suficiente luz y cámaras alrededor del escenario donde se desea hacer el seguimiento, además de marcadores sobre el objeto que se desea seguir. El sistema por ultrasonido es susceptible a la pérdida de información por obstrucciones. Y el seguimiento por sensores inerciales requiere de algoritmos que sean capaces de contrarrestar los problemas de deriva de los giróscopos.

Tabla 1. Comparativa del desempeño de los diferentes tipos de seguimiento de movimiento (Zhou and Hu 2008).

Tipo	Precisión	Compacidad	Procesamiento	Costo	Desventajas
Inercial	Alta/Media	Alta	Eficiente	Bajo	Deriva
Mecánico	Media	Media	Eficiente	Medio	Inflexibilidad
Ultrasonido	Media	Baja	Eficiente	Bajo	Oclusión
Magnético	Media	Alta	Eficiente	Bajo	Objetos metálicos
Óptico	Alta	Baja	Ineficiente	Alto	Oclusión

Como se puede observar en la tabla 1 el seguimiento por medio de sensores inerciales tiene el mejor desempeño, ya que tiene el mayor número de ventajas, sobre los demás sistemas.

1.4 JUSTIFICACIÓN DE LA TESIS

En la Unidad de Investigación y Desarrollo Tecnológico del CCADET en el Hospital General de México “Dr. Eduardo Liceaga”, existen diversas aplicaciones clínicas en las que es necesario realizar estudios de las capacidades de movilidad de los pacientes. Entre estas pueden citarse las áreas de reumatismo, geriatría, medicina física y rehabilitación. Desafortunadamente, los equipos comerciales para la evaluación del movimiento humano son muy costosos y en la

actualidad la UIDT-HGM no cuenta con ninguno, mientras que los dispositivos comerciales disponibles de bajo costo (Kinect), son limitados en cuanto al tipo de movimientos que permiten monitorear y no cuentan con validación clínica. Por tal motivo, existe la necesidad en la UIDT-HGM de desarrollar un sistema propio que permita la evaluación de movimiento de pacientes, de manera robusta y económica.

Para ello se realizó una investigación sobre los métodos que existen para el seguimiento de movimiento; se hizo una comparación entre estos y se determinó que debido al tamaño, costo y precisión en las mediciones, el mejor método para nuestros objetivos es el sistema basado en sensores inerciales. Así, en este trabajo de tesis se planteó desarrollar un sistema de estimación de orientación basado en sistemas inerciales de bajo costo, como base para el desarrollo a corto plazo de un sistema completo de evaluación de movimientos humanos que cubra esta necesidad de la UIDT-HGM del CCADET.

1.5 OBJETIVOS

1.5.1 OBJETIVO PRINCIPAL

Desarrollar el prototipo de un sistema que haga la estimación de la orientación de dos sensores inerciales montados en el brazo y en el antebrazo; para que, utilizando estas orientaciones se realice el cálculo del ángulo que forma la articulación del codo.

1.5.2 OBJETIVOS ESPECÍFICOS

- Desarrollar el algoritmo que estime la orientación de cada sensor inercial; basado en un filtro de Kalman que nos ayude a corregir los errores en la estimación, debido a la deriva de los giróscopos.
- Realizar la adquisición de las señales entregadas por los sensores para hacer su análisis y caracterizar el ruido presente en ellas.
- Implementar la electrónica para establecer una comunicación inalámbrica entre los sensores y la computadora.
- Diseñar la tarjeta de control, que integre el microcontrolador, la alimentación y los módulos de comunicación inalámbrica; buscando que todo quede integrado en una unidad de control que al ser montado en el cuerpo humano, no limite su movimiento.
- Realizar el rediseño de la tarjeta de desarrollo comercial del sensor MPU-6050, para reducir su tamaño.

- Hacer pruebas del sistema propuesto, calculando el ángulo formado por la articulación del codo y hacer una comparación de los resultados obtenidos con los datos que se obtienen de un sistema óptico para validar el sistema.

1.6 METODOLOGÍA

- **IDENTIFICACIÓN DE UNA NECESIDAD**

En el área de rehabilitación y medicina física en general, debido a los grandes avances en la tecnología, cada vez hay más interés en aplicar estas nuevas herramientas y técnicas en apoyo de la medicina para el desarrollo de nuevos métodos de diagnóstico y cuidado, ya que existe una gran demanda de servicios médicos y se busca tener una mayor eficiencia en el tiempo de diagnóstico, además de mayor comodidad para los pacientes que requieran algún tipo de tratamiento o fisioterapia, por déficits de movilidad asociadas a condiciones neuromotoras o musculoesqueléticas (Huiyu Zhou and Huosheng Hu 2005).

- **INVESTIGACIÓN DE PRODUCTOS EN EL MERCADO**

Se realizó una búsqueda de sensores inerciales, sus características, costo, aplicaciones y su disponibilidad en el mercado para ser adquiridos. Además, con base en nuestros objetivos, se analizaron las tecnologías disponibles para la comunicación inalámbrica y los microcontroladores existentes que permitan la integración del sistema, con mayor facilidad.

- **SELECCIÓN E INTEGRACIÓN DE LOS COMPONENTES**

El siguiente paso fue la selección e integración de los componentes de manera que el microcontrolador obtenga los datos de los sensores y por medio de la antena de radiofrecuencia envíe los datos a una computadora para su procesamiento y de esta forma determinar la orientación de cada uno de los sensores conectados.

- **PRUEBAS DE COMUNICACIÓN Y CARACTERIZACIÓN DEL SENSOR**

Una vez hecha la integración, se realizaron pruebas de comunicación inalámbrica entre los sensores y la computadora. Además se analizaron las señales recibidas de los sensores para caracterizarlos.

- **PROCESAMIENTO DE LOS DATOS DEL SENSOR**

Para realizar el procesamiento de los datos del sensor se implementó un algoritmo para hacer la estimación de la orientación, basado en un filtro de Kalman que corrige los errores ocasionados por la deriva del sensor.

- DISEÑO DE LA TARJETA DE PRUEBA

Una vez realizadas todas las pruebas y cambios en la electrónica del sistema propuesto, se diseñó la tarjeta que incluye todos los dispositivos en un modelo compacto y fácil de portar por el usuario.

- VALIDACIÓN DE LOS RESULTADOS

Para validar la orientación obtenida del algoritmo implementado, se hizo una comparación con los datos de referencia que arroja el sistema óptico Kinect de la marca *Microsoft*.

1.7 INFRAESTRUCTURA

A continuación se enlista el material disponible para el desarrollo de este proyecto:

- Fuente de alimentación variable.
- Material eléctrico para implementar el circuito como: resistencias, capacitores, osciladores de cristal, diodos, transistores, etc.
- Osciloscopio.
- Sistema óptico Kinect de *Microsoft*.
- Software:
 - Arduino Software para el desarrollo del programa de adquisición de datos.
 - Proteus 8 Professional para el diseño de las tarjetas electrónicas.
 - MatlabR2013a para la calibración de los sensores y para la implementación de los algoritmos de estimación de movimientos.
 - XCTU-NG para la configuración de las antenas de radiofrecuencia.
 - NX 8.5 SIEMENS para el diseño mecánico del instrumento de calibración de los sensores.

1.8 ESTRUCTURA DE LA TESIS

El contenido de esta tesis quedó estructurado en cuatro partes principales:

1. ANTECEDENTES Y ESTADO DEL ARTE DEL SEGUIMIENTO DEL MOVIMIENTO HUMANO CON SENSORES INERCIALES

En el capítulo 2 se hace la introducción a los sensores inerciales y al seguimiento del movimiento humano con este tipo de sensores. Además se da un panorama general de los antecedentes y el estado del arte de los sistemas que utilizan sensores inerciales para hacer seguimiento de movimiento humano y estimación de orientación.

2. INTRODUCCIÓN A LOS FILTROS DE KALMAN

En el capítulo 3 se da una introducción del filtro de Kalman, sus características y aplicaciones en los sistemas de control de los sensores inerciales.

3. DESARROLLO DEL SISTEMA

El capítulo 4, explica todo el desarrollo llevado a cabo para la obtención del sistema que hace la estimación de la orientación de dos sensores inerciales, que en conjunto nos dan el ángulo que forma la articulación del codo; desde la selección de los componentes hasta la calibración de los sensores seleccionados y la implementación del filtro de Kalman que realizará las correcciones para una correcta estimación.

4. RESULTADOS Y VALIDACIÓN DEL SISTEMA

El capítulo 5 muestra los resultados obtenidos con el sistema y se realiza la validación haciendo una comparación con un sistema de seguimiento óptico.

5. CONCLUSIONES Y TRABAJO A FUTURO

En el capítulo 6 se presentan las conclusiones y se hace un análisis de las expectativas del trabajo a futuro.

2 SEGUIMIENTO DEL MOVIMIENTO HUMANO CON SENSORES INERCIALES

2.1 SENSORES INERCIALES

Los Sensores Inerciales IMU (Inertial Measuring Unit) miden la variación de rotación (giróscopos) y la aceleración lineal (acelerómetros). Un giróscopo de 3 ejes es un dispositivo inercial para detectar la rotación en el espacio 3D mediante la medición del ángulo de giro, pero tiene un problema inherente de deriva progresiva y ello limita su uso independiente. Para combatirlo, se puede usar de forma complementaria un acelerómetro de 3 ejes. El acelerómetro funciona basándose en la segunda ley de Newton; en condiciones de equilibrio o casi equilibrio, la salida puede ser aproximada a la gravedad, por lo que es posible usar un acelerómetro para la estimación de la inclinación (Atallah et al. 2010).

Mediante una doble integración matemática de los datos obtenidos del acelerómetro, se puede obtener la posición. Mientras que para el giróscopo, por medio de la integración de los datos es posible obtener el ángulo de giro.

2.2 ANTECEDENTES

Actualmente se encuentra comercialmente un traje que fue desarrollado por la empresa Xsens, cuenta con 17 sensores de inercia, los cuales constan de acelerómetros, magnetómetros y giróscopos de tres ejes; alimentado por 8 pilas AA; con un costo aproximado de 20, 000 dólares (**figura 4**) (Roetenberg, Luinge, and Slycke 2015). Este aparato es un sistema portátil de captura del movimiento humano, que puede utilizarse en exteriores mientras se practica un deporte (incluso deportes acuáticos), para medir la posición de todos los segmentos corporales y reproducirlos en un ordenador visualmente, midiendo posiciones, ángulos, velocidades en tres

dimensiones y mostrando esos datos en gráficos y tablas sincronizados con otros sistemas de medición como EMG (Electromiografía o actividad muscular eléctrica).



Figura 4. Traje desarrollado por Xsens, consta de 17 sensores y es alimentado por 8 pilas AA (Roetenberg, Luinge, and Slycke 2015).

Este sistema se puede utilizar en aplicaciones de rehabilitación, ciencias del deporte, animación cine y videojuegos, biomédica y ergonomía.



Figura 5. Aplicación de los sensores inerciales.

También hay aplicaciones militares para simuladores de realidad virtual que utilizan soldados como entrenamiento (**figura 6**). Otras aplicaciones son ya industriales para valorar el movimiento, aceleración y velocidad de vehículos, robots industriales y otros.

Existe otro sistema de rehabilitación Riablo de CoRehab, el cual utiliza sensores de inercia para evaluar el ángulo de la rodilla y la inclinación del torax, además tiene retroalimentación audio-visual en un video juego. Este sistema sirve para rehabilitación de las articulaciones de las extremidades inferiores (Leardini et al. 2014).



Figura 6. Aplicación en la milicia de los sensores inerciales.

2.3 ESTADO DEL ARTE

La estimación de la orientación tiene aplicaciones clínicas en reumatismo, medicina física, del deporte y rehabilitación; en las que es necesario realizar estudios de las capacidades de movilidad de los pacientes. Otra de las áreas de aplicación, es la de monitoreo y cuidados a distancia; por ejemplo en geriatría, en donde un seguimiento constante de la postura puede ayudar a detectar riesgos de caída.

Como ya se mencionó, el mayor reto para tener una buena estimación de la orientación al utilizar sensores inerciales, es el desarrollo de los algoritmos que hagan la corrección de los errores debido a la deriva que presentan los giróscopos.

Se encuentran reportados en el estado del arte dos métodos que miden orientación con respecto a la horizontal utilizando un solo sensor de inercia:

- El primer método hace el diseño y desarrollo de un filtro de Kalman para calcular la inclinación respecto a la vertical en condiciones en donde están presentes aceleraciones externas, diferentes a la aceleración de la gravedad. En este método hacen la fusión de un giróscopo de tres ejes y un acelerómetro de tres ejes por medio del filtro de Kalman (Ligorio and Sabatini 2015).

Para validar su sistema, montaron el sensor en la muñeca de 10 participantes y realizaron mediciones mientras realizaban 2 tareas: una secuencia de siete movimientos manuales con una duración de 60 segundos y 90 segundos de caminata a una velocidad natural. Como referencia utilizaron un sistema óptico.

Como resultados obtuvieron un error cuadrático medio en la estimación de la orientación, de 3.6° y 1.8° en las tareas manuales y la caminata, respectivamente.

- En el segundo método, también se hace la propuesta de un filtro de Kalman para calcular la inclinación en condiciones dinámicas, utilizando un giróscopo y un acelerómetro de tres ejes cada uno. En este método hacen la propuesta de separar en las señales recibidas de los acelerómetros, la aceleración de la gravedad de las aceleraciones externas (Lee, Park, and Robinovitch 2012).

En los resultados reportan que en periodos cortos de aceleración, su algoritmo es capaz de calcular con buena precisión la inclinación; pero al prolongarse estos periodos el algoritmo comienza a exhibir errores que van incrementando gradualmente.

3 INTRODUCCIÓN A LOS FILTROS DE KALMAN

El filtro de *Kalman* es un estimador de estados que produce una estimación óptima. Se asume que el sistema, del cual se están estimando los estados, es excitado por perturbaciones aleatorias (ruido blanco) y que las mediciones son ruidosas (Elguera Garcia 2015).

Sea un sistema de una sola entrada de control $u(t)$ y una sola salida $y(t)$, cuya descripción de estado es:

$$\dot{x} = Ax + Bu + Fv \quad (3.1)$$

$$y = Cx + w \quad (3.2)$$

donde x es el vector de estado, u es una entrada escalar conocida, y es la salida escalar, v y w son procesos escalares aleatorios gaussianos del tipo de ruido blanco, cuyas densidades espectrales de potencia son V y W , respectivamente.

Para diseñar un observador o estimador óptimo de Kalman para el vector de estado, se recurre a la estructura básica del observador. La ecuación que describe a este observador es:

$$\dot{\underline{x}} = A\underline{x} + Bu + K_k(y - C\underline{x}) \quad (3.3)$$

siendo \underline{x} el vector de estado estimado y K_k la matriz de diseño del estimador.

Si las señales v y w no están correlacionadas, el valor de K_k se determina mediante la fórmula:

$$K_k = PC^T W^{-1} \quad (3.4)$$

Siendo P la matriz de covariancia óptima del error del estimador de estado.

4 DESARROLLO

En este capítulo se explicará el método que se siguió para desarrollar el prototipo del sistema que estima la orientación de dos sensores inerciales, que en conjunto nos dan el ángulo de la articulación del codo.

4.1 SELECCIÓN DE COMPONENTES

Como resultado de la investigación de los productos disponibles en el mercado se seleccionó el sensor comercial MPU-6050 de la marca *Invensense* que contiene: tres giróscopos y tres acelerómetros; es decir, tiene un acelerómetro y un giróscopo para cada uno de los ejes (X, Y, Z). Cada giróscopo cuenta con un rango de escala programable de ± 250 , ± 500 , ± 1000 y $\pm 2000^\circ/\text{seg}$; mientras que los acelerómetros tienen un rango de escala de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$. Además utiliza un convertidor analógico-digital de 16 bits para cada uno de los acelerómetros y giróscopos, lo que nos favorece con el muestreo en tiempo real.

Otras consideraciones importantes son: un costo accesible de 100.00 pesos M.N. aproximadamente, gran disponibilidad en el mercado nacional y el tamaño de la tarjeta comercial es de 2x1.6 cm (**figura 7**), lo que permite utilizarla en casi cualquier parte del cuerpo sin limitar el movimiento.



Figura 7. Tarjeta con sensor MPU-6050 de la marca *Invensense* y una referencia de su tamaño.

La adquisición de los datos digitales del sensor se realiza por medio del protocolo de comunicación I²C (*Inter Integrated Circuit*); la comunicación se realiza a través de solo dos líneas de transmisión: **SCL** (*Serial Clock*) que es la línea de reloj para la sincronización de los datos y **SDA** (*Serial Data*), la línea a través de la cual se transmiten los datos (Herrera Zuleta, n.d.). Este protocolo permite tener en paralelo varios sensores, ya que trabaja por medio del manejo de direcciones; es decir, cada sensor tiene asociada una dirección y el dispositivo maestro (microcontrolador) determina cual esclavo leer (sensor), de esta forma se puede trabajar de forma sencilla con más de un sensor a la vez (**figura 8**).

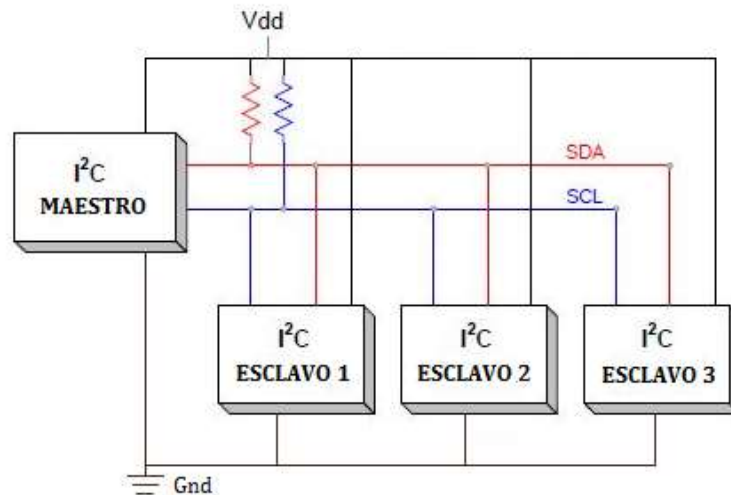


Figura 8. Esquema de la comunicación por I²C.

Para la comunicación inalámbrica se eligieron las antenas de radiofrecuencia Xbee S2 de la marca *Digi International*, elegidas por su tamaño y facilidad de uso (**figura 9**). Estas antenas emulan un cable serial para hacer la comunicación entre el microcontrolador, que controla los sensores, y la computadora en donde se realizará el procesamiento de los datos. Estos dispositivos están diseñados para aplicaciones que requieren un alto tráfico de datos y tienen un alcance aproximado de 30 metros.



Figura 9. Antenas XBee de radiofrecuencia de la marca Digi International.

El microcontrolador elegido fue el ATmega 328, ya que tiene las librerías y la arquitectura necesaria para utilizar el protocolo I2C con los sensores y a su vez controlar las antenas de radiofrecuencia. Para las pruebas se utilizó la tarjeta de desarrollo Arduino Nano, que contiene el microcontrolador seleccionado (**figura 10**).



Figura 10. Microcontrolador ATmega 328 en tarjeta de desarrollo Arduino Nano.

4.2 ALGORITMO DE ESTIMACIÓN DE ORIENTACIÓN

Para la implementación del algoritmo de estimación de la orientación de los sensores, se desarrolló un algoritmo basado en filtros de Kalman que nos ayudó en el procesamiento de los datos arrojados por los sensores inerciales. Como se explicó en el capítulo 2, los giróscopos entregan datos de velocidad angular que pueden integrarse matemáticamente y de esta forma obtener ángulos u orientación. El acelerómetro por su parte nos entrega datos de aceleración lineal y nos proporciona una referencia vertical y fija, que es la dirección de la fuerza de aceleración gravitacional.

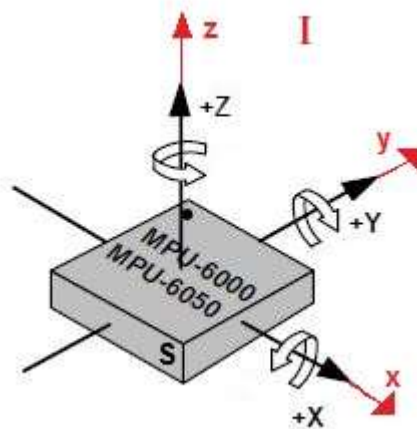


Figura 11. Sensor MPU6050 de la marca Invensense con referencia de la orientación de sus ejes y sentido de giro.

Es importante destacar que en condiciones dinámicas el acelerómetro es sensible no solo a la gravedad, también lo es a aceleraciones externas (en este caso, la aceleración del objeto en movimiento). Por tanto, la señal del acelerómetro es la suma de la aceleración gravitacional y la aceleración externa.

Como se ha mencionado anteriormente, los sensores de velocidad angular tienen un problema inherente de deriva, que a lo largo del tiempo va acumulando errores en la estimación de la orientación del sistema. Este algoritmo está diseñado para hacer la estimación de la orientación en condiciones en donde están presentes aceleraciones externas, diferentes a la producida por la gravedad (Ligorio and Sabatini 2015). Todo lo anterior hace necesaria una fusión correcta del giróscopo que proporciona la orientación y del acelerómetro que nos da la referencia y nos ayuda a corregir los errores de estimación.

Descripción del método utilizado en el algoritmo.

En la figura 11 se muestra un sistema de referencia inercial I xyz y un sistema de referencia XYZ embebido en un sensor S . La transformación de coordenadas de un vector x de (3×1) entre el sistema de referencia del sensor S y el sistema de referencia inercial I se expresa:

$${}^I x = {}^I_S R {}^S x \quad (4.1)$$

en donde ${}^I_S R$ es la matriz de rotación del sistema de referencia del sensor S respecto al sistema de referencia inercial I .

Por otro lado, la matriz de rotación ${}^I_S R$

$${}^I_S R = [{}^S X \quad {}^S Y \quad {}^S Z]^T \quad (4.2)$$

se puede expresar, utilizando los ángulos de Euler, como:

$${}^I_S R = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix} \quad (4.3)$$

en donde α (yaw), β (pitch) y γ (roll) son los ángulos de rotación alrededor de los ejes Z , Y y X respectivamente.

Cómo se puede observar en la matriz (4.3), la última fila de ${}^I_S R$ esta expresada solo en términos de γ y β ; por lo tanto, los ángulos roll y pitch pueden calcularse usando solo ${}^S Z$, de la siguiente forma:

$$\gamma = \tan^{-1} \left(\frac{Z_2}{Z_3} \right) \quad \text{y} \quad \beta = \tan^{-1} \left(\frac{-Z_1}{Z_2 / \sin \gamma} \right) \quad (4.4)$$

Teniendo en cuenta todo lo anterior, se puede entender que el propósito del algoritmo es estimar ${}^S Z$.

Descripción del algoritmo.

Modelo del sensor

Las señales del giróscopo (G) y del acelerómetro (A) se modelan de la siguiente forma:

$$y_G = {}^S \omega + n_G \quad (4.5a)$$

$$y_A = {}^S g + {}^S a + n_A \quad (4.5b)$$

En donde ω es la velocidad angular, a es la aceleración externa, g es la aceleración de la gravedad, n_G y n_A son los ruidos de las mediciones del giróscopo y el acelerómetro respectivamente, los cuales se asume que son no correlacionados y son ruido blanco Gaussiano de media cero.

De (4.5b), ${}^S a$ se modela como un proceso de ruido blanco, filtrado por un paso bajas de primer orden:

$$a_t = c_a a_{t-1} + \epsilon_t \quad (4.6)$$

donde c_a es una constante adimensional entre 0 y 1 que determina la frecuencia de corte, y ϵ_t es el error variable en el tiempo del modelo del proceso de aceleración.

Diseño del Filtro de Kalman

El filtro de Kalman puede definirse por medio del modelo del proceso y el modelo de la medición:

$$x_t = \Phi_{t-1} x_{t-1} + w_{t-1} \quad (4.7a)$$

$$z_t = Hx_t + v_t \quad (4.7b)$$

En donde x es el vector de estado, z es el vector de medición, Φ es la matriz de transición del estado, H es la matriz de observación, w es el ruido blanco Gaussiano del proceso y v el ruido blanco Gaussiano de las mediciones.

Como se mencionó antes, el propósito del algoritmo es estimar ${}^S Z$, por lo tanto el vector de estado se define como $x = {}^S Z$.

Ahora, el modelo del proceso está basado en las mediciones del giróscopo, como se mencionó. Por lo tanto, la orientación puede encontrarse integrando de la siguiente manera:

$$R_t = R_{t-1}(I + \Delta t \tilde{\omega}_{t-1}) \quad (4.8)$$

En la ecuación (4.8), Δt es el intervalo de tiempo de muestreo y $\tilde{\omega}$ es una matriz de 3×3 que denota la matriz anti-simétrica del vector ω .

La matriz anti-simétrica de $\tilde{\omega}$ se define como:

$$[{}^S \omega \times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

En donde ω_x , ω_y y ω_z son las velocidades angulares de los giróscopos de los ejes X, Y y Z, respectivamente.

La forma específica de (4.8) para ${}^S Z$ es

$${}^S Z_{.t} = (I + \Delta t \tilde{\omega}_{t-1})^T {}^S Z_{.t-1} \quad (4.9)$$

Sin embargo debemos considerar la salida real del giróscopo $y_G = \omega + n_G$, por lo tanto

$${}^S Z_{.t} = (I - \Delta t \tilde{y}_{G,t-1}) {}^S Z_{.t-1} + \Delta t (-{}^S \tilde{Z}_{.t-1}) n_G \quad (4.10)$$

De (4.10) tenemos que:

$$\Phi_{t-1} = I - \Delta t \tilde{y}_{G,t-1} \quad (4.11)$$

$$w_{t-1} = \Delta t (- {}^S \tilde{Z}_{\cdot,t-1}) n_G \quad (4.12)$$

Ahora, la matriz de covarianza del ruido del proceso está dada como el primer momento estadístico del ruido del proceso:

$$Q_{t-1} = E[w_{t-1} w_{t-1}^T] \quad (4.13)$$

Por lo que, usando (4.12), Q_{t-1} se define como:

$$Q_{t-1} = -\Delta t^2 {}^S \tilde{Z}_{\cdot,t-1} \Sigma_G {}^S \tilde{Z}_{\cdot,t-1} \quad (4.14)$$

Donde Σ_G es la matriz de covarianza del ruido de la medición del giróscopo, definida como $E[n_G n_G^T]$. Σ_G se establece como $\Sigma_G = \sigma_G^2 I_3$, en donde σ_G^2 es la varianza del ruido, caracterizada en la fase de calibración que se describirá más adelante.

Por otra parte, como el modelo de medición está basado en las mediciones del acelerómetro, es utilizado para corregir el error de estimación propagado en el tiempo, en el modelo del proceso. El modelo de medición debe lograr separar de buena forma la aceleración externa de la aceleración gravitacional, para alcanzar gran precisión en la estimación, en condiciones dinámicas.

El modelo de la aceleración externa en (4.6) se sustituyó en el modelo del acelerómetro (4.5b). Para este propósito, primero definamos el error de la predicción de aceleración como:

$${}^S a_{\varepsilon,t}^- = {}^S a_t^- - {}^S a_t \quad (4.15)$$

Cabe hacer notar que el signo de menos (-) en el superíndice denota la estimación *a priori*.

En (4.15) la aceleración externa *a priori* en el tiempo actual ${}^S a_t^-$ es $C_a {}^S a_{t-1}^+$. Entonces (4.5b) puede reescribirse como:

$$Y_{A,t} - C_a {}^S a_{t-1}^+ = g {}^S \tilde{Z}_{\cdot,t} - {}^S a_{\varepsilon,t}^- + n_A \quad (4.16)$$

Se observa que el lado izquierdo de la igualdad es la resta de la aceleración externa de la señal del acelerómetro.

De (4.16) tenemos que:

$$z_t = Y_{A,t} - C_a \cdot^S a_{t-1}^+ \quad (4.17)$$

$$H = gI \quad (4.18)$$

$$v_t = -\cdot^S a_{\varepsilon,t}^- + n_A \quad (4.19)$$

La matriz de covarianza del ruido de la medición M_t está definida por el primer momento estadístico del ruido de la medición:

$$M_t = E[v_t v_t^T] \quad (4.20)$$

Ya que $\cdot^S a_{\varepsilon,t}^-$ y n_A no están correlacionadas, (4.20) puede dividirse en

$$M_t = \Sigma_{acc} + \Sigma_A \quad (4.21)$$

en donde Σ_{acc} es la matriz de covarianza del error del modelo de aceleración y se define como $E[(\cdot^S a_{\varepsilon,t}^-)(\cdot^S a_{\varepsilon,t}^-)^T]$ y Σ_A es la matriz de covarianza del ruido de las mediciones del acelerómetro definida como: $\Sigma_A = E[n_A n_A^T]$

Ahora usando (4.15) y asumiendo que $\cdot^S a_t^-$ y $\cdot^S a_t^+$ no están correlacionadas y que, en promedio, la expectativa de aceleración externa es cero, Σ_{acc} se puede simplificar como:

$$\Sigma_{acc} = E[(C_a \cdot^S a_{t-1}^+)(C_a \cdot^S a_{t-1}^+)^T] \quad (4.22)$$

Debe notarse que, ya que Σ_{acc} es variante en el tiempo y no puede obtenerse analíticamente, se aproxima a tener una matriz diagonal en donde el cuadrado de la norma de $C_a \cdot^S a_{t-1}^+$ esta distribuido en cada elemento de la diagonal principal, es decir: $\Sigma_{acc} = 3^{-1} C_a^2 \|\cdot^S a_{t-1}^+\|^2 I$

Si se compara (4.16) con la ecuación equivalente en condiciones estáticas (es decir, $y_{A,t} = g^S Z_t + n_A$), se observa que la matriz de covarianza Σ_{acc} desempeña un papel muy importante compensando el efecto de la aceleración externa $C_a \cdot^S a_{t-1}^+$, incrementando M_t .

Ahora Σ_A se establece como $\sigma_A^2 I$, en donde σ_A^2 es la varianza del ruido del acelerómetro, obtenida en la fase de calibración del sensor.

Una vez definidos los modelos del proceso y de la medición, el procedimiento del FK propuesto se puede resumir como **(ver figura 12)**:

1. Calcular la estimación del estado *a priori*:

$$x_t^- = \Phi_{t-1} x_{t-1}^+ \quad (4.23)$$

2. Calcular la matriz de covarianza del error *a priori*:

$$P_t^- = \Phi_{t-1} P_{t-1}^+ \Phi_{t-1}^T + Q_{t-1} \quad (4.24)$$

3. Calcular la ganancia del Filtro de Kalman:

$$K_t = P_t^- H^T (H P_t^- H^T + M_t)^{-1} \quad (4.25)$$

4. Calcular la estimación del estado *a posteriori*:

$$x_t^+ = x_t^- + K_t (z_t - H x_t^-) \quad (4.26)$$

5. Calcular la matriz de covarianza del error *a posteriori*:

$$P_t^+ = (I - K_t H) P_t^- \quad (4.27)$$

Una vez estimada ${}^S Z_t^+$, la aceleración externa ${}^S a_t^+$ se obtiene de:

$${}^S a_t^+ = y_{A,t} - g^S Z_t^+$$

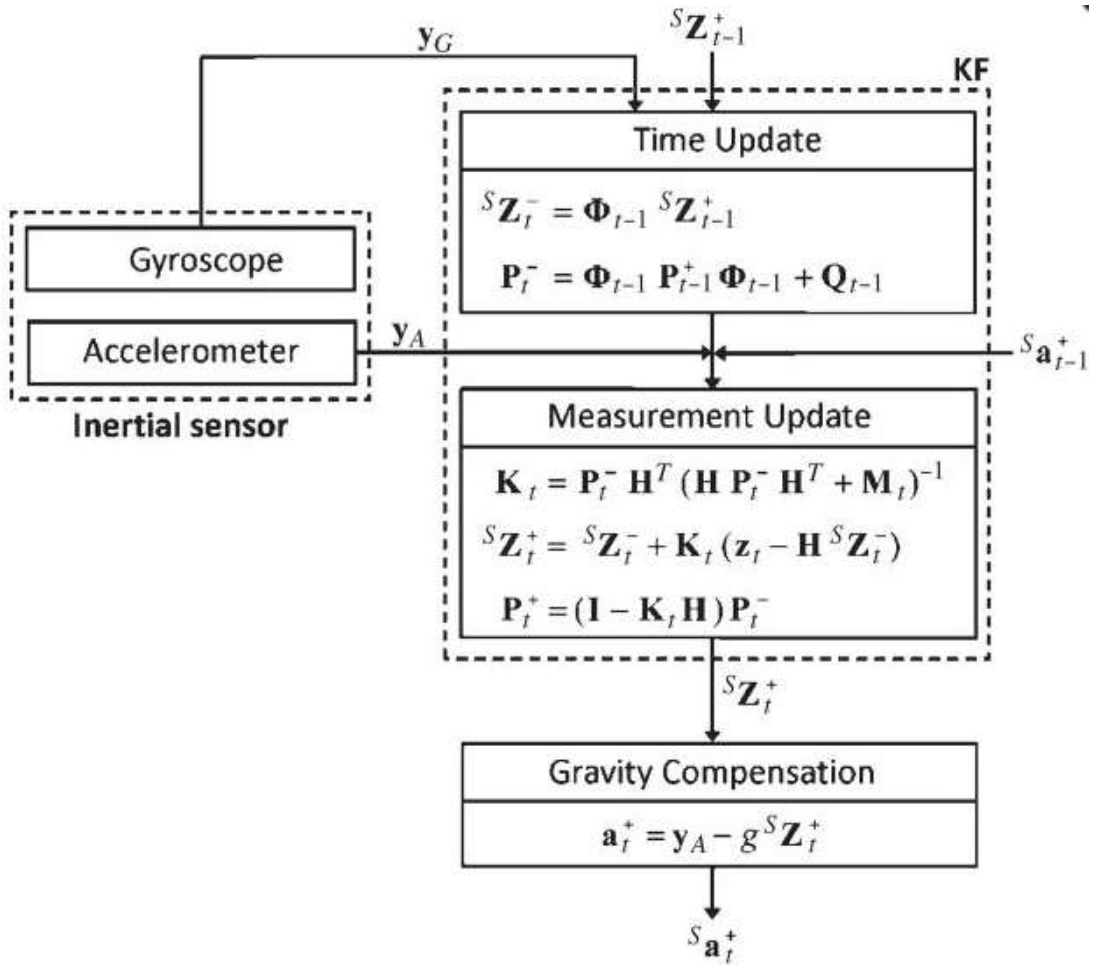


Figura 12. Esquema del filtro de Kalman utilizado (Lee, Park, and Robinovitch 2012).

4.3 ARQUITECTURA DEL SISTEMA

4.3.1 COMUNICACIÓN ENTRE MICROCONTROLADOR Y SENSORES

Como se mencionó anteriormente, para fines de probar el concepto, se implementó un sistema compuesto por dos IMU con el objetivo de medir un grado de libertad de una articulación. Cada IMU contienen: tres acelerómetros y tres giróscopos; los acelerómetros arrojan datos digitales de 16 bits equivalentes a la aceleración lineal a la que están sujetos cada uno de los ejes del sensor (X, Y, Z); y por su parte los giróscopos entregan datos digitales de 16 bits equivalentes a la velocidad angular alrededor de cada uno de los ejes del sensor.

El microcontrolador, utilizando el protocolo de comunicación I2C, realiza el control de la lectura de cada uno de los sensores. Este protocolo con el MPU-6050 alcanza una velocidad de comunicación de 400 KHz; para poder leer y alimentar los sensores comparten cuatro líneas:

1. *Línea de Alimentación Vcc.* Los sensores se alimentan con 3.3 volts que son tomados del Arduino.
2. *Tierra Gnd.* Conectada a la misma tierra del microcontrolador.
3. *Línea de reloj SCL.* Para sincronizar el envío de datos.
4. *Línea de datos SDA.* Por esta línea se envían los datos al microcontrolador

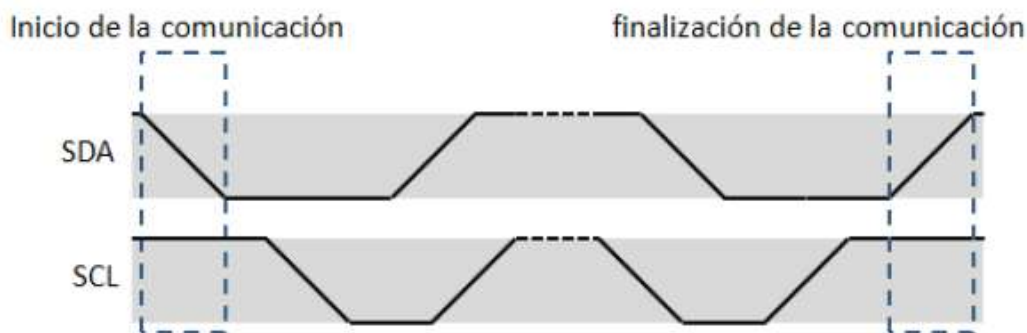


Figura 13. Esquema del protocolo de comunicación I2C

En este protocolo solo los dispositivos maestros pueden iniciar comunicación con los esclavos, que como ya se mencionó, tienen una dirección propia y única con la cual pueden ser llamados e iniciar comunicación. Para establecer comunicación se necesita la condición de BUS LIBRE, esta se cumple al estar las dos líneas de comunicación (**SDA** y **SCL**) en estado lógico alto. En este estado cualquier maestro puede ocupar el bus estableciendo la condición de INICIO (ver **figura**

13), esta consiste en poner en estado lógico bajo la línea de datos (SDA) y dejar en alto la de reloj (SCL).

Una vez que se da la condición de INICIO, el primer byte que se transmite contiene los 7 bits que componen la dirección del dispositivo con el que se requiere comunicar y un bit que corresponde a la operación que se desea realizar: lectura o escritura (R/W). Si el dispositivo cuya dirección corresponde al byte enviado se encuentra en el bus, este responde con un bit en bajo llamado BIT DE RECONOCIMIENTO (ACK).

Cuando el bit de lectura/escritura se pone en un nivel lógico bajo (escritura), el maestro envía datos al esclavo y la comunicación se mantiene siempre que reciba señales de reconocimiento (bit en bajo). Si por el contrario, el bit de lectura/escritura se envía en nivel lógico alto (lectura), el maestro genera pulsos de reloj para que el dispositivo esclavo pueda enviarle datos; además, después de cada byte recibido, el maestro genera el correspondiente pulso de reconocimiento.

Para terminar la comunicación y dejar libre el bus, el maestro debe generar una condición de parada o FIN, esta consiste en poner ambas líneas de comunicación en estado lógico alto. En este momento el maestro puede reiniciar la comunicación con otra condición de inicio y cambiar de dispositivo esclavo o cambiar el estado del bit lectura/escritura.

Los sensores utilizados (MPU-6050) siempre actúan como esclavos cuando se comunican con el microcontrolador y solo tienen dos posibles opciones de dirección: 0X68 o 0X69 en hexadecimal (1101000 y 1101001 en binario, respectivamente). El bit menos significativo de la dirección de los sensores, se establece por medio del pin 9 (AD0); poniendo en alto este bit, es decir, conectándolo a VDD (3.3V) tendremos la dirección 0X69 y si, por el contrario, ponemos en bajo o a Tierra el pin, tendremos la dirección 0x68.

Teniendo todo esto en cuenta, queda claro que si se desea trabajar con más de dos sensores, se tiene que trabajar con algún tipo de multiplexor para poder controlarlos. Con el microcontrolador seleccionado se puede trabajar hasta con 12 sensores de manera multiplexada.

4.3.2 COMUNICACIÓN ENTRE MICROCONTROLADOR Y COMPUTADORA

Por otra parte, el microcontrolador envía los datos obtenidos del sensor a la computadora por medio de una comunicación serial asíncrona, a través de las antenas de radiofrecuencia XBee (figura 14).

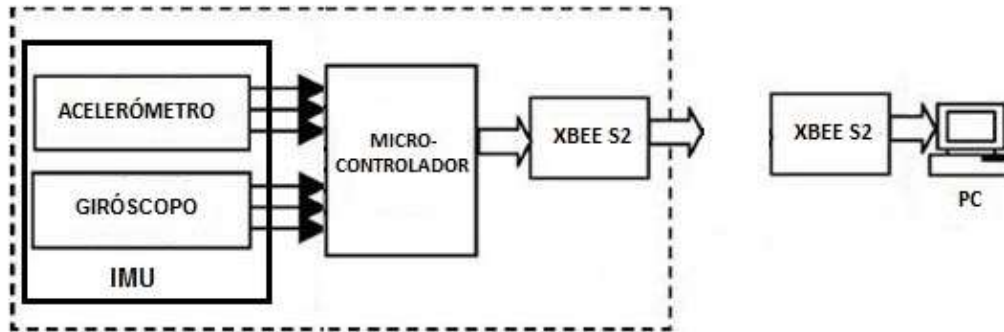


Figura 14. Esquema del sistema propuesto.

Las antenas XBee trabajan en la región de radiofrecuencia del espectro electromagnético, en la banda de 2.4 GHz. Tienen una tasa de transferencia de datos de 250 kbps con un bajo consumo de energía.

La serie 2 de antenas Xbee puede trabajar en tres diferentes tipos de comunicación: punto a punto, punto a multipunto y redes *mesh* (malla). Los elementos de una red Xbee son:

- **Coordinador.** Es el encargado de controlar la red.
- **Router.** Interconecta dispositivos alejados.
- **Dispositivo final.** Puede establecer comunicación con su coordinador o router pero no puede transmitir información a otro dispositivo.

El tipo de comunicación a usar en nuestro sistema, es de punto a punto, ya que solo necesitamos dos antenas: la que estará conectada directamente a la computadora (coordinador) y la que estará conectada con el microcontrolador para enviar los datos del sensor (dispositivo final). Para poder utilizar las antenas es necesario realizar su configuración por medio del software X-CTU.

Al utilizar una comunicación serial asíncrona (RS-232) entre las antenas, es necesario configurar los parámetros como son: velocidad, paridad, bits de datos y bits de parada. Estos parámetros son configurados para que sean los mismos en cada antena, con el software X-CTU. Además, cada antena cuenta con un número de “dirección de destino” compuesto por DH (Destination

Address High) y DL (Destination Address Low) (**figura 15**). Estos deben ser anotados para la configuración de las antenas; el número de dirección del coordinador debe guardarse en la configuración del dispositivo final y viceversa.



Figura 15. Número de dirección de destino de las antenas de radiofrecuencia Xbee.

Para comunicar el microcontrolador con la antena, es necesario realizar la conexión que se observa en la **figura 16**. El Arduino envía a la antena los datos que obtuvo del sensor, por medio del pin de comunicación serial Tx que va conectado al pin Data In (Rx) del XBee; además se debe conectar la antena a la salida de 3.3 volts del Arduino para su alimentación y conectar las tierras.

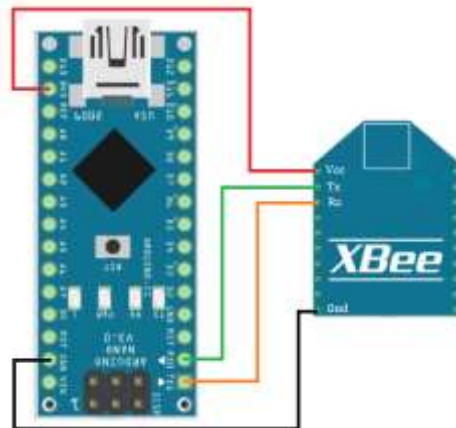


Figura 16. Conexión entre el Arduino y la antena Xbee.

De igual forma, la comunicación entre el microcontrolador y la antena que enviará los datos del sensor, es bajo el protocolo RS-232; por lo que los parámetros, con los valores que se configuraron en las antenas, deben ser los mismos para el Arduino.

4.4 PRUEBAS DE COMUNICACIÓN

Una vez realizada la integración de los componentes, se hizo un programa en Matlab para recibir y guardar los datos enviados por el sensor. Un acelerómetro y un giróscopo por cada eje, por lo tanto, se reciben en cada cadena (muestra) 7 datos (3 del giróscopo + 3 del acelerómetro + 1 identificador del IMU).

En Matlab se desarrolló el algoritmo para hacer un muestreo de las cadenas recibidas por un cierto periodo de tiempo (pudiendo ser infinito); tanto el tiempo, como la frecuencia de muestreo pueden ser modificados dentro del programa (ver Anexo A). Además se implementó en el mismo algoritmo una parte de filtrado de datos corruptos, es decir, se descartan las muestras que estén incompletas o que contengan datos de más; esto debido a que se pueden perder o añadir datos por ruido en la comunicación inalámbrica.

La velocidad utilizada en la comunicación serial entre las antenas, es de 9,600 baudios; a esta velocidad, la frecuencia de muestreo máxima obtenida, integrando las antenas Xbee y el algoritmo en Matlab, es aproximadamente de 30 Hz, velocidad suficiente para monitorizar movimientos humanos naturales a velocidad moderada.

Sabemos que la frecuencia máxima obtenida es de 30 Hz, ya que se hicieron pruebas de 60 segundos variando la frecuencia de muestreo en cada una; siendo el máximo de muestras obtenidas igual a 1800, aun ajustando la frecuencia de muestreo a más de 30 Hz.

$$f_{muestreo} = \frac{\text{no.de muestras}}{60 \text{ segundos}} \quad (4.28)$$

Además se realizaron pruebas de comunicación y de alcance con las antenas para ver que tanto se modifica el número de muestras recibidas en la computadora al modificar la distancia entre las antenas.

Posteriormente, se modificaron la electrónica y los algoritmos para hacer la comunicación y adquisición de datos de forma inalámbrica de dos sensores (ver Anexo A). Como ya se explicó anteriormente, los sensores comparten el mismo bus para su comunicación y alimentación. Además, cada uno de estos está diseñado para trabajar con dos posibles direcciones, de esta forma se hace relativamente sencilla la comunicación en paralelo; ya que cada uno se configura con una dirección diferente y el microcontrolador decide cual llamar.

4.5 CALIBRACIÓN DE LOS SENSORES

Entendiendo el funcionamiento de las IMU se tiene claro que si el sensor está en posición estática sobre una superficie plana y sin inclinaciones, el acelerómetro del eje que este alineado con la dirección de la fuerza de gravedad (eje -Z, en la **figura 19**) arrojará un valor equivalente a 1 G (9.8 m/s^2), mientras que los otros dos acelerómetros de los ejes perpendiculares a la dirección de la fuerza gravedad deberán dar un valor de 0 (cero). Del mismo modo los tres giróscopos deberán arrojar un valor de cero al no estar sujetos a ninguna velocidad angular.

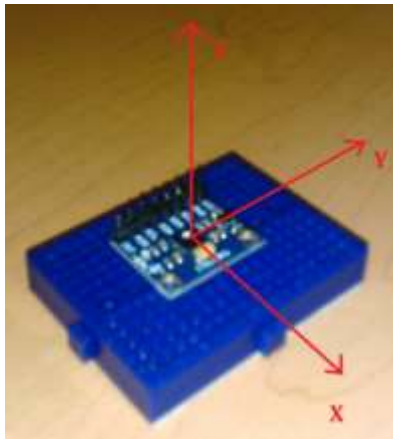


Figura 17. Tarjeta con sensor MPU-6050 indicando el sentido de sus ejes.

Sin embargo, una de las desventajas de los sensores inerciales es que no dan lecturas exactas, es decir tienen un valor de *offset* y oscilaciones debidas a ruido. Para el correcto procesamiento de los datos y tener resultados precisos de orientación, deben ser caracterizados estos valores y determinar las matrices de covarianza del ruido de las señales, necesarias para la fase de corrección del filtro de Kalman.

En la caracterización se realizaron dos pruebas diferentes:

1. En la primera prueba se alinearon cada uno de los ejes del sensor con la vertical que apunta en la dirección de la fuerza de gravedad (en ambos sentidos: positivo y negativo) y se tomaron muestras del sensor estático por 60 segundos a 30 Hz; esta prueba se realizó 4 veces para cada eje y sentido, modificando la orientación de los ejes perpendiculares a la gravedad (Ligorio and Sabatini 2015). De esta caracterización se obtienen los valores digitales equivalentes a 1G (9.8 m/s^2) de cada acelerómetro, el valor de offset de cada acelerómetro y el valor de offset de cada giróscopo.

2. En la segunda prueba se caracterizaron los valores digitales máximos para cada uno de los giróscopos; se colocó el sensor con el eje Z alineado con la gravedad y se puso a girar el sensor alrededor de este para generar un valor en el giróscopo del eje Z. Esta prueba se realizó para los otros dos ejes, X y Y.

La primera prueba, para caracterizar los acelerómetros, se realizó en diferentes momentos del día, en diferentes días y en diferentes lugares; esto para tener un número más grande de muestras y así tener estimaciones más precisas con menos error, mediante una mejor caracterización estadística del ruido:

- Se trazó una línea, sobre una superficie plana y vertical, apuntando directamente en dirección de la gravedad con ayuda de una plomada y se colocó el sensor sobre dicha superficie cuidando de alinear el eje de estudio con la línea de gravedad (Ligorio and Sabatini 2015)(**ver figura 20**).

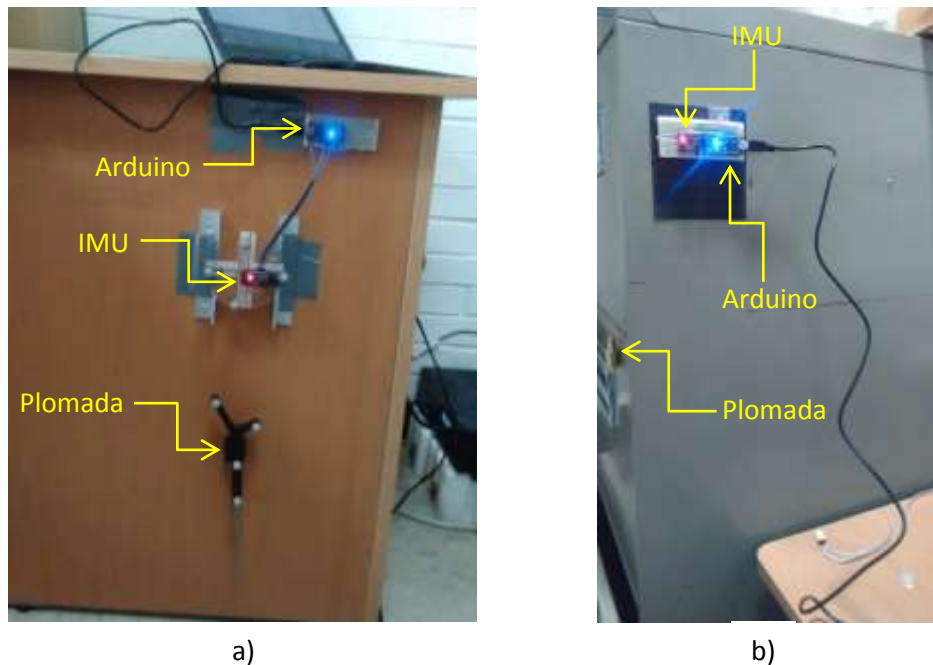


Figura 18. Diseño experimental tomado del artículo de la referencia. Las pruebas se realizaron sobre dos superficies verticales diferentes: a) un escritorio y b) un archivero.

- Se colocó el sensor sobre una mesa, considerando que está completamente perpendicular a la línea que apunta a la gravedad (**ver figura 19**).



Figura 19. Muestras tomadas sobre la mesa.

La segunda prueba, como se mencionó, se utilizó para calibrar los giróscopos; por lo que se necesitaba un sistema de referencia que rotara y que la velocidad de rotación pudiera ser controlada. Con ayuda del software NX 8.5 de Siemens para diseño CAD, se diseñó el sistema con un motorreductor y un rodamiento para tener el giro en un eje y poder controlar su velocidad con voltaje (**figuras 20 - 22**).

La manufactura del sistema de calibración de los giróscopos se realizó en impresión 3D, en el Laboratorio Nacional de Manufactura Aditiva, Digitalización 3D y Tomografía Computarizada (MADiT) del CCADET. El material utilizado fue ABS (Acrilonitrilo Butadieno Estireno).

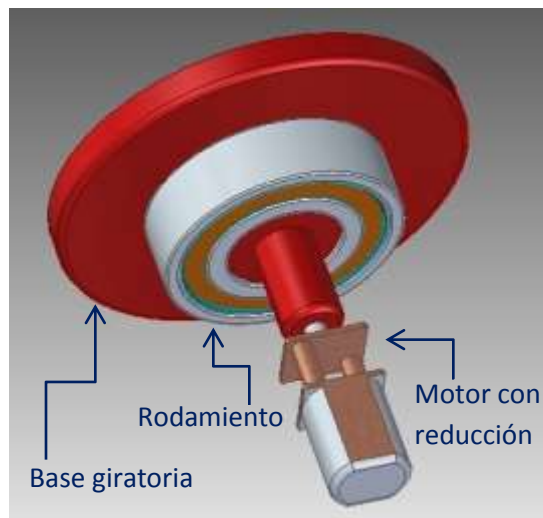


Figura 20. Motorreductor y rodamiento con plataforma para la calibración de los sensores.

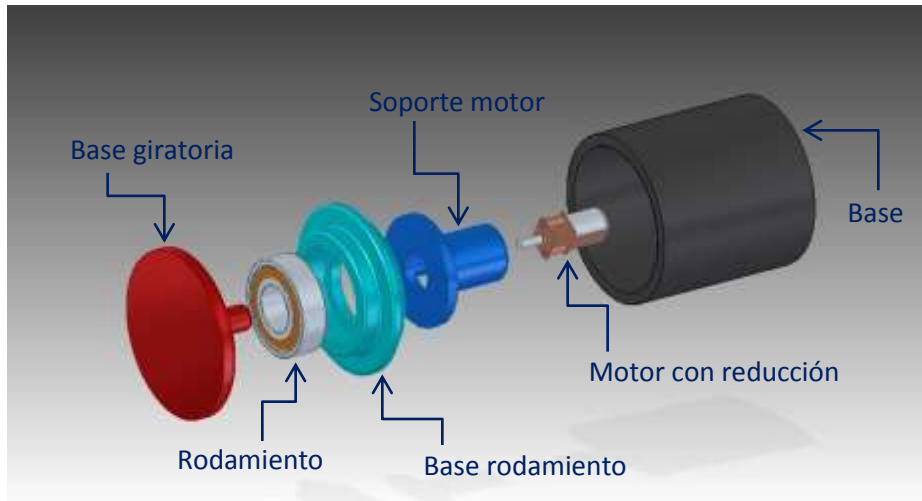


Figura 21. Explosivo del sistema de calibración de los giróscopos, completo.



Figura 22. Ensamble del sistema de calibración de los giróscopos.

4.6 PROCESAMIENTO DE LOS DATOS

Una vez hecha la calibración de los acelerómetros y giróscopos, se puede hacer un análisis estadístico de las señales arrojadas por cada uno de ellos. Obteniendo la media y la varianza de las señales se pueden formar las matrices de covarianza del error que son necesarias para el filtro de Kalman del algoritmo implementado, para corregir los errores en el cálculo de la orientación.

Como ya se mencionó, la calibración también es necesaria para obtener los valores de *offset* de los acelerómetros y giróscopos; así como los valores de referencia para poder escalar las señales a dimensiones con unidades reales, ya que el sensor arroja datos digitales de 16 bits (con un valor de 0 - 65,536).

Establecida la comunicación entre el Arduino y la computadora se hace un muestreo de los datos arrojados a una frecuencia 30 Hz, que es una tasa de muestreo suficiente para hacer la captura de movimientos naturales del cuerpo humano en condiciones moderadas de movilidad, como es usual en rehabilitación. El procesamiento de los datos de cada uno de los sensores, también se realiza con Matlab, en dónde se programó el filtro de Kalman que hará las correcciones necesarias para obtener una orientación precisa. Una vez obtenida la orientación (θ_1 y θ_2) de cada uno de los sensores (S_1 y S_2) y tomando en cuenta que estarán montados en el brazo y en el antebrazo, se realiza el cálculo de la cinemática para encontrar en ángulo que forma la articulación del codo (θ_3) (ver **figura 23**).

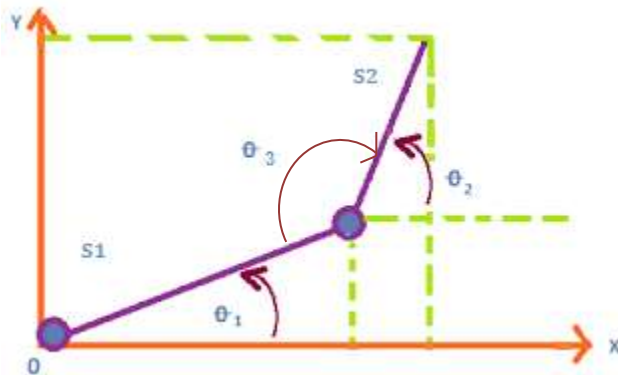


Figura 23. Cálculo de la cinemática para encontrar el ángulo que forma una articulación con un grado de libertad.

4.7 DISEÑO DE LA TARJETA DE CONTROL

Uno de los objetivos del proyecto es tener un sistema que sea inalámbrico, ergonómico, fácil de portar, que no limite el movimiento del usuario y que pueda leer varios sensores simultáneamente; todo esto hizo necesario que el diseño de la unidad de control, fuese contenido en un espacio pequeño: el microcontrolador, la antena de XBee, las baterías que alimenten el sistema y el bus para los sensores.

Se diseñó la tarjeta de manera que puedan ser conectados hasta doce sensores, esto pensando en poder ampliar el sistema. Como se mencionó anteriormente, los sensores trabajan solo con dos direcciones que pueden ser manipuladas con el pin ADO del sensor; por tanto si se desea trabajar con más de dos sensores es necesario implementar un *multiplexor*.

Para implementar el *multiplexor* se manipuló el pin ADO de cada sensor por medio de las salidas digitales del Arduino, todas las líneas se colocan a un nivel lógico bajo y solo se pone en nivel alto la salida digital conectada al sensor que se desea leer; de esta forma, siempre se está leyendo el sensor con la dirección 0X69.

Ya que los pines digitales del controlador tienen una salida de 5 volts y el sensor solo acepta un voltaje de 3.3 volts, es necesario implementar una parte de acondicionamiento para bajar el voltaje. Para esto se utilizó el regulador de voltaje CD4550 y así pasar los 5 volts a 3.3 volts de todas las salidas digitales del Arduino.

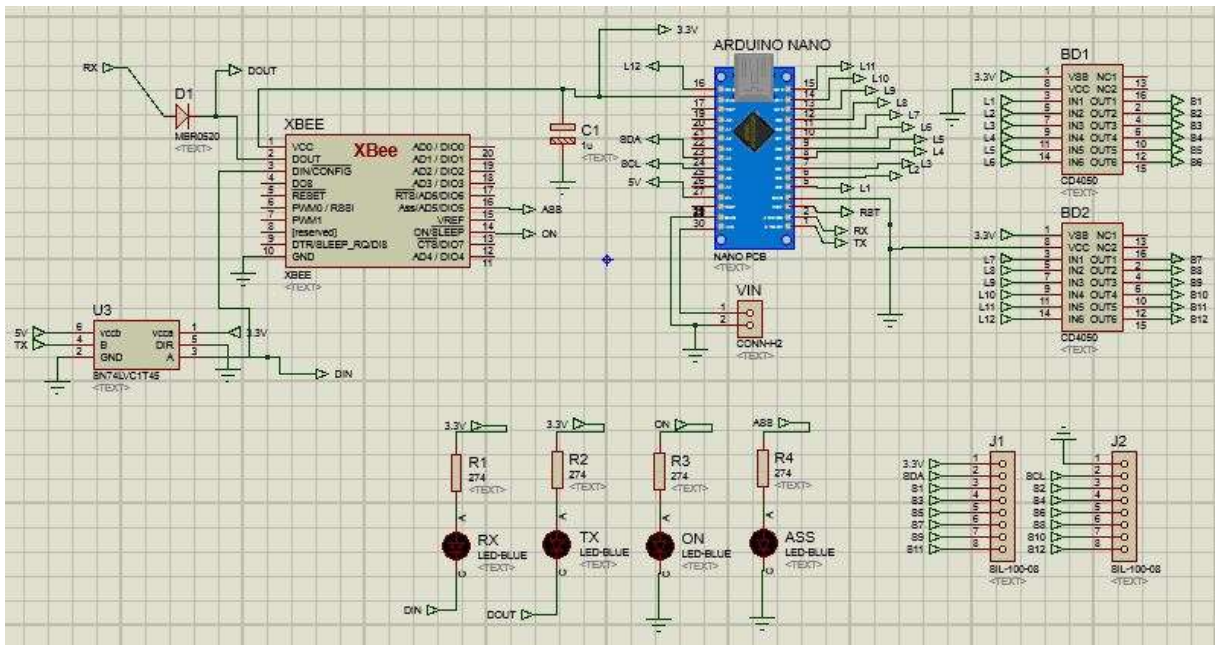


Figura 24. Esquemático del diseño de la tarjeta de prueba en el software Proteus.

El diseño de la tarjeta se realizó con ayuda del software Proteus y se intentó reducir lo más posible el tamaño final, quedando de 3.9 X 5.1 centímetros (ver figuras 24-25).

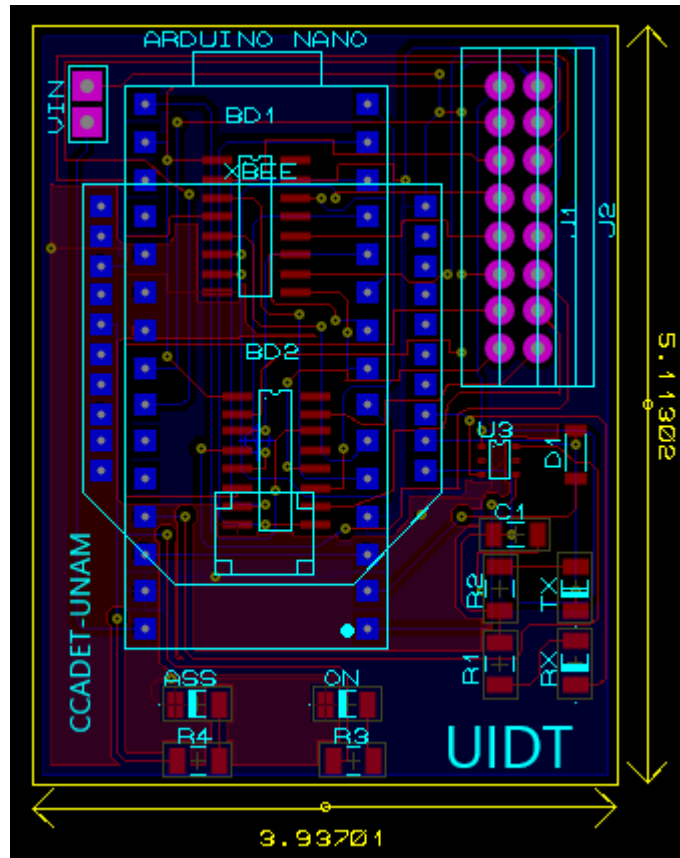


Figura 25. Diseño final de la tarjeta de control en Proteus Ares.

5 RESULTADOS EXPERIMENTALES Y VALIDACIÓN DEL SISTEMA

5.1 IMPLEMENTACIÓN DE LA ELECTRÓNICA

En la **figura 26**, se observan las conexiones, en una placa de pruebas, de la electrónica propuesta para comunicar un sensor de inercia por medio del microcontrolador y enviar los datos de forma inalámbrica a una computadora. Se implementó un divisor de voltaje para las líneas de transmisión entre el Arduino y la antena XBee, ya que como se mencionó en el capítulo anterior, las salidas del microcontrolador tienen un nivel de voltaje de 5 V y el Xbee solo puede recibir 3.3 V. Este divisor de voltaje fue sustituido por el regulador de voltaje CD4550 en el circuito impreso. En la **figura 27** se puede observar que el sensor y la computadora están comunicados de forma inalámbrica.

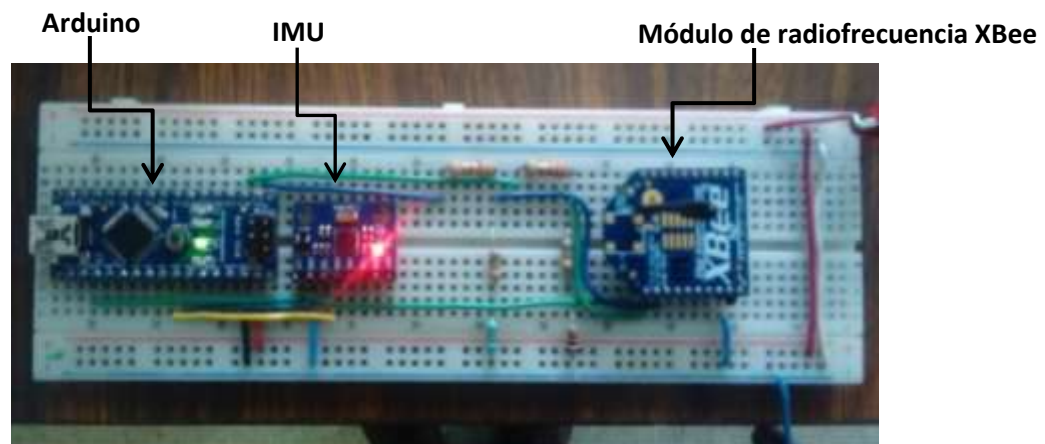


Figura 26. Circuito de la electrónica propuesta.

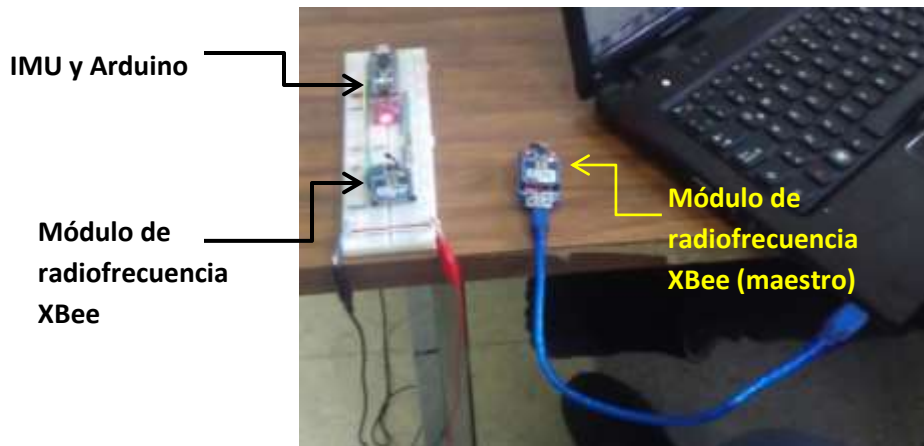


Figura 27. Comunicación inalámbrica sensor-computadora.

De las pruebas realizadas para comprobar el alcance de las antenas, se obtuvo la **tabla 2** en la que se puede observar que mientras más se aleje el sensor de la computadora, disminuye un poco el número de muestras que se obtienen haciendo el muestreo a 30 Hz por 60 segundos. Como se mencionó en el desarrollo, debido al ruido que existe en la comunicación por radiofrecuencia, se pueden obtener muestras con un mayor o menor número de datos, y estas cadenas son descartadas por nuestro algoritmo en Matlab.

Como se observa en la tabla 2, a mayor distancia existe más ruido, por lo tanto existe un mayor número de datos corruptos que son descartados por el algoritmo. Pero esta disminución no es significativa mientras se mantenga el rango de distancia especificado por el fabricante de las antenas.

Tabla 2. Pruebas de ruido y alcance de las antenas XBee.

Prueba	Distancia x	No. de muestras n
1	$x < 1 \text{ m}$	$n = 1800$
2	$1 \text{ m} < x < 5 \text{ m}$	$n = 1800$
3	$5 \text{ m} < x < 10 \text{ m}$	$1750 < n < 1800$
4	$10 \text{ m} < x < 20 \text{ m}$	$1720 < n < 1760$
5	$20 \text{ m} < x < 30 \text{ m}$	$1700 < n < 1750$
6	$x > 30 \text{ m}$	$300 < n < 600$

Además se realizaron pruebas con un programa en lenguaje C, modificando la velocidad de comunicación de las antenas (baudios), para observar como varía el número de muestras que se pueden recibir en un minuto y así aumentar la frecuencia de muestreo para realizar el seguimiento de movimientos más rápidos. En la **figura 28** se observa un comportamiento lineal del número de muestras recibidas al aumentar los baudios. Se puede observar que al llevar la

velocidad de comunicación a un valor de 74 880 baudios, las muestras llega a 14 479; por lo tanto, la frecuencia de muestreo aumenta hasta 240 Hz. Esta frecuencia de muestreo se obtuvo con la ecuación 5.28 del capítulo anterior.

Cabe mencionar que estas pruebas se realizaron sin el filtro implementado en Matlab para quitar las cadenas de datos que estén incompletas o que tengan algún dato de más; el objetivo de estas pruebas fue comprobar si utilizando un lenguaje diferente se podía obtener una mayor velocidad de muestreo.

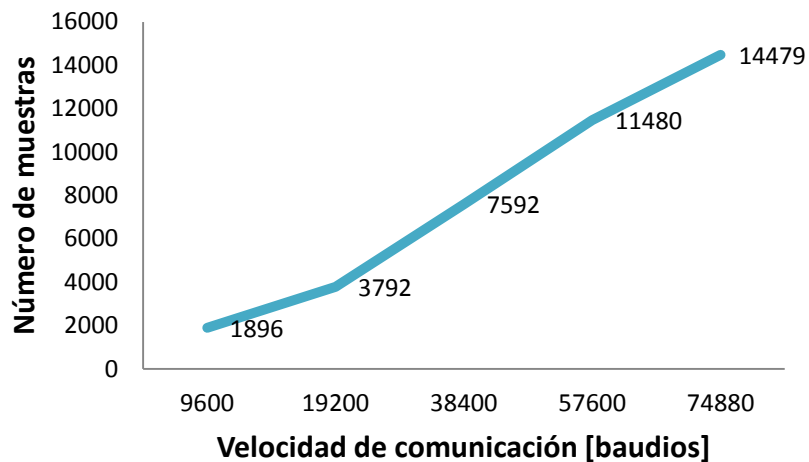


Figura 28. Gráfica de Velocidad de comunicación vs Número de muestras, utilizando un sensor inercial.

Para fines del alcance de esta tesis, como ya se mencionó, se utilizaron solo dos sensores (IMU 1 y IMU 2) (ver **figura 29**) para estimar el ángulo de una articulación; sin embargo la electrónica y el algoritmo se pueden extender con facilidad para un mayor número de sensores. Por ello, pensando en poder ampliar el sistema para hacer el seguimiento en todo el cuerpo humano, se trabajó en la electrónica y en el algoritmo para comunicar más de dos sensores en el futuro (**figura 30**).

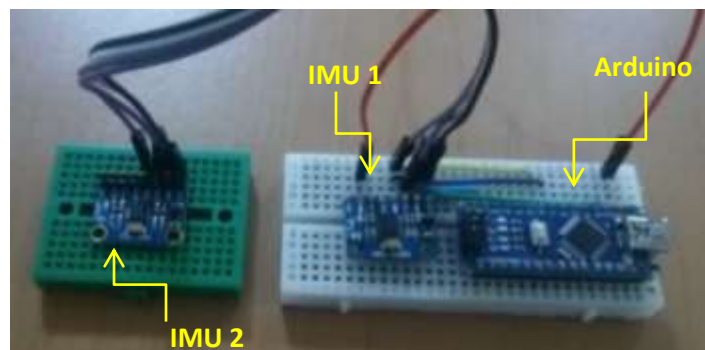


Figura 29. Electrónica de la integración de dos sensores.

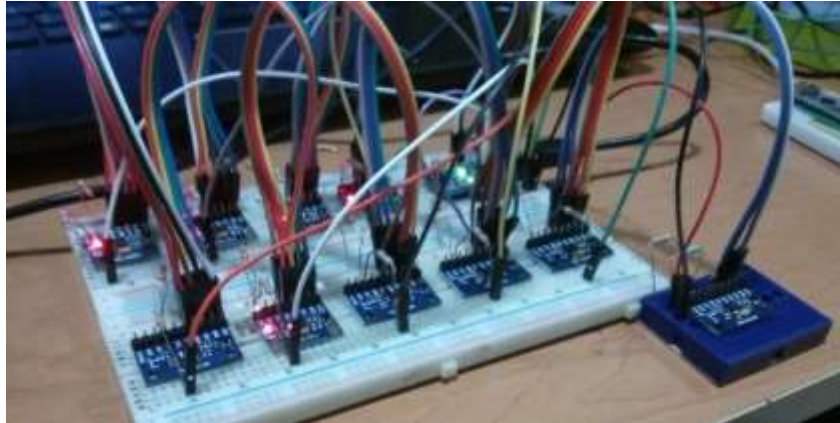


Figura 30. Electrónica propuesta para la integración de diez sensores inerciales.

En la **figura 31** se observa la gráfica del comportamiento de las muestras obtenidas comunicando dos sensores, en donde modificando la velocidad de comunicación en C, las muestras descenden casi a la mitad de las muestras reportadas con un solo sensor. Utilizando una velocidad de 74 880 baudios, obtenemos 7938 muestras; estas muestras deben ser divididas entre 2 ya que las muestras recibidas son de ambos sensores. Así, tenemos 3969 muestras por sensor y una frecuencia de muestreo alrededor de 66 Hz (utilizando ecuación 5.28).

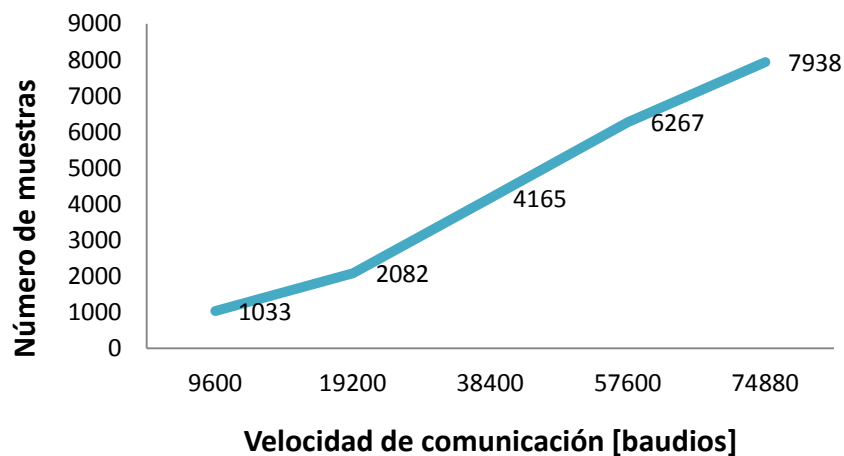


Figura 31. Comportamiento del número de muestras obtenidas, al variar la velocidad de comunicación para dos sensores inerciales.

Al realizar las pruebas de comunicación para diez sensores, se obtuvo la gráfica de la **figura 32**. En esta gráfica se puede observar una disminución bastante significativa en el número de muestras; las 9788 muestras obtenidas a una velocidad de 74 880 baudios, deben ser divididas

entre los diez sensores dejando un número de 978 muestras por cada sensor, disminuyendo la frecuencia de muestreo hasta 16 Hz aproximadamente (utilizando ecuación 5.28).

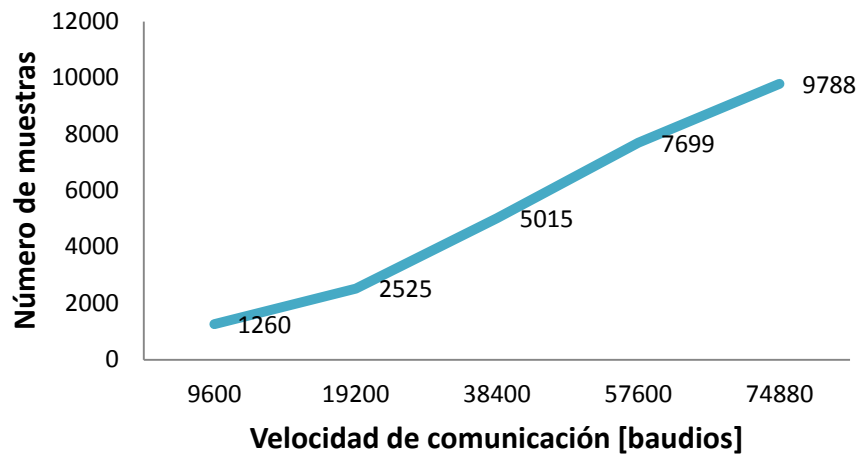


Figura 32. Gráfica de velocidad de comunicación vs número de muestras utilizando diez sensores inerciales.

5.2 TARJETA DE CONTROL

Una vez que se realizaron todas las pruebas y correcciones de la electrónica, y se comprobó el desempeño de la comunicación por medio de las antenas de radiofrecuencia; se desarrolló la tarjeta de control para utilizar hasta 12 sensores en paralelo. La placa de circuito impreso (PCB), se fabricó con ayuda del departamento de Electrónica del CCADET (ver **figura 33**). Se diseñó de forma que en un espacio de aproximadamente 4x5 cm se tiene el controlador, la antena y el circuito de acondicionamiento.

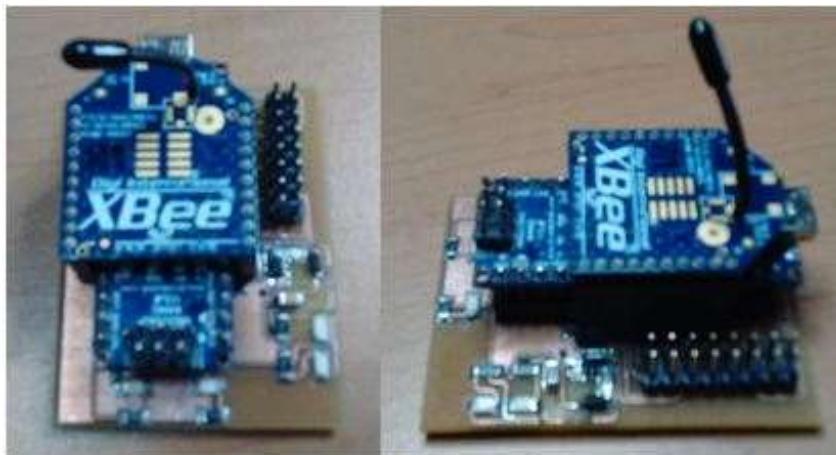


Figura 33. Tarjeta de control diseñada para comunicar hasta 12 sensores inerciales inalámbricamente.

Para que el sistema sea completamente portátil e inalámbrico, se realizó una búsqueda de las baterías disponibles en el mercado para su alimentación. El circuito se diseñó para ser alimentado hasta con 12 V, ya que las líneas de alimentación van conectadas directamente al regulador de voltaje que tiene el Arduino. Las baterías que se eligieron por su tamaño, ligereza, elevada capacidad energética y resistencia a la descarga, son las de iones de litio (Li-Ion) (ver **figura 34**). Se implementaron dos baterías en paralelo de 3.7 V de 550mAh, corriente suficiente para alimentar todo el sistema.



Figura 34. Batería de iones de Litio utilizada para alimentar el sistema.

Para poder colocar la tarjeta de control sobre el cuerpo y así minimizar las conexiones con los sensores, se diseñó un armazón que contenga la tarjeta de control y la alimentación (**figuras 35-36**); además de tener la función de ser utilizado para la fase de calibración de los giróscopos.

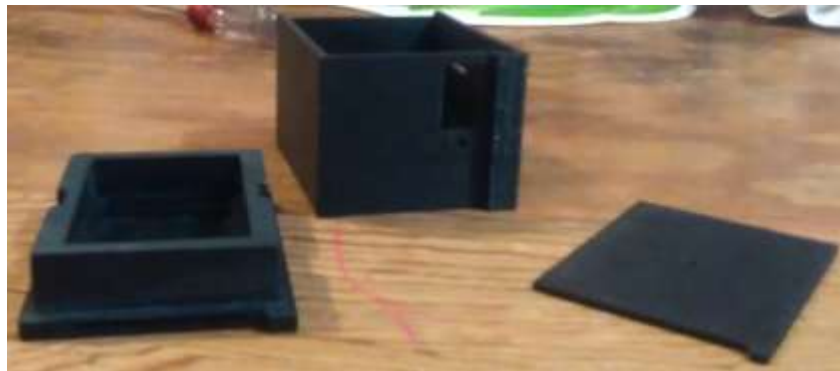


Figura 35. Armazón para la tarjeta de control y el sistema de alimentación, diseñado en NX 8.5 y fabricado en el MADiT del CCADET de la UNAM.



Figura 36. Unidad de control del sistema de estimación de orientación.

5.3 CALIBRACIÓN DE LOS SENSORES

5.3.1 ESTABILIZACIÓN DE LOS SENSORES

Para empezar a calibrar los sensores y conocer de mejor forma su funcionamiento, se realizó una prueba para comprobar el tiempo en que se estabilizan los valores que arroja dependiendo de la temperatura del mismo. Se colocó el sensor en posición estática sobre una mesa y se realizó el seguimiento del acelerómetro del eje Z, desde el momento en que se conecta y se dejó trabajar por 10 minutos continuos. Como se observa en la gráfica de la **figura 37**, el tiempo en el que se estabilizan los valores es muy corto, 0.4 segundos en promedio; por lo tanto se puede concluir que no es necesario tener un tiempo de espera para que los datos que arroja el sensor puedan ser utilizados.

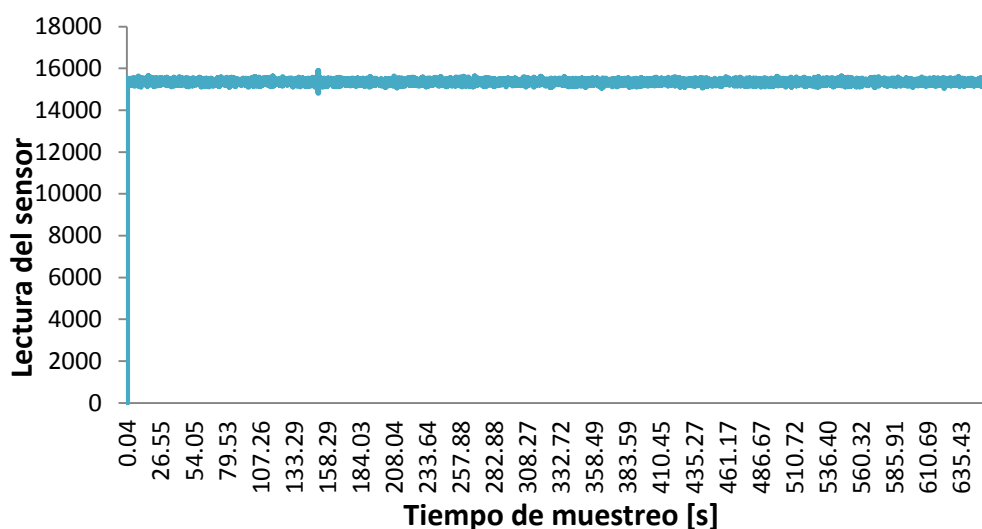


Figura 37. Gráfica de prueba de estabilización de los valores del sensor, respecto al tiempo.

5.3.2 CALIBRACIÓN DE LOS ACELERÓMETROS

De la caracterización con la primera prueba descrita en el capítulo 4 de Desarrollo, se obtuvieron gráficas como las de la **figura 38**; en estas se observa que el acelerómetro del eje que está alineado con la dirección de la fuerza de gravedad tiene un valor mucho mayor a los otros dos que están perpendiculares a esta. Como se mencionó, el valor digital arrojado por el acelerómetro alineado con la dirección de la fuerza de gravedad es equivalente a 9.8 m/s^2 , este valor nos servirá para escalar los datos que se obtengan del sensor al hacer el cálculo de la orientación.

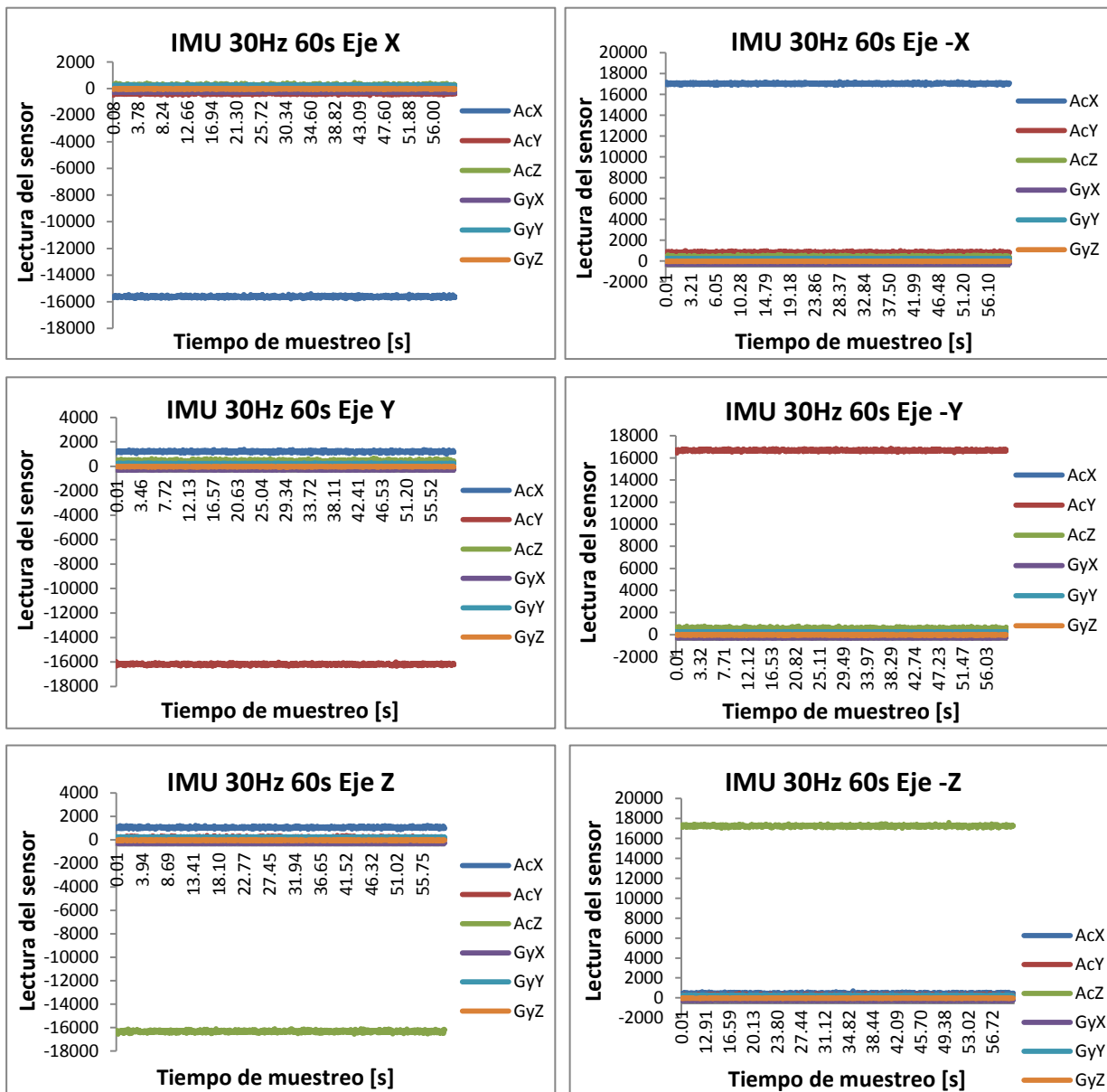


Figura 38. Gráficas de los datos obtenidos de la prueba número 1 de caracterización de los sensores.

La lectura que arrojan los acelerómetros de los ejes perpendiculares a la dirección de la fuerza de gravedad, como se puede observar en las gráficas, no dan un valor de cero aunque no están sujetos a ninguna aceleración; esto se debe al ruido inherente del sensor. La media de estos valores es el *offset* que deberá ser restado a cada valor obtenido de los acelerómetros al hacer el procesamiento de los datos y calcular la orientación.

El mismo fenómeno se observa con los giróscopos, al estar estático el sensor no se debería registrar valor alguno en estos, ya que no están sujetos a ninguna velocidad angular; sin embargo se tienen valores diferentes de cero. Del mismo modo, la media de estos valores es el *offset* que tendrá que ser compensado en el cálculo de la orientación.

5.3.3 Calibración de los giróscopos

Para obtener los valores de referencia de los giróscopos, se realizaron las pruebas con el sistema de calibración diseñado (**figura 22**). Se colocaron esferas de alta luminiscencia sobre la plataforma del sistema (**figura 39**), con el fin de utilizarlas como marco de referencia para realizar el seguimiento del movimiento de rotación del dispositivo, por medio del sistema óptico Kinect y así realizar el cálculo de la velocidad angular a la que estará girando el sensor colocado justo en el eje de rotación del mecanismo de calibración.



Figura 39. Sistema de calibración para los giróscopos, diseñado en NX 8.5 y fabricado con ayuda del MADiT de la UNAM.



Figura 40. Sistema de calibración de los giróscopos con la unidad de control y sensor montados.

Para realizar la calibración de los giróscopos, se puso a girar el sistema por 25 segundos a diferentes velocidades (**figura 40**), variando el voltaje del motorreductor; durante estos 25 segundos se hizo la captura del movimiento del sistema de calibración con el Kinect y también se guardaron los datos digitales arrojados por el sensor. Esta prueba se realizó para cada uno de los ejes del sensor con nueve velocidades diferentes.

De esta prueba de calibración se obtuvieron gráficas como las de la **figura 41**, en esta se puede observar el valor digital que arroja el giróscopo del eje Z cuando se aplican 2 volts al motor y el ruido que aparece en los otros dos giróscopos que no están sujetos a ninguna velocidad angular.

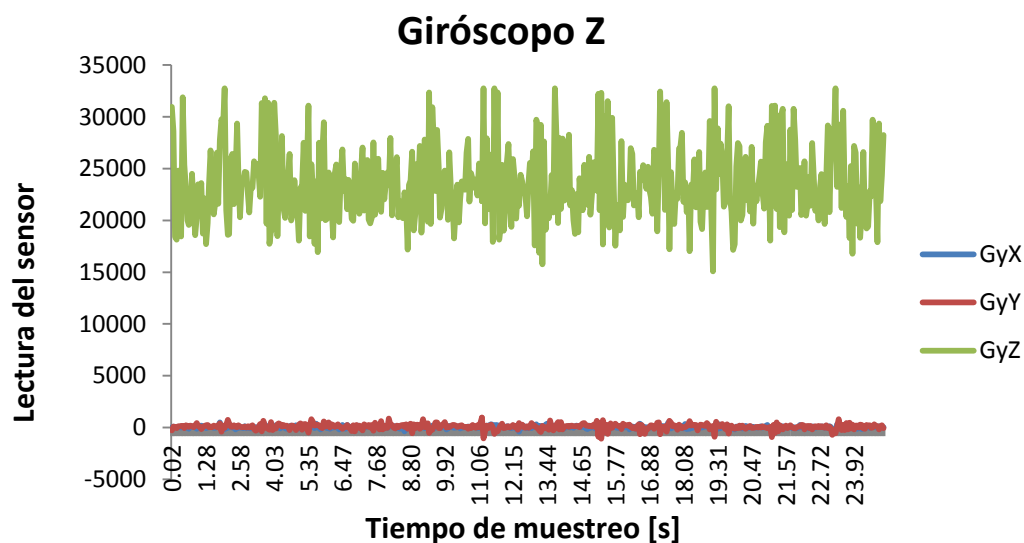


Figura 41. Gráfica de los datos obtenidos de la prueba 2 de caracterización de los sensores inerciales.

Del sistema óptico se obtuvo la velocidad angular en radianes por segundo de todo el sistema y, como se mencionó, también se guardaron los datos digitales que arroja el sensor con cada velocidad. En la tabla 3, se registraron los valores que se obtuvieron con las 9 velocidades utilizadas con uno de los sensores.

Tabla 3. Set-up para la calibración de los valores de referencia de los giróscopos.

Voltaje	X		Y		Z	
	V. Angular [rad/s]	Datos Digitales	V. Angular [rad/s]	Datos Digitales	V. Angular [rad/s]	Datos Digitales
0.7	0.672	4866.88	0.7518	6133.13	0.7022	5405.09
1	1.1567	8811.35	1.0725	8622.19	1.1193	8833.83
1.3	1.6066	12422.85	1.6066	12546.51	1.7241	13388.59
1.7	2.3591	19061.02	2.4969	19459.06	2.4514	19470.85
2	3.0909	24202.21	2.8919	22505.69	2.9844	23846.6
2.3	3.4986	28026.94	3.4896	27829.78	3.545	27825.33
2.6	4.0426	31307.8	4.1452	31583.70	4.0929	31055.56
3	4.7218	32763.59	4.7779	32682.75	4.8938	32486.04
3.5	5.5941	32767	5.5493	32767	5.6636	32767



Figura 42. Arreglo para la calibración de los valores de referencia de los giróscopos.

Como se puede observar en la tabla 3 y en las gráficas de la **figura 43**, existe un punto en el que el sensor se satura, es decir, ya no detecta cambio alguno en la velocidad y tiene un valor máximo de 32767. Conociendo la velocidad a la que gira el sensor cuando llega a ese punto podemos sacar el factor de escala para pasar a unidades reales los datos del giróscopo en la parte del procesamiento para la estimación de la orientación.

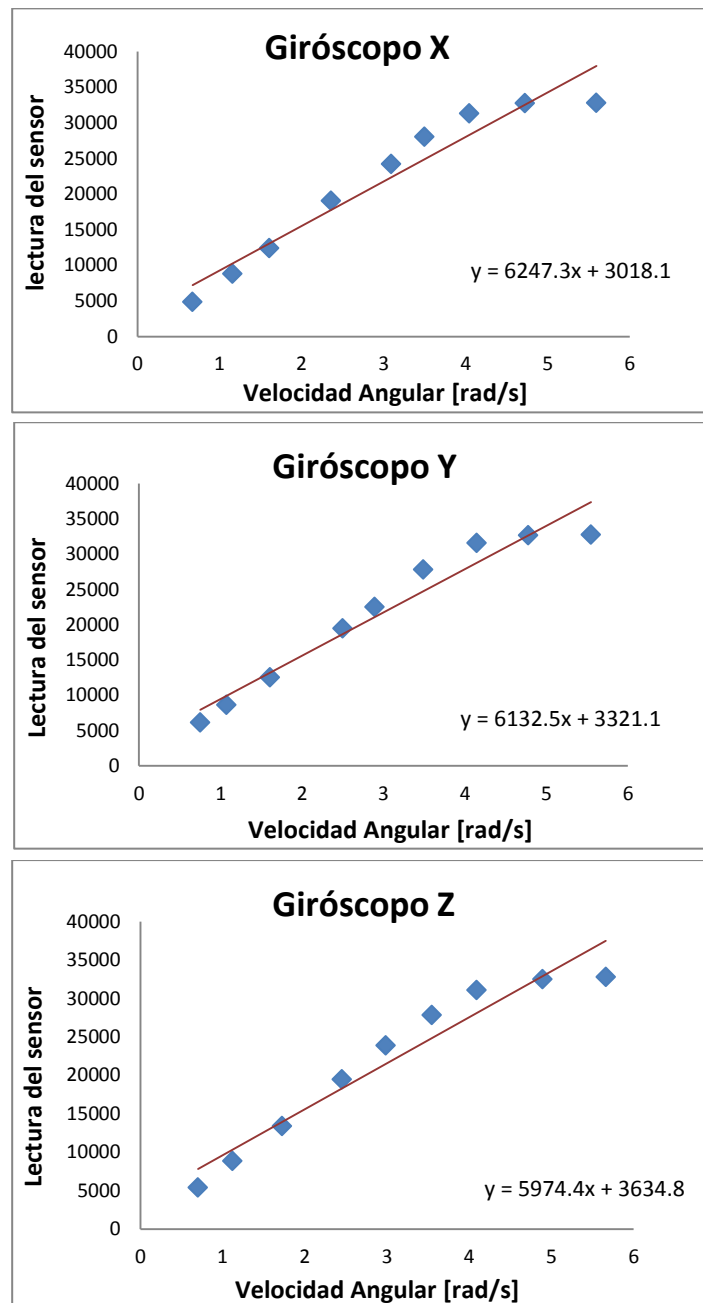


Figura 43. Gráficas de los datos obtenidos de la prueba número 2 de caracterización de los sensores inerciales.

5.4 MINIATURIZACIÓN DEL SENSOR

Una vez entendido el comportamiento de los sensores MPU-6050 y la forma de caracterizarlos, se pensó en reducir el tamaño de estos, con el fin de poder estimar movimientos finos en articulaciones complejas como las manos de manera no invasiva. El tamaño de la tarjeta de desarrollo comercial más pequeña es de 2x1.6 cm, este tamaño es conveniente para montarlo en casi cualquier parte del cuerpo; pero si se necesita hacer el seguimiento de las manos, específicamente del movimiento de los dedos, este tamaño puede limitar el movimiento natural de estos. Por esta razón se realizó la miniaturización del sensor, pensando en un trabajo futuro de implementar un sistema en forma de guante, para múltiples aplicaciones en biomecánica o rehabilitación de las manos, o bien en forma de cinturones para la evaluación de la curvatura de la espalda a nivel fino, por ejemplo en reumatismo.

El diseño de nuestra tarjeta se realizó con ayuda del software Proteus para diseño electrónico (**figura 44**). El tamaño final de nuestro diseño resultó de 1.5x1.3 cm, la manufactura del circuito impreso (PCB) del sensor prototipo se realizó con el equipo del Laboratorio de Electrónica del CCADET y quedando lista para su manufactura industrial.

Además se diseñó otra tarjeta, incluyendo los conectores necesarios para hacer la conexión de los 12 sensores que se implementaron en la unidad de control (**figura 45**). Esta diseño incluye un MPU-6050 y tiene un tamaño de 3.7x3 cm.

Estas dos tarjetas (**figura 46**) son las que se utilizaron para hacer las pruebas de estimación de orientación y la validación del sistema.

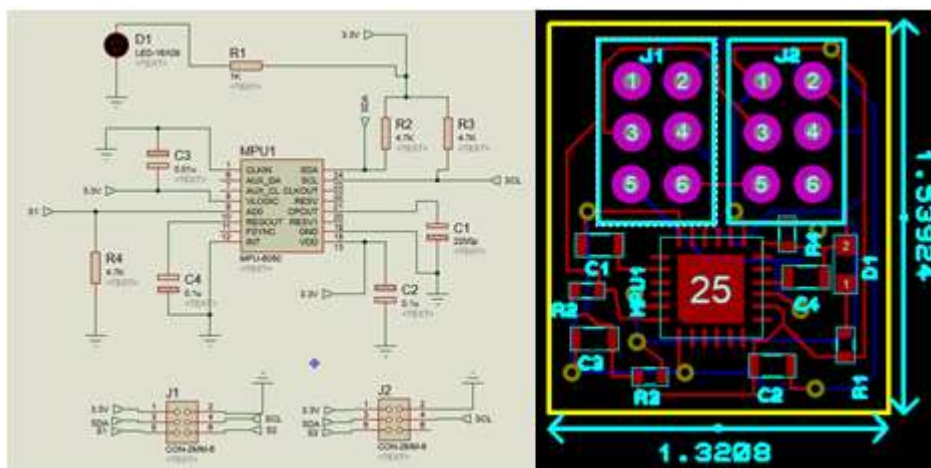


Figura 44. Diseño para la miniaturización de la tarjeta de desarrollo del sensor MPU-6050.

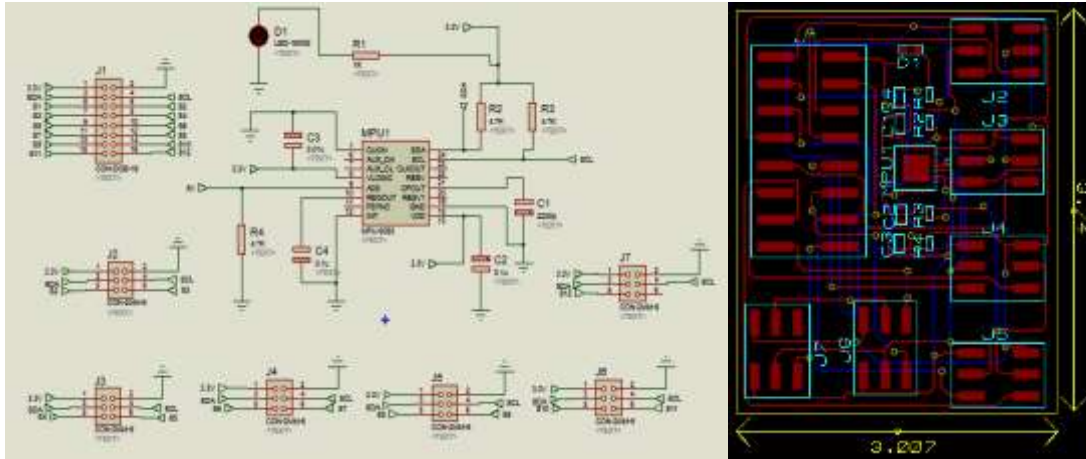


Figura 45. Diseño de la tarjeta con MPU-6050 y conectores para implementar hasta 12 sensores.

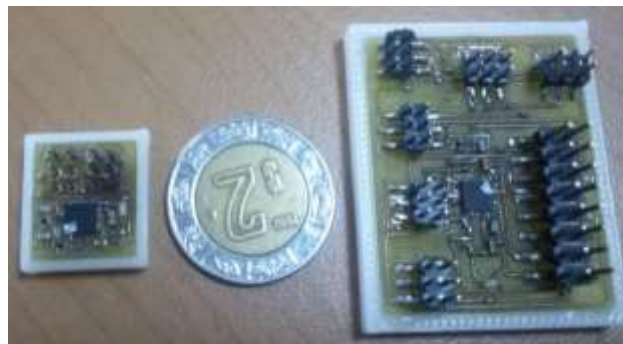


Figura 46. Tarjetas diseñadas para hacer las pruebas de la estimación de la orientación y validación del sistema, con una referencia de su tamaño.

Como se observa en la **figura 46**, las tarjetas de los sensores están montadas sobre unas bases de material plástico. Estas fueron diseñadas para tener una superficie completamente plana que nos permitiera calibrar los sensores y además montarlos en el cuerpo de manera sencilla. Se diseñaron con ayuda del software NX 8.5 para diseño CAD y se fabricaron en el MADiT del CCADDET de la UNAM (**figura 47**).



Figura 47. Bases para las tarjetas diseñadas de los sensores MPU-6050.

Pensando en el trabajo a futuro de ampliar los algoritmos para poder hacer la estimación del ángulo “Yaw” y un seguimiento del movimiento en 3 dimensiones, se hizo la miniaturización del sensor MPU-9250. Este sensor además de contar con tres acelerómetros y tres giróscopos, cuenta con tres magnetómetros. Este nuevo sensor nos proporciona una referencia para corregir la deriva del giróscopo en el eje Z.

El diseño del circuito se realizó en el software Proteus, quedando con un tamaño de 1.2 x 1.4 cm (**figura 48**). El tamaño de la tarjeta comercial es de aproximadamente 2.5x1.5 cm.



Figura 48. Tarjeta diseñada para el sensor MPU-9250 y una referencia de su tamaño.

5.5 PROCESAMIENTO DE LOS DATOS

Si se realiza un acercamiento a cualquiera de las señales obtenidas de la caracterización de los sensores, se puede observar que esta no es continua (**figura 49**), tiene oscilaciones como consecuencia del ruido. Este ruido, como ya se ha mencionado, debe ser caracterizado con el valor medio y la varianza de los datos; esto con el propósito de calcular la matriz de covarianzas de los acelerómetros y de los giróscopos. Estas matrices serán utilizadas por el filtro de Kalman implementado, para hacer la corrección de la deriva del sensor de velocidad angular.

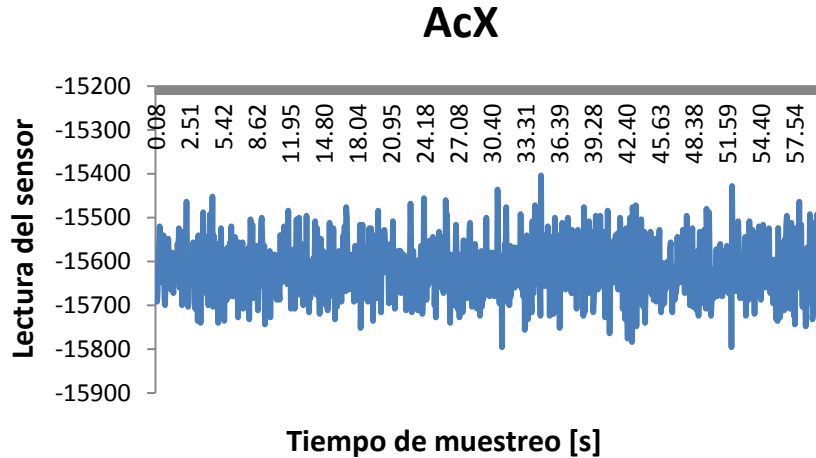


Figura 49. Gráficas de las mediciones del eje X.

Con ayuda del software Matlab, se implementó un programa para analizar las señales. Se obtuvieron las tablas 4 y 5 con los datos estadísticos de cada uno de los acelerómetros y los giróscopos de los sensores (**figura 46**). La tarjeta que solo contiene el sensor MPU-6050 será llamada Sensor 1 y nos referiremos a la tarjeta que incluye los conectores, como Sensor 2.

Tabla 4. Valores estadísticos de las señales obtenidas en el Sensor 1.

	Máximo	Mínimo	Media	Varianza
AcX	736	-180	283.87	26904.35
AcY	396	-140	127.77	6474.20
AcZ	1632	584	1132.98	21626.28
GyX	-162	-375	-239.59	1073.19
GyY	-578	-801	-666.34	2545.16
GyZ	787	667	725.32	213.22

Tabla 5. Valores estadísticos de las señales obtenidas en el sensor 2.

	Máximo	Mínimo	Media	Varianza
AcX	1564	4	699.36	170088.22
AcY	1044	-536	212.85	177165.53
AcZ	952	-36	441.65	25667.29
GyX	-195	-348	-267.71	405.70
GyY	290	168	228.06	198.21
GyZ	41	-81	-28.65	199.62

De las tablas 4 y 5 se forman las siguientes matrices de covarianza:

Sensor 1

Matriz de covarianza de los acelerómetros

$$\Sigma_A^1 = \begin{bmatrix} 26904.35 & 0 & 0 \\ 0 & 6474.20 & 0 \\ 0 & 0 & 21626.28 \end{bmatrix}$$

Matriz de covarianza de los giróscopos

$$\Sigma_G^1 = \begin{bmatrix} 1073.19 & 0 & 0 \\ 0 & 2545.16 & 0 \\ 0 & 0 & 213.22 \end{bmatrix}$$

Sensor 2

Matriz de covarianza de los acelerómetros

$$\Sigma_A^2 = \begin{bmatrix} 170088.22 & 0 & 0 \\ 0 & 177165.53 & 0 \\ 0 & 0 & 25667.29 \end{bmatrix}$$

Matriz de covarianza de los giróscopos

$$\Sigma_G^2 = \begin{bmatrix} 405.70 & 0 & 0 \\ 0 & 198.21 & 0 \\ 0 & 0 & 199.62 \end{bmatrix}$$

En la **figura 50** se observan las gráficas del comportamiento de los ángulos roll y pitch del Sensor 2 con el Filtro de Kalman implementado. Se colocó el sensor en posición estática por 10 minutos y se hizo la estimación de la orientación (cambio en el ángulo roll y pitch), se puede observar que ambos tienen una variación menor a $\pm 0.8^\circ$.

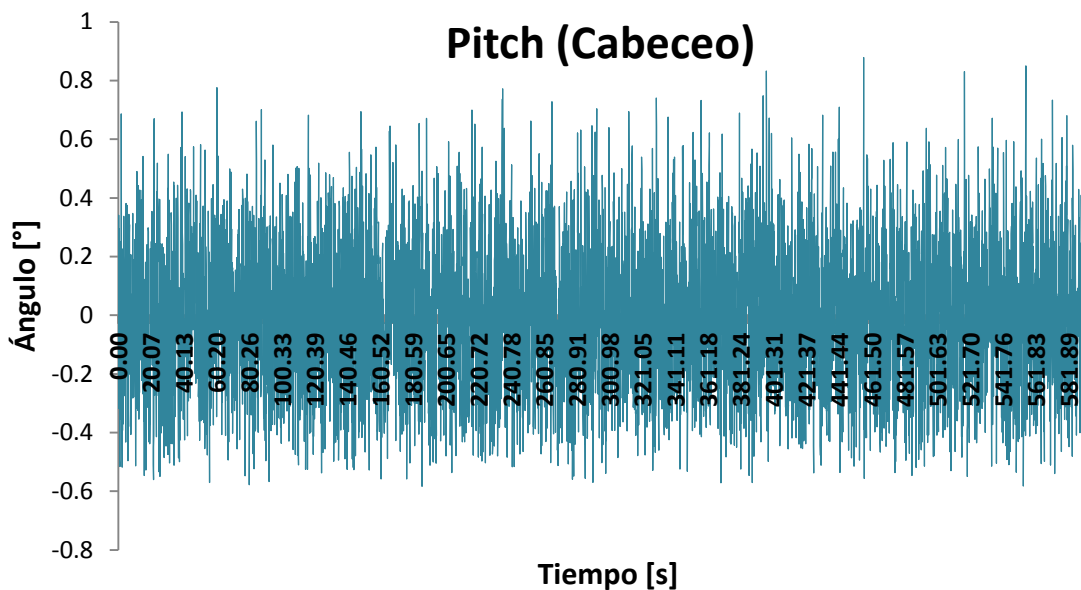
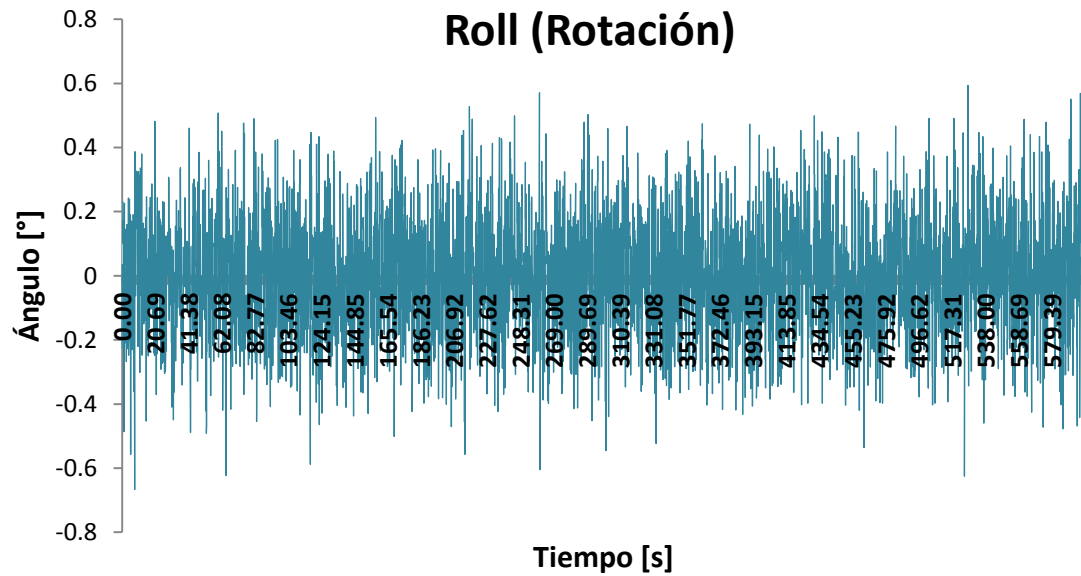


Figura 50. Comportamiento de los ángulos roll y pitch del Sensor 2 con el Filtro de Kalman implementado.

Además se realizó una prueba sin el Filtro de Kalman para comparar el comportamiento que tiene la estimación de la orientación sin el algoritmo propuesto en este trabajo. Se colocó, del mismo modo que en la prueba anterior, el Sensor 2 en posición estática y se hizo la estimación de la orientación por 10 minutos sin el filtro. En la **figura 51** se observa la comparativa de la estimación de la orientación con y sin Filtro de Kalman.

Comparativa de la estimación de la orientación

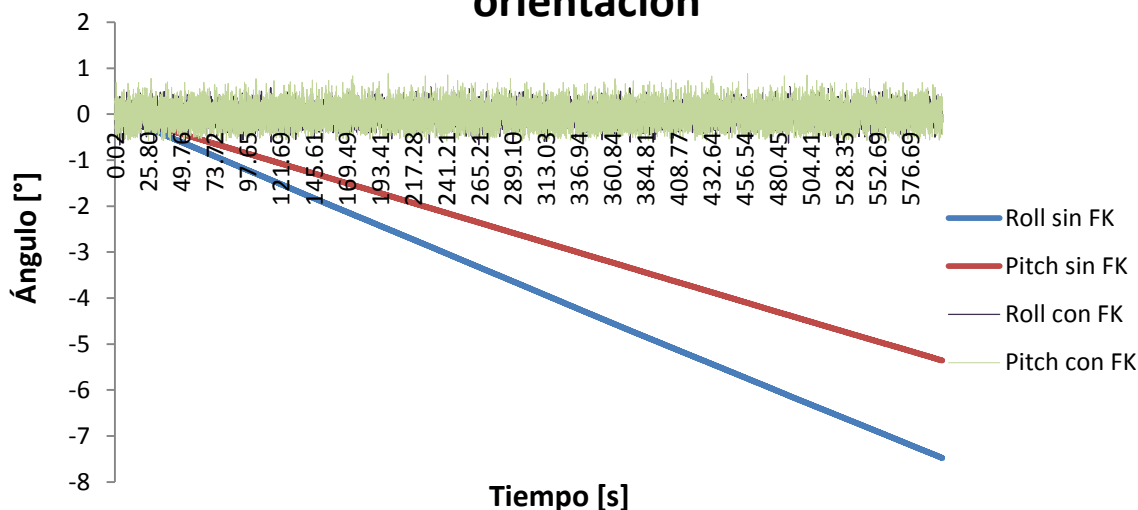


Figura 51. Comparativa de la estimación de la orientación con y sin Filtro de Kalman.

Como se puede observar en la gráfica anterior, sin el filtro de Kalman, en el transcurso de 10 minutos el cálculo de los ángulos roll y pitch tiene una deriva bastante considerable, el roll tiene una desviación de -8° y el pitch de -6° . Por tanto se puede afirmar que el algoritmo propuesto e implementado hace una correcta corrección de las derivas; ya que, como se mencionó, el roll y el pitch con la ayuda del filtro tiene una desviación de $\pm 0.8^\circ$.

5.6 VALIDACIÓN DEL SISTEMA

Para realizar la validación del sistema se utilizó la articulación del codo, se montaron los sensores inerciales en el brazo y en el antebrazo como se observa en la **figura 52**. Para la prueba de validación se hicieron movimientos de extensión y flexión hasta los 90° , frente a la cámara del sistema óptico Kinect (**figura 53**) y así poder comparar los ángulos que entrega nuestro sistema contra los datos que nos muestra el Kinect.

Para la estimación del ángulo por medio del sistema de referencia, se utilizó un programa de captura de secuencia de imágenes en infrarojo a partir del Kinect, a una frecuencia de captura de 30 Hz. El programa es capaz de detectar automáticamente en tiempo real una serie de marcadores ópticos retroreflectivos colocados en el objeto a seguir (Gastelum y Padilla 2016). De esta forma se capturaron de manera simultánea la secuencia de movimientos tanto con el sistema de referencia, como con el sistema de sensores inerciales. En el caso de los sensores

inerciales, el ángulo de la articulación se computó en tiempo real, directamente con el algoritmo de estimación de Kalman, mientras que para el sistema de referencia se realizó fuera de línea mediante un programa desarrollado en Matlab; este último calcula los vectores de orientación de los segmentos del brazo formado por los marcadores colocados en el brazo, codo y antebrazo en cada una de las imágenes de la secuencia, y luego a partir de ahí calcula la variación de posición angular en la articulación durante la secuencia.

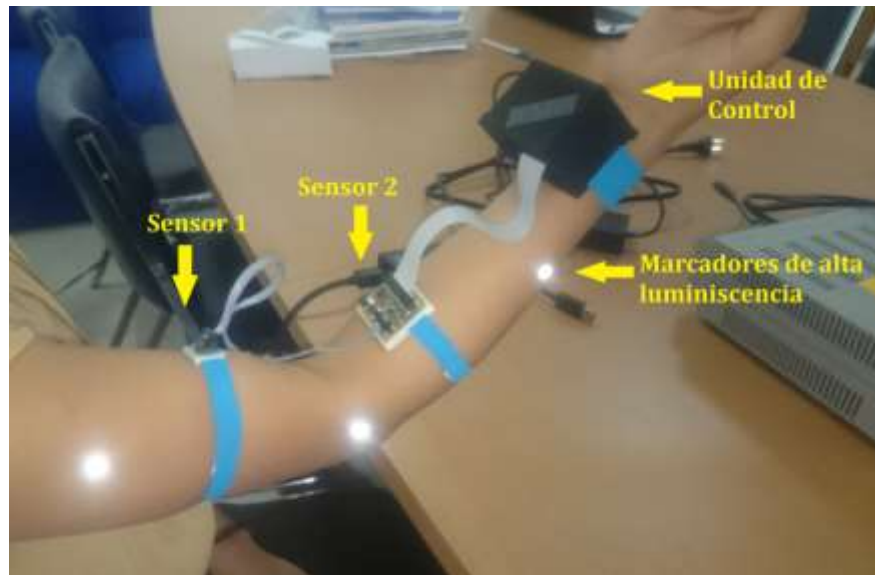


Figura 52. Prueba de validación del sistema de estimación de orientación propuesto.



Figura 53. Movimientos de extensión y flexión hasta los 90° frente al sistema óptico Kinect.

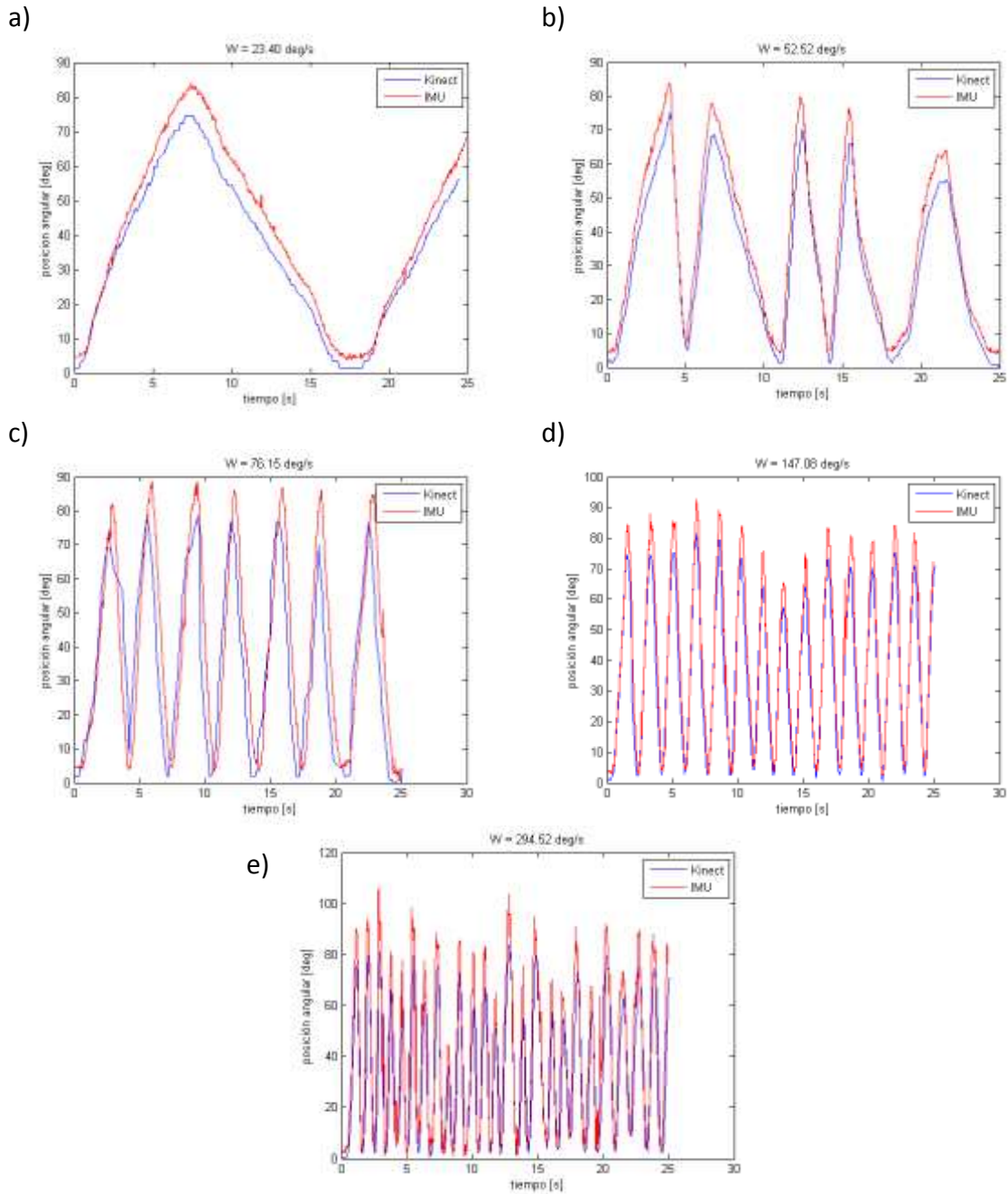


Figura 54. Estimación de la posición angular de una articulación mediante un par de IMUs (curvas en rojo) y su comparación con la estimación con un sistema de referencia usando marcadores ópticos (Kinect, curvas en azul) a diferentes perfiles de velocidad de movimientos: a) velocidad angular (W) baja; b) velocidad angular (W) media baja; c) velocidad angular (W) media; d) velocidad angular (W) media alta; e) velocidad angular (W) alta.

El valor que se reporta en las gráficas de la **figura 54**, es el ángulo suplementario del ángulo que se forma entre el brazo y el antebrazo al realizar los movimientos de flexión-extensión. Los movimientos fueron realizados por un sujeto de prueba, con el brazo suspendido en el aire y sin ninguna referencia física o visual. Para cada una de las velocidades se le pidió que realizara el movimiento cada vez más rápido, intentando mantener esta velocidad constante durante 25 segundos.

En cada gráfica se observa el comportamiento del sistema propuesto en este trabajo (línea roja) y su comparación con un sistema de referencia óptico (Kinect, línea azul). En la gráfica a) se aprecia que el ángulo no llega a los 90° , esto debido a que la persona no tiene una referencia para que el movimiento sea preciso; pero se puede ver que mientras aumenta la velocidad, los movimientos son más cercanos a este y para una velocidad muy alta (gráfica e), se sobrepasan los 90° . Mientras más rápido sea el movimiento, menos control del ángulo que forma la articulación, tiene la persona.

En cada gráfica se tiene una velocidad diferente para realizar los movimientos, siendo: a) una velocidad angular baja, b) velocidad angular media baja, c) velocidad angular media, d) velocidad angular media alta y e) velocidad angular alta. En cada una se observa que tiene un comportamiento similar al que reporta el sistema óptico, la variación observada en torno al límite del rango de movimiento alrededor de los 90° se debe a las diferencias de estimación cinemática debido a la posición de los marcadores ópticos para el Kinect; y para el sistema propuesto de sensores inerciales, se debe a que nuestro sujeto de prueba realiza los movimientos sin referencia alguna.

De hecho, la precisión del sistema óptico que se está tomando como referencia, depende de diversos factores, como la correcta alineación de los ejes imaginarios a lo largo de las articulaciones, formados por las posiciones de los marcadores en las articulaciones; la alineación de la cámara y la calibración de la cámara.

Por otro lado, al aumentar la velocidad de los movimientos (gráficas d y e) se puede ver que hay más error en el movimiento al no alcanzar los 90° en repetidas ocasiones, esto es debido a que los movimientos tienen una velocidad mayor que la velocidad a la que el sistema puede hacer la adquisición de los datos (como se planteó en el desarrollo, la velocidad de muestreo es de 30 Hz), además el tiempo de procesamiento y cálculo de la orientación de cada sensor puede aumentar el error al entregar el ángulo suplementario de la articulación. Todo esto aunado al error que la persona agrega al realizar los ejercicios.

En el caso del sistema óptico, la frecuencia a la que trabaja también es de 30 Hz y al hacer el seguimiento de movimientos más rápidos sufre de igual forma, pérdidas de información y por lo tanto errores en el cálculo del ángulo.

Por todo lo anterior, se puede considerar que una vez bien calibrados los sensores, la correcta estimación de las posiciones angulares de las articulaciones mediante nuestro sistema, depende de menos factores de error debido a que, en este caso, la orientación de cada segmento del cuerpo se realiza en el sistema de coordenadas local de cada sensor inercial y no debido a la calibración de un sistema externo.

Para mejorar la estimación del ángulo y eliminar errores del sistema de sensores inerciales propuesto, se puede cambiar el modo de realizar las pruebas de validación, cambiando el sujeto de prueba por un sistema mecánico con el que se tenga un mayor control de los movimientos.

6 CONCLUSIONES Y TRABAJO A FUTURO

En este trabajo se realizó el diseño electrónico y la implementación de los algoritmos para un sistema portátil e inalámbrico que hace la estimación de la orientación de dos sensores inerciales, que montados en alguna articulación del cuerpo humano, nos indica el ángulo que forma ésta. El objetivo de este sistema es tener una herramienta que brinde apoyo en el diagnóstico y evaluación de condiciones musculoesqueléticas o neuromotoras, permitiendo hacer una monitorización de los movimientos en condiciones y situaciones cotidianas, sin limitar el movimiento de los pacientes.

Además de haber obtenido un sistema que tiene muy buena precisión al hacer el cálculo del ángulo que forma alguna articulación, todo el trabajo realizado con la caracterización de los sensores, nos deja una metodología para posteriores trabajos y experimentos con sensores inerciales, ya que aunque pueden variar en ciertas características dependiendo del modelo y la marca, siempre es necesario conocer los valores de iniciales de estos sistemas.

Uno de los principales objetivos del sistema es que trabaje en tiempo real para tener evaluaciones inmediatas de los movimientos que realiza el paciente. En este trabajo se utilizó el software Matlab para hacer la adquisición de los datos de forma inalámbrica y el procesamiento de éstos; y aunque se realizaron pruebas de comunicación en lenguaje C, obteniendo mejores resultados en cuanto a velocidad de muestreo, los algoritmos solo se implementaron en Matlab. Quedando como trabajo a futuro implementarlos en otro lenguaje que permita una mayor velocidad de muestreo.

Por otra parte, si solo se hace el procesamiento de dos sensores (utilizando lenguaje C para la adquisición de datos), el sistema alcanza frecuencias de muestreo bastante altas que permiten hacer la monitorización de movimientos rápidos. Sin embargo, al realizar el muestreo de los doce

sensores que se implementaron en la tarjeta de control, la frecuencia de muestreo no llega a más de 16Hz, la cual no es suficiente para hacer el seguimiento.

El tipo de comunicación utilizada en este sistema, por medio de antenas de radiofrecuencia, es un medio muy utilizado y ya probado; esto nos permitió implementar de forma rápida el objetivo de que el sistema sea inalámbrico, sin embargo, existen otros medios y tecnologías que tal vez puedan aportar un canal de comunicación de mayor velocidad y menor ruido. Esto podría ser un gran factor que ayude a cumplir con el objetivo del procesamiento en tiempo real.

Como se mencionó en el capítulo anterior, en la parte de validación del sistema, es necesario hacer más pruebas con un mayor control de los movimientos para poder eliminar errores y conocer mejor la precisión que alcanza el sistema propuesto. Además se necesitan realizar pruebas para estimar la sensibilidad del sistema.

Por otra parte, la principal motivación de este trabajo fue el deseo de hacer un traje completo que realice el seguimiento en 3 dimensiones del cuerpo humano, sin embargo, por cuestiones de tiempo solo se lograron desarrollar los algoritmos para estimar orientación. Se deja para trabajo a futuro ampliar los algoritmos para hacer la estimación de la posición y la estimación del ángulo de rotación “yaw”; para realizar esto se necesita implementar un sensor que además de los acelerómetros y giróscopos, contenga otro sensor que nos proporcione una nueva señal para los movimientos que no puedan ser referenciados con la gravedad. El más utilizado para estos casos es el magnetómetro, que proporciona una referencia respecto al campo magnético de la Tierra.

Para poder utilizar estos sensores (magnetómetros), también se requiere hacer una caracterización para conocer su comportamiento y los valores equivalentes al campo magnético de la Tierra. Esto requerirá incluir en la metodología propuesta una caracterización similar en la dirección del campo magnético terrestre. Sin embargo toda la electrónica desarrollada en este proyecto, puede ser utilizada también para incluir estos nuevos sensores y alcances.

BIBLIOGRAFÍA

Atallah, L., J. Zhang, B. P. L. Lo, D. Shrikrishna, J. L. Kelly, A. Jackson, M. I. Polkey, G. -. Z. Yang, and N. S. Hopkinson. 2010. "Validation of an Ear Worn Sensor for Activity Monitoring in COPD." *Am J Respir Crit Care Med* 181.

Elguera Garcia, Jesús Felipe. 2015. "CALIBRACIÓN DE GIRÓSCOPOS PARA ELIMINAR LA DERIVA EN UN SISTEMA DE NAVEGACIÓN INERCIAL DE UN NANOSATÉLITE." Ciudad Universitaria: Universidad Nacional Autónoma de México.

Eston, R., M. Hawes, and T Reilly. n.d. *Kinanthropometry and Exercise Physiology Laboratory Manual: Tests, Procedures and Data. Volume One: Anthropometry*. 3ra. New York: Routledge Taylor & Francis Group.

MENACHE, A. & STURZA, M. A.,2003. MOTION TRACKING SYSTEM AND METHOD. *PATENTSCOPE*, Issue WO/2003/079672.

Gowitzke, Barbara, and Morris Milner. 1999. *EL Cuerpo Y Sus Movimientos. Bases Científicas*. Paidotribo.

Herrera Zuleta, Oscar Eduardo. n.d. "Comunicación I2C, Prototipo Brazo Robótico." <http://docplayer.es/18986556-Comunicacion-i2c-prototipo-brazo-robotico-bm001.html>.

Huiyu Zhou, and Huosheng Hu. 2005. "Inertial Motion Tracking of Human Arm Movements in Stroke Rehabilitation." In *IEEE International Conference Mechatronics and Automation, 2005*, 3:1306–1311 Vol. 3. doi:10.1109/ICMA.2005.1626742.

Lear dini, Alberto, Giada Lullini, Sandro Giannini, Lisa Berti, Maurizio Ortolani, and Paolo Caravaggi. 2014. "Validation of the Angular Measurements of a New Inertial-Measurement-Unit Based Rehabilitation System: Comparison with State-of-the-Art Gait Analysis." *Journal of Neuroengineering and Rehabilitation* 11 (September): 136. doi:10.1186/1743-0003-11-136.

Lee, Jung Keun, Edward J. Park, and Stephen N. Robinovitch. 2012. "Estimation of Attitude and External Acceleration Using Inertial Sensor Measurement During Various Dynamic Conditions." *IEEE Transactions on Instrumentation and Measurement* 61 (8): 2262–73.

Ligorio, Gabriele, and Angelo M. Sabatini. 2015. "A Novel Kalman Filter for Human Motion Tracking With an Inertial-Based Dynamic Inclinometer." *IEEE Transactions on Bio-Medical Engineering* 62 (8): 2033–43. doi:10.1109/TBME.2015.2411431.

Roetenberg, Daniel, Henk Luinge, and Per Slycke. 2015. "Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors." <http://www.researchgate.net/publication/239920367>.

Zhou, Huiyu, and Huosheng Hu. 2008. "Human Motion Tracking for rehabilitation—A Survey." *Biomedical Signal Processing and Control* 3 (1): 1–18. doi:10.1016/j.bspc.2007.09.001.

Zhu, Rong, and Zhaoying Zhou. 2004. "A Real-Time Articulated Human Motion Tracking Using Tri-Axis Inertial/magnetic Sensors Package." *IEEE Transactions on Neural Systems and Rehabilitation Engineering : A Publication of the IEEE Engineering in Medicine and Biology Society* 12 (2): 295–302. doi:10.1109/TNSRE.2004.827825.

A. Gastelum Strozzi, M.A. Padilla Castañeda, R. Bernardini, J.S Perez Lomeli, F. Arambula Cosío, J. Marquez Flores and R. Burgos-Vargas. Computer vision system for evaluating the Schober's test. XIV Mexican Symposium of Medical Physics. México D.F, March 18-21 2016. ISBN: 978-0-7354-1404-4

Xsens Technologies, <http://www.xsens.com>.

Invensense, <http://www.invensense.com/>

Digi International, www.digi.com/es/

Xbee, Hoja de especificaciones, <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>

Anexo A

Programa en Arduino para leer datos de una IMU

```
#include <Wire.h>

//Direccion I2C de la IMU
#define MPU 0x68

//MPU-6050 da los valores en enteros de 16 bits
//Valores sin refinar
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;

void setup()
{
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
}

void loop()
{
  //Leer los valores del Acelerometro de la IMU
  Wire.beginTransmission(MPU);
  Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true); //A partir del 0x3B, se piden 6 registros
  AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
  AcY=Wire.read()<<8|Wire.read();
  AcZ=Wire.read()<<8|Wire.read();
  //AcX=100; //Cada valor ocupa 2 registros
  //AcY=200;
  //AcZ=300;

  //Leer los valores del Gir6scopo
  Wire.beginTransmission(MPU);
  Wire.write(0x43);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true); //A diferencia del Acelerometro, solo se piden 4 registros
  GyX=Wire.read()<<8|Wire.read();
  GyY=Wire.read()<<8|Wire.read();
```



```

GyZ=Wire.read()<<8|Wire.read();
//GyX=400;
//GyY=500;
//GyZ=600;

//Mostrar los valores por consola
/*
Serial.print("#");
//Serial.print(AcX);
Serial.print(","); Serial.print(AcY); Serial.print(","); Serial.print(AcZ); Serial.print(",");
//Serial.print(GyX);
Serial.print(","); Serial.print(GyY); Serial.print(","); Serial.print(GyZ); Serial.print(",");
Serial.print("$");
//Serial.print("");
*/
Serial.print("#");
Serial.print(AcX); Serial.print(",");
Serial.print(AcY);Serial.print(",");
Serial.print(AcZ); Serial.print(",");
Serial.print(GyX); Serial.print(",");
Serial.print(GyY); Serial.print(",");
Serial.print(GyZ); Serial.print(",");
Serial.print("$");
}

```

Programa en Arduino para leer datos de dos IMU

```

#include <Wire.h>

//Direccion I2C de la IMU
#define MPU1 0x68
#define MPU2 0x69

//MPU-6050 da los valores en enteros de 16 bits
//Valores sin refinar
int16_t AcX1, AcY1, AcZ1, GyX1, GyY1, GyZ1, Temp1, AcX2, AcY2, AcZ2, GyX2, GyY2, GyZ2, Temp2;

void setup()
{
  Wire.begin();
  Wire.beginTransmission(MPU1);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Wire.beginTransmission(MPU2);
  Wire.write(0x6B);

```

```

Wire.write(0);
Wire.endTransmission(true);
Serial.begin(9600);
}

void loop()
{
  //Leer valor del sensor de Temperatura del IMU 1
  Wire.beginTransmission(MPU1);
  Wire.write(0x41); //Pedir el registro 0x41 - corresponde al sensor de temperatura
  Wire.endTransmission(false);
  Wire.requestFrom(MPU1,2,true); //A partir del 0x41, se piden 2 registros
  Temp1=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros

  //Leer los valores del Acelerometro de la IMU 1
  Wire.beginTransmission(MPU1);
  Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
  Wire.endTransmission(false);
  Wire.requestFrom(MPU1,6,true); //A partir del 0x3B, se piden 6 registros
  AcX1=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
  AcY1=Wire.read()<<8|Wire.read();
  AcZ1=Wire.read()<<8|Wire.read();

  //Leer los valores del Gir6scopo de la IMU 1
  Wire.beginTransmission(MPU1);
  Wire.write(0x43);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU1,6,true);
  GyX1=Wire.read()<<8|Wire.read();
  GyY1=Wire.read()<<8|Wire.read();
  GyZ1=Wire.read()<<8|Wire.read();
  Wire.endTransmission(true);

  //Leer valor del sensor de Temperatura del IMU 2
  Wire.beginTransmission(MPU2);
  Wire.write(0x41); //Pedir el registro 0x41 - corresponde al sensor de temperatura
  Wire.endTransmission(false);
  Wire.requestFrom(MPU2,2,true); //A partir del 0x41, se piden 2 registros
  Temp2=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros

  //Leer los valores del Acelerometro de la IMU 2
  Wire.beginTransmission(MPU2);
  Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
  Wire.endTransmission(false);
  Wire.requestFrom(MPU2,6,true); //A partir del 0x3B, se piden 6 registros

```

```

AcX2=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
AcY2=Wire.read()<<8|Wire.read();
AcZ2=Wire.read()<<8|Wire.read();

//Leer los valores del Gir6scopo de la IMU 2
Wire.beginTransmission(MPU2);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(MPU2,6,true);
GyX2=Wire.read()<<8|Wire.read();
GyY2=Wire.read()<<8|Wire.read();
GyZ2=Wire.read()<<8|Wire.read();
Wire.endTransmission(true);

//Mostrar los valores por consola
Serial.print("#");
Serial.print("1"); Serial.print(",");
Serial.print(Temp1); Serial.print(",");
Serial.print(AcX1); Serial.print(",");
Serial.print(AcY1);Serial.print(",");
Serial.print(AcZ1); Serial.print(",");
Serial.print(GyX1); Serial.print(",");
Serial.print(GyY1); Serial.print(",");
Serial.print(GyZ1); Serial.print(",");
Serial.print("$");
Serial.print("#");
Serial.print("2"); Serial.print(",");
Serial.print(Temp2); Serial.print(",");
Serial.print(AcX2); Serial.print(",");
Serial.print(AcY2);Serial.print(",");
Serial.print(AcZ2); Serial.print(",");
Serial.print(GyX2); Serial.print(",");
Serial.print(GyY2); Serial.print(",");
Serial.print(GyZ2); Serial.print(",");
Serial.print("$");
}

```

Programa en Matlab para obtener y guardar datos del sensor.

```

%Lee datos del puerto serial y guardarlos en un archivo de excel.

clear all
clc

%borrar previos
delete(instrfind({'Port'}, {'COM3'}));

```

```

%crear objeto serie
ardu = serial('COM3','BaudRate',9600);
%warning('off','MATLAB:serial:fscanf:unsuccessfulRead');

%abrir puerto
fopen(ardu);

% parámetros de medidas
tmax = 15; % tiempo de captura en s
fm = 43; % frecuencia de muestreo en Hz
tm = 1/fm;

% inicializar
titulos = {'dt','AcX','AcY','AcZ','GyX','GyY','GyZ'};
t = 0;
%d=[0 0 0 0 0 0];
a=[0 0 0 0 0 0];
%a=[];

%creación del archivo
xlswrite('IMUGy_30Hz_30s_axisY_3.5V.xls',titulos);

% ejecutar bucle cronometrado
tic
D = [];
FRAME = [0 0 0 0 0 0];
i=1;
primerdato = 0;
ts = toc;
while t<tmax
    t = toc;
    dt = t - ts;
    if dt >= tm,
        imudata = readIMU( ardu );
        if length(imudata)>0,
            if primerdato == 0,
                imudataold = imudata;
                primerdato = 1;
            end;
            if ( abs(imudata - imudataold) < 300000 ),
                %D = [D; [t imudata]];
                D(i,:) = [t imudata];
                i=i+1;
                imudataold =imudata;
            end;
        end;
        ts = toc;
    end;
end

%res = d';

% mandar a excel
xlswrite('IMUGy_30Hz_30s_axisY_3.5V.xls',D,1,'A2');

```

```

%Limpiar
fclose(ardu);
delete(ardu);
%clear ardu;

```

Programa en Matlab con algoritmo de estimación de orientación.

```

%Lee datos del puerto serial y guardarlos en un archivo de excel.

```

```

clear all
clc
%hold on

```

```

%borrar previos
delete(instrfind({'Port'},{'COM4'}));

```

```

%crear objeto serie
ardu = serial('COM4','BaudRate',9600);
%warning('off','MATLAB:serial:fscanf:unsuccessfulRead');

```

```

%abrir puerto
fopen(ardu);

```

```

% parámetros de medidas
tmax = 60; % tiempo de captura en s
fm = 43; % frecuencia de muestreo en Hz
tm = 1/fm;

```

```

% inicializar
%titulos = {'dt','AcX','AcY','AcZ','GyX','GyY','GyZ'};
t = 0;
%d=[0 0 0 0 0 0];
a=[0 0 0 0 0 0];
%a=[];

```

```

%creación del archivo
%xlswrite('IMU_30Hz_60s_axisZ_up.xls',titulos);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Filter initialization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
g = 9.8; %Gravity constant
dt = tm;
Zero33 = zeros(3,3);
I33 = eye(3); %Identity Matrix
Ca = 0.0000001;

```

```

scaleG = 1/10;
SIGMAg = (scaleG*scaleG)*[1073.19685 0 0; 0 2545.16405 0; 0 0 213.222899];
STDg = sqrt(SIGMAg);

```

```

SIGMAa = [0.00960523 0 0; 0 0.00229897 0; 0 0 0.00774327];
STDa = sqrt(SIGMAa);

Ya = zeros(3,1);           %Accelerometer signals from the IMU sensor
Yg = zeros(3,1);           %Gyroscope signals from the IMU sensor

priorX = zeros(3,1);       %A priori State vector at time step k
priorX_prev = zeros(3,1);  %A priori State vector at time step k-1
postX = zeros(3,1);        %A posteriori State vector at time step k
postX_prev = zeros(3,1);   %A posteriori State vector at time step k-1

PHI_prev = zeros(3,3);     %Transition Matrix

w_prev = zeros(3,1);       %Process Noise
v_prev = zeros(3,1);       %Measurement Noise

H = g * I33;               %The observation matrix
M = zeros (3,3);           %Measurement noise covariance matrix
z = zeros (3,1);           %The mesurement Vector

priorP = zeros(3,3);       %The a priori error covariance matrix
postP = zeros(3,3);        %The a posteriori error covariance matrix
postP_prev = zeros(3,3);   %The a posteriori error covariance matrix

Q_prev = zeros (3,3);      %The process noise covariance matrix

postZ = [0 0 1];           %A posteriori kinematic orientation vector (from
rotation matrix of the sensor R = [X Y Z]'
postA = zeros(3,1);        %A posteriori external acceleration (body) with
respect of the reference frame of the sensor
postV = zeros(3,1);
postPos = zeros(3,1);
rollr=0;
roll=0;
pitchr=0;
pitch=0;
R = [cos(pitch) sin(pitch)*sin(roll) sin(pitch)*cos(roll); 0 cos(roll) -
sin(roll); -sin(pitch) cos(pitch)*sin(roll) cos(pitch)*cos(roll)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%loop = 0;

% ejecutar bucle cronometrado

D = [];
FRAME = [0 0 0 0 0 0];
i=1;
primerdato = 0;

% imudata = readIMU( ardu );
% while length(imudata)==0,
%     imudata = readIMU( ardu );
% end;
%
% AcANT = imudata(1:3);
% GyANT = imudata(4:6);

```

```

%
% YanANT = [AcANT(1)-591.783601 AcANT(2) AcANT(3)-(-2091.83804)];
% YgnANT = [GyANT(1)-(-882.840321) GyANT(2)-(-235.982418) GyANT(3)-113.804818];
%
% YaANT = [YanANT(1)*0.00059845 YanANT(2)*0.00059723 YanANT(3)*0.0005845];
% YgANT = (1/(scaleG))*[YgnANT(1) YgnANT(2) YgnANT(3)];

V=[0 0 0];
tic
ts = toc;
%while t<tmax
while 1==1
    t = toc;
    dt = t - ts;
    if dt >= tm,
        imudata = readIMU( ardu,6 );
        validData = -1;
        if length(imudata)>0,
            validData = 0;
        end;

        if validData == 0,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %%%----- IMU Sensor signals -----
        -----%%
                Ac = imudata(1:3);
                Gy = imudata(4:6);
%
                IMUID = imudata(1);
%
                Temp = imudata(2);
%
                Ac = imudata(3:5);
%
                Gy = imudata(6:8);

                Yan = [Ac(1)-283.876232 Ac(2)-127.778091 Ac(3)-1132.98212];
                Ygn = [Gy(1)-(-239.592861) Gy(2)-(-666.342263) Gy(3)-(-725.328957)];

                Ya = [Yan(1)*0.00059751 Yan(2)*0.0005959 Yan(3)*0.00059837];
                Yg = (scaleG)*[Ygn(1) Ygn(2) Ygn(3)];

%
                if IMUID == 2,
%
                    validData = -1;
%
                end;
        end;
        if validData == 0,

                YaANT=Ya;
                YgANT=Yg;

                %Ya = imudata(1:3)
                %Yg = imudata(4:6)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %%%----- A Priori Phase (TIME UPDATE)%%
        ---%%

```

```

Z = postZ;
Zskew = [0 -Z(3) Z(2); Z(3) 0 Z(1); -Z(2) Z(1) 0];
PHI_prev = I33-(dt*[0 -Yg(3) Yg(2); Yg(3) 0 Yg(1); -Yg(2) Yg(1)
0]); %Skew-Symmetric matrix of the vector Yg
Q_prev = -(dt*dt) * Zskew * SIGMAg * Zskew;

%%% Compute the a priori state estimate
priorX = PHI_prev * postX_prev;
%%% Compute the a priori error covariance matrix
priorP = PHI_prev * postP_prev * PHI_prev' + Q_prev;

SIGMAacc = (1/3) * Ca * Ca * norm(postA) * norm(postA) * I33; %The
covariance matrix of the acceleration model error
M = SIGMAacc + SIGMAa;

%%% Compute de Kalman Gain
J = (H * priorP * H' + M);
Jinv = inv(J);
K = priorP * H' * Jinv;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%----- A Posteriori Phase (MEASUREMENT UPDATE) -----
-----%%%

z = Ya' - Ca * postA;

%%% Compute de a posteriori state estimate
postX = priorX + K * (z - H * priorX);
postX_prev = postX;

%%% Compute the a posteriori error covariance matrix
postP= (I33 - K * H) * priorP;
postP_prev = postP;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%----- GRAVITY COMPENSATION -----
-----%%%

%postZ = postX;
postZ = (1/norm(postX))*postX;
postA = Ya' - g * postZ;

postV = postV + tm * postA;
postPos = postPos + tm * postV;

%
% rollr = atan(postZ(2)/postZ(3));
% roll = (180*rollr)/pi;
% pitchr = atan(-postZ(1)/(postZ(2)/sin(rollr)));
% pitch = (180*pitchr)/pi;
%

rollr = atan2(postZ(2),postZ(3));
roll = (180*rollr)/pi;
pitchr = atan2(-postZ(1),(postZ(2)/sin(rollr)));
pitch = (180*pitchr)/pi;

```



```

        Rant = R;
        R = [cos(pitchr) sin(pitchr)*sin(rollr) sin(pitchr)*cos(rollr); 0
cos(rollr) -sin(rollr); -sin(pitchr) cos(pitchr)*sin(rollr)
cos(pitchr)*cos(rollr)];

        W = (1/tm)*(Rant' * R - I33);           %velocidad angular
        Wx = W(3,2);
        Wy = W(1,3);
        Wz = W(2,1);
        T = [Ya postZ' roll pitch];

        fprintf('Z =
%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f\n',Z(1),Z(2),Z(3),Ya(1),Ya(2),Ya(3)
),Yg(1),Yg(2),Yg(3));
        fprintf('Z =
%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f\n',Z(1),Z(2),Z(3),Ya(1),
Ya(2),Ya(3),Yg(1),Yg(2),Yg(3),roll,pitch);
        fprintf('Z =
%.3f,%.3f,%.3f,%.3f,%.3f\n',Z(1),Z(2),Z(3),roll,pitch);
        fprintf('Z = %.3f,%.3f,%.3f\n',Z(1),Z(2),Z(3));
        fprintf('W = %.3f,%.3f,%.3f\n',Wx,Wy,Wz);
        fprintf('Pos =
%.3f,%.3f,%.3f\n',postPos(1),postPos(2),postPos(3));
        fprintf('T = %.3f, %.3f, %.3f, %.3f, %.3f, %.3f, %.3f, %.3f
\n',T(1),T(2),T(3),T(4),T(5),T(6),T(7),T(8));
        fprintf('A = %.3f, %.3f, %.3f \n',postA(1),postA(2),postA(3));
        fprintf('roll = %.3f \n',roll);
        fprintf('pitch = %.3f \n',pitch);

        %loop = loop+1;
    end;
    ts = toc;
end;

end

%Limpiar
fclose(ardu);
delete(ardu);
%hold off
%clear ardu;

```