

**UNIVERSIDAD CATÓLICA DEL MAULE**

Facultad de Ciencias de la Ingeniería

Escuela de Ingeniería Civil Informática

**PROFESOR GUÍA**

Mary Carmen Jarur

**NOMBRE DE LA TESIS LOREM IPSUM DOLOR SIR  
AMET, CONSECTETUR ADIPISCING ELIT**

**HÉCTOR GABRIEL PEREDO URBINA**

Tesis para optar al  
Título Profesional de Ingeniero Civil Informático

TALCA, JULIO 2016

**UNIVERSIDAD CATÓLICA DEL MAULE  
FACULTAD DE CIENCIAS DE LA INGENIERÍA  
ESCUELA DE INGENIERÍA CIVIL INFORMÁTICA**

**TESIS PARA OPTAR AL  
AL TÍTULO PROFESIONAL DE INGENIERO CIVIL INFORMÁTICO**

**NOMBRE DE LA TESIS LOREM IPSUM DOLOR SIR AMET, CONSECTETUR  
ADIPISCING ELIT**

**HECTOR GABRIEL PEREDO URBINA**

**COMISIÓN EXAMINADORA**

**FIRMA**

**PROFESOR GUÍA**

**MARY CARMEN JARUR MUÑOZ**

\_\_\_\_\_

**PROFESOR COMISIÓN**

**DR. HERNÁN MAUREIRA PAREJA**

\_\_\_\_\_

**PROFESOR COMISIÓN**

**DR. MARCO ANTONIO MORA COFRE**

\_\_\_\_\_

**NOTA FINAL EXAMEN DE TÍTULO**

\_\_\_\_\_

**TALCA, JULIO 2016**

## Índice

<b>1. Introducción</b>	<b>5</b>
1.1. Objetivos Generales . . . . .	5
1.2. Objetivos Específicos . . . . .	5
1.3. Contribución Esperada . . . . .	6
1.4. Organización de la Tesis . . . . .	6
<b>2. Estado del Arte</b>	<b>7</b>
2.1. Balance . . . . .	7
<b>3. Marco Teórico</b>	<b>8</b>
3.1. Protocolo de Comunicación I <sup>2</sup> C . . . . .	8
3.2. Sistema Coordinado . . . . .	9
3.3. Sistema microelectromecánico (MPU6050) . . . . .	9
3.3.1. Características Técnicas . . . . .	10
3.3.2. Configuración de MPU6050 . . . . .	11
3.3.3. Obtención mediciones Giroscopio y Acelerómetro . . . . .	12
3.4. Arduino . . . . .	13
3.5. Framework QT . . . . .	14
<b>4. Desarrollo</b>	<b>15</b>
4.1. Introducción . . . . .	15
4.2. Programación en Arduino . . . . .	16
4.3. Creación del Software . . . . .	17
4.4. Requisitos Funcionales . . . . .	20
4.5. Requisitos No funcionales . . . . .	20
4.6. Características . . . . .	20
4.7. Cálculo ángulo . . . . .	22
4.7.1. Cálculo ángulo con acelerómetro . . . . .	22
4.7.2. Cálculo ángulo con giroscopio . . . . .	23
4.8. Fusión de Datos Sensores . . . . .	23
4.8.1. El Filtro Complementario . . . . .	23
4.8.2. Filtro de Kalman discreto . . . . .	24
4.8.3. Comparativa de Filtros . . . . .	25
4.9. Cálculo proyección del Ángulo . . . . .	26
4.10. Cálculo del Recorrido Curvo . . . . .	27

4.11. Principales Dificultades . . . . .	27
4.11.1. Representación Tiempo Real . . . . .	27
4.11.2. Orientación Sensores . . . . .	28
4.11.3. Data Terminal Ready . . . . .	28
4.11.4. Latencia en Controlador FTDI . . . . .	29
<b>5. Conclusiones y Trabajos Futuros</b>	<b>30</b>
5.1. Conclusiones . . . . .	30
5.2. Trabajos Futuros . . . . .	30
<b>6. Bibliografía</b>	<b>31</b>

## Índice de figuras

1. Diagrama I <sup>2</sup> C. . . . .	8
2. Diagrama MPU6050 . . . . .	10
3. Ejemplo curva calibración $\pm 2g$ . . . . .	13
4. Diagrama Sistema . . . . .	16
5. Diagrama Código Arduino . . . . .	17
6. Ventana Ajustes Sensores . . . . .	18
7. Ventana Principal V0.1 . . . . .	19
8. Gráficos de los sensores . . . . .	19
9. Ejemplo Herramienta de Análisis . . . . .	21
10. Gráfico de Prueba . . . . .	22
11. Diagrama Filtro Complementario . . . . .	24
12. Diagrama estados Filtro Kalman . . . . .	25
13. Comparativa Angulo Acelerómetro vs Filtro Kalman vs Filtro Complementario	26
14. Ejemplo calculo proyección Angulo . . . . .	27

## Índice de tablas

1. Características de los sensores . . . . .	10
--	----

## 1. Introducción

La comprensión de las dinámicas y patrones de movimiento humano, se facilitan enormemente con el apoyo de tecnologías adecuadas, que permitan medir, registrar y analizar variables cinemáticas. Particularmente, el estudio del movimiento humano (Kinesiología) depende hoy en día de las facilidades tecnológicas que ofrece la electrónica (sensores, microcontroladores, sistema de adquisición de señales) y las ciencias de la computación (procesamiento, análisis y posterior reporte adecuado de variables cinemáticas y cinéticas).

Las soluciones tecnológicas que hoy existen, no solo permiten realizar el seguimiento ('tracking') y representación gráfica, sino que además facilitan la representación en tiempo real de las variables de interés, conformando con esto los denominados sistemas de Bio-feedback (retroalimentación de señales biológicas). Aunque existen alternativas adecuadas para estos fines, poseen el inconveniente en lo prohibitivo de su valor comercial junto con ser soluciones cerradas tanto en hardware como software, limita las funciones según las características propuestas por fabricante, hace prácticamente imposible la adición de nuevas funcionalidades y el uso para otros fines u obtención de la información capturada por los sensores incorporados.

Frente a este desafío, está la posibilidad de integrar tecnología para conseguir soluciones de menores costos, que permitan realizar investigación y al mismo tiempo explorar posibilidades de innovación tecnológica que den paso al desarrollo de tecnología generadora de impacto no tan sólo en el ámbito nacional probablemente de carácter global.

Particularmente, este proyecto propone desarrollar un sistema de integración hardware-software que permita el registro del comportamiento pendular de la posición bípeda quieta (variables de posición y velocidad angular en el tiempo), habilitando funciones de bio-feedback que permitan replicar evaluaciones de balance postural.

### 1.1. Objetivos Generales

Diseñar e implementar un prototipo de software-hardware basado en un microcontrolador Arduino y un sensor de velocidad angular y acelerometría de 3 ejes, para el registro y representación gráfica del centro de masa y bio-realimentación.

### 1.2. Objetivos Específicos

- Integrar microcontrolador Arduino con sensor (giroscopio-acelerómetro).
- Diseñar sistema que permita el registro y visualización de todas las variables cinemáticas (posición y velocidad angular) del centro de masa.

- Construcción de un sistema que facilite mediante bio-realimentación la posición del centro de presión (proyección del centro de masa).

### **1.3. Contribución Esperada**

El desarrollo del sistema permitirá medir y registrar el comportamiento del centro de masa y su respectiva proyección (centro de presión) durante la posición bípeda quieta, al mismo tiempo que realimentará la señal como mecanismo de evaluación de balance postural e implementación de biofeedback.

El principal alcance de la propuesta es la posibilidad de contar con una solución abierta, que facilita la importación de todos los registros además del procesamiento de las señales recogidas de las evaluaciones, permitiendo además integrar la solución a otros sistemas.

El costo del sistema, respecto a las soluciones de mercado (Biodex- Balance SD), es bajo, lo que le da viabilidad económica al proyecto, permitiendo además dejar el prototipo en operación en el Laboratorio de Biomecánica.

Desde el punto de vista disciplinar (Ciencias de la Ingeniería), este proyecto materializa trabajos de investigación y colaboración transdisciplinario entre la Facultad de Ingeniería y la Facultad de Salud de la UCM. Además posiciona a los Ingenieros Informáticos como profesionales capaces de adecuarse a diferentes contextos mediante soluciones tecnológicas.

### **1.4. Organización de la Tesis**

## **2. Estado del Arte**

### **2.1. Balance**

El control del equilibrio es complejo e implica el mantenimiento de posturas [MAN10], lo que facilita el movimiento y recuperar el equilibrio. Control de balance consiste en controlar el centro de masa del cuerpo por su entre los límites de estabilidad. Evaluaciones clínicas del equilibrio pueden ayudar a evaluar el riesgo de caídas y / o determinar las causas de los trastornos del equilibrio subyacente. La mayoría de las escalas de valoración del equilibrio funcional evalúan el riesgo de caída y la necesidad de r...(line truncated)... Un tercio a la mitad de la población mayor de 65 años notifica algunos problemas con el balance o deambulación. Los pacientes con trastornos neurológicos o musculoesqueléticos son aún más propensos a tener problemas de equilibrio que afectan a su movilidad. La complejidad del control del equilibrio resultados en muchos tipos diferentes de problemas de equilibrio que necesitan la evaluación clínica sistemática para un tratamiento eficaz.

### 3. Marco Teórico

#### 3.1. Protocolo de Comunicación I<sup>2</sup>C

Para la comunicación entre el Arduino y los Sensores, y debido a la limitante del tipo de protocolo usados por el MPU6050, se debe utilizar protocolo de comunicación I<sup>2</sup>C.

El protocolo I<sup>2</sup>C fue desarrollado en 1982 por Philips Semiconductors (hoy NXP Semiconductors) para la comunicación interna entre circuitos integrados como por ejemplo juegos de CD y televisiones.

Las principales características del protocolo de Comunicación I<sup>2</sup>C [SEM14] son:

- Solo dos líneas para la comunicación son necesarias; una línea para la información serial (SDA) y otra para el reloj serial (SCL).
- Cada dispositivo conectado al bus es direccionado por software por un ID único y usando relación Esclavo/Maestro todo el tiempo.
- Posee un sistema de detección de colisiones lo que permite usar el modo multi-maestro previniendo la corrupción de los datos si dos o mas maestros comienzan a transferir datos de forma simultánea.
- Orientado a transferencias de datos en 8-bit de forma bi-direccional pueden ser realizadas hasta los 100 kbit/s en el modo Estándar, hasta 400 kbit/s en el modo Rápido, hasta 1 Mbit/s en el modo Mas Rápido, o hasta 3.4 Mbit/s en el modo de Alta-Velocidad.
- En el chip de filtrado se rechazan los picos en la línea de datos del bus de preservar la integridad de datos.

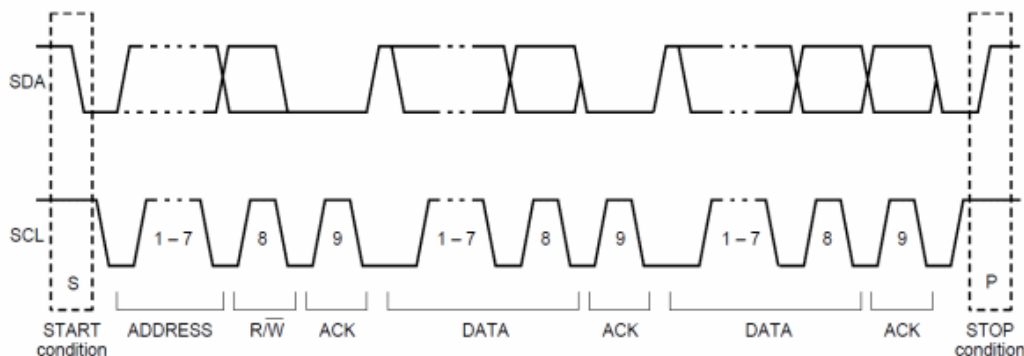


Figura 1: Diagrama I<sup>2</sup>C.



### 3.2. Sistema Coordinado

Toda la información obtenida debe estar asociado segun un sistema especifico de coordenadas, que sirva de referencia clara para la interpretacion de la información.

Para ello se tomaron en consideracion los principales sistemas a utilizar para referencias de sistemas coordinados durante el desarrollo:

- **Plano Terrestre:** El eje Z es el eje paralelo a la fuerza de gravedad, los ejes X e Y son ortogonales a éste.
- **Ejes del sensor:**
- **Plano Sagital:** Antero-Posterior y Medio-Lateral

### 3.3. Sistema microelectromecánico (MPU6050)

El MPU6050[INV13a] es un sistema Microelectromecánico (MEMS), es un dispositivo capaz de medir la velocidad angular y fuerza de gravedad gracias a estar compuesto por Giroscópico y Acelerómetro ambos de 3 Ejes, el hecho de poseer estos 2 sensores de movimiento indica que existen almenos 6 grados de libertad para utilizar al momento de obtener las mediciones provenientes de éste.

La tecnología utilizada MEMS patentada por InverSense, en el caso del giroscopio consiste en masas vibratorias verticalmente alineadas las cuales son usadas para crear un chip de bajo costo.

El MPU6050 esta diseñado para realizar seguimiento (Tracking) gracias a que posee un bajo consumo, es de bajo costo, y cumple con los requisitos de los teléfonos inteligentes, tabletas y sensores portátiles. La transmisión de datos con el microcontrolador es realizada únicamente mediante el protocolo I2C, el cual requiere de 2 líneas de conexión una para el envío de datos (SDA) y otra para el clock (SCL), las cuales varían dependiendo del modelo de microcontrolador, junto con la alimentación que consiste en la línea de voltaje(VCC) y conexión a tierra (GND).

El acelerómetro incluido mide el movimiento linear respecto a cada uno de sus tres ejes ver Figura (2), es decir la aceleración percibida por el movimiento y esta es expresada en proporción a la fuerza de gravedad (veces de g).

El giroscopio mide la rotación alrededor de cada uno de los ejes ver Figura (2), es decir la velocidad angular expresada en °/seg.

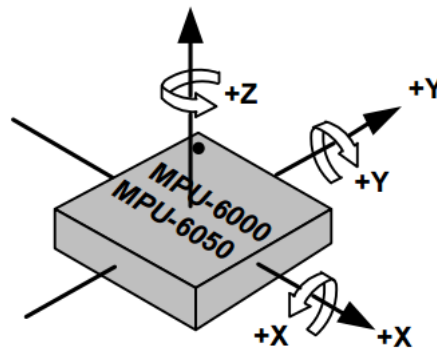


Figura 2: Diagrama MPU6050

### 3.3.1. Características Técnicas

Acelerómetro		Giroscopio	
Rango (g)	Sensibilidad (LSB/g)	Rango (°/seg)	Sensibilidad (LSB/°/seg)
±2	16384	250	131
±4	8192	500	65.5
±8	4096	1000	32.8
±16	2048	2000	16.4

Tabla 1: Características de los sensores

- **Rango:** Posee un rango configurable en los 2 sensores incluidos, tanto acelerómetro como giroscopio, el rango nos indica el dominio sobre el cual son adquiridos los datos, más concretamente los valores máximos y mínimos donde fluctúan la información obtenida.
- **Sensibilidad:** La sensibilidad describe la mínima variación medible, es decir la razón de cambio entre una variable entrante y el resultado de la variable saliente.
- **Variables:** Las variables entregadas por el Acelerómetro consiste en la aceleración lineal percibida en cada eje expresada en veces de g, y en el Giroscopio corresponde a la velocidad angular dada por la variación de los grados en un instante de tiempo al girar sobre cada eje.
- **Frecuencia de Muestreo:** Nos describe la cantidad de datos que pueden ser obtenidos en un intervalo de tiempo, a su vez nos describe la diferencia de tiempo entre cada dato obtenido.

- **Resolución:** Consiste en la mínima variación generada en la variable medida y la cual los sensores son capaces de informar.

### 3.3.2. Configuración de MPU6050

Por defecto el MPU6050 viene configurada con los parámetros básicos para realizar la captura de información, lo cual puede perfectamente ser útil según sea el caso de estudio, pero si se desea que los sensores operen en rangos específicos y/o los datos capturados por de los sensores sean obtenidos a frecuencias específicas se debe comprender como llevar a cabo la configuración en cada uno de los registros internos presentes, el mapa de los registros [INV13b] proporcionado en el sitio de InverSense es buena guía para entender como se realiza esta labor.

En el interior del MPU6050 se encuentran mas de 70 registros cada uno de 8 Bit, para la configuraciones listadas a continuación se deben ser usados solo algunos pocos.

- **Configurar frecuencia muestreo:** Para obtener la cantidad de muestras que se obtendrán finalmente “En Teoría” se calcula usando la ecuación:

$$SampleRate = \frac{GyroscopeOutputRate}{(1 + SMP_LRT\_DIV)} \quad (1)$$

En la ecuación (1) factores, tales como la velocidad del puerto serial pueden influir en el resultado, pero cambiando los datos almacenados en el registro **SMPRT\_DIV**, con valores entre 0 y 255, el valor de *GyroscopeOutputRate* depende directamente del estado del filtro pasa-bajo, en caso de estar desactivado, el valor de *GyroscopeOutputRate* sería 8Khz, en cambio si está activado el *GyroscopeOutputRate* tomaría el valor de 1Khz con lo cual haciendo los cálculos necesarios al estar desactivado el filtro pasa-bajo la frecuencia de muestreo variará entre las 8000 y 31 muestras/seg de forma teórica, al contrario en caso de estar activado el filtro, la fórmula para el calculo de la frecuencia de muestreo fluctuaría entre las 1000 y 3.9 muestras/seg, por cada uno de los 6 datos obtenibles.

Nota: Hay que tener en consideración que el valor obtenido para la frecuencia de muestreo es válido principalmente para el giroscopio, ya que para el acelerómetro se maneja con su propia frecuencia de que es siempre de 1Khz por ende si la frecuencia de muestreo es mayor a 1Khz se pueden repetir valores de en la salida del acelerómetro.

- **Configurar Giroscopio:** El registro GYRO\_CONFIG permite el cambio de los rangos para la obtención de la velocidad angular, con lo cual aumentamos o disminuimos de

forma inversa la sensibilidad de el Giroscopio. En el interior de este registro específicamente en los bits 3 y 4 del registro deben ser ingresados los valores en binario de los números 0, 1, 2 o 3 para configurar los cuatro rangos disponibles 250°/seg, 500°/seg, 1000°/seg y 2000°/seg, respectivamente.

- **Configurar Acelerómetro:** La configuración para el el rango del acelometro se realiza de forma similar, solo que en cambio el utilizado el registro ACCEL\_CONFIG, en donde también en los bits 3 y 4 se deben ingresar los valores en binario de 0, 1, 2 o 3 que representarán  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ ,  $\pm 16g$  respectivamente.

### 3.3.3. Obtención mediciones Giroscopio y Acelerómetro

Luego de realizar la configuración de los registros del MPU6050, se deben leer ciertos registros para la obtención de la información del movimiento registrado por los sensores inerciales:

- **Obtención de la Velocidad Angular:** Para obtener la variación del ángulo por unidad de tiempo medida por el giroscópico, deben ser leídos 6 registros en donde son usados por cada eje medible 2 registros de 8 bit,

- Eje X: *GYRO\_XOUT\_H*, *GYRO\_XOUT\_L*.
- Eje Y: *GYRO\_YOUT\_H*, *GYRO\_YOUT\_L*.
- Eje Z: *GYRO\_ZOUT\_H*, *GYRO\_ZOUT\_L*

Posteriormente de ser leídos ambos registros de cada eje estos deben ser sumados usando aritmética binaria en donde el valor obtenido del primer registro leído le es aplicado un corrimiento de 8 bit para luego sumar el segundo obtenido, además se utiliza el complemento a 2 para la obtención de valores negativos, que genera como dato resultante un valor de entre aproximadamente  $\pm 2^{15}$ , este valor resultante debe ser dividido según la sensibilidad asociada a el rango escogido del giroscopio, para transformar el dato RAW del giroscopio y obtener la información física en (°/seg).

- **Obtención de la Aceleración:** Para obtener la información del acelerómetro al igual que en Giroscopio deben ser leídos los 6 pertenecientes a las mediciones:
  - Eje X: *ACCEL\_XOUT\_H*, *ACCEL\_XOUT\_L*.
  - Eje Y: *ACCEL\_YOUT\_H*, *ACCEL\_YOUT\_L*.
  - Eje Z: *ACCEL\_ZOUT\_H*, *ACCEL\_ZOUT\_L*

Para luego unir los registros y usar el complemento a 2 acorde a cada eje, cada uno de estos valores deben ser dividido según el rango escogido, que determina la sensibilidad a usar para obtener la información física de la aceleración.

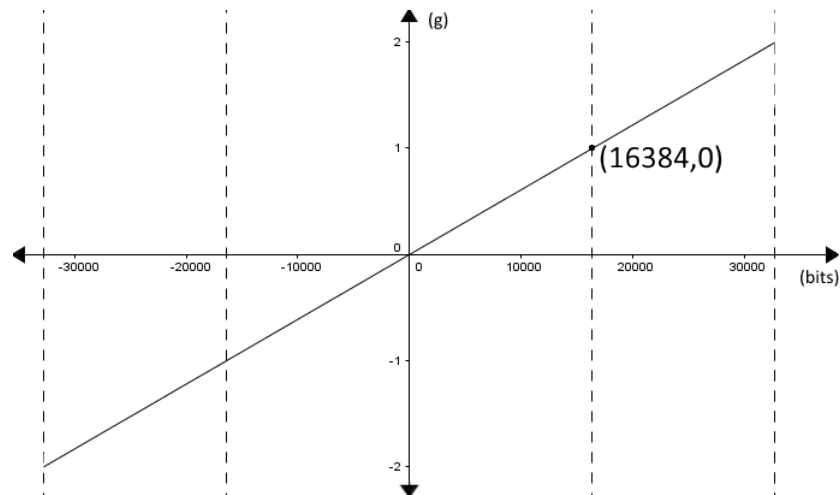


Figura 3: Ejemplo curva calibración  $\pm 2g$

### 3.4. Arduino

**Editar** Arduino[ARD13] es una plataforma de hardware de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, con un entorno de desarrollo basado en el lenguaje de programación Processing, donde sus principales características son:

- **Barato:** Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50\$.
- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- **Entorno de programación simple y claro:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino. Código abierto y software extensible: El software Arduino está publicado como

herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.

- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender como funciona y ahorrar dinero.

### 3.5. Framework QT

Qt es un Framework basado en C++, tiene disponibilidad multiplataforma- escritorio, móvil y sistemas embebidos. Posee las bibliotecas bases de C++ junto a librerías y herramientas de desarrollo propias de Qt, en donde se incluye IDE Qt Creator , y herramientas de productividad . El entorno de desarrollo (IDE) del framework de C++ Qt es Qt Creator en donde se incluyen las herramientas para la administración de los ficheros .h y .cpp usados en C++ junto con las opciones de compilación, ejecución y depuración del software.

## **4. Desarrollo**

### **4.1. Introducción**

A partir de la noción general que intrínsecamente cada uno posee de un estudio del balance y complementado con las ideas de herramientas o técnicas ya existentes que intentan solventar mediante diversos puntos de vista la problemática.

Usando de referencia principalmente los equipamientos disponibles en el Laboratorio de Bio-mecánica que sirvieron de base para el desarrollo de la solución Hardware-Software fueron la plataforma de fuerza Kistler 9286b [KIS16] y balance SD, que en ambos casos dan una representación del movimiento del centro de masa/presión y utilizando la representación entregada se ideó una solución que opera de forma análoga a los dispositivos que ya son utilizados en el laboratorio con un software que permita entregar información de manera similar a lo que entregan estos sistemas, para aprovechar los conocimientos de los usuarios con el fin de disminuir la curva de aprendizaje del sistema implementado para que estos comprendan el software sin demasiado entrenamiento y a su vez la transición sea mas satisfactoria para estos.

En lo referente a Hardware se utilizó un sensor inercial MPU6050, el cual será puesto en el sujeto de prueba en una posición aproximada al centro de masa que éste a su vez captura la información del movimiento generado por el sujeto y lo trasmite al microcontrolador Arduino en donde se recibe y traduce la información obtenida del sensor ademas de transmitir al ordenador los datos obtenidos utilizando variables físicas, tanto aceleración y velocidad angular en donde finalmente el software del ordenador recibe la información y realiza el procesamiento para obtener el ángulo y desplazamiento junto con brindar la opción de efectuar análisis específicos de la información utilizando interfaces gráficas sencillas.

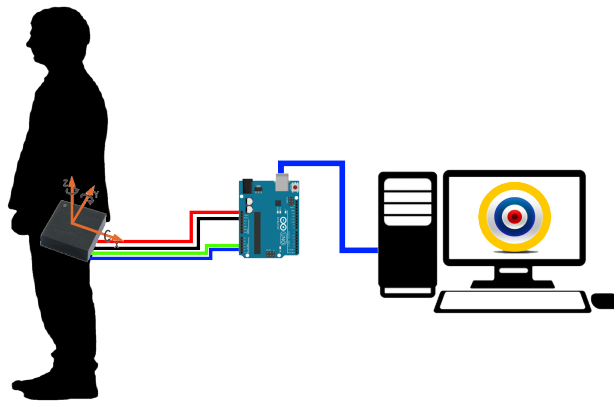


Figura 4: Diagrama Sistema

## 4.2. Programación en Arduino

Uno de los primeros pasos en el desarrollo de la solución consiste en utilizar el IDE de Arduino para realizar la programación inicial del microcontrolador, en donde se debe configurar las direcciones de los registros internos del MPU6050 usados, junto con la configuración inicial, además de definir el formato de envío para desplegar los datos por el puerto serial. En el Loop de ejecución de Arduino se obliga a configurar el MPU6050, mediante la escritura de comandos en el puerto serial, para ajustar las configuraciones de los sensores a conforme las necesidades del usuario ver Figura (5), para dar comienzo a la escritura serial de Arduino hacia PC.



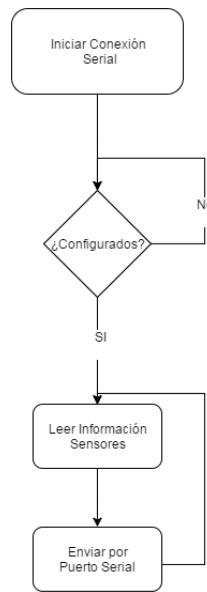


Figura 5: Diagrama Código Arduino

Existen 4 parámetros a ser configurados de forma obligatoria Frecuencia de Muestreo, Filtro pasa bajo, rango Acelerómetro y rango Giroscopio, antes de comenzar con la lectura de las de la información registros. Para configurar los 4 parámetros requeridos debe ser recibida una cadena por el puerto serial en donde se analiza e interpretan los valores a utilizar para cada configuración. Una vez se completa la configuración se procede a la lectura de los registros para obtener los valores RAW tanto acelerómetro como giroscopio para finalmente enviar mediante comunicación serial el valor resultante expresado en su unidad física correspondiente al ser dividido por la sensibilidad según el rango previamente configurado.

### 4.3. Creación del Software

Una parte importante del sistema es la interfaz que permite al usuario interactuar con el micro-controlador sin conocer mas a fondo su funcionamiento, con este proposito fue implementado utilizando en Framework QT(C++)[QT16] en su versión Open-Source, al estar basado en C++ en cuanto a estabilidad y rendimiento se refiere es una de las mejores opciones a tener en consideración.

El software permite la captura y procesamiento de información proveniente de la comunicación serial entre el micro-controlador Arduino y el ordenador, El usar un framework como Qt facilita en gran parte la creación de interfaces gráficas, capa de funcionalidad que C++ no contiene por defecto, esto posibilita una rápida integración entre la parte lógica y visual del software a desarrollar. Debido a el tipo de proyecto a implementar la metodología usada para

el desarrollo fue el método ágil, al ser una solución de software con requisitos funcionales poco definidos, más que nada pautas del resultado esperado en donde el programador pasa a ser el principal diseñador definiendo por sí mismo la distribución y comportamiento de los componentes del software.

En etapas tempranas del desarrollo se implementó una interfaz que se tan solo se comunicara con el puerto serial, pudiendo conectarse a este, y obtener la información que se encontrara disponible. Durante el desarrollo se incluyeron más opciones para una visualización mas agradable, junto con la parametrización de todos los elementos mostrados, permitiendo al usuario activar y desactivar opciones según sea el caso.

La comunicación serial entre el dispositivo-ordenador no solo es usada para leer los datos en el puerto, también es posible enviar los parámetros para realizar la configuración del sensor, para definir las opciones que consideradas más relevantes tales como: la frecuencia de muestreo, el uso del filtro Pasa-Bajo y los rangos tanto Acelerómetro como Giroscopio, usando una interfaz sencilla e intuitiva ver Figura 6.

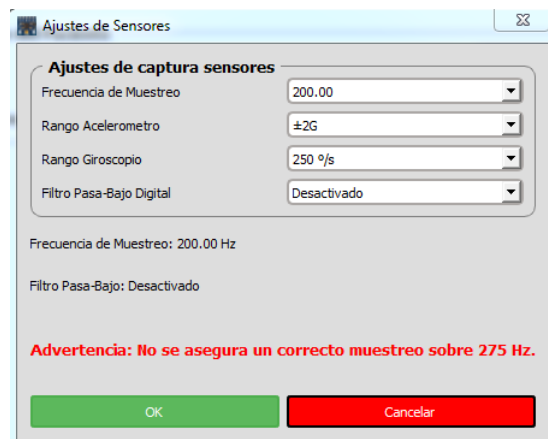


Figura 6: Ventana Ajustes Sensores

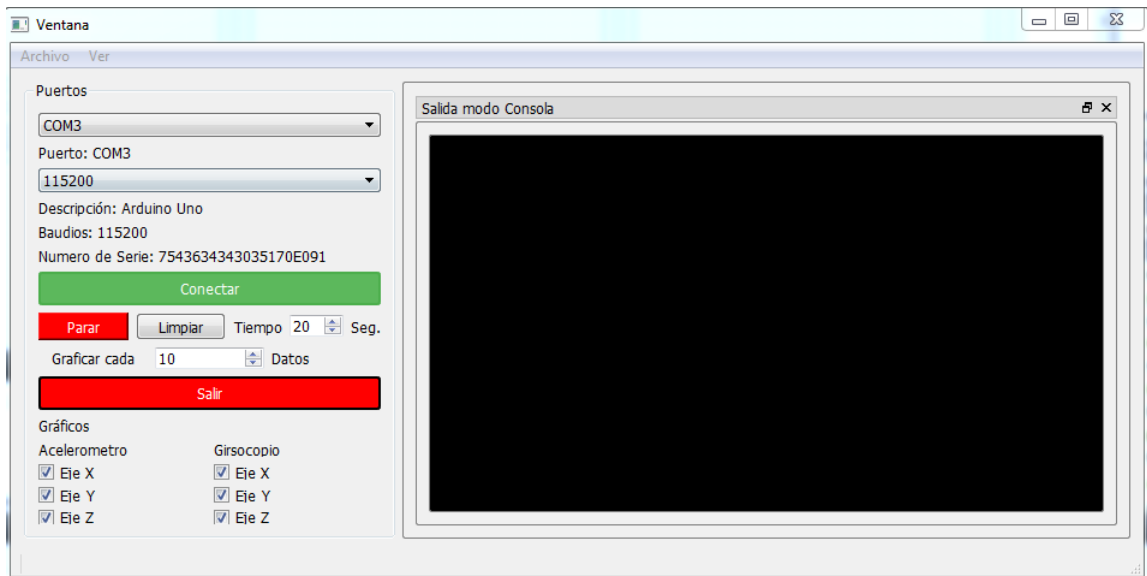


Figura 7: Ventana Principal V0.1

Luego de realizar la conexión de forma exitosa, se procede a desplegar la información obtenida, en forma de gráficos.

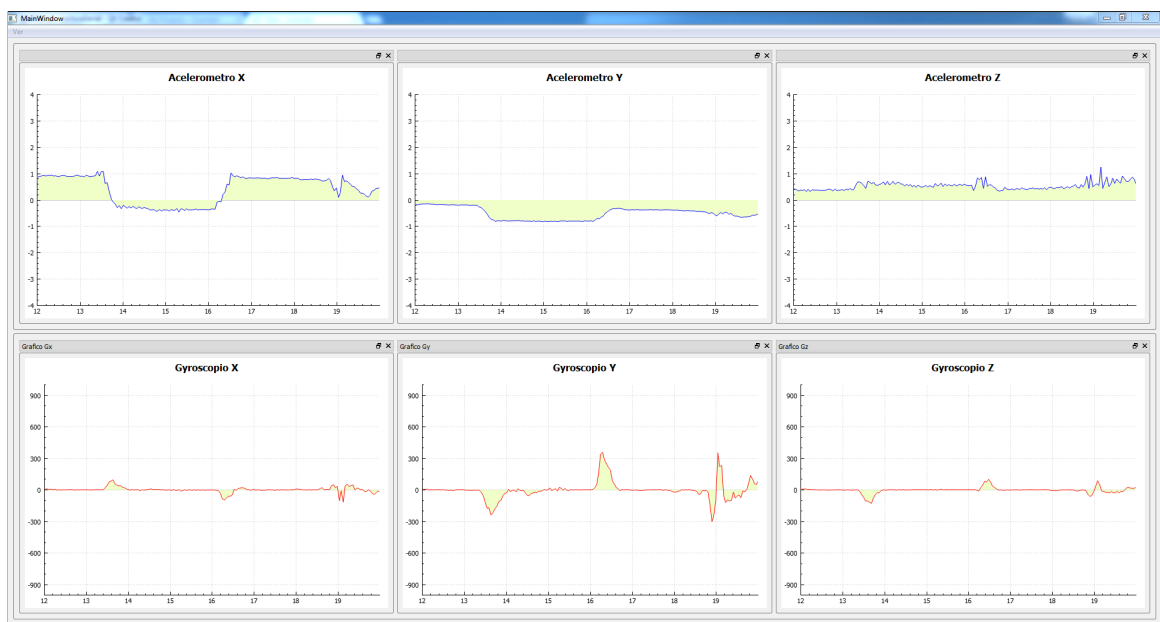


Figura 8: Gráficos de los sensores

#### 4.4. Requisitos Funcionales

- El sistema debe ser capaz de describir mediante el estudio de variables cinemáticas (aceleración y velocidad angular), procesarlas y realizar análisis kinesiólogos (desplazamiento del Centro de Masa).
- El sistema debe ser capaz de representar en Tiempo Real el desplazamiento del centro de masa.
- El sistema debe permitir la generación de reportes, con todas las variables estudiadas y analizadas durante la prueba.
- El sistema debe permitir la configuración de los Sensores, ajustar su sensibilidad rango.
- Se debe permitir la configuración de las variables involucradas en cada pruebas a realizar, partiendo por la altura desde la que fue puesta el sensor, la orientación de éste, tiempo de prueba, etc.
- En lo referente a análisis de los gráficos de resultados, debe ser posible calcular estadísticos, como media, desviación estándar y junto con los valores máximos y mínimos.

#### 4.5. Requisitos No funcionales

- El sistema debe presentar una interfaz clara y sencilla, tanto en el manejo, como para la presentación de resultados.
- El sistema debe tener un nombre representativo.
- El sistema debe permitir análisis usando los gráficos y mostrando los cambios (recortes o ajustes) en tiempo real.

#### 4.6. Características

El software permite configurar la gran mayoría de opciones de tanto velocidades de muestreo, los rangos para la captura de información del movimiento, así como lo que se representaran en pantalla, tamaños de los elementos a graficar, limitar geométricamente el gráfico, entre otros.

Se uso bastante la geometría analítica, para la generación y representación gráfica tanto de los objetivos en las pruebas, como para la intersección del movimiento realizado con cada

uno de estos. De igual manera la trigonométrica para limitar el área del movimiento según el radio que sea definido para el examen.

Reportes: Cada uno de los gráficos, los que despliegan las medidas tomadas o los que representan los resultados de el ejercicio, pueden ser almacenados en muestras de cada uno de los gráficos, o también generar una imagen de estos.

Herramientas de Análisis Gráficas: ventanas desplegables acompañadas a un gráfico que permiten de forma interactiva seleccionar partes del gráfico, para analizar intervalos según se estime conveniente y a su vez obtener de este datos como, los valores máximos y mínimos, la media, desviación estándar de los datos, y en el caso del desplazamiento obtener la velocidad media del intervalo analizado.

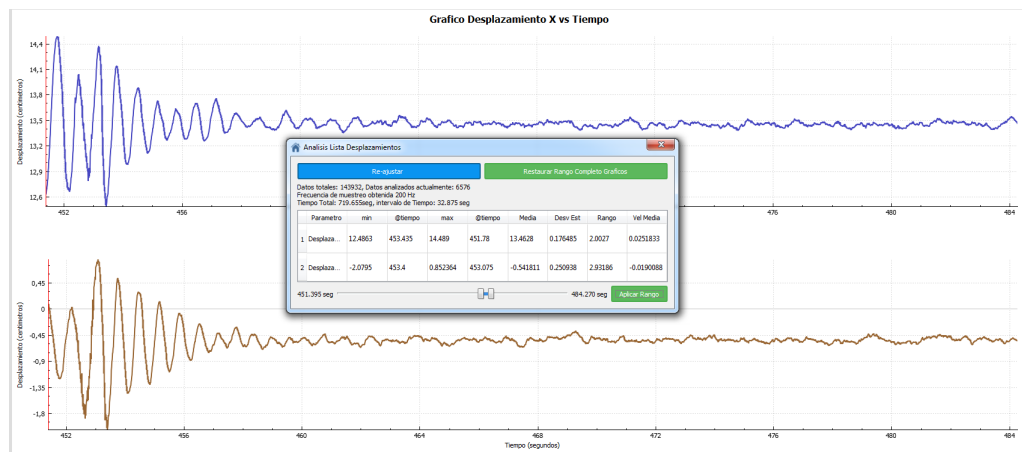


Figura 9: Ejemplo Herramienta de Análisis

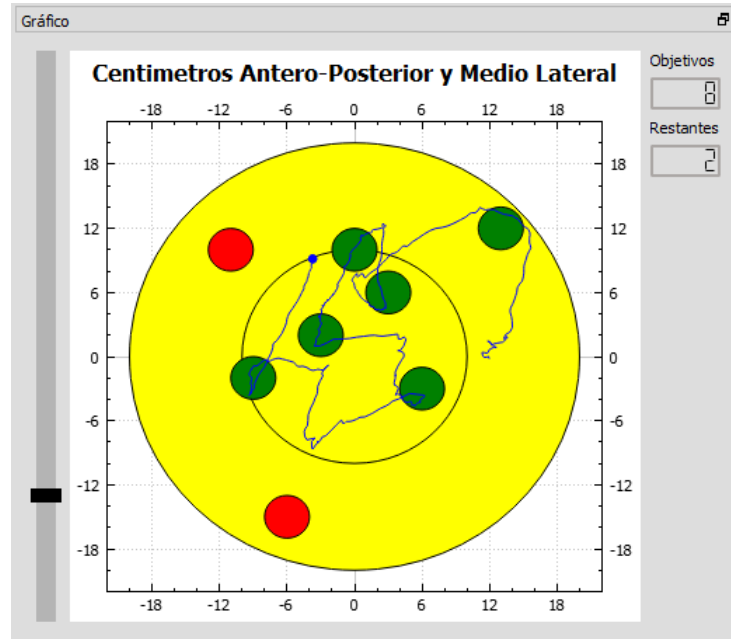


Figura 10: Gráfico de Prueba

#### 4.7. Cálculo ángulo

Para efectos de análisis o representación final, a partir de las variaciones de aceleración y velocidad angular detectadas por los sensores existe la opción de calcular el ángulo, que es un indicador equivalente a la inclinación que ha realizado el sujeto.

Para ello existen distintos métodos usando la información obtenida, tales como:

##### 4.7.1. Cálculo ángulo con acelerómetro

Si es utilizado acelerómetro este nos entrega el ángulo exacto pero al ser producida cualquier variación en la aceleración en cualquiera de sus 3 ejes de medición que no sea producto tan solo de la fuerza gravitacional será interpretada como un cambio de rotación, por lo cual se generan medida erráticas debido a los cambios de aceleración constantes que son generados al girar la IMU como se muestra en el ejemplo para el cálculo de los ángulos X ver ecuación (2) e Y ver ecuación (3) usando el sensor en posición vertical.

$$\text{anguloX} = \tan \left( \frac{\text{aceleracionX}}{\sqrt{\text{aceleracionZ}^2 + \text{aceleracionY}^2}} \right) \quad (2)$$

$$\text{anguloY} = \tan \left( \frac{\text{aceleracionZ}}{\sqrt{\text{aceleracionX}^2 + \text{aceleracionY}^2}} \right) \quad (3)$$

#### 4.7.2. Cálculo ángulo con giroscopio

En cambio, si utilizamos sólo el giroscopio a diferencia del acelerómetro, podemos realizar los cálculos del ángulo con mayor precisión pero al no tener una referencia clara del origen de las mediciones, es difícil de contrastar la información con el movimiento de un sujeto y dificulta enormemente la interpretación de esta información, además con el tiempo los datos capturados comienzan a acumular un pequeño error, a estos errores acumulados se les conoce como drift o deriva.

### 4.8. Fusión de Datos Sensores

La Fusión de Datos es un método para integrar señales provenientes de múltiples fuentes. Permite extraer información de varias fuentes diferentes para integrarlas en una sola señal o información.

Para el análisis del desplazamiento del centro de masa se requiere el cálculo del ángulo y para ello se usará la información proveniente de ambos sensores con el fin de obtener el ángulo lo más refinado posible, con este fin se estudiaron e implementaron 2 filtros que permiten una correcta integración de los datos de aceleración con velocidad angular.

#### 4.8.1. El Filtro Complementario

En síntesis tenemos un acelerómetro que entrega ángulos con referencia en la posición del sensor, pero muy susceptible a la variación de cambios de aceleración del IMU y en cambio el giroscopio nos permite calcular el ángulo de manera exacta, pero si tener una clara referencia de la posición donde está el sensor.

Para unificar la información proveniente de ambos sensores existe un filtro, que nació principalmente de la práctica, conocido como el Filtro Complementario [CAV10, Capítulo 3, p. 44]. El filtro requiere un ángulo de punto de partida y para esto se utiliza el acelerómetro para recoger el ángulo inicial, ya que, al poseer la referencia de la gravedad, se puede tener una mejor estimación para comenzar a realizar las iteraciones del filtro.

El Filtro complementario es ampliamente usado en unidades inerciales y sistemas de visión. Este filtro es en sí un filtro de Kalman de estado estacionario para una cierta clase de problemas de filtrado. Está compuesto por la unión de dos filtros diferentes:

- **Filtro Pasa-Bajo:** Se usa para eliminar las frecuencias altas del acelerómetro, por lo cual dejamos pasar bajo ciertas frecuencias, para eliminar el ruido detectado por el cambio de aceleración (vibración principalmente) en los ejes del acelerómetro.

- **Filtro Pasa-Alto:** Se aplica en las mediciones obtenidas por el giroscopio, para eliminar el drift acumulado en el tiempo, dejando pasar solo las de una frecuencia más alta.

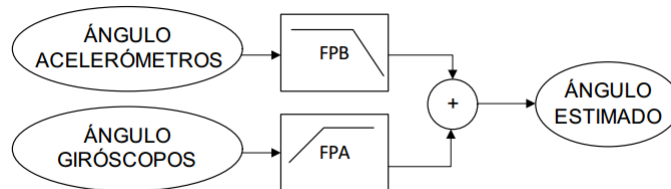


Figura 11: Diagrama Filtro Complementario

Todo lo anterior queda descrito usando la siguiente Ecuación (4):

$$angle = (0,98) * (angle + gyro * dt) + (0,02) * angle\_acc \quad (4)$$

En donde cada uno de las variables representa lo siguiente:

- **gyro:** Es el ángulo del Giroscopio que hemos calculado previamente.
- **angle\_acc:** El ángulo calculado por los datos del Acelerómetro.
- **dt:** Corresponde a la variación en segundos desde la ultima vez que se aplico el filtro para calcular el angulo.
- **Coeficientes:** 0.98 y 0.02 pueden ser personalizados para aumentar o disminuir la contribución de cada sensor, pero se debe tener en consideración que la suma de ambos debe ser 1.

#### 4.8.2. Filtro de Kalman discreto

El filtro de Kalman [ASD05] es un estimador lineal, insesgado y óptimo del estado de un proceso. En él se ha impuesto la condición de que el proceso a ser estimado es regido por una dinámica lineal y que el ruido que lo perturba es blanco y gaussiano. Aún cuando la condición del comportamiento probabilístico gaussiano del ruido se omite, el filtro de Kalman sigue siendo el mejor filtro recursivo lineal (error de menor varianza ) e insesgado.

Su propósito es emplear las mediciones obtenidas en un período de tiempo que a su vez son afectadas por variaciones aleatorias (ruido) junto con el conocimiento del comportamiento del sistema, para producir estimaciones que tiendan a estar más cerca del valor real



del proceso en cuestión. Todas las mediciones y cálculos basados en modelos son aproximaciones en cierto grado, datos ruidosos provenientes de los sensores [CAV10, Capítulo 3, p. 45].

EL filtro de Kalman posee flexibilidad para ser utilizado tanto en tiempo continuo o en tiempos discretos. El modelo del filtro permite la flexibilidad para ser utilizado en éstos casos.

La principal diferencia es la notación que utilizada en cada uno de los casos. Debido a que en la actualidad la mayoría de las aplicaciones de filtros es realizada en plataformas de tipo digital, y considerando el uso del filtro para la fusión de los sensores en este proyecto.

El Filtro consiste en 2 estados Predicción y Actualización.

aSDASD ASDASDSA LOREM

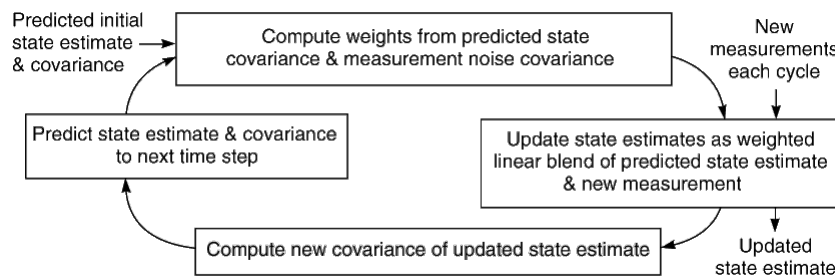


Figura 12: Diagrama estados Filtro Kalman

#### 4.8.3. Comparativa de Filtros

Luego de la integración de los filtros en el Software es posible utilizar el deseado para la prueba, pero se deben tener en consideración las ventajas de usar o no el filtro razón por la cual se realizó una breve comparativa para ver el comportamiento entre el cálculo del ángulo sin utilizar filtros, usando filtro complementario y Kalman para la obtención durante la misma prueba. Los resultados demuestran la gran corrección que realizan los filtros frente a el cálculo sin estos ver Figura (13).

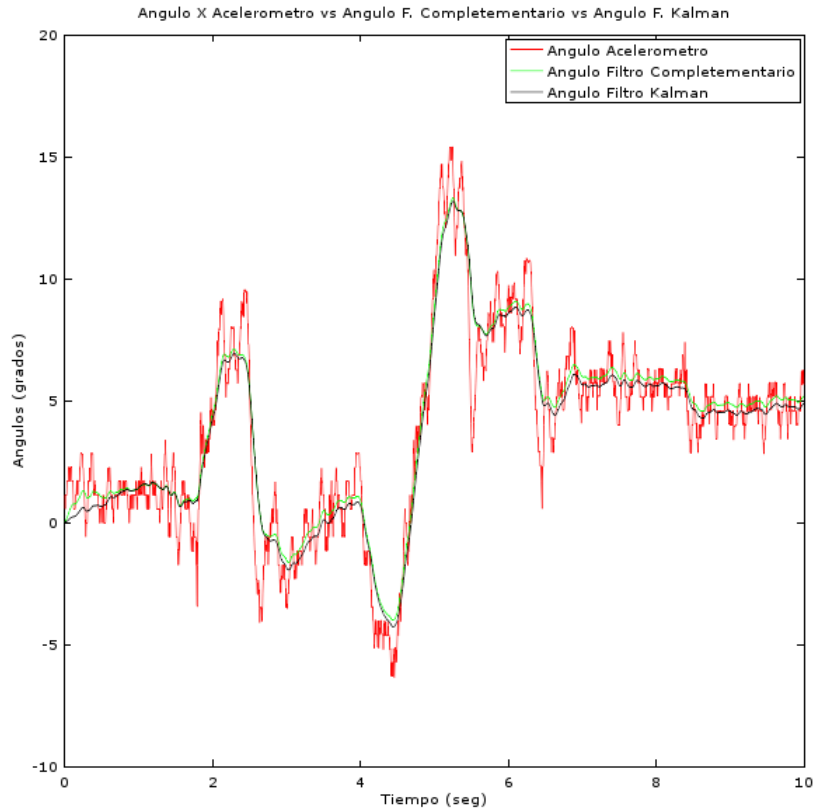


Figura 13: Comparativa Angulo Acelerómetro vs Filtro Kalman vs Filtro Complementario

#### 4.9. Cálculo proyección del Ángulo

Basándonos en que la posición bípeda-quieta puede ser representada como un péndulo invertido, por lo tanto considerando la variación angular del movimiento pendular de un sujeto en las coordenadas antero-posterior y medio-lateral es aplicable la trigonometría para la obtención del desplazamiento, ésta variación angular es directamente proporcional a el desplazamiento en el eje horizontal, tomando la altura  $h$  como la distancia desde el suelo hasta donde fue puesto el sensor como el equivalente a la hipotenusa de un triángulo rectángulo, como se muestra en la Figura (14), al aplicar la función sin del ángulo en grados obtenido mediante el filtro aplicado, junto con ponderar el valor obtenido por la altura  $h$  (hipotenusa del triángulo), obtendríamos la distancia  $d$  correspondiente al desplazamiento del centro de masa estimado, en la cual su unidad de medida es la misma usada para describir la altura  $h$  en general centímetros.

$$d = \sin(\alpha) \quad (5)$$

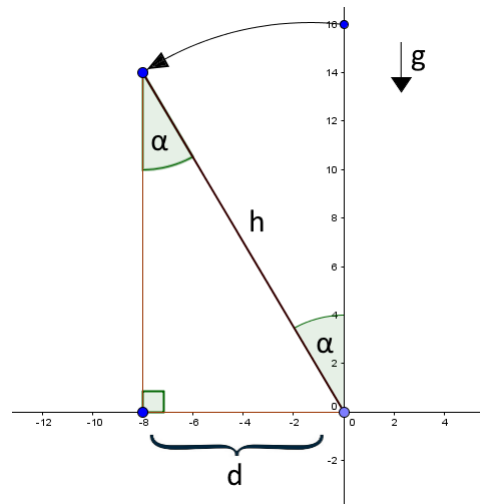


Figura 14: Ejemplo calculo proyección Angulo

#### 4.10. Cálculo del Recorrido Curvo

#### 4.11. Principales Dificultades

##### 4.11.1. Representación Tiempo Real

La interfaz gráfica es una parte integral de el sistema, pasando a ser la principal fuente de información para la retroalimentación, junto con el lugar donde mayormente interviene el usuario de este tipo de sistemas, por lo tanto el no poder desplegar la información a tiempo, o que esta sea poco fiable es un gran problema. Se planteo inicialmente en graficar cada una de las salidas tanto del acelerómetro como del giroscopio en donde se representarían 6 variables a una velocidad igual a la velocidad de muestreo, lo que en la práctica no resulta demasiado bien. Al representar en gráficos a frecuencias muy altas es decir 100hz o superior, multiplicado por la cantidad de datos que se estén representando, junto con el procesamiento interno para mostrar esos datos, se genera una sobrecarga en la parte gráfica, lo que conlleva a ralentizar el proceso, generando perdidas de información y fidelidad al desplegar la información en tiempo real.

Para dar solución al problema, se implemento la opción de graficar un porcentaje de la información recibida en base a la frecuencia de muestreo seleccionada para la captura de información, con esto se asegura que la representación gráfica en tiempo real no sea un problema.

#### 4.11.2. Orientación Sensores

Al momento de realizar la captura de información sensorial, junto con la obtención del ángulo de inclinación deben ser consideradas referenciado tales como:

- **Referencia ejes del Sensores:** La obtención de información de aceleración y velocidad angular debe estar correctamente alineada con la referencia presente en los ejes del sensor, junto con la posición respecto al eje de gravedad de la tierra, para la correcta obtención de la información en base al movimiento ejercido por los sensores, y que estos sean descritos por los ejes que correspondan, para su posterior análisis o uso.
- **Posición en el Sujeto:** Junto con la referencia del sensor, se debe tomar en consideración en la posición que el prototipo o sistema final será puesta en el sujeto de estudio, ya que al cambiar la posición, las referencias internas del sensor dependen directamente de la posición, sino todo lo resultante va a cambiar, y con ello la información entregada comenzaría a describir el movimiento desde una perspectiva distinta a la planteada para los cálculos y estudio posterior.

Para dar solución a este inconveniente se incorporó en el software la opción de describir mediante la interfaz como será puesto el prototipo al momento de realizar las mediciones, usando un indicador de referencia clara, con el fin de no generar problemas para conocer la posición del sensor y facilitar al usuario final la tarea. Según la orientación del sensor se generan distintas conjugaciones de los ejes a utilizar tanto para el cálculo del ángulo, como en el uso de la velocidad angular para la aplicación del filtro (en caso de ser utilizado), éstos se configuraron estudiando los ejes que debían ser usados según la orientación junto con el sentido en que varía la información de Acelerómetro y giroscopio, para así realizar el cálculo final.

#### 4.11.3. Data Terminal Ready

Según el tipo de semiconductor del puerto USB de Arduino puede venir por defecto el Pin DTR(Data Terminal Ready) la función Reset de Arduino puede venir activada o desactivada. La función Reset de Arduino permite al microcontrolador reiniciar las variables internas según las configuraciones encontradas en el Setup del código de programación, para así eliminar cualquier cambio realizado en la ejecución del Loop, lo que es complicación al usar modelos de Arduino que tengan o no esta característica.

Para solucionar este problema se realiza una estandarización activando por Software el Pin DTR para que sin importar el modelo de semiconductor operen todos de forma igual en

el software, para así no preocuparse de configuraciones o errores producidos debido a el PIN DTR.

#### **4.11.4. Latencia en Controlador FTDI**

Por defecto el controlador del puerto USB FTDI para el Sistema Operativo Windows, en su configuración contenía una Latencia de 10 ms pre-configurada, lo que afecta considerablemente la captura de información sobre 100hz. El principal inconveniente que genero este error es la detección propia de éste, ya que al ser un error completamente aislado de los elementos que comúnmente son manipulados Arduino y Software, y a su vez es relativo a un modelo concreto de controlador de sistema y a su vez sistema operativo lo cual era de partida complejo de percibir y/o predecir.

La solución fue a partir de la información de configuración FTDI[INT10] para configurar directamente en el registro de Windows (regedit) en la dirección asociada al controlador el parámetro de latencia deseado, para que éste sea lo más bajo posible.

## **5. Conclusiones y Trabajos Futuros**

### **5.1. Conclusiones**

### **5.2. Trabajos Futuros**

## 6. Bibliografía

- [ARD13] Arduino. *Arduino Home*. 2013. URL: <https://www.arduino.cc/> (visitado 10-05-2015).
- [ASD05] asda. *asd*. 2005. URL: <http://see-out.com/sandramau/doc/Mau05MathKalman.pdf> (visitado 05-04-2016).
- [CAV10] Damian Pierre Vigouroux Cavolina. «Implementación de unidad de mediciones inerciales (IMU) para robótica utilizando Filtro de Kalman». Tesis para optar al título Ingeniero Electrónico. Universidad Simón Bolívar, 2010.
- [INT10] Future Technology Devices International. *FTDI Knowledgebase*. 2010. URL: <http://www.ftdichip.com/Support/Knowledgebase/index.html?settingacustomdefaultlaten.htm> (visitado 10-08-2015).
- [INV13a] InvenSense®. *Datasheet MPU6050*. 2013. URL: <http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050> (visitado 10-05-2015).
- [INV13b] InvenSense. *Map Register and Descriptions*. 2013. URL: <http://43zrtwysvxb2gf29rwpengine.netdna-cdn.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf> (visitado 10-05-2015).
- [KIS16] KISTLER. *Kistler 9286B DataSheet*. 2016. URL: <https://www.ntnu.edu/documents/221360533/221362168/KISTLER.pdf/6e0bf322-368e-46b2-b125-c7251046a81c> (visitado 05-04-2016).
- [MAN10] Mancini. «The relevance of clinical balance assessment tools to differentiate balance deficits». En: *European journal of physical and rehabilitation medicine* 46.2 (2010), pág. 239. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3033730/> (visitado 28-06-2016).
- [QT16] QT. *QT Home*. 2016. URL: <http://www.qt.io/> (visitado 10-05-2015).
- [SEM14] NXP Semiconductors. *I2C bus Specification and User Manual*. 2014. URL: <http://www.i2c-bus.org/specification> (visitado 13-12-2015).