

UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA

CARRERA DE INGENIERÍA ELECTRÓNICA

Tesis previa a la Obtención del
Título de Ingeniero Electrónico

TÍTULO:

**“DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN SISTEMA
EMBEBIDO DE ADQUISICIÓN DE PARÁMETROS CINEMÁTICOS
DE LA MARCHA HUMANA EN TOBILLO, RODILLA Y CADERA”**

AUTORES:

**BYRON GERMAN CONTRERAS BERMEO
VÍCTOR HUGO FLORES MORALES**

DIRECTOR:

ING. FREDDY LEONARDO BUENO PALOMEQUE

CUENCA, FEBRERO 2015

Los conceptos, análisis y conclusiones desarrollados en el presente trabajo, son de completa responsabilidad de los autores.



Byron Contreras Bermeo



Victor Flores Morales

DECLARACIÓN

Nosotros, Byron German Contreras Bermeo y Víctor Hugo Flores Morales, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Byron Contreras Bermeo



Víctor Flores Morales

CERTIFICACIÓN

Certifico que el trabajo “DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN SISTEMA EMBEBIDO DE ADQUISICIÓN DE PARÁMETROS CINEMÁTICOS DE LA MARCHA HUMANA EN TOBILLO, RODILLA Y CADERA” realizado por Byron German Contreras Bermeo y Víctor Hugo Flores Morales, fue desarrollado bajo mi supervisión.

Cuenca, febrero de 2015



Ing. Freddy Bueno Palomeque

DEDICATORIA

Dedico esta tesis a mis padres Lauro Contreras y Olga Bermeo quienes con su amor y ejemplo me enseñaron a ser una persona de bien y me brindaron todo el apoyo a pesar de las dificultades para que pueda culminar esta etapa de mi vida.

A mis hermanas Mónica y Verónica por sus consejos y apoyo en todo momento.

AGRADECIMIENTO

Agradezco primeramente a Dios por guiar mi vida, a toda mi familia y compañeros por el apoyo brindado, finalmente a las personas quienes de una u otra forma nos colaboraron de forma desinteresada en la elaboración de esta tesis.

Byron

DEDICATORIA

Dedica a todos quienes hicieron posible este sueño en especial a mis padres Víctor Alfonso Flores y María Carmita Morales con su apoyo incondicional durante todas las etapas de mi vida.

Mis hermanos Xavier y Karina quienes me alentaron para continuar.

Y a mis tías y tíos quienes fueron como unos segundos padres brindándome su apoyo y consejo durante toda mi vida.

AGRADECIMIENTO

Mis más sinceros agradecimientos a todo mi familia, compañeros, amigos y la persona amada Ximena Llanés que me brindaron su apoyo desinteresado durante este largo camino universitario.

Desde lo más profundo de mi ser les agradezco infinitamente.

Hugo

CONTENIDO

GLOSARIO DE TERMINOS.....	xii
PRÓLOGO.....	1
INTRODUCCIÓN.....	2
CAPÍTULO 1.....	3
1. BIOMECÁNICA DE LA MARCHA HUMANA.....	3
1.1 ANÁLISIS CINEMÁTICO DE LA MARCHA HUMANA.....	3
1.1.1 COMPONENTES ÓSEOS DEL MIEMBRO INFERIOR.....	4
1.1.2 PLANOS DE REFERENCIA.....	4
1.1.3 BIOMECÁNICA DE LA CADERA.....	7
1.1.4 BIOMECÁNICA DE LA RODILLA.....	11
1.1.5 BIOMECÁNICA DEL TOBILLO.....	12
1.1.6 CICLO DE LA MARCHA HUMANA.....	13
1.1.7 ÁNGULOS DEL MIEMBRO INFERIOR DURANTE UN CICLO DE MARCHA.....	15
1.2 SISTEMAS EMBEBIDOS PARA EL ANÁLISIS BIOMECÁNICO.....	16
1.2.1 SENSORES PORTÁTILES.....	17
1.3 DIPOSITIVOS INALÁMBRICOS PARA EL ANÁLISIS BIOMECÁNICO.....	17
CAPÍTULO 2.....	23
2. SISTEMA EMBEBIDO.....	23
2.1 DISEÑO DE HARDWARE.....	25
2.1.1 COMUNICACIÓN I2C ENTRE MPU 6050 – CONTROLADORES.....	26
2.1.2 COMUNICACIÓN SERIAL ENTRE INTEGRADOS ATMEGA 328P-PU.....	27
2.1.3 COMUNICACIÓN SPI ENTRE ATMEGA 328P-PU Y TARJETA MICRO SD.....	28
2.1.4 DISEÑO Y CONTRUCCIÓN DE LAS TARJETAS ELECTRÓNICAS.....	30
2.2 DISEÑO DEL SOFTWARE.....	32
2.2.1 CÁLCULO DE LOS ÁNGULOS DE NAVEGACIÓN.....	33
2.2.2 DISEÑO DE SOFTWARE DEL MICROCONTROLADOR ATMEGA 328P-PU.....	37
2.2.3 CÓDIGO DE PROGRAMA DEL ESCLAVO.....	38
2.2.4 CÓDIGO DE PROGRAMA DEL MAESTRO.....	54
2.3 CREACIÓN DEL ARCHIVO .MOT.....	60

2.3.1 ARCHIVO.MOT.....	60
2.4 RECREACIÓN DEL MOVIMIENTO.....	68
2.4.1 CARGAR LA TRAYECTORIA.....	69
CAPÍTULO 3.....	71
3. PRUEBAS Y RESULTADOS.....	70
3.1 PRUEBAS DE FUNCIONAMIENTO.....	71
3.1.1 MUESTRAS DE ESTABILIZACIÓN INICIAL.....	71
3.1.2 PRUEBA DE ESTABILIDAD INERCIAL.....	72
3.1.3 PRUEBA DE FILTRADO.....	73
3.1.4 PRUEBA PARA EL CÁLCULO DEL ERROR.....	74
3.1.5 PRUEBAS DE FUNCIONAMIENTO.....	82
3.2 FALLAS DEL SISTEMA.....	88
3.3 DISCUSIÓN.....	89
CAPÍTULO 4.....	92
4. COSTOS DEL PROYECTO.....	91
ANEXOS.....	95
ANEXO 1: TABLA DE ERROR EN ESTADO INERCIAL.....	96
ANEXO 2: TABLA DE ERROR EN YAW.....	100
ANEXO 3: TABLA DE ERROR EN PITCH.....	107
ANEXO 4: TABLA DE ERROR DE ROLL.....	114
ANEXO 5: TABLA PARA EL CÁLCULO DE PROBABILIDAD MEDIANTE LA DISTRIBUCIÓN NORMAL.....	120
ANEXO 6: MANUAL DE FUNCIONAMIENTO.....	121

Índice de Figuras

Figura 1. 1: Huesos y articulaciones de los miembros inferiores	4
Figura 1. 2: Plano sagital, plano frontal, plano transversal.....	5
Figura 1. 3: a) Flexión de la cadera, b)Extensión de la cadera, c) Flexión de la rodilla, d) Extensión de la rodilla, e) Flexión dorsal, f) Flexión plantar	6
Figura 1. 4: a) Abducción, b) Aducción.....	7
Figura 1. 5: Rotación.....	7
Figura 1. 6: Vista frontal de las dos articulaciones de la cadera	8
Figura 1. 7: a) Flexión de la cadera con extensión de la rodilla, b) Flexión de la cadera con flexión de la rodilla	8
Figura 1. 8: Extensión de la cadera con extensión de la rodilla b) Extensión de la cadera con flexión de la rodilla	9
Figura 1. 9: a) Abducción activa	10
Figura 1. 10: Abducción pasiva	10
Figura 1. 11: Rodilla	11
Figura 1. 12: a) Flexión máxima de la rodilla, b) Extensión máxima de la rodilla....	11
Figura 1. 13: Movimiento giratorio de la rodilla.....	12
Figura 1. 14: Tobillo	12
Figura 1. 15: a) Posición neutra del tobillo, b) Ángulo de flexión dorsal, c) Ángulo de flexión plantar	13
Figura 1. 16: Ciclo de la marcha	14
Figura 1. 17: Apoyo simple y doble apoyo.....	14
Figura 1. 18: Ángulos del ciclo de marcha de las tres articulaciones del miembro inferior correspondientes al plano sagital	16
Figura 1. 19: Diagrama de bloques de los dispositivo inalámbricos.....	17
Figura 1. 20: Ubicación de los sensores de acuerdo al trabajo de Qi An et al.....	18
Figura 1. 21: Orientación del antebrazo mediante un sensor inercial	20
Figura 1. 22: Sistema airbag humano.....	20
Figura 1. 23: Sistema MVN Biomech Awinda	21
Figura 1. 24: Sistema MVN y graficas de los ángulos de la cadera.....	22
Figura 2. 1: Traje ergonómico.....	24
Figura 2. 2: Colocación de los sensores inerciales.....	25
Figura 2. 3: Sensor MPU-6050	26
Figura 2. 4:Diagrama de la comunicación I2C.	26
Figura 2. 5: Diagrama de conexión de los diferentes integrados y sensores.....	27
Figura 2. 6: Diagrama de conexión entre el integrado ATMEGA 328P-PU y el lector micro SD	29
Figura 2. 7: Diagrama de conexión del circuito de procesamiento.....	29
Figura 2. 8: Diagrama de conexión del circuito de mando	29
Figura 2. 9: PCB tarjeta de procesamiento.....	30
Figura 2. 10: PCB tarjeta de mando	31
Figura 2. 11: Tarjeta de procesamiento.....	31

Figura 2. 12: Tarjeta de mando	32
Figura 2. 13: Ángulos de navegación.....	33
Figura 2. 14: Denotación de los ángulos de Euler	34
Figura 2. 15: Manejo de cuaterniones	36
Figura 2. 16: Cuaternion de gravedad.....	37
Figura 2. 17: Cálculo del ángulo de navegación	37
Figura 2. 18: Diagrama de flujo del esclavo ATMEGA 328P-PU	39
Figura 2. 19: Diagrama de flujo del cálculo del ángulo del giroscopio 1	40
Figura 2. 20: Diagrama de flujo del cálculo del ángulo del giroscopio 2	41
Figura 2. 21: Librerías utilizadas para el sketch.....	42
Figura 2. 22: #define y direcciones del MPU	42
Figura 2. 23: Variables del sketch 1	43
Figura 2. 24: Variables del sketch 2	44
Figura 2. 25: Código de la variable de clasificación.....	45
Figura 2. 26: Código de las subrutinas activadas por las interrupciones	45
Figura 2. 27: Código de subrutina de envío de datos del esclavo 2	45
Figura 2. 28: Código de subrutina de envío de datos del esclavo 3	46
Figura 2. 29: Código de subrutina de envío de datos del esclavo 4	46
Figura 2. 30: Código de la subrutina de seteo	47
Figura 2. 31: Inicialización de los puertos I2C y serial en los esclavos.....	47
Figura 2. 32: Inicialización de los giroscopios.....	48
Figura 2. 33: Código de programa inicio del loop	49
Figura 2. 34: Código para la extracción y verificación de datos del MPU 6050	50
Figura 2. 35: Código para la extracción de los ángulos de navegación	50
Figura 2. 36: Código de programa para el set y preparación de la variable para el envío.....	51
Figura 2. 37: Cambio de testigo y reseteo de la memoria FIFO	51
Figura 2. 38: Código del proceso de verificación Nan.....	52
Figura 2. 39: Código del proceso de verificación del funcionamiento	52
Figura 2. 40: Lectura del puerto serial	53
Figura 2. 41: Diagrama de Flujo del maestro.....	53
Figura 2. 42: Librerías del sketch del maestro	54
Figura 2. 43: Variables del sketch del maestro	54
Figura 2. 44: Subrutina para el esclavo 2	55
Figura 2. 45: Subrutina para el esclavo 3	55
Figura 2. 46: Subrutina para el esclavo 4.....	56
Figura 2. 47: Proceso de inicialización de librerías y declaración de pines de salida y entrada	56
Figura 2. 48: Proceso de inicialización del puerto serial y verificación de conexión de la tarjeta SD.....	57
Figura 2. 49: Verificación de archivos y eliminación de archivos en la tarjeta SD... ..	57
Figura 2. 50: Proceso de verificación.....	58
Figura 2. 51: Proceso de envío de seteo	58
Figura 2. 52: Petición y recepción de ángulo	59

Figura 2. 53: Archivo .mot.....	59
Figura 2. 54: Diagrama de flujo para la creación del archivo .mot.....	60
Figura 2. 55: Interfaz visual para la creación del archivo .mot.....	61
Figura 2. 56: Exportación de los datos desde el archivo de texto a Matlab.....	62
Figura 2. 57: Longitud de cada vector y longitud mínima.....	63
Figura 2. 58: Vectores de igual longitud.....	64
Figura 2. 59: Código para el encabezado.....	65
Figura 2. 60: Vector de ceros.....	65
Figura 2. 61: Orientación de Ejes.....	65
Figura 2. 62: Creación del Filtro IIR Butterworth pasabajo mediante fdatool.....	66
Figura 2. 63: Respuesta en frecuencia del filtro Butterworth diseñado.....	66
Figura 2. 64: Datos válidos para la recreación del movimiento.....	67
Figura 2. 65: Modelo biomecánico y coordenadas de OpenSim.....	68
Figura 2. 66: Grafica Obtenida en OpenSim.....	69
Figura 3. 1: Prueba de estabilización.....	70
Figura 3. 2: Prueba de estabilidad inercial.....	71
Figura 3. 3: Prueba de filtrado con un solo giroscopio.....	72
Figura 3. 4: Prueba de filtrado con dos giroscopios.....	73
Figura 3. 5: Instrumento patrón.....	74
Figura 3. 6: Captura de datos mediante el instrumento patrón.....	74
Figura 3. 7: Distribución normal del ángulo yaw.....	76
Figura 3. 8: Distribución normal acumulada del ángulo yaw.....	76
Figura 3. 9: Frecuencia de datos del ángulo yaw.....	77
Figura 3. 10: Ángulos reales y medidos del ángulo yaw.....	77
Figura 3. 11: Distribución normal del ángulo pitch.....	78
Figura 3. 12: Distribución normal acumulada del ángulo pitch.....	78
Figura 3. 13: Frecuencia de datos del ángulo pitch.....	79
Figura 3. 14: Ángulos reales y medidos del ángulo pitch.....	79
Figura 3. 15: Distribución normal del ángulo roll.....	80
Figura 3. 16: Distribución acumulada del ángulo roll.....	80
Figura 3. 17: Frecuencia de datos del ángulo roll.....	81
Figura 3. 18: Ángulos reales y medidos del ángulo roll.....	81
Figura 3. 19: a) Sujeto de prueba, b) Avatar de OpenSim, c) Curva que describe el movimiento de flexión en la cadera.....	82
Figura 3. 20: a) Sujeto de prueba, b) Avatar de OpenSim, c) Curva que describe el movimiento de abducción y aducción.....	83
Figura 3. 21: a) Sujeto de prueba, b) Avatar de OpenSim, c) Curva que describe el movimiento rotación en la cadera.....	84
Figura 3. 22: a) Sujeto de prueba b) Avatar de OpenSim c) Curva que describe el movimiento flexión de la rodilla.....	85
Figura 3. 23: a) Sujeto de prueba, b) Avatar de OpenSim, c) Curva que describe deflexión-extensión del tobillo.....	86

Figura 3. 24: Ciclo de marcha.....	87
------------------------------------	----

Índice de Tablas

Tabla 3. 1: Muestras de estabilización	71
Tabla 4. 1: Costos del prototipo	91
Tabla 4.2: Costos de producción del sistema	92

GLOSARIO DE TÉRMINOS

ADC	Analog-to-Digital Converter
CS	Chip Selector
DMP	Digital Motion Processor
I2C	Inter Integrated Circuits
IDE	Integrated development environment
MPU	Multiple Process Unit
PCB	Printed Circuit Board
RAM	Random Access Memory
SCL	System Clock
SD	Secure Digital
SDA	System Data
SPI	Serial Peripheral Interface

PRÓLOGO

En esta tesis se presenta una herramienta para la medición de las variaciones angulares en las articulaciones de las caderas, rodillas y tobillos, mediante la utilización de sensores inerciales. Los datos adquiridos son almacenados en una memoria SD para posteriormente crear un archivo compatible con el software *OpenSim*, herramienta ampliamente utilizada para modelamiento biomecánico. Nuestro proyecto busca brindar a los especialistas en el tema la posibilidad de detectar diferencias en el patrón de marcha de personas y de analizar más objetivamente la ejecución de una determinada técnica en deportistas.

A lo largo de los capítulos se muestra aspectos importantes de la biomecánica y sistemas embebidos actuales utilizados para adquisición de parámetros biomecánicos, el diseño y construcción tanto del software como del hardware y finalmente las pruebas, análisis y resultados de todo el sistema embebido creado.

INTRODUCCIÓN

El mejoramiento de una técnica deportiva está estrictamente ligado al descubrimiento de nuevas soluciones motrices a problemas determinados dentro de un marco reglamentario. Esto se traduce en el desarrollo de nuevas formas de ejecutar una acción determinada o bien aumentar la eficacia de los movimientos, es aquí donde los ángulos de inclinación, rotación y tiempo en el cual se produce el movimiento de cada extremidad, son los factores primordiales para lograr alcanzar velocidades o fuerzas que le permiten sobresalir al atleta en su disciplina deportiva.

El movimiento puede ser recreado mediante equipos recolectores de datos basados en la combinación de hardware y software con algoritmos óptimos que efectúen dichas tareas.

Existen centros especializados para el análisis de la marcha, compuestos por un espacio de alta tecnología que permite realizar un diagnóstico preciso acerca del estado del movimiento de los deportistas, los cuales se han visto limitados al espacio de estudio, el peso del equipo, limitaciones de movimientos en el individuo y sobre todo al elevado costo de sus equipos.

Ante esta situación, nuestro trabajo presenta un sistema de captura de movimiento con el propósito de describir y registrar la técnica del deportista al momento de ejecutar el gesto mediante el análisis de las curvas y los rangos de movimiento presentando un método de cuantificación que permita al entrenador diseñar y aplicar las correcciones necesarias de la técnica, una vez que se obtengan la información cinemática del atleta.

CAPÍTULO 1.

1. BIOMECÁNICA DE LA MARCHA HUMANA

La biomecánica de la marcha humana es la encargada del estudio de los movimientos del cuerpo humano producido por fuerzas tanto externas como internas que actúan sobre él. Existen diferentes causas que afectan la capacidad de caminar en las personas como por ejemplo factores hereditarios, congénitos, traumas, afecciones neurológicas y músculo-esqueléticas como: dismetría, inestabilidad y reducción en los ángulos de rotación articular, limitación de la amplitud y pie equino [1]. La biomecánica de la marcha humana es utilizada para diagnosticar determinadas patologías en las personas y de esta manera planificar el tratamiento necesario [2].

El estudio de la biomecánica se basa en la mecánica básica y la física; en el caso de la estática mediante ecuaciones de equilibrio se puede determinar las fuerzas que se producen en cierto músculo o articulación para mantener una determinada posición, por otro lado la dinámica es utilizada para la evaluación de los movimientos, análisis de la marcha, y el análisis del movimiento segmentario [3]. Sus aplicaciones en el ámbito deportivo es de gran utilidad debido a que permite corregir errores técnicos con la finalidad de ejecutar movimientos precisos que permitan un mayor rendimiento en el deportista, así como el de evitar lesiones [4] [5].

1.1 ANÁLISIS CINEMÁTICO DE LA MARCHA HUMANA

El término de cinemática dentro de la marcha humana hace referencia a la posición, velocidad y aceleración de los diferentes segmentos del cuerpo humano. Este trabajo está centrado en el análisis cinemático de una persona para conocer los rangos de variación de las distintas articulaciones durante la ejecución de cierto movimiento [6].

1.1.1 COMPONENTES ÓSEOS DEL MIEMBRO INFERIOR

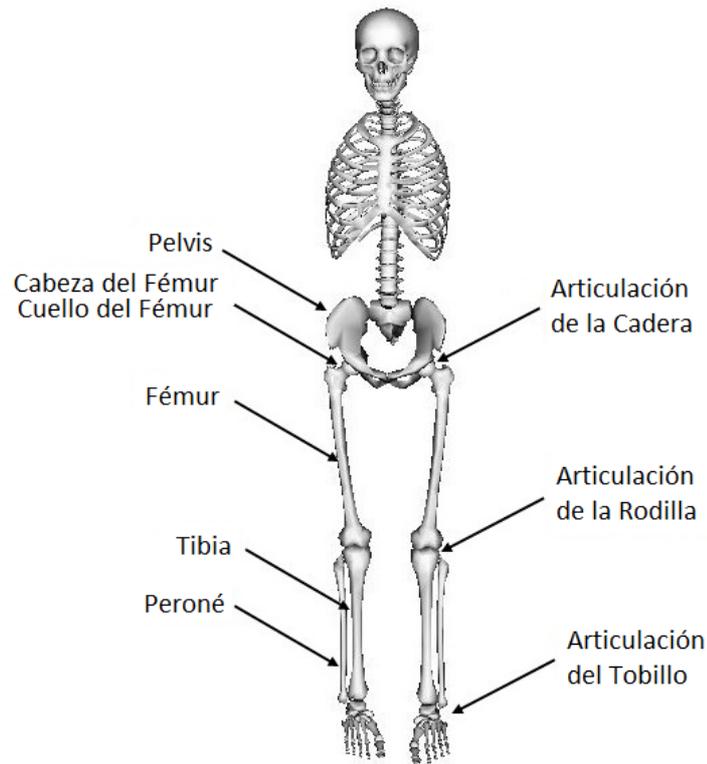


Figura 1.1: Huesos y articulaciones de los miembros inferiores

Para un análisis práctico de la marcha humana es necesario considerar los huesos de la pelvis y de las piernas como se puede ver en la figura 1.1 dichos huesos se encuentran sometidos a fuertes cargas durante actividades fisiológicas diarias como caminar o trotar [2] [3].

1.1.2 PLANOS DE REFERENCIA

El movimiento de las articulaciones se describe utilizando tres planos de referencia que se intersectan entre sí: sagital, frontal y vertical; dicha intersección se define como el centro de masa del cuerpo humano y el eje de rotación de cada articulación es perpendicular al plano de referencia [7].

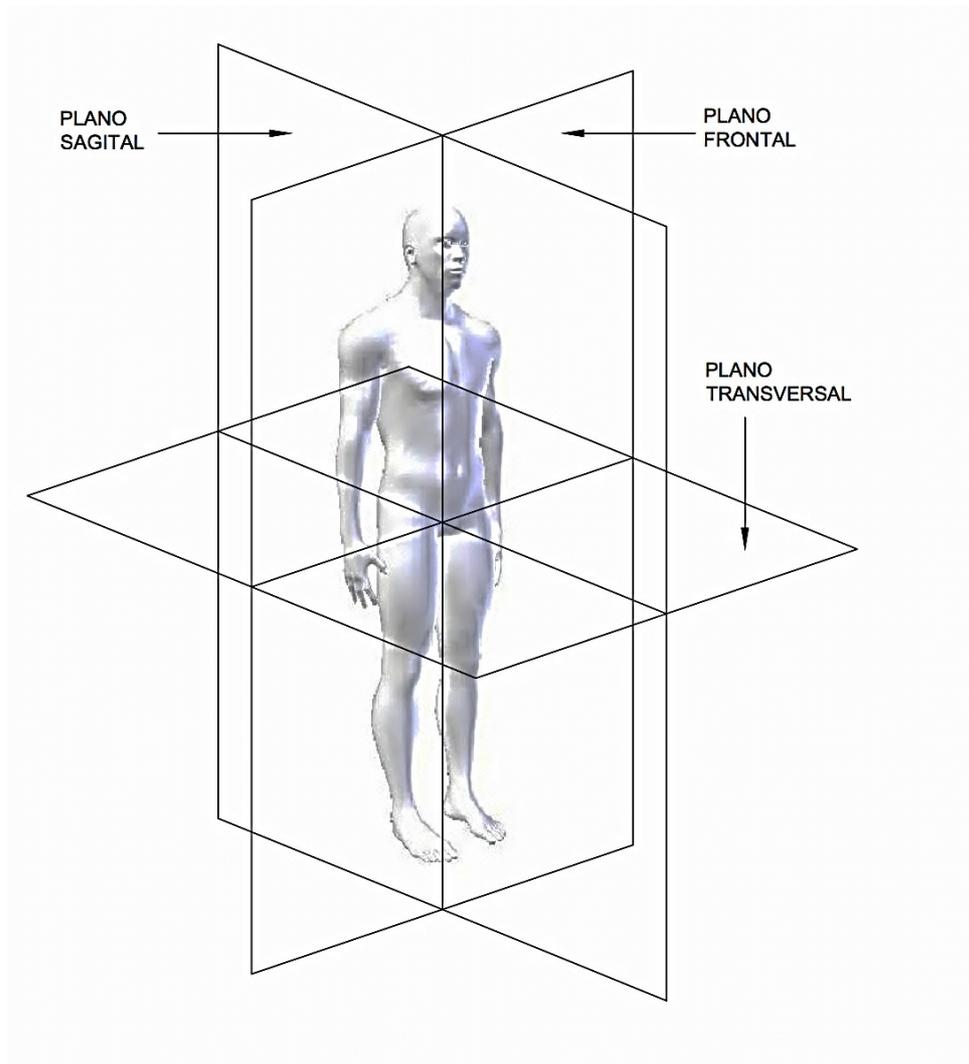


Figura 1.2: Plano sagital, plano frontal, plano trasversal

Plano sagital: Es el que pasa desde la parte posterior a la parte anterior o frontal y divide al cuerpo en mitades derecha e izquierda [2].

Plano frontal: Divide el cuerpo en dos partes delantera y trasera [2].

Plano transversal: Divide al cuerpo en partes inferior y superior [2].

Los movimientos producidos por las diferentes articulaciones del miembro inferior: cadera, rodilla y tobillo se producen máximo en dos de los planos antes mencionados, los cuales describen si el individuo está ejecutando correctamente su movimiento o a su vez presenta alguna patología en el caso de los deportistas el análisis en cada uno de estos planos presenta una información detallada acerca de cómo el individuo está ejecutando una determinada técnica.

Los movimientos de flexión y extensión se producen en el plano sagital (Figura 1.3) la flexión en la cadera se produce al llevar la pierna hacia adelante, con la extensión ocurre lo contrario, llevando la pierna hacia atrás, en el caso de la rodilla la flexión se produce al llevar el pie hacia los glúteos a su vez la extensión consiste en llevar el pie hacia adelante [2].

En el tobillo los movimientos reciben el nombre de flexión dorsal o dorsiflexión cuando el pie se mueve hacia arriba es decir cuando el dorso se dirige hacia la parte anterior de la pierna y flexión plantar cuando el pie se mueve hacia abajo alejando el dorso de la pierna [2].

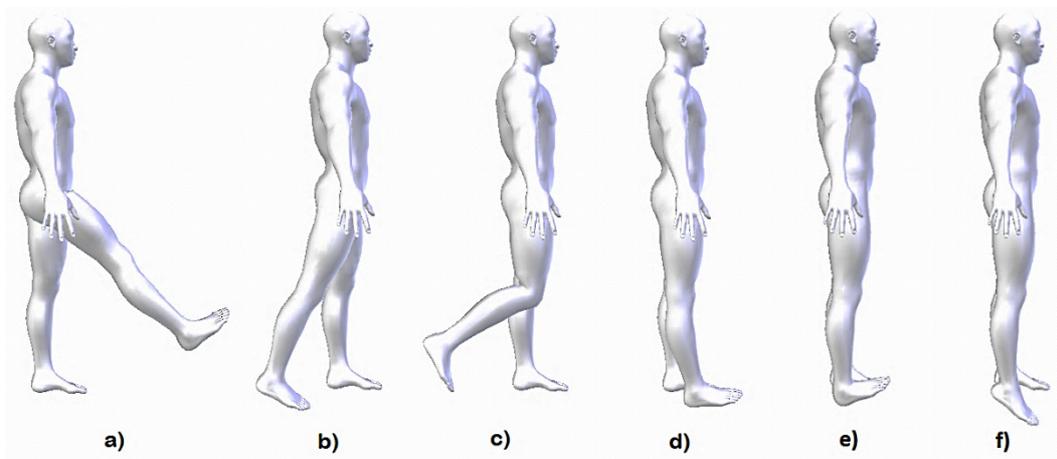


Figura 1.3: a) Flexión de la cadera, b) Extensión de la cadera, c) Flexión de la rodilla, d) Extensión de la rodilla, e) Flexión dorsal, f) Flexión plantar

Los movimientos que se producen en el plano frontal reciben el nombre de abducción cuando la pierna es separada de la posición normal y aducción al regresar la pierna a su sitio original [2].

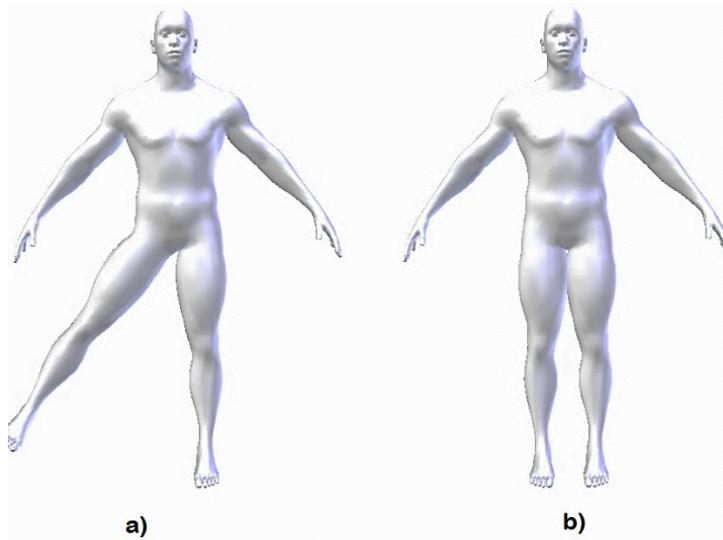


Figura 1.4: a) Abducción, b) Aducción

En el plano transversal se producen los movimientos de rotación externa y rotación interna como se puede ver en la Figura 1.5.

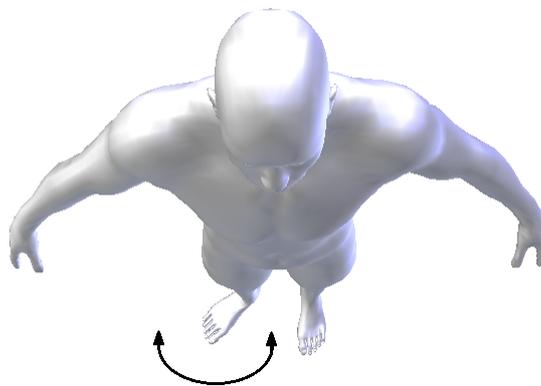


Figura 1.5: Rotación

1.1.3 BIOMECÁNICA DE LA CADERA

La articulación de la cadera o también denominada articulación coxofemoral está compuesta por la cabeza del fémur y el acetábulo de la pelvis, es la encargada de unir el miembro superior con el miembro inferior. La variación de los ángulos de rotación depende de la combinación de movimientos con otras articulaciones, en la cadera se tiene dos tipos de movimientos de flexión, la activa y la pasiva la diferencia entre estas dos es la amplitud de los ángulos que se producen [8].

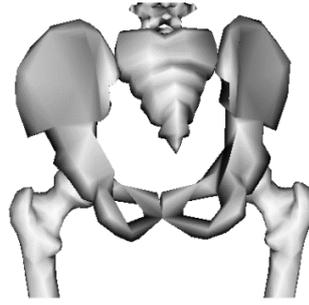


Figura 1.6: Vista frontal de las dos articulaciones de la cadera

Flexión activa: Depende del movimiento efectuado por la rodilla, cuando se produce una extensión en la rodilla el ángulo de flexión de la cadera puede llegar a 90 grados, mientras que cuando se produce una flexión en la rodilla el ángulo de flexión de la cadera puede ser mayor a 120 grados [8].

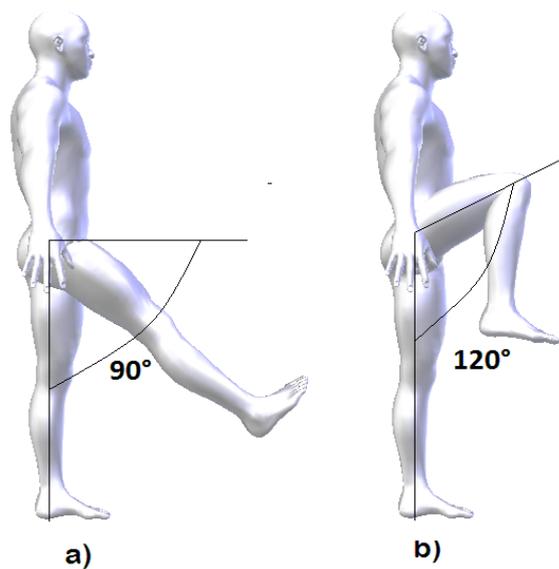


Figura 1.7: a) Flexión de la cadera con extensión de la rodilla, b) Flexión de la cadera con flexión de la rodilla

Flexión pasiva: La amplitud del ángulo para la flexión pasiva es mayor que el de la activa debido a que se requiere de un apoyo para efectuarla. Cuando se produce una flexión en la rodilla el ángulo de flexión de la cadera es de aproximadamente 140 grados, si se produce una extensión de la rodilla el ángulo de flexión de la cadera es mucho menor, otro punto importante es cuando se flexiona las dos caderas y las dos rodilla hasta que estas estén muy cerca del contacto con el tórax, el ángulo de flexión es aún mayor [8].

La extensión es otro de los movimientos producidos en la cadera que consiste en llevar la pierna hacia la parte posterior del cuerpo, se divide en dos tipos extensión activa y extensión pasiva.

Extensión activa: Cuando la rodilla se encuentra extendida la extensión de la cadera puede llegar a tomar un ángulo de 20 grados mientras que cuando la rodilla se flexiona el ángulo de extensión de la cadera puede llegar a los 10 grados aproximadamente [8].

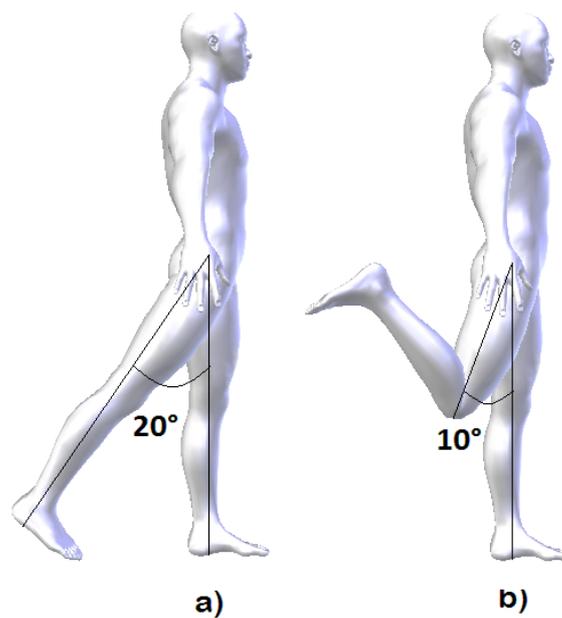


Figura 1.8: a) Extensión de la cadera con extensión de la rodilla, b) Extensión de la cadera con flexión de la rodilla

Extensión pasiva: la amplitud del ángulo de la extensión pasiva es mayor al de la activa y dependiendo de la flexibilidad de cada persona puede llegar a los 20 grados al realizar un paso hacia adelante y un ángulo de 30 grados al flexionar completamente tanto la rodilla como la cadera [8].

Los movimientos de abducción hacen referencia a la rotación de la pierna hacia la parte externa del cuerpo, sobre un eje perpendicular al plano frontal, la abducción de una sola cadera es posible hasta los 30 grados, a partir de aquí se produce una abducción idéntica en la otra cadera debido a que la pelvis sufre una inclinación el movimiento concluye cuando el ángulo de separación de las dos piernas alcanza los 90 grados, debido al movimiento simétrico de ambas caderas se podría decir que la amplitud del ángulo de una sola cadera es de 45 grados [8].

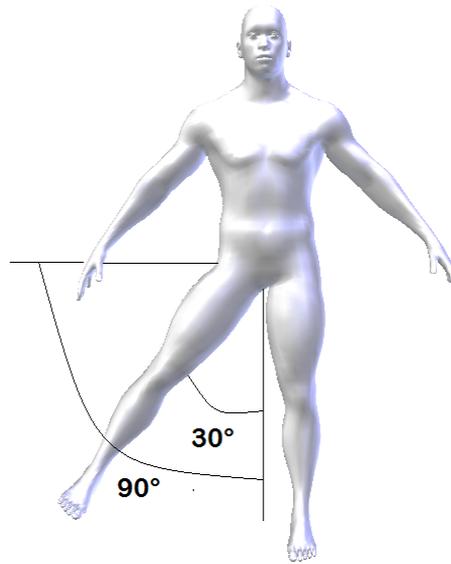


Figura 1.9: Abducción activa

Músculos aductores y ligamentos limitan el movimiento de abducción de la cadera en el individuo, mediante la práctica se puede conseguir una amplitud mayor, es por ello que deportistas de gimnasia o bailarines entre otros, mediante sus entrenamientos pueden llegar a sobrepasar los 90 grados, sin ningún apoyo, llegando a alcanzar ángulos que varían entre 120 a 130 grados aproximadamente [8].

La abducción pasiva se requiere de un apoyo, mediante el entrenamiento se puede lograr ejecutar este movimiento en el cual la amplitud del ángulo puede llegar a alcanzar los 180 grados [8].

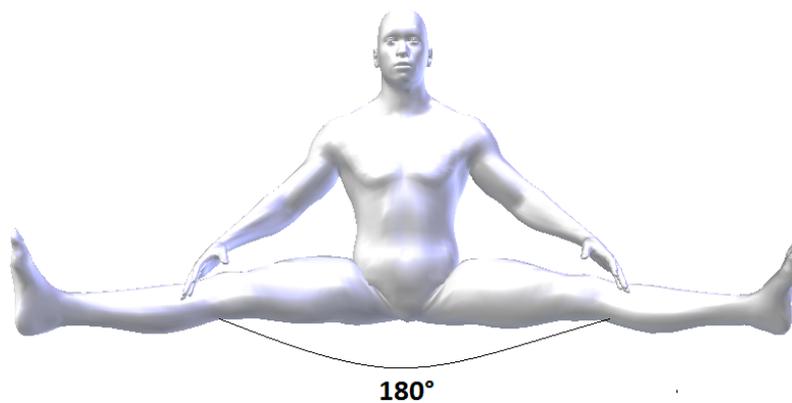


Figura 1.10: Abducción pasiva

1.1.4 BIOMECÁNICA DE LA RODILLA

La rodilla es la articulación intermedia del miembro inferior, encargada de unir el fémur con la tibia, posee dos grados de libertad el primero es el que se produce al llevar la rodilla de la flexión a la extensión y un segundo que se produce de manera dependiente cuando la rodilla se encuentra flexionada [8].



Figura 1.11: Rodilla

Si tomamos como referencia el eje longitudinal de la pierna al realizar una extensión máxima de la rodilla el ángulo puede llegar a ser de 10 grados mientras que al realizar una flexión máxima el ángulo podría llegar a tomar un valor de 120 grados [8].

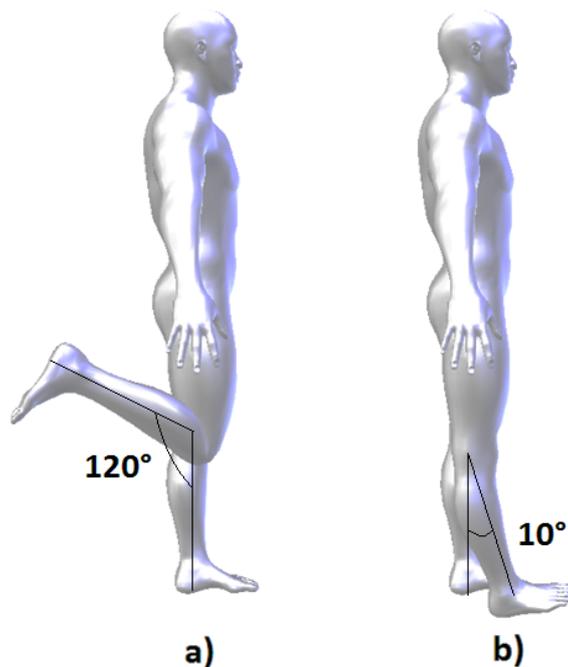


Figura 1.12: a) Flexión máxima de la rodilla, b) Extensión máxima de la rodilla

El segundo grado de libertad correspondiente a la rodilla se produce únicamente cuando la rodilla se encuentra flexionada. En la Figura 1.13 se puede observar un individuo sentado realizando un movimiento giratorio de la rodilla sobre el eje longitudinal de la pierna [8].

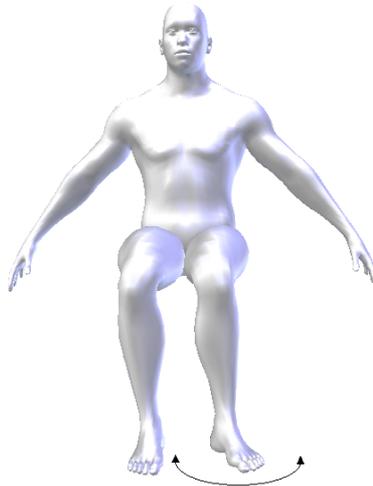


Figura 1.13: Movimiento giratorio de la rodilla

1.1.5 BIOMECÁNICA DEL TOBILLO

El tobillo es la articulación que une el pie con la pierna, posee un solo grado de libertad, su movimiento es llevado a cabo en el plano sagital. Cuando el individuo se encuentra apoyado con un solo pie esta articulación recibe todo el peso del cuerpo e incluso aumenta por la energía cinética producida en la fase de apoyo simple durante la caminata normal [8].

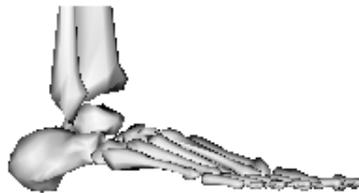


Figura 1.14: Tobillo

Los movimientos producidos en el plano sagital reciben el nombre de flexión dorsal o flexión y flexión plantar o extensión. Tomando como referencia la figura 1.16a en la cual se puede observar como el pie es perpendicular el eje longitudinal de la pierna al realizar una flexión dorsal el ángulo al cual se puede llegar es de 20 a 30 grados mientras que para la flexión plantar la variación del ángulo puede ser de 30 a 50 grados [8].

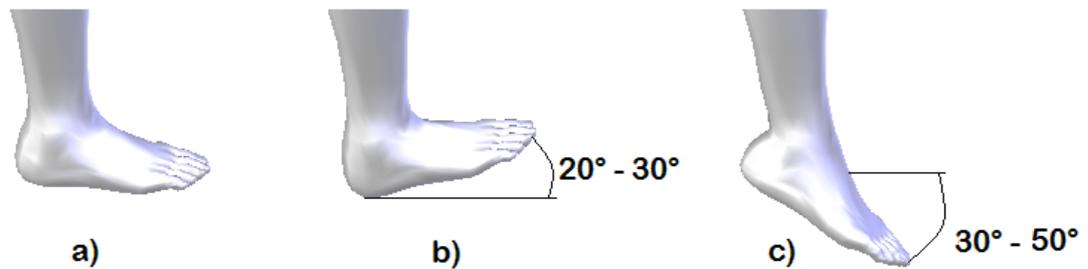


Figura 1.15: a) Posición neutra del tobillo, b) Ángulo de flexión dorsal, b) Ángulo de flexión plantar

Cabe recalcar que en el caso de la cadera y la rodilla al realizar movimientos de flexión los ángulos formados son mayores que en la extensión, excepto en el tobillo la Figura 1.15 muestra como la amplitud del ángulo durante la extensión del tobillo es mayor que en la flexión [8].

1.1.6 CICLO DE LA MARCHA HUMANA

El ciclo de la marcha humana es una combinación de movimientos alternativos de las dos piernas, inicia cuando el pie entra en contacto con el suelo y termina con el siguiente contacto del mismo pie, dentro del ciclo de la marcha humana se destacan dos componente importantes como son: la fase de apoyo, cuando el pie está en contacto con el suelo y la fase de balanceo, cuando pie se desplaza en el aire [2] [7].

Los siguientes términos se utilizan para identificar los eventos que se producen durante el ciclo de la marcha [2].

1. Contacto inicial
2. Despegue del dedo del pie opuesto
3. Elevación del talón
4. Contacto inicial del pie opuesto
5. Despegue del dedo del pie
6. Pies adyacentes
7. Tibia vertical
 1. (Contacto inicial)

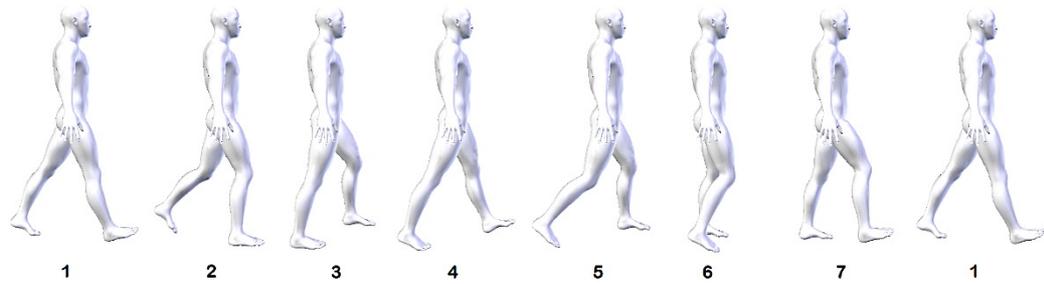


Figura 1.16: Ciclo de la marcha

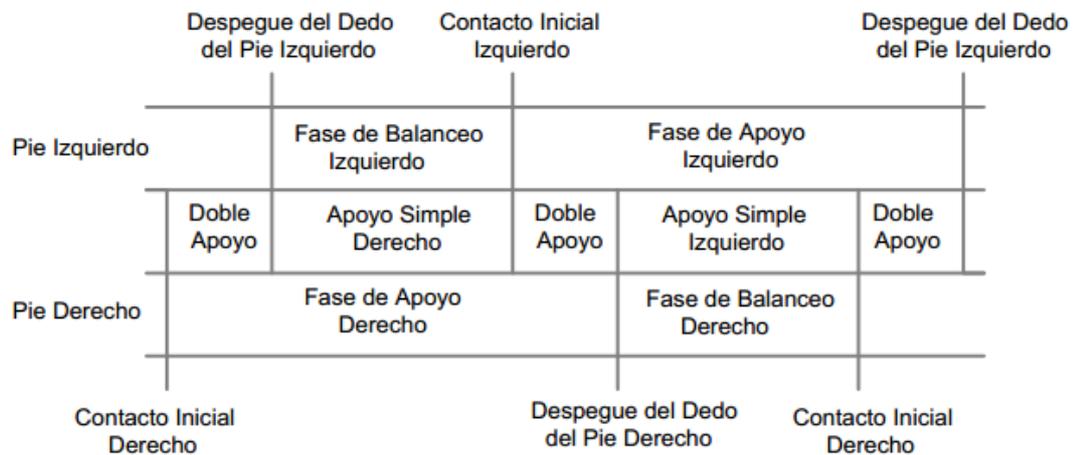


Figura 1.17: Apoyo simple y doble apoyo [2]

Estos siete eventos subdividen el ciclo de la marcha en siete periodos de los cuales cuatro se producen en la fase de apoyo y tres en la de balanceo y se los muestra a continuación.

Fase de apoyo:

1. ***Respuesta de carga:*** Se produce entre el contacto inicial y el apoyo completo del pie con el suelo manteniendo el equilibrio del cuerpo al amortiguar su descenso.
2. ***Posición media:*** Se produce en el instante del despegue de los dedos del pie opuesto y concluye en el despegue del talón del mismo pie.
3. ***Posición final:*** Inicia cuando con el despegue del talón del pie opuesto y termina cuando este vuelve a tener contacto con el suelo.
4. ***Pre-oscilación:*** Instante en el que se produce el segundo doble apoyo el pie se prepara para fase de balanceo.

Fase de balanceo:

5. ***Oscilación inicial:*** Inicia en el momento que despega el pie y termina cuando este alcanza la pierna opuesta.
6. ***Oscilación media:*** Inicia en el instante en que los dos miembros inferiores se cruzan y termina cuando la tibia de la pierna oscilante toma una posición vertical.
7. ***Oscilación Final:*** Es la parte final de la etapa de oscilación de la pierna la cual termina en el momento del contacto inicial.

Durante el ciclo de la marcha se producen dos periodos de doble apoyo y dos periodos de apoyo simple. La fase de apoyo dura alrededor del 60% del ciclo mientras que la fase de balanceo alrededor del 40% y cada periodo del doble apoyo alrededor del 10 % sin embargo estos datos pueden variar con la velocidad que se lleve a cabo el ciclo de marcha [2].

1.1.7 ÁNGULOS DEL MIEMBRO INFERIOR DURANTE UN CICLO DE MARCHA

Para detallar el comportamiento de los ángulos de las tres articulaciones del miembro inferior nos hemos basado en un estudio realizado por Michael Whittle denominado *Gait Analysis an Introduction* en el cual muestra las curvas de los ángulos de una pierna que se producen durante un ciclo completo de marcha sobre un terreno plano (Figura 1.18). La numeración de la parte superior indica los siete eventos producidos con los respectivos periodos en los cuales se subdivide dichos eventos, además se muestra el instante en el que se producen movimientos tanto de flexión como de extensión de las tres articulaciones [2].

El análisis está realizado sobre el plano sagital ya que es aquí en donde se produce una mayor variabilidad de los ángulos, aunque también existe una leve variación en los otros planos [2].

Las curvas de la figura 1.18 corresponden a un ciclo de marcha normal cuyos datos fueron obtenidos dentro de un laboratorio, los valores pueden variar debido a las características propias de cada persona ya sea por el tranco, velocidad con la que ejecute el movimiento, entre otras, sin embargo hay que considerar el patrón de la señal la cual indica los cambios que se producen durante el ciclo de marcha.

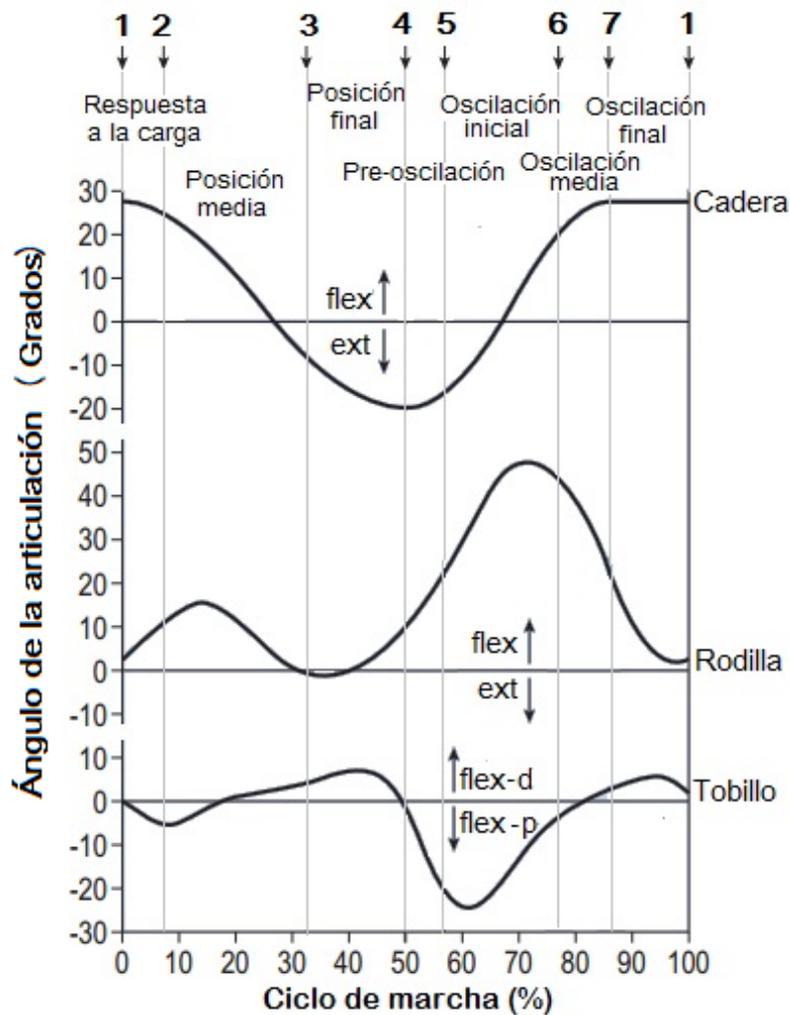


Figura 1.18: Ángulos del ciclo de marcha de las tres articulaciones de miembro inferior sobre el plano sagital [2]

1.2 SISTEMAS EMBEBIDOS PARA EL ANÁLISIS BIOMECÁNICO

Existen centros especializados en el análisis de la marcha, compuesto por un espacio de alta tecnología que permite hacer un diagnóstico preciso acerca del estado de movimiento de las personas que presentan problemas los miembros inferiores. Algunos de estos laboratorios de análisis de movimiento cuentan con cámaras que emiten luz infrarroja sobre marcadores colocados previamente en el cuerpo del paciente, iluminándolos y enviando los datos recopilados a la computadora para su respectivo análisis, sin embargo estos están limitados a un área y condiciones que se deben cumplir estrictamente. Se cita el caso de la luz ambiente, el tipo de caminata que debe realizar el paciente además cabe destacar la influencia que la conexión de equipos de instrumentación puede ocasionar.

Ante esta situación se optó por el uso de dispositivos baratos y fáciles de utilizar como son los sensores portátiles debido a su aplicación en el análisis y diagnóstico de enfermedades patológicas, actividades deportivas, y rehabilitación.

1.2.1 **SENSORES PORTÁTILES**

Son algunos los tipos de sensores utilizados para el análisis de la marcha humana, entre ellos se tiene giroscopios, acelerómetros, sensores magnéticos, de fuerza entre otros, cada uno de estos presenta diferentes características que serán descritos a continuación.

Acelerómetros: Son sensores inerciales su principio de funcionamiento está basado en la segunda ley de Newton la cual establece que la fuerza = masa x aceleración. Este dispositivo permite medir la aceleración en los tres ejes [9].

Giroscopios: Son sensores inerciales que miden la velocidad angular en los tres ejes y su principio de funcionamiento se basa en la medición del efecto Coriolis que se produce entre la estructura del giroscopio y una segunda estructura existente dentro del dispositivo [10].

Magnetómetros: estos dispositivos miden el campo magnético, son utilizados para la determinar la orientación de un objeto en cualquiera de los tres ejes del sistema de coordenadas [10].

1.3 **DIPOSITIVOS INALÁMBRICOS PARA EL ANÁLISIS BIOMECÁNICO**

Los equipos inalámbricos para el análisis de la marcha humana son sistemas en los cuales se produce un trabajo conjunto de diferentes dispositivos como son: sensores inerciales para la medición del movimiento, microcontroladores para la adquisición de los datos, módulos inalámbricos para transmitir los datos hacia el computador ya sean de radio frecuencia, bluetooth o wifi, y un computador para el procesamiento y visualización de los datos.

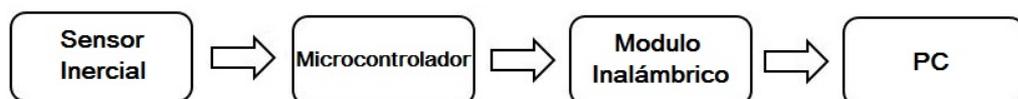


Figura 1.19: Diagrama de bloques de los dispositivos inalámbricos

El uso de los sensores inerciales en el análisis de la marcha humana ha ido creciendo en los últimos años y podemos encontrar una serie de trabajos realizados utilizados para el monitoreo del movimiento.

El trabajo realizado por Qi An et al. [11], consistió en ubicar tres giroscopios en la cadera, rodilla y tobillo además de un sensor ubicado en la parte posterior del individuo con el propósito de medir los ángulos de dichas articulaciones. Los datos fueron transmitidos mediante un módulo bluetooth y mediante cable además de contar con un software el cual permite la visualización de las curvas de los datos obtenidos y de un modelo musculoesquelético para la recreación del movimiento.

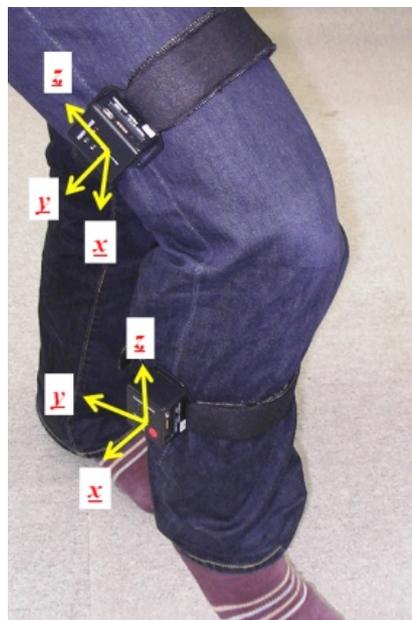


Figura 1.20: Ubicación de los sensores de acuerdo de Qi An et al [11]

Los datos obtenidos fueron comparados con otro experimento realizado mediante un sistema de captura de movimiento óptico en un laboratorio implementado de cámaras y marcadores. Se obtuvieron los coeficientes de correlación entre los dos sistemas, los resultados fueron alentadores demostrando que el uso de sensores inerciales puede ser implementado en la evaluación de parámetros biomecánicos del cuerpo humano.

Las aplicaciones de sensores inerciales no están centradas únicamente en la marcha humana y se han ido expandiendo hacia el campo deportivo debido a su tamaño reducido y sobre todo la gran ventaja de que el análisis puede ser realizado en cualquier espacio ya sea pistas de atletismo, velódromos, parques etc. Lo cual

permite a los entrenadores planificar correctamente los entrenamientos de los deportistas.

En el año 2013 se presentó un trabajo realizado por: Lattes, M. et al. [12] aplicado a los deportistas que practican ciclismo en pista. Se ubicaron acelerómetros que proporcionaban información acerca de aspectos como la velocidad, aceleración y cadencia para monitorear el desenvolvimiento del deportista dentro de su área de trabajo.

Por otro lado en la natación, el análisis mediante cámaras de video que captan el movimiento posee ciertas desventajas: la refracción de la luz en el agua, el oleaje y burbujas producidas debido a las características propias del agua dificultan el análisis, es por ello que se encontró en los sensores inerciales un método óptimo que permite un monitoreo adecuado de la ejecución de los movimientos, velocidades y tiempos. Todo esto con el propósito de poder planificar adecuadamente los entrenamientos que permitan alcanzar un mejor rendimiento en el deportista [13].

Una de las técnicas más evaluadas en la natación es el estilo crol o comúnmente conocido como estilo libre, en donde el uso de los brazos es muy importante debido a la ejecución de cuatro eventos los cuales reciben el nombre de: Entrada, Agarre, Tirón, y Empuje. Dichos eventos producen cambios de la aceleración del deportista durante su deslizamiento en el agua [14].

Uno de los estudios sobre la aplicación de sensores inerciales en la natación fue el realizado por J. B. Lee et al. [15]. En el estudio se hizo una comparación de los sensores inerciales con sistemas de cámaras infrarrojos, los sensores fueron ubicados en el antebrazo del deportistas para el análisis de las aceleraciones y desaceleraciones del deportista durante los eventos de entrada y salida.

Otro estudio basado en la utilización de estos sensores pertenece a Farzin Dadashi et al. [14]. El propósito de ese trabajo fue detectar las características temporales durante la práctica de la natación mediante algoritmos fiables para el cálculo de velocidades angulares y aceleraciones. El estudio fue comparado con un sistema de video, en el cual se pudo comprobar que los resultados obtenidos mediante ambos sistemas fueron similares.

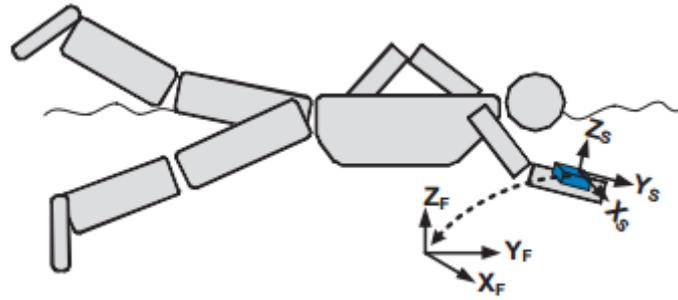


Figura 1.21: Orientación del antebrazo mediante un sensor inercial [14]

Uno de los trabajos más innovadores que se pudo conocer es el realizado por: Guangyi Shi et al. [16]. El cual se trata de un Sistema Airbag Humano para la protección contra caídas aplicado a los ancianos, debido a que estas personas con el pasar de los años se vuelven frágiles y se encuentran más propensos a sufrir lesiones debido a caídas. Ante ello un grupo de investigadores desarrollo dicho sistema que consiste en un cinturón (Figura 1.22), en el cual se encuentran los sensores inerciales y un cilindro con aire comprimido que infla dos bolsas de aire que protegen del golpe en la cadera cuando los sensores detectan la caída de la persona. El reducido tamaño de los sensores inerciales así como del resto de dispositivos lo convierten en sistema ergonómico fácil de utilizarlo.

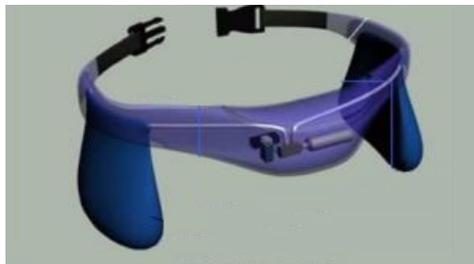


Figura 1.22: Sistema de airbag humano [16]

El sistema hace un seguimiento inalámbrico en tiempo real de la variación de los movimientos del cuerpo humano detectando cuando se produce una desorientación, para ello se tiene una base de datos con aceleraciones y velocidades angulares de caídas. El sistema además cuenta con un filtro SVM (Maquina de Vectores de Soporte) el cual es un algoritmo de reconocimiento de patrones que tiene la finalidad de reconocer cuando una persona está cayendo para activar las bolsas de aire instantes antes de que se produzca el impacto con el suelo.

Los sistemas inalámbricos mostrados anteriormente son algunos de los experimentos realizados mediante la construcción de sistemas basados en sensores inerciales para la adquisición del movimiento humano. Ahora nos referiremos a un equipo sofisticado, que se encuentra disponible en el mercado para el análisis de varios parámetros cinemáticos del cuerpo humano.

Se trata del sistema MVN Biomech Awinda, un traje completo, innovador y ergonómico para el seguimiento de movimiento humano en tiempo real, construido por la empresa XSSENS [17].



Figura 1.23: Sistema MVN Biomech Awinda[17]

El traje está formado por correas elásticas que contienen 17 sensores inerciales ubicados en los diferentes segmentos del cuerpo como son las piernas, brazos, hombros y cabeza. Los sensores obtienen las mediciones de los ángulos de las articulaciones, centro de masa, parámetros temporales, y aceleración. Para que la adquisición de los datos sea óptima el sistema debe ser calibrado y el tiempo requerido para ejecutar esta configuración tarda alrededor de 10 minutos. Los datos se transmiten inalámbricamente al ordenador para posteriormente ser procesados mediante el software MVN estudio que cuenta con algoritmos de sistema de navegación inercial (INS) lo cual permite la animación en 3D en tiempo real de los

movimientos mediante un modelo biomecánico incluido, además de la visualización de las curvas de todos los parámetros obtenidos [18].

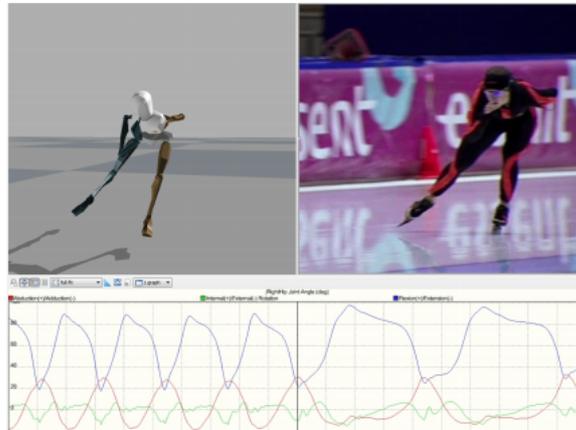


Figura 1.24 Sistema MVN y graficas de los ángulos de la cadera [18]

El traje puede ser implementado en cualquier disciplina deportiva o bien en el análisis de la marcha humana, incluso puede ser implementado sobre animales lo único que se debería cambiar es el modelo biomecánico [18].

CAPÍTULO 2.

2. SISTEMA EMBEBIDO

La construcción de un sistema para la adquisición de los ángulos de los miembros inferiores nace con el objeto de ser un complemento para los laboratorios de marcha con el cual se pueda realizar la evaluación respectiva en distintos espacios físicos donde los deportistas lleven a cabo sus actividades.

La estructura ergonómica hará posible que al momento de hacer el análisis biomecánico de los miembros inferiores el rendimiento del atleta no se vea afectado en mayor medida, ante ello se ha elaborado un traje cómodo que se adapte a las características físicas y anatómicas del deportista.

Se eligió como material el neopreno el cual nos brinda propiedades dieléctricas importantes para mantener a los sensores aislados de diferentes fuentes electroestáticas que pueden producirse con el roce de la piel. Este material concede al deportista propiedades térmicas las cuales evitan lesiones a causa de sobre esfuerzos en el músculo causados por la carencia de un calentamiento previo [19].

El neopreno es un material semielástico que facilita la sujeción y fijación necesaria para que los sensores no varíen de posición de cuerdo a la actividad y en el ámbito biológico relaja al músculo y ayuda a la absorción de la sobre carga en las articulaciones y músculos. La elasticidad del neopreno depende de las costuras que se realicen en este.

Los sujetadores de los sensores en el traje están hechos de algodón el cual nos brinda propiedades dieléctricas y su manejo de elasticidad es más maleable que el neopreno [19].

Para que el traje pueda ser utilizado por diferentes personas de distinta talla se incluyó un regulador de medida mediante correas y velcro, adaptable desde la talla *small* hasta la talla *large*.

El traje está compuesto por:

Una faja de cadera con sujeción en la articulación con el fin de evitar deslizamientos de su posición.

Dos anillos subrotulianos sin almohadilla para mayor movilidad sobre la rodilla.

Dos tobilleras abiertas para evitar deslizamientos debido al movimiento que realiza el tobillo.

Debemos tener en cuenta que toda sujeción por mínima que sea crea un limitante que varía entre los $\pm 5^\circ$ [19].



Figura 2.1: Traje ergonómico.

Los sensores deben ser ubicados en los lugares estratégicos del miembro inferior si bien no existe un estándar para la ubicación de los sensores como si existe para los marcadores utilizados en los laboratorios, la ubicación de estos se los realizará en puntos donde se presente un mayor giro (Figura 2.2).

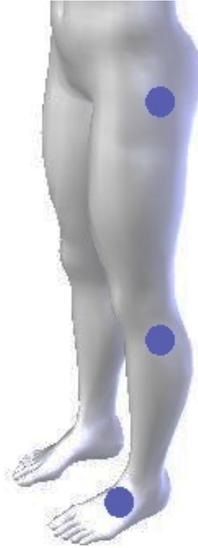


Figura 2.2 Colocación de sensores inerciales.

2.1 DISEÑO DE HARDWARE

Se utilizaron los sensores inerciales MPU 6050 como base del sistema para la adquisición de ángulos de navegación, los integrados ATMEGA 328P-PU como cerebros, Max 485 como intercomunicadores entre los integrados y una tarjeta SD como periférico de almacenamiento.

MPU 6050

Los dispositivos utilizados para la adquisición de datos son los sensores MPU-6050 que trabajan a 5VCC. Estos integrados de 6 ejes en su interior combinan un acelerómetro y un giroscopio. La transmisión de los datos hacia el ATMEGA 328P-PU se da mediante la comunicación I2C. Poseen 3 ADCs de 16 bits cada uno para el giroscopio y 3 ADCs para el acelerómetro los cuales digitalizan la señal de salida, una memoria FIFO de 1024bytes además de un DMP para algunos cálculos complejos en el chip reduciendo de esta manera la carga en el controlador [20].



Figura 2.3: Sensor MPU 6050[20].

2.1.1 COMUNICACIÓN I2C ENTRE MPU 6050 – CONTROLADORES

La transmisión de datos del sensor MPU 6050 hacia los integrados se da mediante la comunicación I2C la cual transmite los datos de forma serial mediante dos conductores SDA y SCL en este tipo de comunicación el integrado toma el papel de maestro iniciando la transferencia y generando la señal de *clock* mientras que los sensores MPU 6050 son los esclavos. En la figura 2.4 podemos apreciar el diagrama a bloques de la red I2C entre el integrado y los giroscopios.

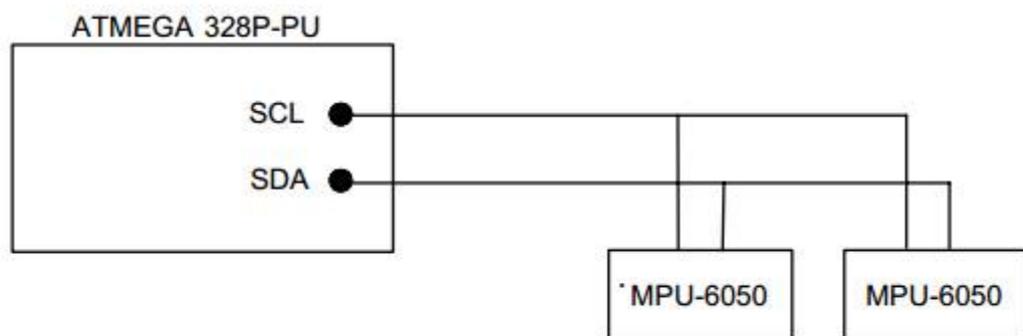


Figura 2.4: Diagrama de la comunicación I2C

Una las características dentro de la comunicación I2C es que cada dispositivo es reconocido mediante una única dirección, los sensores MPU6050 únicamente poseen dos direcciones 0X68 y 0X69 las cuales se las puede seleccionar mediante el pin ADO en el sensor, por lo tanto cada integrado maneja tan solo dos sensores [20].

El pin INT del sensor, entrega una señal de interrupción que indica que la memoria FIFO se encuentra con datos para la transmisión.

Debemos recalcar que el integrado ATMEGA 328P-PU tan solo tiene dos interrupciones posibles en sus pines 4 y 5.

Debido a que son 6 articulaciones a evaluar se optó por la utilización de 4 integrados ATMEGA 328P-PU.

2.1.2 COMUNICACIÓN SERIAL ENTRE INTEGRADOS ATMEGA 328P-PU

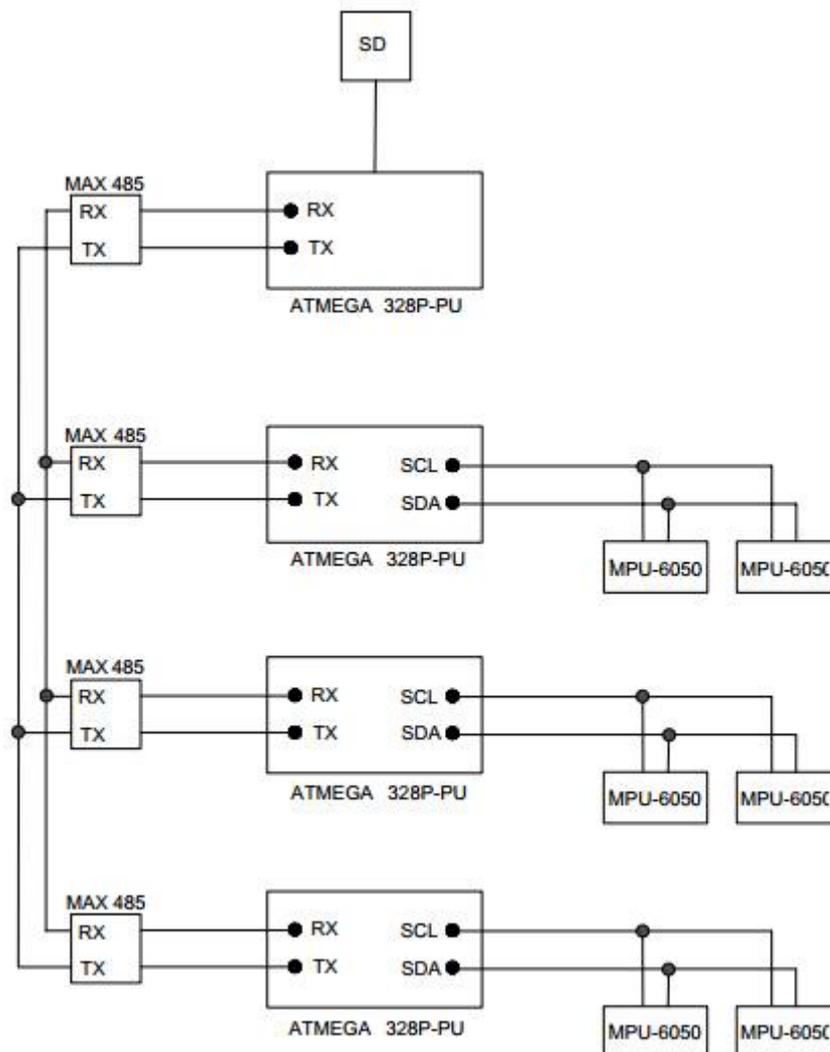


Figura 2.5: Diagrama de conexión de los diferentes integrados y sensores.

Se utilizó una comunicación serial bidireccional para el envío y recepción de datos de forma alterna, el cuarto integrado ATMEGA es el encargado de coordinar la

comunicación. Para dicha comunicación se utilizaron los integrados Max 485 con el propósito de aislar y evitar la colisión de datos mediante el manejo de su pin *enable* el cual es activado por cada uno de los integrados al momento de transmitir los datos. Cada uno de los datos recolectados luego de ser procesados son enviados al cuarto integrado ATMEGA para que este los almacene.

Se podría decir que se tiene una estructura maestro esclavo, pero debemos aclarar que en esta comunicación serial cualquiera podría ser el maestro.

La utilización de un cuarto integrado para el almacenamiento y la coordinación se debe a la velocidad de transmisión y a la cantidad de datos recolectados; además, un solo integrado ATMEGA 328P-PU no puede manejar comunicación serial, comunicación I2C, comunicación SPI y procesamiento de datos.

Cada integrado tiene una dirección dentro de la red serial. Las direcciones son simples números 2, 3 y 4 respectivamente para los esclavos y el 1 para el maestro o dicho de mejor manera el integrado coordinador y almacenador de datos. La velocidad al momento de transmitir por el puerto serial es de 115200 b/s en un caso ideal.

2.1.3 COMUNICACIÓN SPI ENTRE ATMEGA 328P-PU Y TARJETA MICRO SD

Para poder almacenar los datos adquiridos por los sensores MPU6050 y procesados por los integrados ATMEGA se empleó una tarjeta SD cuyo tamaño de almacenamiento puede variar de acuerdo a las necesidades del usuario.

La tarjeta SD es el dispositivo utilizado para el almacenamiento de datos, esta maneja el protocolo de comunicación síncrono SPI, el cual es utilizado para la comunicación con periféricos externos a distancias cortas y de forma rápida.

Los datos adquiridos por cada uno de los integrados son almacenados en archivos separados para poder reconocerlos fácilmente, estos llevan el nombre de: GIRO1.txt, GIRO2.txt, GIRO3.txt los cuales contienen seis columnas cada una correspondiente a un grado de libertad de los giroscopios. La figura 2.6 muestra la conexión entre el integrado ATMEGA328P-PU y el lector de la tarjeta micro SD.

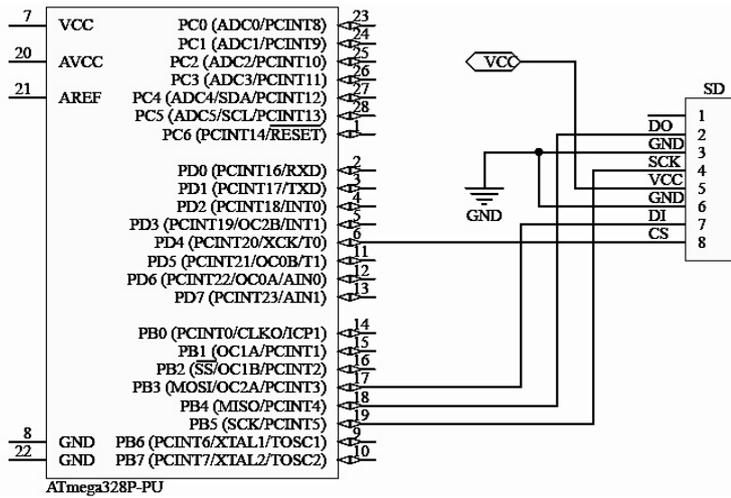


Figura 2.6: Diagrama de conexión entre el integrado ATMEGA 328P-PU y lector micro SD

En las figuras 2.7 y 2.8 se puede observar el esquema de conexión entre cada uno de los integrados, sus respectivas alimentaciones y elementos necesarios para su funcionamiento.

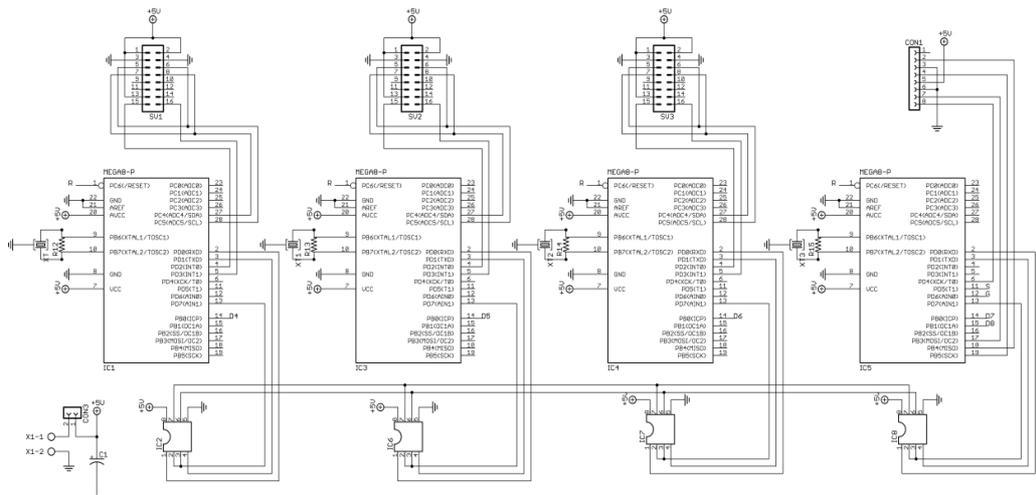


Figura 2.7: Diagrama de conexión del circuito de procesamiento.

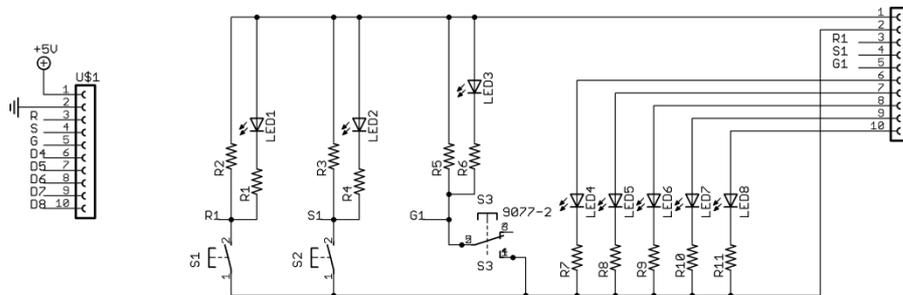


Figura 2.8: Diagrama de conexión del circuito de mando

2.1.4 DISEÑO Y CONSTRUCCIÓN DE LAS TARJETAS ELECTRÓNICAS

Para el Diseño del PCB se redujo su tamaño al mínimo posible para que el usuario no tuviera problemas al momento de utilizarlo.

Se construyó dos Tarjetas Electrónicas, la de procesamiento que cuenta con todas las conexiones de los integrados así como los puertos de salida para la conexión de los MPU 6050 y la tarjeta de mando en donde se encuentran los indicadores leds, pulsantes e interruptor para el control del funcionamiento de todo el sistema.

En las siguientes figuras observamos los PCBs de cada tarjeta, y el acabado final de estas.

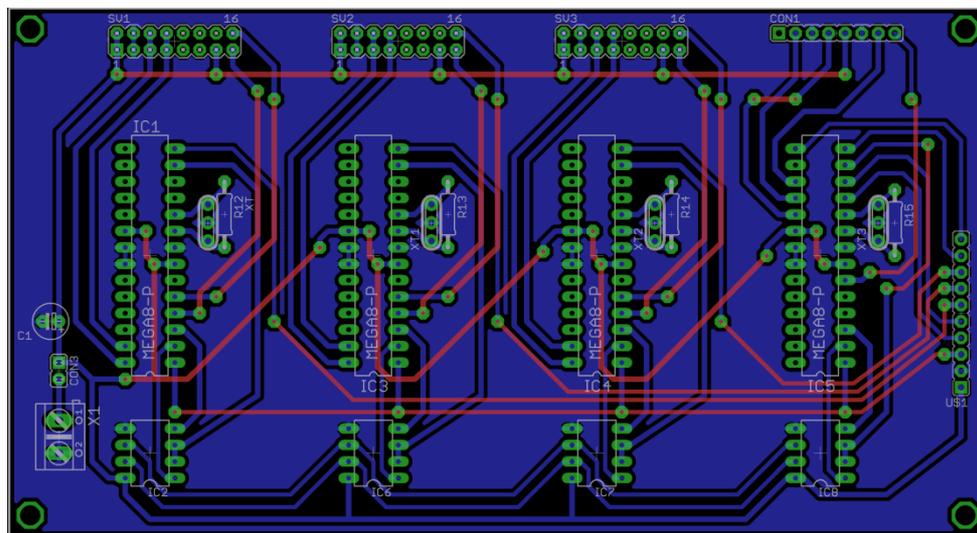


Figura 2.9: PCB tarjeta de procesamiento

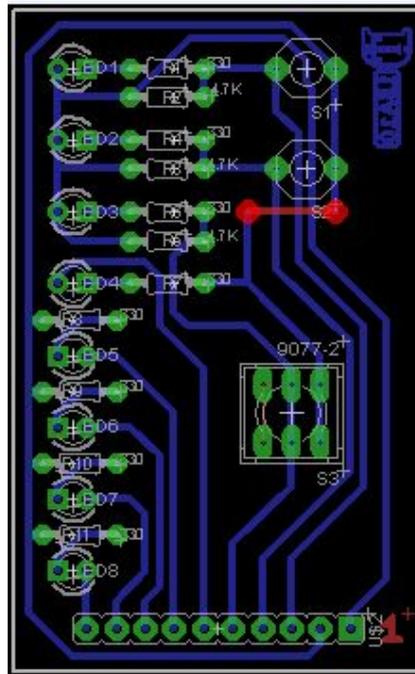


Figura 2.10: PCB tarjeta de mando

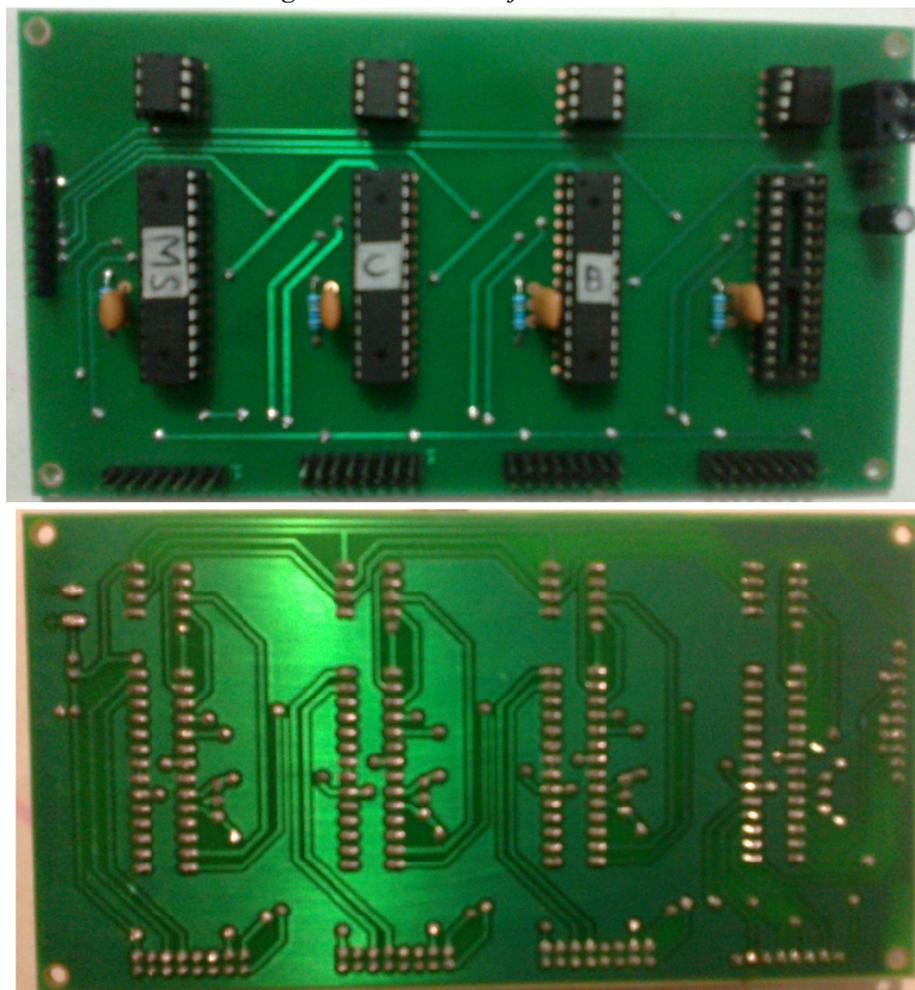


Figura 2.11: Tarjeta de procesamiento

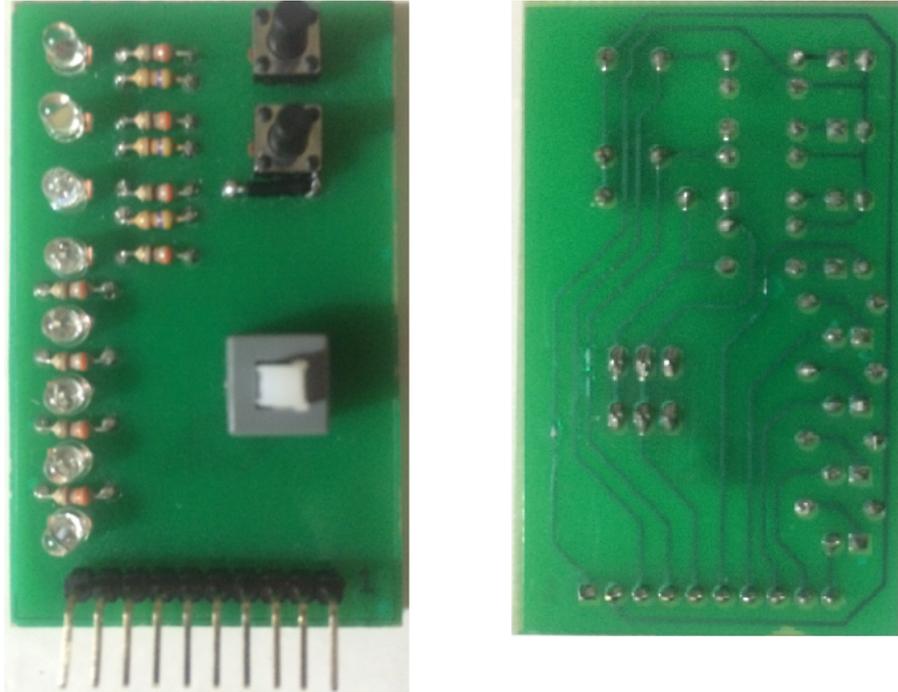


Figura 2:12: Tarjeta de mando

La tarjeta de mando consta de ocho leds. El primero indica el reseteo de los integrados, el segundo indica el seteo de los ángulos, el tercer led indica la acción de grabación, los siguientes 4 leds indican el funcionamiento de cada integrado y finalmente el octavo led indica que los datos se están grabando en la tarjeta SD.

La disposición de esta cantidad de leds se la estipuló para que los futuros desarrolladores del sistema pueden identificar de una manera más sencilla la ubicación de errores y para que el usuario denote cuando debe pulsar el interruptor de grabación (El interruptor de grabación debe ser pulsado siempre y cuando los 4 leds de funcionamiento estén parpadeando). También se cuenta con un *switch* de encendido y apagado. Un pulsante de *set* otro de reseteo y un interruptor de grabación.

2.2 DISEÑO DEL SOFTWARE

El sistema embebido detector de ángulos consiste en la recolección de datos enviados por los sensores inerciales mediante una red I2C, los cuales llegan a un integrado que los transforma en ángulos de navegación y los transmite de forma serial al integrado encargado de ordenar y almacenar los datos en archivos diferentes dentro de una tarjeta SD. Estos datos de ángulos son utilizados en el programa de

matlab para generar un archivo .mot el cual nos permite simular los movimientos realizados mediante el programa de *OpenSim*.

2.2.1 CÁLCULO DE LOS ÁNGULOS DE NAVEGACIÓN

Los ángulos de navegación son un utilizados para describir la orientación de un objeto en tres dimensiones. Si se tiene un sistema de coordenadas móvil respecto de uno fijo, en tres dimensiones, y se desea dar la posición del sistema móvil en un momento dado se lo puede lograr con estos ángulos. Los tres ángulos son la dirección o yaw (ψ), elevación o pitch (θ) y ángulo de alabeo o roll (ϕ), corresponden a los valores de estas tres rotaciones principales. (Figura 2.13)[21].

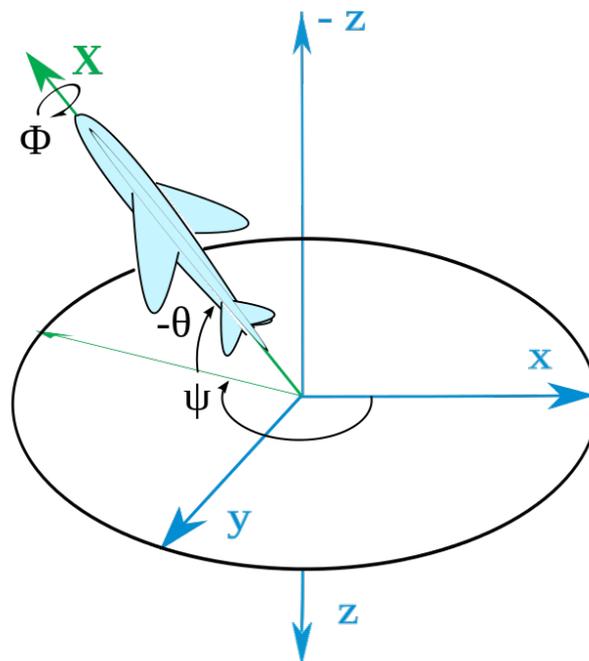


Figura 2.13: Ángulos de navegación[21]

Para poder calcular los ángulos de navegación recordemos que el integrado MPU6050 contiene un giroscopio que entrega la velocidad angular sobre cada uno de sus tres ejes independientes. Con estos valores se puede calcular el ángulo de giro sobre cada eje mediante la ecuación de movimiento uniformemente variado.

$$\theta = \theta_0 + \omega_0 t + \frac{(\omega - \omega_0)t}{2} \quad (1)[22]$$

Donde:

ω es la velocidad angular dada por el giroscopio

ω_0 es la velocidad angular inicial

θ es el ángulo de giro sobre los eje
 θ_0 es el ángulo de giro inicial

Los ángulos obtenidos anteriormente son también conocidos como ángulos de Euler φ , θ y ψ , que son un conjunto de tres coordenadas angulares que sirven para especificar la orientación de un sistema de referencia de ejes ortogonales, normalmente móvil, respecto a otro sistema de referencia de ejes ortogonales normalmente fijos [23]. En la figura 2.14 se observa los ángulos de Euler.

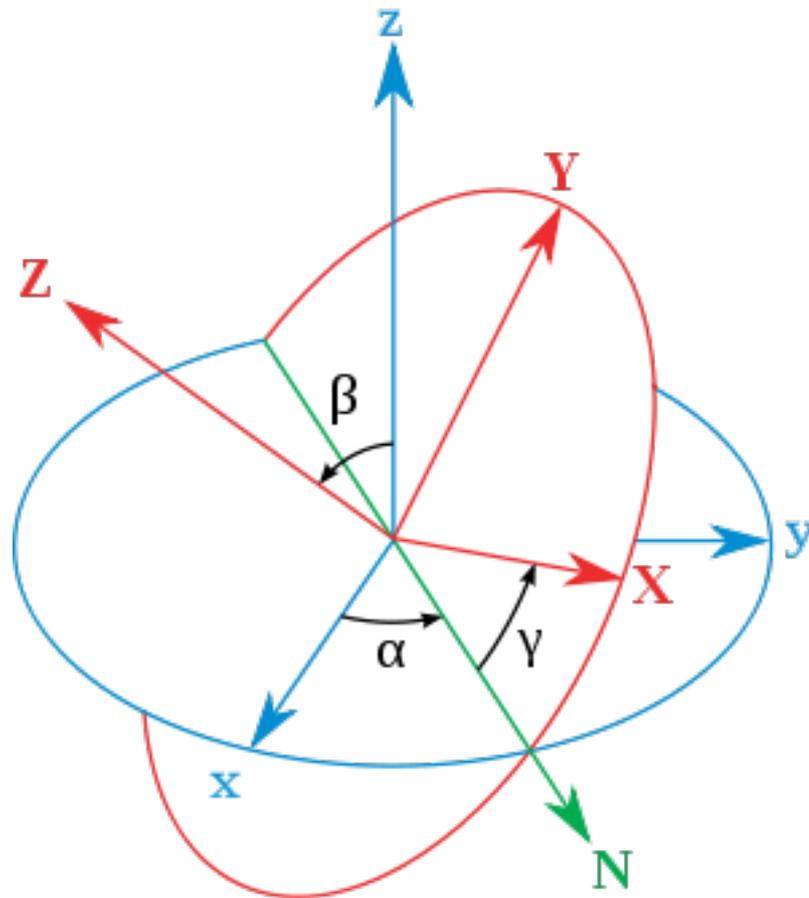


Figura 2.14: Denotación de los ángulos de Euler [21]

Estos ángulos los representaremos como cuaterniones para representar las orientaciones y las rotaciones de objetos en tres dimensiones[23]. Tiene componentes en i , j , k que representan los ejes alrededor del cual se producirán una rotación. También tiene componente q_0 , que representa la cantidad de rotación que se producirá alrededor de este eje[24][25].

Un cuaternion está representado de la siguiente manera

$$q = q_0 + q_1i + q_3j + q_4k \quad (2)[26]$$

Los cuaterniones se los puede calcular con las siguientes ecuaciones [27].

$$q_0 = \cos \frac{1}{2} \theta \cdot \cos \frac{1}{2} (\varphi + \psi) \quad (3)$$

$$q_1 = \sin \frac{1}{2} \theta \cdot \cos \frac{1}{2} (\varphi - \psi) \quad (4)$$

$$q_3 = \cos \frac{1}{2} \theta \cdot \sin \frac{1}{2} (\varphi - \psi) \quad (5)$$

$$q_4 = \sin \frac{1}{2} \theta \cdot \sin \frac{1}{2} (\varphi - \psi) \quad (6)$$

En la figura 2.15 se observa el tratamiento que se le da a los cuaterniones en lenguaje C++ y a la matemática necesaria para que represente un punto en el espacio.

```
class Quaternion {
public:
    float w;
    float x;
    float y;
    float z;

    Quaternion() {
        w = 1.0f;
        x = 0.0f;
        y = 0.0f;
        z = 0.0f;
    }

    Quaternion(float nw, float nx, float ny, float nz) {
        w = nw;
        x = nx;
        y = ny;
        z = nz;
    }

    Quaternion getProduct(Quaternion q) {

        return Quaternion(
            w*q.w - x*q.x - y*q.y - z*q.z, // new w
            w*q.x + x*q.w + y*q.z - z*q.y, // new x
            w*q.y - x*q.z + y*q.w + z*q.x, // new y
            w*q.z + x*q.y - y*q.x + z*q.w); // new z
    }
}
```

```

Quaternion getConjugate() {
    return Quaternion(w, -x, -y, -z);
}

float getMagnititude() {
    return sqrt(w*w + x*x + y*y + z*z);
}

void normalize() {
    float m = getMagnititude();
    w /= m;
    x /= m;
    y /= m;
    z /= m;
}

Quaternion getNormalized() {
    Quaternion r(w, x, y, z);
    r.normalize();
    return r;
}

```

Figura 2.15: Manejo de los cuaterniones[28].

El cálculo de los cuaterniones a partir del giroscopio no es del todo confiable debido a que con el transcurso del tiempo va acarreado y sumando un error, este error se lo conoce con el nombre de drift. Es necesario complementar estos datos con un marco de referencia global en nuestro caso solo podemos valernos de la gravedad, generamos un vector de gravedad que nos proveerá de un valor real en cada instante de tiempo. Para generar dicho vector se parte de las aceleraciones entregadas por el acelerómetro de las cuales podemos calcular los ángulos pitch y roll mediante las siguientes ecuaciones [29].

$$roll = \sin\left(\frac{a_y}{\sqrt{(a_y^2 + a_z^2)}}\right) \quad (7)$$

$$pitch = \sin\left(\frac{a_z}{\sqrt{(a_z^2 + a_x^2)}}\right) \quad (8)$$

Estos ángulos pueden ser representados en forma de cuaternion mediante las ecuaciones [3][4][5][6]. La figura 2.16 muestra el cuaternion de gravedad.

```
uint8_t MPU6050::dmpGetGravity(VectorFloat *v, Quaternion *q) {
    v -> x = 2 * (q -> x*q -> z - q -> w*q -> y);
    v -> y = 2 * (q -> w*q -> x + q -> y*q -> z);
    v -> z = q -> w*q -> w - q -> x*q -> x - q -> y*q -> y + q -> z*q -> z;
```

Figura 2.16: Cuaternion de gravedad [28]

El cuaternion de gravedad contiene la información de la orientación del marco de la tierra con respecto al bastidor del sensor y de los datos provenientes del acelerómetro.

Con el cuaternion resultante del giroscopio podemos calcular los ángulos de navegación y mediante:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan 2(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \arctan 2(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix} \quad (9)[26]$$

El cuaternion consiguiente del acelerómetro nos ayuda a compensar el error del drift. En la siguiente gráfica podemos apreciar el cálculo de los ángulos de navegación compensados

```
uint8_t MPU6050::dmpGetYawPitchRoll(float *data, Quaternion *q, VectorFloat *gravity) {
    // yaw: (about Z axis)
    data[0] = atan2(2*q -> x*q -> y - 2*q -> w*q -> z, 2*q -> w*q -> w + 2*q -> x*q -> x - 1);
    // pitch: (nose up/down, about Y axis)
    data[1] = atan(gravity -> x / sqrt(gravity -> y*gravity -> y + gravity -> z*gravity -> z));
    // roll: (tilt left/right, about X axis)
    data[2] = atan(gravity -> y / sqrt(gravity -> x*gravity -> x + gravity -> z*gravity -> z));
    return 0;
```

Figura 2.17: Cálculo del ángulo de navegación[28].

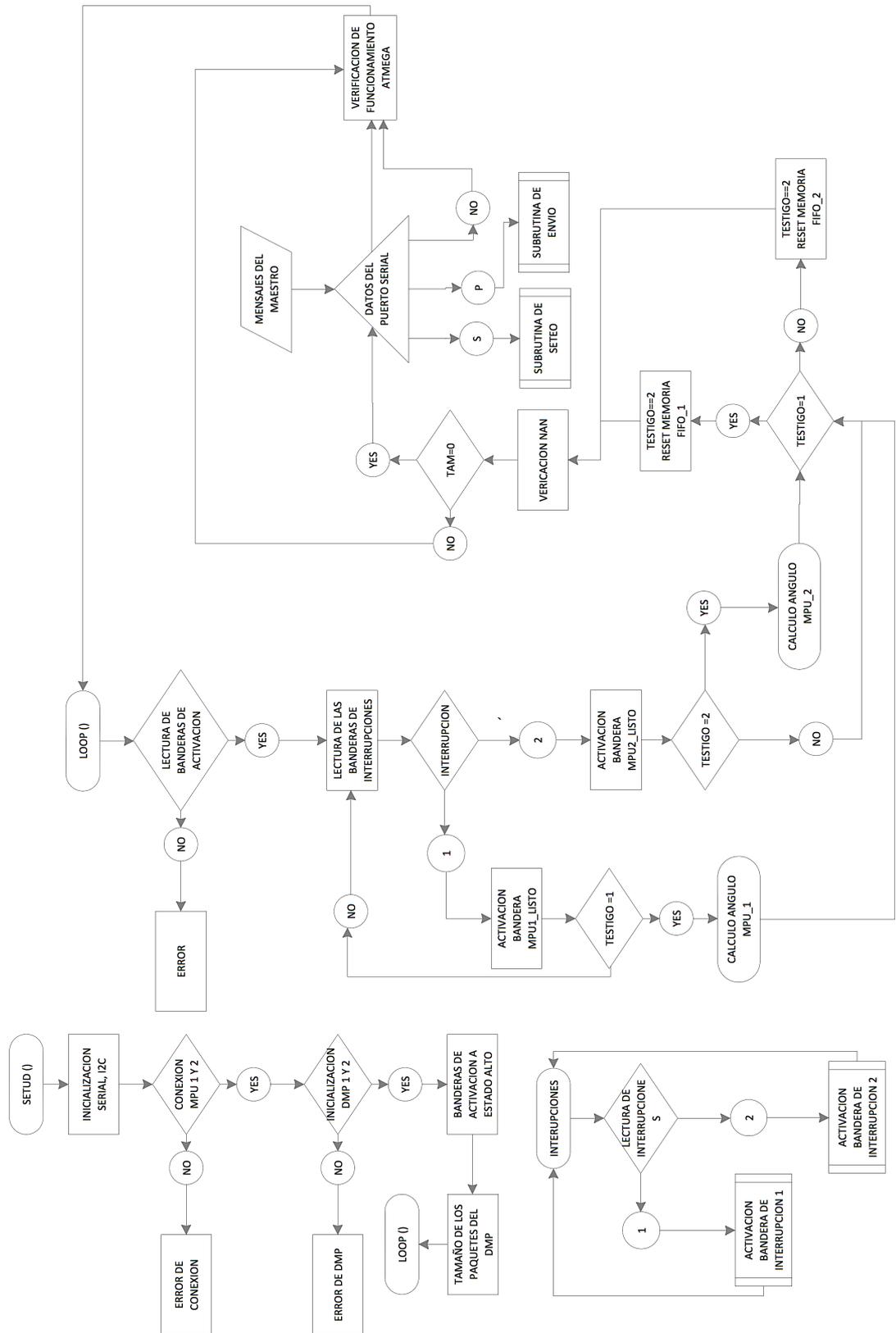
En base a los ángulos de navegación se planteó la posibilidad de recolectar los ángulos de giro de las articulaciones de los miembros inferiores como se los puede apreciar en el capítulo uno.

2.2.2 DISEÑO DE SOFTWARE DEL MICROCONTROLADOR ATMEGA 328P-PU

En este punto se explica el desarrollo del programa para los integrados ATMEGA 328P-PU el objetivo es dotar al sistema embebido de la recolección de datos de los ángulos de navegación y de la configuración de la comunicación interna

que permita al integrado un intercambio de datos de manera segura y ordenada eliminando los datos que puedan causar conflictos en la creación del archivo .mot

2.2.3 CÓDIGO DE PROGRAMA DEL ESCLAVO



Observemos el diagrama de flujo del programa del integrado ATMEGA 328P-PU que ejecuta las funciones de esclavos en la figura 2.18. Debemos recalcar que la programación sobre el IDE de arduino se basa en dos funciones *Void Set up ()* la cual

se ejecuta una sola vez, es aquí donde se inicializa puerto serie, I2C, SPI etc. y la función de *Void Loop ()* que se ejecuta de manera repetitiva. Las librerías, #define y las variables son declaradas y agregadas antes de las dos funciones mencionadas anteriormente. Y las subrutinas se encuentran fuera de las dos funciones principales.

Figura 2.18: Diagrama de flujo del esclavo ATMEGA 328P-PU

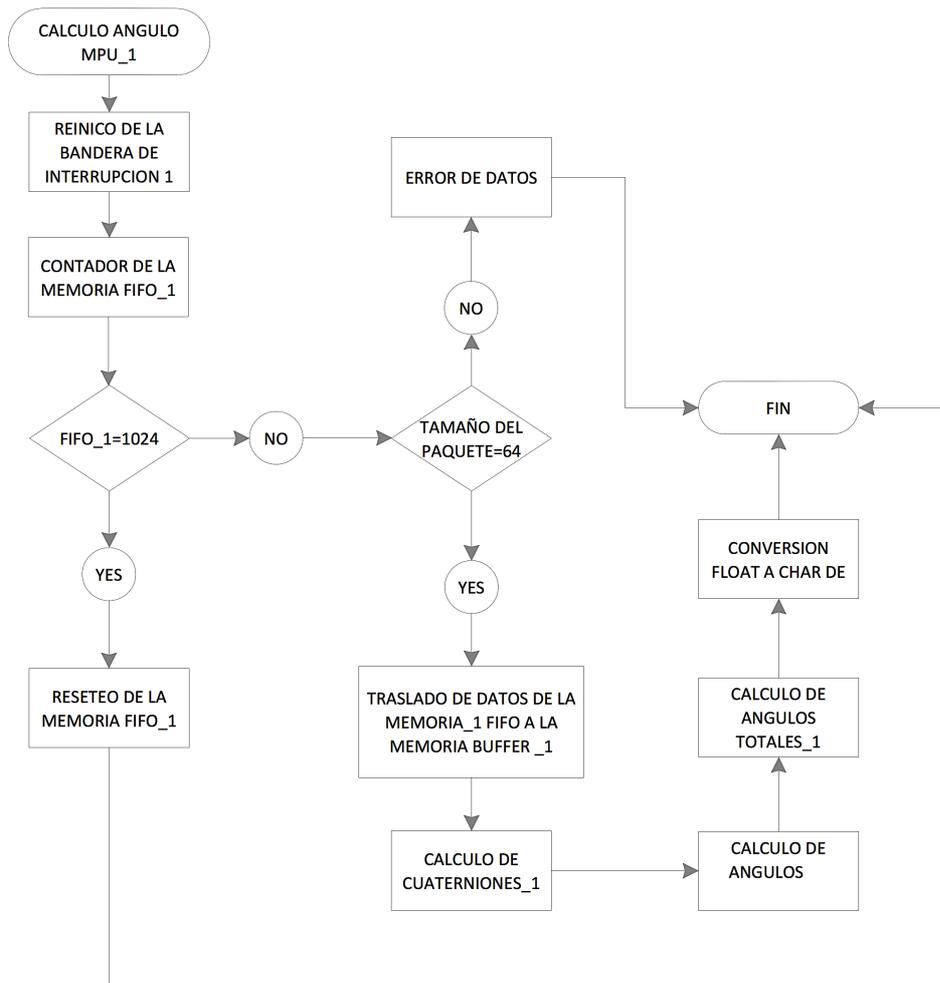


Figura 2.19: Diagrama de flujo del cálculo del ángulo del giroscopio 1

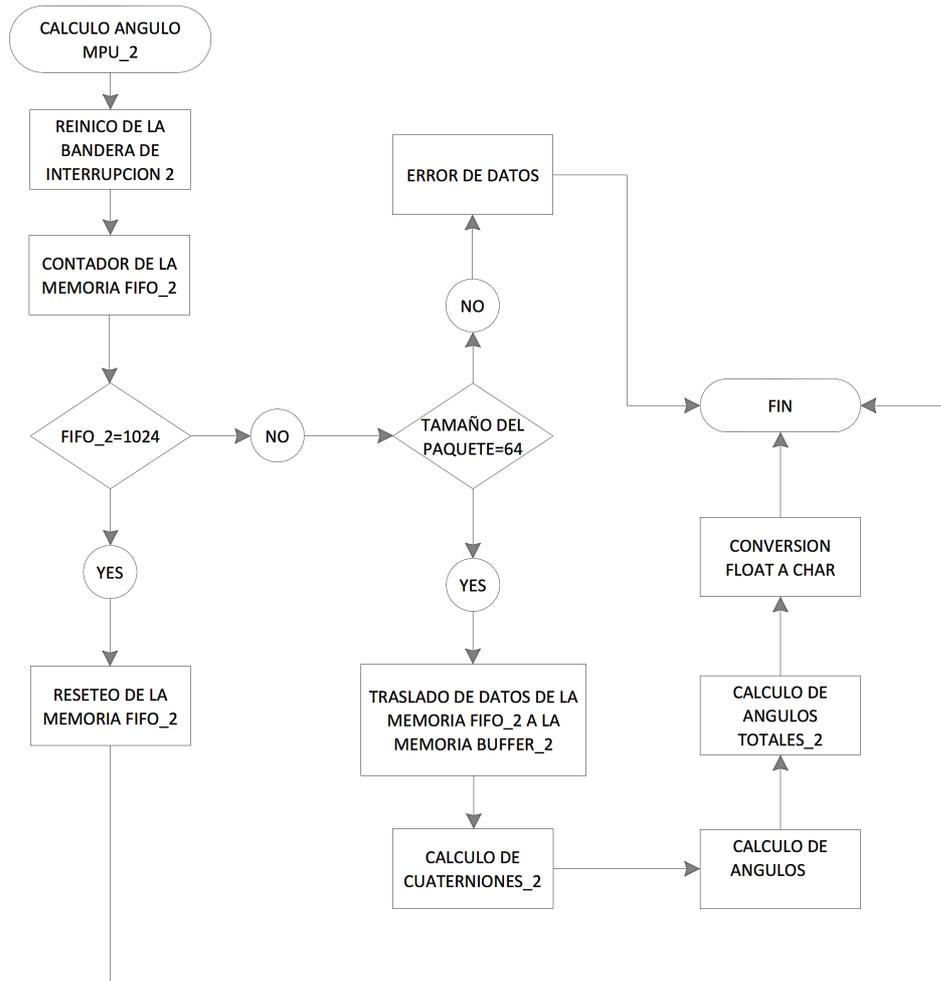


Figura 2.20: Diagrama de flujo del cálculo del ángulo del giroscopio 2

Las bibliotecas proporcionan funcionalidad adicional para el uso en bocetos, por ejemplo, trabajar con el hardware o la manipulación de los datos. Un número de bibliotecas viene instalada con el IDE, pero también se puede descargar o crear uno propio [30].

Una librería es un subproceso que se le añade al programa principal con tan solo llamarlo. Cabe mencionar que algunas de las librerías son creadas por cualquier persona, debido a que la plataforma de arduino es de carácter libre y dichas librerías pueden ser usadas por el programador con total libertad.

Las librerías que se ocuparon para la creación del sketch se las ve en la figura 2.21

```

//Codigo de Programa ATMEGA 328P-PU
//Universidad Politecnica Salesiana
//Flores y Contreras

//Librerias

#include "Wire.h"// Permite la comunicacion I2c /twi dispositivos
#include "I2Cdev.h"//abstrae la comunicacion de bits
#include "MPU6050_6Axis_MotionApps20.h"// libreria del giroscopio
#include <ICSC.h>// Comunicacion serial alterna
#include <math.h>//libreria para verificar un NAN
#include<stdlib.h></span>// libreria DE CONVERTIDOR DE FLOAT A char

```

Figura 2.21: Librerías utilizadas para el sketch

#define es un componente C útil que permite al programador dar un nombre a un valor constante antes de compilar el programa [31].

En la figura 2.22 nótese la definición necesaria para que la librería del giroscopio calcule en su subrutina y nos devuelva los ángulos navegación. Si el *#define* cambia e incrementando las variables necesarias se podría obtener no solo los ángulos de navegación sino también los ángulos de euler, los cuaterniones, las aceleraciones etc.

Obsérvese la declaración de los MPU 6050, las direcciones que se van a utilizar y que están identificadas en la red I2C. Recordemos que los giroscopios MPU 6050 únicamente poseen dos direcciones.

```

//declaracio de la direccion de los giroscopios
MPU6050 mpu_1(0x68); //Primero GY-521
MPU6050 mpu_2(0x69); //Segundo GY-521

//definicion de una constante para que la libreria
//del MPU devuelva angulos navegacion
#define OUTPUT_READABLE_YAWPITCHROLL

```

Figura 2.22: #define y direcciones del MPU

Las variables declaradas son contenedoras de los ángulos, necesarias para el cálculo en la librería “*MPU6050_6Axis_MotionApps20.h*”, y las sobrantes nos sirven para constatar el estado y el tamaño de la memoria FIFO.

Además es necesario usar diferentes tipos de variables y de diferente longitud ya que se debe hacer varios tipos de cálculos y conversiones para su envío.

Finalmente, la verificación de su contenido para que no se envíen datos erróneos por el puerto serial. Cada una de las variables se presentan las en figuras 2.23 y 2.24 con sus correspondientes comentarios, los mismos que son necesarios para identificar su función.

```
// Variables de almacenamiento control y estados

bool dmpReady = false; //Establece true si la unidad Dmp fue exitosa
|
uint8_t mpuIntStatus_1; // sostiene byte de estado de interrupción real de MPU
uint8_t devStatus_1; // devuelve el estado después de cada operación del
//dispositivo (0 = éxito, !=0 = error)
uint16_t packetSize_1; // Tamaño esperado del paquete (default is 42 bytes)
uint16_t fifoCount_1; // recuento de todos los bytes actualmente en FIFO
uint8_t fifoBuffer_1[256]; // Tampón de almacenamiento FIFO

//// se repite las variables para el giroscopio 2
uint8_t mpuIntStatus_2;
uint8_t devStatus_2;
uint16_t packetSize_2;
uint16_t fifoCount_2;
uint8_t fifoBuffer_2[256];

float ypr_1[3]; // coontenedor de angulos
float ypr_2[3]; // [yaw, pitch, roll]
Quaternion q1; // [w, x, Y, z] quaternion container
Quaternion q2; // [w, x, Y, z] quaternion container

uint8_t testigo = 1; // //selecion del giroscopio

VectorFloat gravity1; //vector de gravedad
VectorFloat gravity2;

boolean mpul_listo; //verificacion de los MPU
boolean mpu2 listo;
```

Figura 2.23: Variables del sketch 1

```

float seteo[3]; // ALMACENA EL VALOR D SETEO
float seteo_1[3];

float total[3]; // ALMACENA EL VALOR DEL ANGULO EULER MENOS EL TOTAL
float total_1[3];

int tam=0; // VERIFICADOR QUE NO EXISTA UN NAN EN LAS VARIABLES DEL TOTAL

// VARIABLES QUE CONTIENEN EL VALOR TOTAL PARA SER ENVIADOS
char charVal1[10];
char charVal2[10];
char charVal3[10];
char charVal4[10];
char charVal5[10];
char charVal6[10];

//Parpadeo de un led
const int ledPin = 8;
int ledState = LOW; // estado del LED
long previousMillis = 0; // guarda la ultima vez que el LED se actualiza

long interval = 1000; // intervalo de tiempo cuando el LED parpadea

```

Figura 2.24: Variables del sketch 2

Una variable especial que destaca entre las otras es la variable volátil que sirve como un calificador variable la cual se usa por lo general para modificar la forma en que el compilador y el programa posterior tratasen a la variable. Declarar una variable volátil es una directiva para el compilador. Específicamente, se dirige el compilador para cargar la variable de la RAM y no de un registro de almacenamiento, que es una ubicación de memoria temporal donde se almacenan y manipulan las variables del programa. Una variable debe ser declarada volátil cuando su valor puede cambiarse por algo más allá del control de la sección de código en el que aparece, como un hilo que se ejecuta simultáneamente[32].

En el sketch, los únicos lugares que es posible que esto ocurra es en las secciones de código asociados a las interrupciones [32]. Los MPU 6050 nos entrega una interrupción al momento de que su memoria FIFO se encuentra con datos, muy útil para poder ejecutar las subrutinas. Recordemos que el integrado ATMEGA 328P-PU tan solo posee dos interrupciones. En la figura 2.25 observamos el código de las variables de clasificación

```

// Variable de clasificacion

volatile bool mpuInterrupt_1 = true ; // indica si el pin
volatile bool mpuInterrupt_2 = true; //de interrupción MPU ha ido de alta

```

Figura 2.25: Código de la variable de clasificación

Para nuestro caso se declaró 4 subrutinas, las dos primeras nos indican si el DMP de los MPU 6050 están listos para transmitir los datos. Estas subrutinas son llamadas mediante las interrupciones 1 y 2 en los integrados, en las cuales están conectados el pin de interrupción de los giroscopios.

```

// Sub rutinas

// Función que invoca cuando detecta una interrupción en el MPU 1
void dmp_1_DataReady() {
    mpuInterrupt_1 = true;
}
// Función que invoca cuando detecta una interrupción en el MPU 2
void dmp_2_DataReady() {
    mpuInterrupt_2 = true;
}

```

Figura 2.26: Código de las subrutinas activadas por las interrupciones

La siguiente subrutina es llamada por el puerto serial mediante la lectura de un mensaje transmitido por el ATMEGA 328P-PU que actúa como maestro, el mensaje contiene un encabezado con la letra P. esta subrutina está encargada de enviar los datos de los ángulos al integrado ATMEGA 328P-PU que funciona como maestro mediante el puerto serial con un encabezado A, B o C eso depende de que esclavo está enviando el dato. En las siguientes figuras vemos la subrutina descrita en de cada uno de los esclavos.

```

// Subrutina del envío de datos
void DatosA(unsigned char src, char command, unsigned char len, char *data)
{

    ICSC.send(1, 'A', 40, (char*)&charVall); //espacio de memoria con char*
    //40 por la cantidad de informacion que manda el acortarle menor al tamaño
    //del paquete ocasiona el no envío completo dle paquete
}

```

Figura 2.27: Código de subrutina de envío de datos del esclavo 2

```

// SUBrutina del envio de datos
void DatosA(unsigned char src, char command, unsigned char len, char *data)
{

ICSC.send(1,'B',40,(char*)&charVall);//espacio de memoria con char*

}

```

Figura 2.28: Código de subrutina de envío de datos del esclavo 3

```

// SUBrutina del envio de datos
void DatosA(unsigned char src, char command, unsigned char len, char *data)
{

ICSC.send(1,'C',40,(char*)&charVall);//espacio de memoria con char*

}

```

Figura 2.29: Código de subrutina de envío de datos del esclavo 4

Esta subrutina realiza el envío de paquetes a través del puerto serial en forma de una variable tipo *Char*.

La última subrutina es la encargada del *set* de ángulos esto es necesario debido a que los giroscopios van a variar su posición inicial y de acuerdo a la posición inicial se establece el punto inicial o de ángulos navegación igual a cero. Esta rutina es activada por la lectura en el puerto serial mediante un mensaje proveniente del integrado ATMEGA 328P-PU que sincroniza la comunicación serial, el mensaje contiene un encabezado S.

Una vez activada la subrutina guarda los ángulos de navegación ya calculados en base a los datos que manda los MPU 6050 en las variables de *seteo[]*. Esta subrutina es la misma en todos los esclavos en la red serial.

```

|
//subrutina de Seteo
void DatosB(unsigned char src, char command, unsigned char len, char *data)
{
    seteo[0]=ypr_1[0];
    seteo[1]=ypr_1[1];
    seteo[2]=ypr_1[2] ;

    seteo_1[0]=ypr_2[0];
    seteo_1[1]=ypr_2[1];
    seteo_1[2]=ypr_2[2];

}

```

Figura 2.30: Código de la subrutina de seteo

La función *void setup ()* inicializa la red I2C para los giroscopios, la comunicación serial para los integrados aclarando cual va a ser la velocidad, el pin de activación para el Max485 y por supuesto cual va a ser el nombre del integrado en la red serial. A continuación declaramos las cabeceras de los mensajes que ingresen por el puerto serial y las subrutinas a donde se van a redireccionar de acuerdo a dichas cabeceras. Todo esto se lo puede observar en la figura 2.31.

```

//Inicializacion

void setup() {
    Wire.begin(); //Inizialiacion de la I2C
    // Declaracio de puertos Digitales como salidas
    pinMode(7, OUTPUT);
    pinMode(ledPin, OUTPUT);
    // Inicializacion de la comunicacion serial
    ICSC.begin(2,115200,7);// Nombre del esclavo 2, velocidad de transmision/pin de enable para el max.
    ICSC.registerCommand('P', &DatosA);//Registramos P de cabecera de los mensajes provinientes
    // del maestro y datos A la subrutina para el envio de datos
    ICSC.registerCommand('S', &DatosB);//cabecera S subrutina B para seteo de angulos
}

```

Figura 2.31: Inicialización de los puertos I2C y serial en los esclavos

Mediante la red I2C podremos inicializar los MPU, se hará un test de conexión del MPU 6050 y luego la inicialización de los DMP mediante una llamada a la función dentro de la librería esta nos devolverá un 0 indicando que está trabajando el DMP del giroscopio, ahora encenderemos la bandera del DMP la cual es *“mpu_1.setDMPEEnabled (true);”*.

Esperamos que el dato esté listo, para ello utilizamos a las interrupciones las cuales llamarán a la función *“dmp_1_DataReady”* con esto podremos activar nuestra bandera en el programa *“dmpReady = true;”* esta indicará que el dato está listo para

ser transmitido desde el giroscopio al integrado y puede ser usado en la función del Loop (). Finalmente obtendremos el tamaño del paquete que proviene del DMP para luego ser comparado. Todo lo que no esté de acuerdo a esta lógica nos dará un error ya sea en la conexión o inicialización. Todos los mensajes son enviados por el puerto serial, lo mencionado anteriormente se lo repite para el MPU 6050 dos en el proceso y en la lógica implementada. En las figura 2.32 se muestra el código del programa descrito anteriormente.

```

Serial.println(F("Initializing I2C devices..."));
mpu_1.initialize();//el nombre el giroscopio . llama a la funcion en la libreria se llama la funcion initialize
mpu_2.initialize();

// Verificacion de conecion
//nombre del giroscopio.funcion de conecion
Serial.println(F("Testing device connections..."));
Serial.println(mpu_1.testConnection() ? F("MPU6050_1 connection successful") : F("MPU6050_1 connection failed"));
Serial.println(mpu_2.testConnection() ? F("MPU6050_2 connection successful") : F("MPU6050_2 connection failed"));

Serial.println(F("Initializing DMP..."));
devStatus_1 = mpu_1.dmpInitialize();//Nombre del giroscopio.funcion de inicializacion de dmp
devStatus_2 = mpu_2.dmpInitialize();// si esta bien inicializado devuelve un 0

if (devStatus_1 == 0) {
  // turn on the DMP, now that it's ready
  Serial.println(F("Enabling DMP..."));
  mpu_1.setDMPEnabled(true);//Con esto encendemos la bandera del dmp listo para el programa
  Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));
  attachInterrupt(0, dmp_1_DataReady, RISING); // Utilizamos la primera interrupción externa (número 0) que está en el pin digital 2
  // Cuando la interrupción tiene lugar invoca la función "dmp_1_DataReady"

  mpuIntStatus_1 = mpu_1.getIntStatus();

  Serial.println(F("DMP ready! Waiting for first interrupt..."));
  dmpReady = true;//bandera para indica que se puede ya utilizar del DMP

  packetSize_1 = mpu_1.dmpGetFIFOPageSize();//Se obtiene el tamaño del paquete de datos del dmp para luego ser comparado
}
else {
  Serial.print(F("DMP 1 Initialization failed (code ");
  Serial.print(devStatus_1);          // 1 = carga de memoria inicial fracasó
  // 2 actualizaciones de configuración = DMP fracasaron

  Serial.println(F(")"));
}

if (devStatus_2 == 0) {
  Serial.println(F("Enabling DMP..."));
  mpu_2.setDMPEnabled(true);

  Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));
  attachInterrupt(1, dmp_2_DataReady, RISING);
  mpuIntStatus_2 = mpu_2.getIntStatus();

  Serial.println(F("DMP ready! Waiting for second interrupt..."));
  dmpReady = true;

  packetSize_2 = mpu_2.dmpGetFIFOPageSize();
}
else {
  Serial.print(F("DMP 2 Initialization failed (code ");
  Serial.print(devStatus_2);
  Serial.println(F(")"));
}
}

```

Figura 2.32: Inicialización de los giroscopios

La función *loop ()* se repetirá infinitamente. Inicia leyendo la bandera de activación alzada en el setup. Si todo estuvo correctamente instalado e inicializado se pasará a los siguientes bucles, si esta no se encuentra activada el programa no

avanzará y por lo tanto no hará nada. Una vez comprobada la bandera de “dmpReady” esperamos que cualquiera de las interrupciones tenga lugar para que activen las banderas indicadoras de la existencia de datos en el MPU para ser transmitidos al integrado.

```
void loop() {
  //Comienzo del programa
  if (!dmpReady) return;//se lee si la bandera fue activada en el setup

  // Esperar interrupción MPU o paquete extra (s) disponible
  mpul_listo=!(!mpuInterrupt_1 && fifoCount_1 < packetSize_1 );// Si la interrupcion tiene lugar esta listo el MPU
  mpu2_listo=!(!mpuInterrupt_2 && fifoCount_2 < packetSize_2 );//Para transmitir
  //
```

Figura 2.33: Código de programa inicio del loop

Si el MPU está listo y el testigo se encuentra en la posición del giroscopio uno procederemos a reiniciar la bandera de la interrupción y posteriormente haremos un recuento de la memoria FIFO.

Como ya obtuvimos los datos debemos asegurar su veracidad y que la memoria no esté desbordada. Obtenido ya el recuento de la memoria FIFO en el bucle anterior, comparamos que esta no sea igual a 1024. Si esto ocurre significa que se ha desbordado la memoria y por lo tanto los datos que se encuentra ahí son erróneos, con este concepto claro si la memoria es igual a 1024 se descartaran los datos y se enviará un mensaje de advertencia el cual no será enviado al maestro. Para finalizar este bucle de verificación de desbordamiento resetearemos la memoria FIFO.

Si la memoria no se ha desbordado compararemos la longitud del paquete para asegurar que los datos sean correctos. Un dato es correcto si es menor a 64bytes, si el paquete sobrepasa dicho tamaño lo descartaremos.

Si el dato es correcto pasaremos el paquete a la memoria buffer. Finalmente en este proceso haremos una comparación para la aceleración del proceso, si el contador es menor al paquete leeremos inmediatamente sin esperar la interrupción siguiente. Este proceso se lo aprecia en la figura 2.34.

```

//Chequear si hay desbordamientos en la memori FIFO
if ((mpuIntStatus_1 & 0x10) || fifoCount_1 == 1024)//SI sobrepasa los 1024bytes esta desbordada
{
    mpu_1.resetFIFO();//descartamos el dato y reiniciamos la memoria
    Serial.println(F("FIFO 1 overflow!"));//mensaje de desbordamiento no es transmitido al maestro

// De lo contrario, compruebe los datos de DMP listo interrupción
} else if (mpuIntStatus_1 & 0x02) {
    // Espera de longitud de los datos disponibles correcta
    while (fifoCount_1 < packetSize_1) fifoCount_1 = mpu_1.getFIFOCount();

    // Leemos el paquete de la memoria fifo
    if (packetSize_1 >= 64) ;//este debe ser corto por la longitud de datos
    mpu_1.getFIFOBytes(fifoBuffer_1, packetSize_1);//pasamos los datos a la memoria buffer

    //recuento para ver si es menor al paquete disponible para
    //aceleramos el proceso para leer inmediatamente sin esperar la interuccion
    fifoCount_1 -= packetSize_1;

```

Figura 2.34: Código para la extracción y verificación de datos del MPU 6050

Para calcular los ángulos de navegación dentro de las librerías previamente se precisó mediante el *#define* a que parte de esta debe redirigirse. Los datos necesarios para el cálculo de los cuaterniones y el vector de gravedad se los obtiene de la memoria buffer. En la figura 2.35 se denota el bucle para la obtención de los ángulos

```

#ifdef OUTPUT_READABLE_YAWPITCHROLL//se lo definio al principio del programa para q
//devuelva los angulso de navegacion
mpu_1.dmpGetQuaternion(&q, fifoBuffer_1);//con los datos de la memroia buffer
//calculamos las quaternianes y se las muestra en la variable q
mpu_1.dmpGetGravity(&gravityl, &q);//calculo del vector de gravedad
mpu_1.dmpGetYawPitchRoll(ypr_1, &q, &gravityl);//calculos de angulos de navegacion

ypr_1[0]= ypr_1[0]* 180/M_PI;//calculos de angulos
ypr_1[1]= ypr_1[1]* 180/M_PI;
ypr_1[2]= ypr_1[2]* 180/M_PI;

```

Figura 2.35: Código para la extracción de los ángulos de navegación

El siguiente paso es el calcular los ángulos totales en una sustracción del ángulo de navegación actual y el ángulo de navegación capturado al momento del *set*. Posteriormente en este proceso transformaremos las variables “*float total_1 []*” en variables de tipo *char*. Este paso es estrictamente para el envío de datos ya que la librería del puerto serial envía solo datos de tipo *char*. Posteriormente concatenaremos todas los datos en una sola variable “*charVal1*” La figura 2.36 muestra el código del programa para este proceso descrito.

```

total[0]= ypr_l[0]-seteo[0]; //calculo de los angulos de navegacion menos el seteo capturado
total[1]= ypr_l[1]-seteo[1];
total[2]= ypr_l[2]-seteo[2];

dtostrf(total[0], 4, 2, charVal1); //conversion par enviar
dtostrf(total[1], 4, 2, charVal2);
dtostrf(total[2], 4, 2, charVal3);

strcat(charVal1, "\t");
strcat(charVal1, charVal2);
strcat(charVal1, "\t");
strcat(charVal1, charVal3);
strcat(charVal1, "\t");
strcat(charVal1, charVal4);
strcat(charVal1, "\t");
strcat(charVal1, charVal5);
strcat(charVal1, "\t");
strcat(charVal1, charVal6);

```

Figura 2.36: Código de programa para el set y preparación de la variable para él envío

Una vez extraído los datos y calculado los ángulos se debe resetear la memoria *FIFO* y pasar el testigo al giroscopio dos para que este repita el mismo proceso del giroscopio uno. En la figura 2.37 se ve el reseteo del giroscopio y el cambio de testigo.

```

//Reseteo de la memoria y aumento del testigo
if (testigo==1) {
    testigo=2; //con el testigo a 2 pasamos al otro giroscopio
    mpu_1.resetFIFO(); //una vez obtenido los datos se resetea la memoria
}

```

Figura 2.37: Cambio de testigo y reseteo de la memoria *FIFO*

Al realizar cálculos matemáticos complejos hay casos en el que el lenguaje de programación nos devuelve un “*Nan*” (Not a Number) debido un resultado imposible calcular como el caso de las raíces negativas, indeterminaciones, etc.

Es por ello que se implementó un bucle verificador de este “*Nan*”, si se encontrase alguno dentro de las variables de los giroscopios activaría la bandera de no transmisión de datos logrando con esto descartar los datos incompletos. La figura 2.38 muestra el código del proceso descrito anteriormente.

```

//Proceso de verificacion de NAN
int tam=0;//bandera de no transmision

    if (isnan(euler_1[0]))//Verifica si la variable es una Nan
    {
        tam=1;//pasamos a la variable a uno para que no trasmita
    }
    if (isnan(euler_1[1])) {
        tam=1;
    }
    if (isnan(euler_1[2])) {
        tam=1;
    }
    if (isnan(euler_2[0])) {
        tam=1;
    }
    if (isnan(euler_2[1])) {
        tam=1;
    }
    if (isnan(euler_2[2])) {
        tam=1;
    }
}

```

Figura 2.38: Código del proceso de verificación Nan

Posteriormente se creó un proceso de verificación de funcionamiento el cual consiste en el parpadeo de un led cuando esté funcionando el ATMEGA328P-PU. Esto se lo obtuvo anclando el estado del led de *Low* a *High* o viceversa en cada repetición del bucle Loop ().La figura 2.39 muestra el código de este proceso.

```

// esto es para parpadear un led para observar q esta funcionando
unsigned long currentMillis = millis();

if(currentMillis - previousMillis > interval) {
// guarda la ultima vez que el LED parpadea
previousMillis = currentMillis;

// si el LED está apagado, lo prende y viceversa
if (ledState == LOW)
ledState = HIGH;
else
ledState = LOW;

// establecer el LED con el estado de la variable
digitalWrite(ledPin, ledState);
}

```

Figura 2.39: Código del proceso de verificación del funcionamiento.

Finalmente si la bandera de no transmisión esta desactivada se lee el puerto serial para recibir los paquetes provenientes del maestro estos pueden tener un encabezado P o S ya sea para el *set* o envío de datos. El código se encuentra en la Figura 2.40.

```
//condicioanal de transmision
if (tam== 0){
    ICSC.process();//lectura del puerto serial.
}
}
```

Figura 2.40: Lectura del puerto serial

2.2.4 CÓDIGO DE PROGRAMA DEL MAESTRO

A diferencia del esclavo el maestro no tendrá dispositivos conectados en su puerto I2C pero si un dispositivo de almacenamiento masivo en el puerto SPI. En la figura 2.41 observamos el diagrama de flujo del programa.

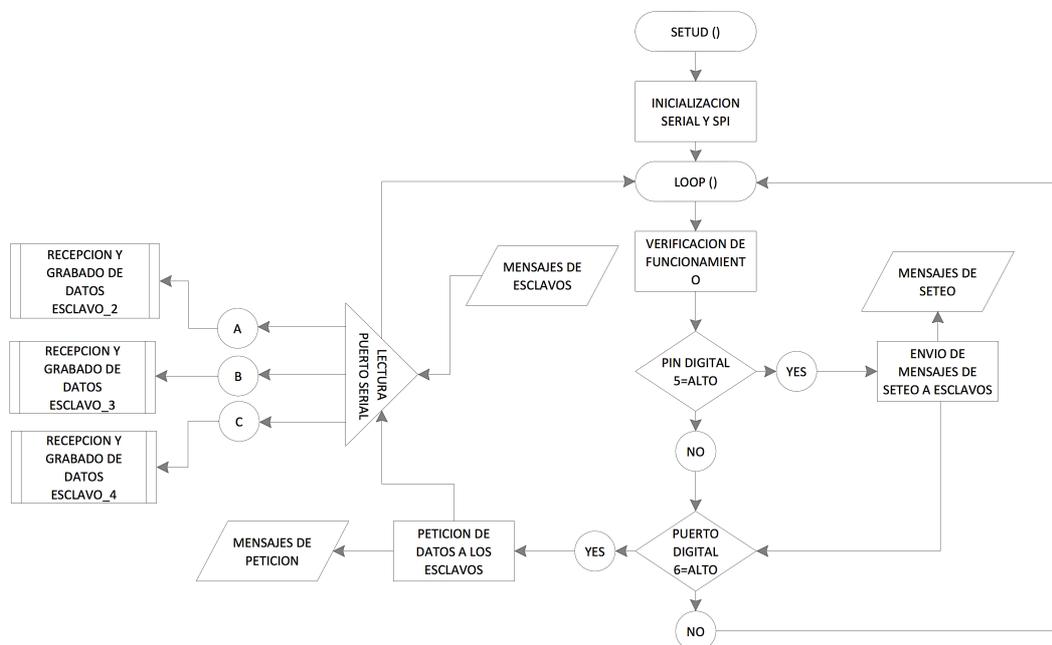


Figura 2.41: Diagrama de flujo del maestro

En este sketch, el grupo de librerías será menor que en el esclavo pero algunas de ellas se utilizarán en ambos casos. En la Figura 2.42, vemos todas las librerías utilizadas por el sketch junto a sus respectivos comentarios que identifican su funcionamiento.

```
#include "Wire.h"//conexion de dispositivos
#include <ICSC.h> // Comunicacion serial alterna
#include <SPI.h>//COMUNICACION SPI
#include <SD.h>//Libreria para la tarjeta SD
File myFile;//PARA PODER ARCHIVOS
```

Figura 2.42: Librerías del sketch del maestro

Una variable fuera de lo común que resalta sobre las otras es la variable de tipo *char** que más que una variable es una declaración de espacio de memoria que genera un incremento notable en el tamaño bytes del sketch. En la figura 2.43 se muestra el tipo de variables que fueron utilizadas y a la vez con un comentario a lado que explica la función de cada variable.

```
//Variables
const int buttonPin = 5; //Boton de seteo
int buttonState = 1;
int tam=0;

//Para grabar
//almacenar en variables
int i=0;//contadores de datos llegados
int j=0;
int k=0;
char* datos_1[5];//espacio de memoria
char* datos_2[5];
char* datos_3[5];

const int buttonPin_1 = 6; //boton de grabado
int buttonState_1= 1;
const int ledPin_1 = 8; //led de grabado

//Parpadeo de un led
const int ledPin = 9;
int ledState = LOW; // estado del LED
long previousMillis = 0; // guarda la ultima vez que el LED se actualiza

long interval = 1000; // intervalo de tiempo cuando el LED parpadea
```

Figura 2.43: Variables del sketch del maestro

Para este sketch se declaró 4 subrutinas las cuales fueron llamadas por el puerto serial mediante la lectura de un mensaje transmitido por cualquiera los integrados ATMEGA 328P-PU que actúan de esclavos, el mensaje contiene un encabezado con las letra A, B, C dependiendo del esclavo. Estas subrutinas están encargadas primeramente de comprobar el tamaño del paquete recibido por los esclavos si son menores a 30 datos serán descartados. En segunda instancia generan un archivo .txt contenedor de los ángulos de navegación provenientes de los esclavos con el fin de grabar dichos archivos en la tarjeta SD, finalmente cada vez que se grabe un dato en la tarjeta SD parpadeará un led. En las siguientes Figuras vemos las 4 subrutinas.

```
void DatosA(unsigned char src, char command, unsigned char len, char *data)
{
    datos_1=data;//cambiamos de variables para evitar conflictos

    l = strlen(datos_1); //sacamos la longitud del char
    //grabacion de los datos
    if(l>30)//si la longitud del char es mayor a 30 llego completo el paquete
    {
        myFile = SD.open("Giro1.txt", FILE_WRITE);

        myFile.println(datos_1);//mandamos a guardar toda la variable
        myFile.close();
    }
    digitalWrite(ledPin_1, HIGH);//encendido y apagado del led de notificación de datos
    digitalWrite(ledPin_1, LOW);
}
}
```

Figura 2.44: Subrutina para el esclavo 2

```
void DatosB(unsigned char src, char command, unsigned char len, char *data)
{
    datos_2=data;//cambiamos de variables para evitar conflictos

    l = strlen(datos_2); //sacamos la longitud del char
    //grabacion de los datos
    if(l>30)//si la longitud del char es mayor a 30 llego completo el paquete
    {
        myFile = SD.open("Giro2.txt", FILE_WRITE);

        myFile.println(datos_2);//mandamos a guardar toda la variable
        myFile.close();
    }
    digitalWrite(ledPin_1, HIGH);//encendido y apagado del led de notificación de datos
    digitalWrite(ledPin_1, LOW);
}
}
```

Figura 2.45: Subrutina para el esclavo 3

```

void DatosC(unsigned char src, char command, unsigned char len, char *data)
{
    datos_3=data;//cambiamos de variables para evitar conflictos

    l = strlen(datos_3); //sacamos la longitud del char
    //grabacion de los datos
    if(l>30)//si la longitud del char es mayor a 30 llega completo el paquete
    {
        myFile = SD.open("Giro3.txt", FILE_WRITE);

        myFile.println(datos_3);//mandamos a guardar toda la variable
        myFile.close();
    }
    digitalWrite(ledPin_1, HIGH);//encendido y apagado del led de notificación de datos
    digitalWrite(ledPin_1, LOW);
}

```

Figura 2.46: Subrutina para el esclavo 4

En la función *set up ()* declararemos los pines de entrada y salida. Inicializaremos la librería para la conexión del dispositivo de almacenamiento masivo. Este proceso lo vemos en la figura 2.47.

```

void setup() {

    pinMode(buttonPin, INPUT);//seteo
    pinMode(buttonPin_1, INPUT);//grabar
    pinMode(ledPin, OUTPUT); //senal de activo
    pinMode(ledPin_1, OUTPUT);//señal de grabado
    Wire.begin();//iniciamos la libreria

    pinMode(7, OUTPUT);//enable del max
}

```

Figura 2.47: Proceso de inicialización de librerías y declaración de pines de salida y entrada

Luego procederemos a declarar el pin digital 7 como una salida para la activación del *enable* del Max485. Posteriormente inicializamos la comunicación serial en la cual declararemos cual es el nombre del dispositivo dentro de la red, la velocidad de transmisión y el pin de activación al momento de transmitir. El siguiente paso es declarar las cabeceras de los mensajes que ingresen por el puerto serial y las subrutinas a donde se van a re direccionar de acuerdo a las cabeceras de los mensajes.

Con el pin digital 4 al CS de la tarjeta SD realizaremos un test de verificación de la conexión de la tarjeta en mención. El proceso descrito anteriormente se lo observa en la figura 2.48 donde se encuentra el código del sketch.

```

ICSC.begin(1,115200,7);// maestRo , velocidad de transmicion/pin de enable para el max
ICSC.registerCommand('A', &DatosA);//Cabecera A subrutina A para el datos del esclavo 2 en la red
ICSC.registerCommand('B', &DatosB);//Cabecera b subrutina b para el datos del esclavo 3 en la red
ICSC.registerCommand('C', &DatosC);//Cabecera C subrutina C para el datos del esclavo 4 en la red
while (!Serial);

// inicializamos la SD
pinMode(4, OUTPUT);// CS para la tarjeta SD como salida
//verificacion de la tarjeta sd
if (!SD.begin(4)) {
  Serial.println("inicializacion Fallida");//mensaje de error
  return;
}
Serial.println("inicializacion correcta");

```

Figura 2.48: Proceso de inicialización del puerto serial y verificación de conexión de la tarjeta SD

Finalmente comprobaremos si en la tarjeta SD existen archivos si es el caso dichos archivos serán eliminados

```

if (SD.exists("Girol.txt")) {
  // delete the file:
  Serial.println("Removing Girol.txt");
  SD.remove("Girol.txt");
}
if (SD.exists("Giro2.txt")) {
  // delete the file:
  Serial.println("Removing Giro2.txt");
  SD.remove("Giro2.txt");
}
if (SD.exists("Giro3.txt")) {
  // delete the file:
  Serial.println("Removing Giro3.txt");
  SD.remove("Giro3.txt");
}

```

Figura 2.49: Verificación de archivos y eliminación de archivos en la tarjeta SD

Ahora realizamos un proceso de verificación de funcionamiento el cual consiste en el parpadeo de un led cuando esté funcionando el ATMEGA328P-PU. La figura 2.50 muestra el código de este proceso.

```

void loop() {

    // esto es para parpadear un led para observar q esta funcionando
    unsigned long currentMillis = millis();

    if(currentMillis - previousMillis > interval) {
    // guarda la ultima vez que el LED parpadea
    previousMillis = currentMillis;

    // si el LED está apagado, lo prende y viceversa
    if (ledState == LOW)
    ledState = HIGH;
    else
    ledState = LOW;

    // establecer el LED con el estado de la variable
    digitalWrite(ledPin, ledState);
    }
}

```

Figura 2.50: Proceso de verificación

Luego procedemos a leer el pin digital 5 en el cual se encuentra un pulsante al ser presionado se enviará un mensaje con encabezado S a todos los esclavos con el propósito de *setear* el valor de los ángulos. En la Figura 2.51 vemos el proceso descrito anteriormente.

```

buttonState = digitalRead(buttonPin);
if (buttonState == HIGH) {

    //ICSC.process();
    //envio del pulso para el seteo en los otros modulos
    ICSC.send(2, 'S', 5, "1");
    ICSC.send(3, 'S', 5, "1");
    ICSC.send(4, 'S', 5, "1");

}

```

Figura 2.51: Proceso de envío de seteo

Finalmente se lee el puerto digital 6 el cual consta de un pulsante que al ser activado hará que el programa entre en un bucle para el envío de petición de ángulos y lectura del puerto serial para la recepción de los ángulos provenientes de los esclavos.

```

buttonState_1 = digitalRead(buttonPin_1);
if (buttonState_1 == HIGH) {
  //lectura del puerto serial y envio de peticion de datos
  ICSC.send(2, 'P', 5, "1");

  ICSC.process();//lectura del puerto
  ICSC.send(3, 'P', 5, "1");
  //delay (5);
  ICSC.process();
  ICSC.send(4, 'P', 5, "1");
  //delay (5);
  ICSC.process();
}

```

Figura 2.52: Petición y recepción de ángulos

2.3 CREACIÓN DEL ARCHIVO .MOT

2.3.1 ARCHIVO.MOT

El archivo .mot es un archivo de texto delimitado por tabulaciones utilizado para modelamiento biomecánico en *OpenSim*, el mismo que cuenta con dos partes: el encabezamiento de movimiento y datos. Dentro del encabezamiento se tiene *nRows* que indica el número de datos por fila y *nColumns* que indica el número de datos por columna. Bajo el encabezado se tiene las columnas con los datos del tiempo y los ángulos de las diferentes articulaciones del modelo biomecánico.

	A	B	C	D	E	F	G	H	I	J	K
1	Coordinates										
2	version=1										
3	nRows=73										
4	nColumns=24										
5	inDegrees=yes										
6											
7	Units are S.I. units (second, meters, Newtons, ...)										
8	Angles are in degrees.										
9											
10	endheader										
11	time	pelvis_tilt	pelvis_list	pelvis_rotati	pelvis_tx	pelvis_ty	pelvis_tz	hip_flexion	hip_adducti	hip_rotation	knee_angle_a
12	0.4	4.8445593	-1.845593	-0.777721	0.6010203	1.0455988	-0.0216	20.018102	-2.906174	-6.950374	-54.91982
13	0.4166667	4.8288886	-1.775946	-0.637785	0.598305	1.0446577	-0.022922	20.476377	-3.027076	-7.401061	-50.00709
14	0.4333333	4.7308114	-1.479428	-0.691095	0.5960126	1.0430188	-0.023041	20.791107	-3.319371	-7.940413	-44.62673
15	0.45	4.8471411	-1.051324	-0.574416	0.5938475	1.0412447	-0.023252	20.590681	-3.745638	-8.738236	-38.76975
16	0.4666667	4.7028551	-0.36595	-0.525556	0.5913281	1.0386044	-0.023295	20.548927	-4.294762	-9.411697	-32.63493
17	0.4833333	4.6418057	-0.089397	-0.081969	0.5900191	1.0361158	-0.022982	19.886777	-4.3296	-10.40548	-25.85626
18	0.5	4.5193727	0.3497198	-0.073875	0.5890374	1.0329995	-0.02159	19.293404	-4.218716	-10.44641	-19.14777
19	0.5166667	4.389784	0.9247047	0.0362953	0.587765	1.0293696	-0.020823	18.745441	-4.282707	-9.865932	-12.83865
20	0.5333333	4.2569614	1.5169438	0.3612415	0.5867882	1.0253962	-0.020332	18.193444	-4.512723	-8.938594	-7.211551
21	0.55	4.1844582	2.1238654	0.276914	0.5863482	1.0213408	-0.019443	17.723726	-4.775907	-7.367456	-2.568068
22	0.5666667	4.1235382	2.5967151	0.2752686	0.5855002	1.0173968	-0.018731	17.631289	-5.070871	-5.905749	0.3882599
23	0.5833333	4.0000933	2.938857	0.9188036	0.5839692	1.0139728	-0.018546	17.748337	-5.465468	-5.496023	1.8962797
24	0.6	3.9996795	2.9892354	0.96881	0.5846468	1.0113169	-0.0167	17.970798	-5.339973	-4.969961	1.8180376
25	0.6166667	3.9102682	2.7671634	1.2284871	0.5852806	1.0091003	-0.01431	18.608929	-4.969875	-5.002568	0.1219256
26	0.6333333	3.8734401	2.7611927	1.7451344	0.5862714	1.0077172	-0.011912	19.295274	-4.91415	-5.363499	-2.570598
27	0.65	3.7527008	2.6503255	2.2062671	0.5877985	1.0066409	-0.008922	20.08814	-4.747597	-5.805723	-5.740391

Figura 2.53: Archivo .mot

Para la creación del archivo se utiliza una hoja de Excel, en la cual se carga los datos de los archivos de texto generados por el sistema embebido.

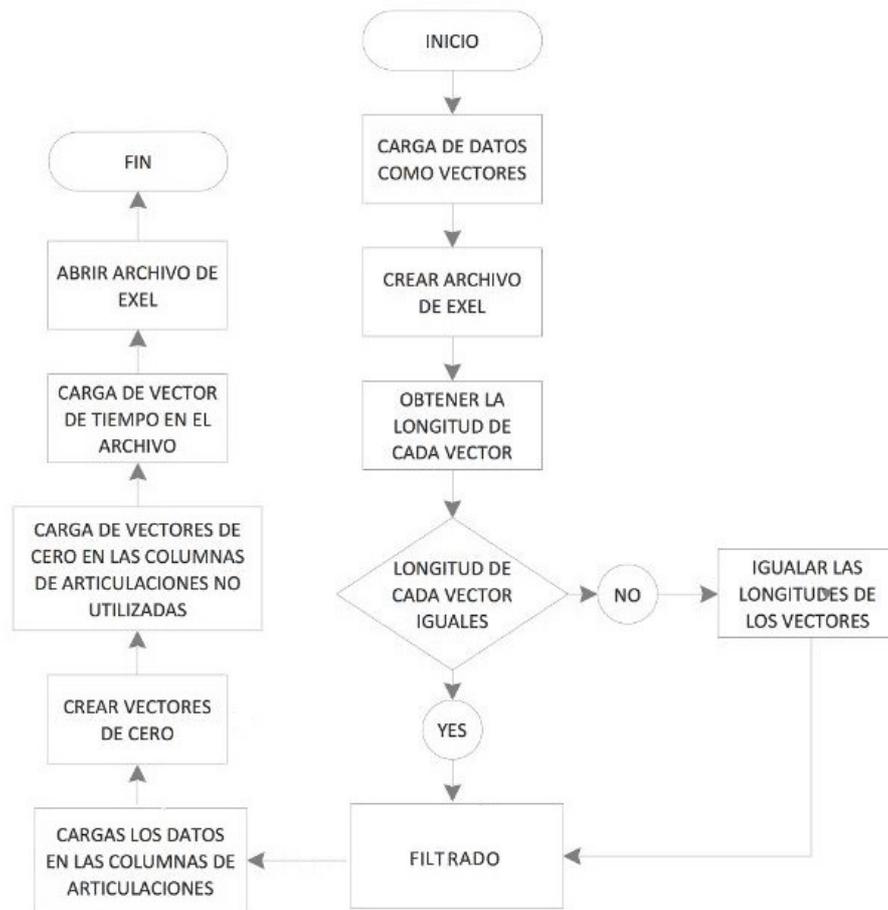


Figura 2.54: Diagrama de flujo para la creación del archivo .mot

Lo primero que se hizo fue crear una interfaz visual que permite al usuario cargar los archivos de texto desde una ruta específica y eliminarlos en caso de no haber elegido los correctos, cada archivo contiene seis columnas que corresponde a cada eje de dos sensores los cuales son cargados en el programa como vectores.



Figura 2.55: Interfaz visual para la creación del archivo .mot

La figura 2.56 muestra cómo cargar los tres archivos de texto generados por los integrados ATMEGA328P-PU y tomar cada columna de datos en forma de vectores, esto con el propósito de poder ordenarlos adecuadamente en las distintas columnas del archivo de Excel.

```
[nombre, ruta] = uigetfile('*.txt', 'Seleccione un archivo de
texto','F:');
dir=strcat(ruta, nombre);% concatena la direccion del txt
set(handles.text3,'String',dir);%muestra la direccion en el textos
estatico

if nombre==0
    return;

else
    m= load([ruta nombre]);%carga los datos desde el archivo de
texto
    z1=m(:,1)%carga la primera columna del primer archivo como
vector
    y1=m(:,2)%carga la segunda columna del primer archivo como
vector
    x1=m(:,3)%carga la tercera columna del primer archivo como
vector
    z2=m(:,4)%carga la cuarta columna del primer archivo como vector
    y2=m(:,5)%carga la quinta columna del primer archivo como vector
    x2=m(:,6)%carga la sexta columna del primer archivo como vector
end

[nombre1, ruta1] = uigetfile('*.txt', 'Seleccione un archivo de
texto','F:');
dir=strcat(ruta1, nombre1);%muestra la direccion del txt
set(handles.text6,'String',dir);

if nombre1==0
```

```

        return;

else
    m1= load([ruta1 nombre1]);
end
    z3=m1(:,1)%carga la primera columna del segundo archivo como
vector
    y3=m1(:,2)%carga la segunda columna del segundo archivo como
vector
    x3=m1(:,3)%carga la tercera columna del segundo archivo como
vector
    z4=m1(:,4)%carga la cuarta columna del segundo archivo como
vector
    y4=m1(:,5)%carga la quinta columna del segundo archivo como
vector
    x4=m1(:,6)%carga la sexta columna del segundo archivo como
vector

[nombre2, ruta2] = uigetfile('*.txt', 'Seleccione un archivo de
texto','F:');
dir=strcat(ruta2, nombre2);%muestra la direccion del txt
set(handles.text4,'String',dir);

if nombre2==0
    return;

else
    m2= load([ruta2 nombre2]);
end
    z5=m2(:,1)%carga la primera columna del tercer archivo como
vector
    y5=m2(:,2)%carga la segunda columna del tercer archivo como
vector
    x5=m2(:,3)%carga la tercera columna del tercer archivo como
vector
    z6=m2(:,4)%carga la cuarta columna del tercer archivo como
vector
    y6=m2(:,5)%carga la quinta columna del tercer archivo como
vector
    x6=m2(:,6)%carga la sexta columna del tercer archivo como
vector

```

Figura 2.56: Exportación de los datos desde el archivo de texto a matlab

Una vez cargados los datos como vectores se procede a sacar la longitud de cada uno de ellos y colocarlos dentro de otro vector, y mediante un comando encontramos la longitud más pequeña, considerando que en ciertos casos las longitudes de cada vector pueden ser distintas lo cual ocasionaría un error al momento de cargarlos en *OpenSim*. Para ello lo que se hace es una comparación de las longitudes de cada vector, con el valor de la longitud mínima, en caso de ser mayor se elimina los datos a partir de la longitud mínima más uno, con esto nos garantizamos de que todas las longitudes sean las mismas.

```

function pushcCREAR_Callback(hObject, eventdata, handles)

load('filtro6.mat')
global x1 y2 x2 x3 y3 z3 x4 y4 z4 x5 y6 x6

    archivo='pruebas.xls'; %carga el archivo xlsx
    hoja='Hoja1';% carga la hoja en la q se esta trabajando

lx1=length(x1);%longitud del vector z1
ly2=length(y2);
lx2=length(x2);
lx3=length(x3);
ly3=length(y3);
lz3=length(z3);
lx4=length(x4);
ly4=length(y4);
lz4=length(z4);
lx5=length(x5);
ly6=length(y6);
lx6=length(x6);

%vector formado por las longitudes de todos los vectores
vl=[lx1; ly2; lx2; lx3; ly3; lz3; lx4; ly4; lz4; lx5; ly6; lx6];
lm=min(vl)% elemento menor del vector

```

Figura 2.57: Longitud de cada vector y longitud mínima

```

lmt=lm+1;
if lx1>lm
    x1(lmt:end) = []
end

if ly2>lm
    y2(lmt:end) = []
end

if lx2>lm
    x2(lmt:end) = []
end

if lx3>lm
    x3(lmt:end) = []
end

if ly3>lm
    y3(lmt:end) = []
end

if lz3>lm
    z3(lmt:end) = []
end

if lx4>lm
    x4(lmt:end) = []
end

```

```

if ly4>lm
    y4(lmt:end) = []
end

if lz4>lm
    z4(lmt:end) = []
end

if lx5>lm
    x5(lmt:end) = []
end

if ly6>lm
    y6(lmt:end) = []
end

if lx6>lm
    x6(lmt:end) = []
end

end

```

Figura 2.58: Vectores de igual longitud

El siguiente paso es llenar el archivo con el encabezado, para este proceso se utilizó el código que se muestra en la figura 2.59.

```

%Encabezado del archivo en las diferentes celdas
a1=xlswrite(archivo,{'Coordinates'},hoja,'a1');
a2={'version=',1};
[texto2,numero2]=xlswrite(archivo,a2,hoja,'a2');
a3={'nRows=',lm};
[texto3,numero3]=xlswrite(archivo,a3,hoja,'a3');
a4={'nColumns=',24};
[texto4,numero4]=xlswrite(archivo,a4,hoja,'a4');
a5=xlswrite(archivo,{'inDegrees=yes'},hoja,'a5');
a7=xlswrite(archivo,{'Units are S.I. units (second, meters, Newtons,
...)'},hoja,'a7');
a8=xlswrite(archivo,{'Angles are in degrees.'},hoja,'a8');
a10=xlswrite(archivo,{'endheader'},hoja,'a10');
a11=xlswrite(archivo,{'time'},hoja,'a11');

b11=xlswrite(archivo,{'pelvis_tilt'},hoja,'b11');
c11=xlswrite(archivo,{'pelvis_list'},hoja,'c11');
d11=xlswrite(archivo,{'pelvis_rotation'},hoja,'d11');
e11=xlswrite(archivo,{'pelvis_tx'},hoja,'e11');
f11=xlswrite(archivo,{'pelvis_ty'},hoja,'f11');
g11=xlswrite(archivo,{'pelvis_tz'},hoja,'g11');
h11=xlswrite(archivo,{'hip_flexion_r'},hoja,'h11');
i11=xlswrite(archivo,{'hip_adduction_r'},hoja,'i11');
j11=xlswrite(archivo,{'hip_rotation_r'},hoja,'j11');
k11=xlswrite(archivo,{'knee_angle_r'},hoja,'k11');
l11=xlswrite(archivo,{'ankle_angle_r'},hoja,'l11');
m11=xlswrite(archivo,{'subtalar_angle_r'},hoja,'m11');
n11=xlswrite(archivo,{'mtp_angle_r'},hoja,'n11');
o11=xlswrite(archivo,{'hip_flexion_l'},hoja,'o11');
p11=xlswrite(archivo,{'hip_adduction_l'},hoja,'p11');
q11=xlswrite(archivo,{'hip_rotation_l'},hoja,'q11');
r11=xlswrite(archivo,{'knee_angle_l'},hoja,'r11');
s11=xlswrite(archivo,{'ankle_angle_l'},hoja,'s11');

```

```

t11=xlswrite(archivo,{'subtalar_angle_1'},hoja,'t11');
u11=xlswrite(archivo,{'mtp_angle_1'},hoja,'u11');
v11=xlswrite(archivo,{'lumbar_extension'},hoja,'v11');
w11=xlswrite(archivo,{'lumbar_bending'},hoja,'w11');
x11=xlswrite(archivo,{'lumbar_rotation'},hoja,'x11');

```

Figura 2.59: Código para el encabezado.

El archivo .mot contiene los datos de todas las articulaciones del cuerpo humano nuestro sistema únicamente hace la captura de los datos correspondientes a los miembros inferiores ante ello es necesario darles un valor de cero al resto de articulaciones, la figura 2.60 se observa como es creado este vector de ceros y cargado en nuestro archivo de Excel.

```

c1=zeros(lm,1);%crea un vector de ceros
%Escribe el vector de ceros en las diferentes celdas
xlswrite(archivo,c1,hoja,'m12');
xlswrite(archivo,c1,hoja,'t12');
xlswrite(archivo,c1,hoja,'b12');
xlswrite(archivo,c1,hoja,'c12');
xlswrite(archivo,c1,hoja,'d12');
xlswrite(archivo,c1,hoja,'e12');
xlswrite(archivo,c1,hoja,'f12');
xlswrite(archivo,c1,hoja,'g12');
xlswrite(archivo,c1,hoja,'n12');
xlswrite(archivo,c1,hoja,'u12');
xlswrite(archivo,c1,hoja,'v12');
xlswrite(archivo,c1,hoja,'w12');
xlswrite(archivo,c1,hoja,'x12');

```

Figura 2.60: Vector de ceros

El siguiente paso es determinar el sentido del ángulo de giro de ciertos ejes debido la ubicación de los giroscopios en el traje.

```

%rodilla-tobillo-p derecha
xx1=x1
yy2=-y2
xx2=-x2
%cadera derecha
xx3=-x3
yy3=-y3
zz3=-z3
%cadera izquierda
xx4=x4
yy4=-y4
zz4=z4 %x
%rodilla tobillo-p izquierda
xx5=-x5
yy6=-y6
xx6=x6

```

Figura 2.61: Orientación de ejes

Es necesario filtrar los datos recibidos con el propósito de que la señal sea más estable para ello se utilizó el *fdatool* una herramienta de *matlab* para la creación de filtros digitales.

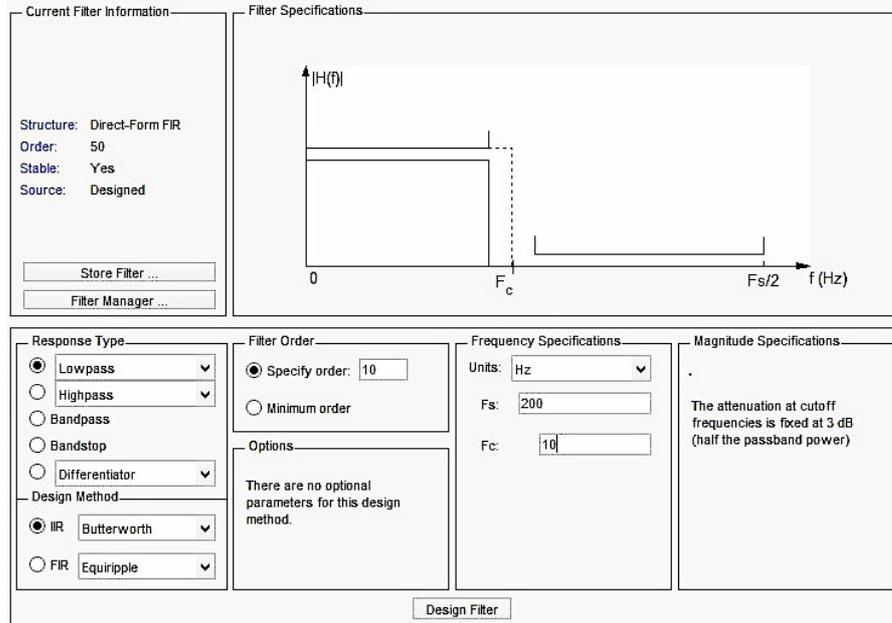


Figura 2.62: Creación del filtro IIR Butterworth Pasabajo mediante *fdatool*

Utilizamos un filtro IIR Butterworth pasa bajo de 10 polos, frecuencia de muestreo de 200 Hz y la frecuencia de corte de 10 Hz, esta última frecuencia se la utiliza debido a que ningún movimiento humano supera este valor.

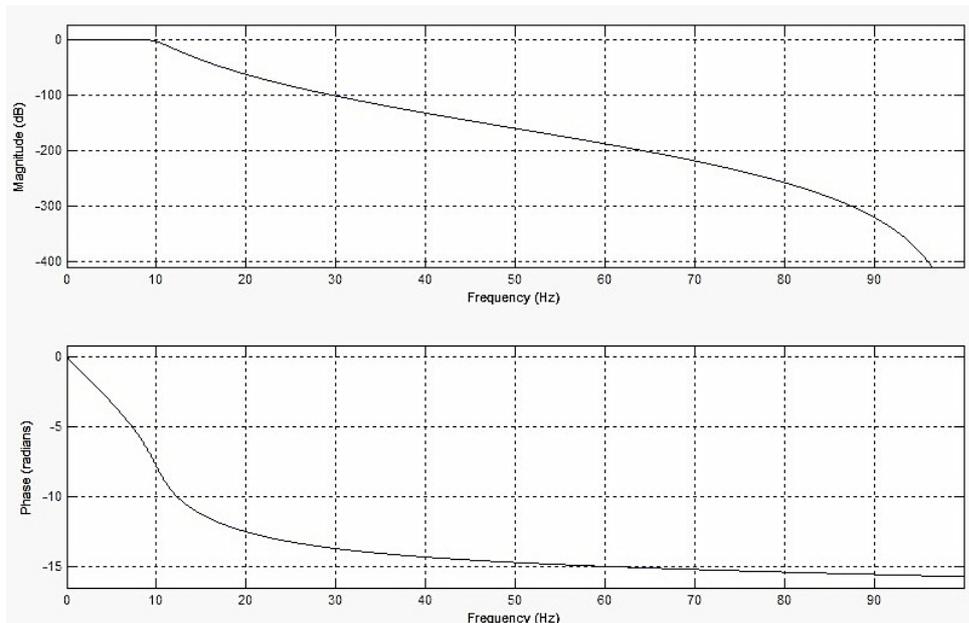


Figura 2.63: Respuesta en frecuencia del filtro Butterworth diseñado

En la siguiente gráfica se muestra el código para cargar el filtro, posteriormente se muestra como se aplica el filtro a cada uno los vectores que contienen los datos de los ángulos y como estos datos son cargados en las diferentes columnas correspondientes a las articulaciones utilizadas para el análisis.

```
load('filtro6.mat')
fxx1=filter(filtro6,xx1)
fyy2=filter(filtro6,yy2)
fxx2=filter(filtro6,xx2)
fxx3=filter(filtro6,xx3)
fyy3=filter(filtro6,yy3)
fzz3=filter(filtro6,zz3)
fxx4=filter(filtro6,xx4)
fyy4=filter(filtro6,yy4)
fzz4=filter(filtro6,zz4)
fxx5=filter(filtro6,xx5)
fyy6=filter(filtro6,yy6)
fxx6=filter(filtro6,xx6)

%Escribe los datos de los vectores en las diferentes celdas
xlswrite(archivo,fxx1,hoja,'k12');
xlswrite(archivo,fyy2,hoja,'l12');
xlswrite(archivo,fxx3,hoja,'h12');
xlswrite(archivo,fyy3,hoja,'i12');
xlswrite(archivo,fxx2,hoja,'j12');
xlswrite(archivo,fxx4,hoja,'o12');
xlswrite(archivo,fyy4,hoja,'p12');
xlswrite(archivo,fxx6,hoja,'q12');
xlswrite(archivo,fxx5,hoja,'r12');
xlswrite(archivo,fyy6,hoja,'s12');
```

Figura 2.64: Datos válidos para la recreación del movimiento

Una vez creado el archivo de Excel con todos los datos necesarios, se guarda como un archivo de texto delimitado por tabulaciones y cambiamos su extensión, con esto el archivo está listo para ser cargado y simulado en *OpenSim*.

2.4 RECREACIÓN DEL MOVIMIENTO

Para la recreación del movimiento utilizamos *OpenSim*, software libre dedicado al modelado biomecánico del cuerpo humano de trayectorias y fuerzas.

Para cargar los modelos biomecánicos tenemos que dirigirnos a la ruta “*File/open/ model*” posteriormente a la librería de *OpenSim* ubicada en “*Equipo/C/OpenSim 3.0/Models*” y cargar los archivos .osim, en los modelos podemos variar las características de peso y tamaño con el propósito de que al

momento de hacer un análisis sobre cierta persona el avatar cuenta con las características propias del individuo.

En la figura 2.65 se puede apreciar en la parte derecha el modelo biomecánico, mientras que en la parte izquierda se encuentra la ventana de coordenadas con el rango de variación de las distintas articulaciones que conforman el cuerpo humano.

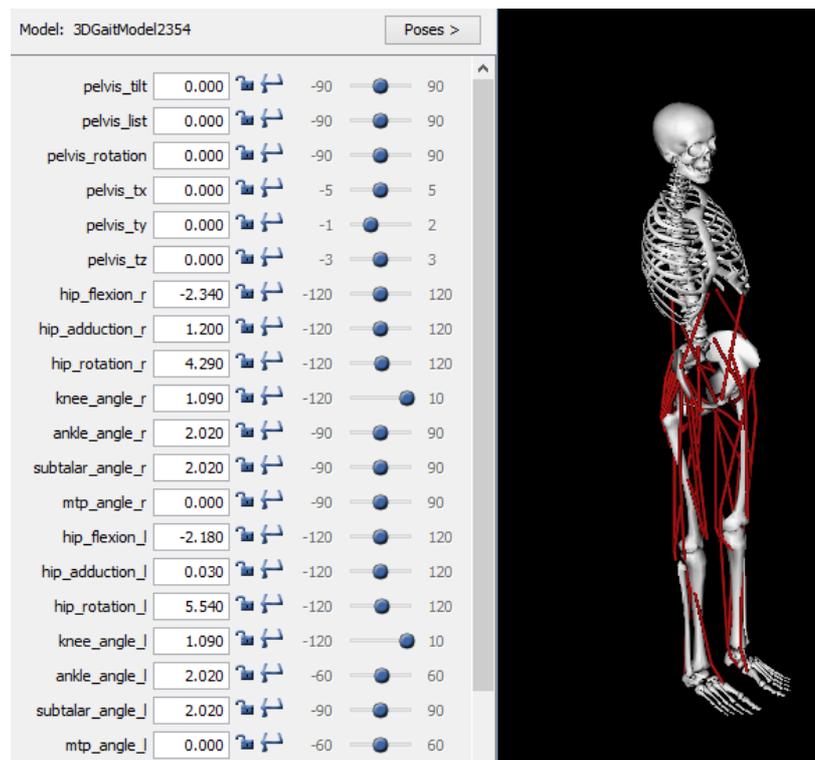


Figura 2.65: Modelo biomecánico y coordenadas de OpenSim

2.4.1 CARGAR LA TRAYECTORIA

Para representar el movimiento de las personas debemos cargar los archivos .mot para ello nos dirigimos a la siguiente ruta: “File/Load Motion” y seleccionamos el archivo requerido para la simulación.

En la ruta “Tools/Plot” podemos obtener las curvas de cada una de las articulaciones con respecto al tiempo o a su vez con respecto a otra articulación.

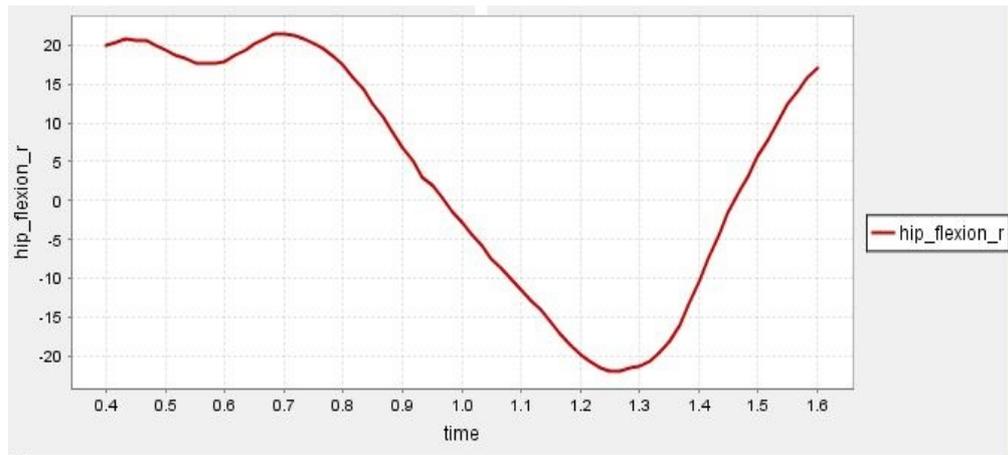


Figura 2.66: Grafica Obtenida en OpenSim

CAPÍTULO 3.

3. PRUEBAS Y RESULTADOS

En este capítulo tratará de las pruebas de funcionamiento que se realizaron al sistema para la comprobación de su funcionamiento así como el análisis de los resultados obtenidos.

3.1 PRUEBAS DE FUNCIONAMIENTO

3.1.1 MUESTRAS DE ESTABILIZACIÓN INICIAL

Para determinar la cantidad de muestras que necesita el giroscopio para que llegue a estabilizarse se tomó un solo integrado ATMEGA 328P-PU con el sketch del esclavo, conectado a dos giroscopios, los datos fueron enviados al puerto serial de un ordenador para ser almacenados y finalmente ser graficados en Matlab. En la siguiente figura podemos apreciar como el ángulo de navegación roll comienza a sumarse hasta estabilizarse en la muestra 1411

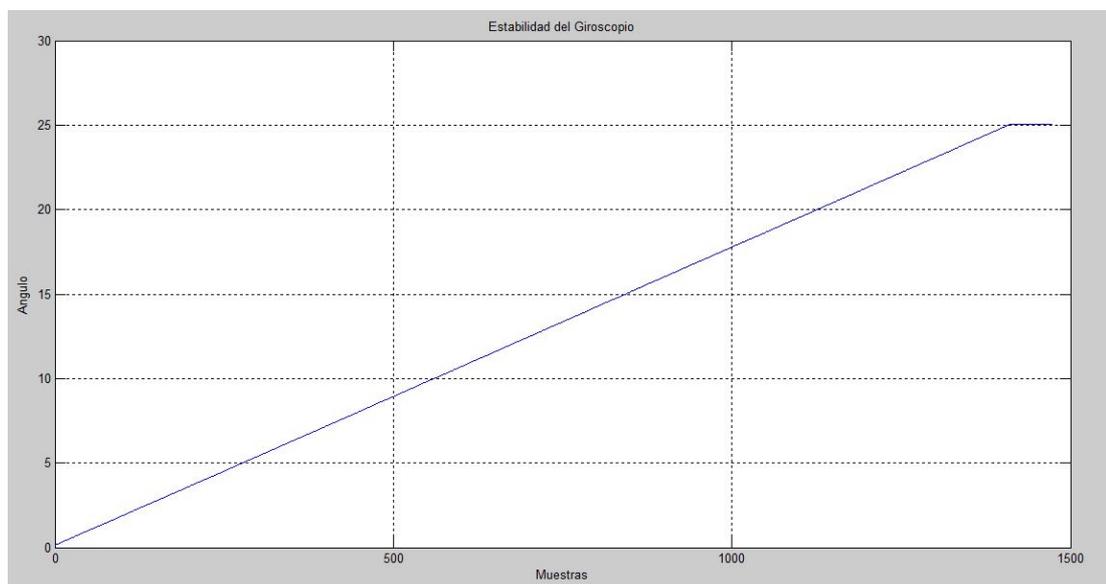


Figura 3.1: Prueba de estabilización

La siguiente tabla se creó en base a las pruebas de estabilización donde se denota la muestra a la cual el giroscopio se estabilizó.

Número de pruebas	Muestra
Prueba #1	1413
Prueba #2	1409
Prueba #3	1412
Prueba #4	1417
Prueba #5	1413
Prueba #6	1410

Tabla 3.1: Muestras de estabilización

Para este caso cada muestra es tomada cada 10ms debido a que se ha omitido la comunicación entre integrados y la comunicación SPI de la grabación del maestro. Dando como una media un valor de 1412.333333 muestras equivalente a 14.12 segundos.

3.1.2 PRUEBA DE ESTABILIDAD INERCIAL

En esta prueba se tomó un solo integrado ATMEGA 328P-PU cargado el sketch del esclavo y conectado a el dos giroscopios, los datos fueron enviados al puerto serial de un ordenador para ser almacenados y luego ser graficados en Matlab. Durante esta prueba los giroscopios se mantuvieron en estado de inercia. En la siguiente gráfica se observa las muestras tomadas y que están representadas por la línea azul mientras que la línea verde denota la misma señal pasada por un filtro pasa bajo butterworth con frecuencia de corte 10Hz.

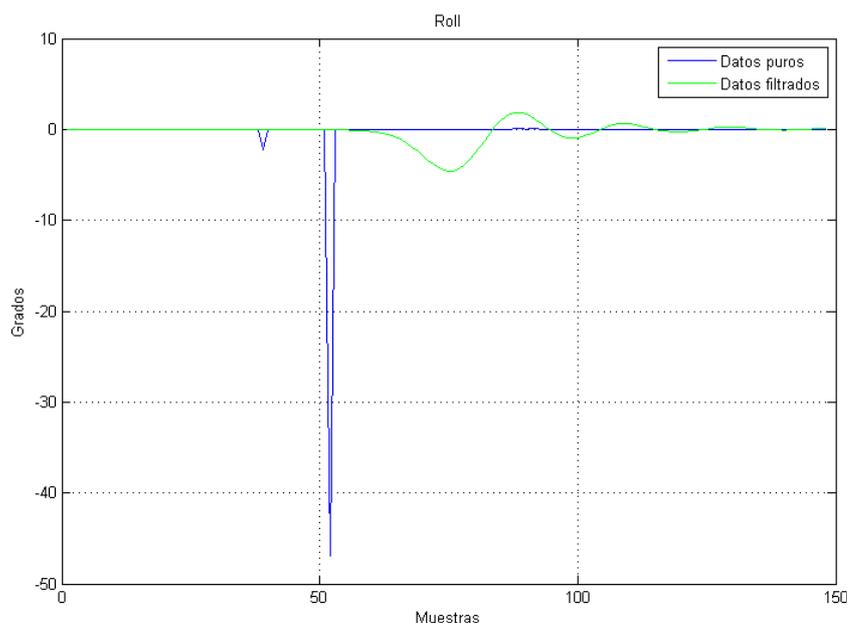


Figura 3.2: Prueba de estabilidad inercial

En el anexo 1 se observa las muestras tomadas, el ángulo denotado en ese instante según el algoritmo, el ángulo real y los respectivos errores.

El error fue calculado mediante la siguiente ecuación matemática.

$$Error = Valor Real - Valor Medido \quad (7)[33]$$

Recordemos que el giroscopio está en estado de reposo por lo tanto el valor real de los ángulos es de cero.

Con los datos del anexo 1 obtenemos una media de error de datos puros de 0.29 y media de error de datos filtrados 0.30 calculados según la ecuación (8).

3.1.3 PRUEBA DE FILTRADO

Para esta prueba se tomó un solo integrado ATMEGA 328P-PU cargado con el sketch del esclavo y conectado un giroscopio, los datos fueron enviados al puerto serial de un ordenador para ser almacenados y luego ser graficados en *matlab*. Durante esta prueba el giroscopio se mantuvo en movimiento rotacional. En la figura 3.3 la línea azul representa los datos tomados de forma pura mientras que la línea roja son las muestras puras pasadas por un filtro pasa bajo butterworth con frecuencia de corte 10Hz.

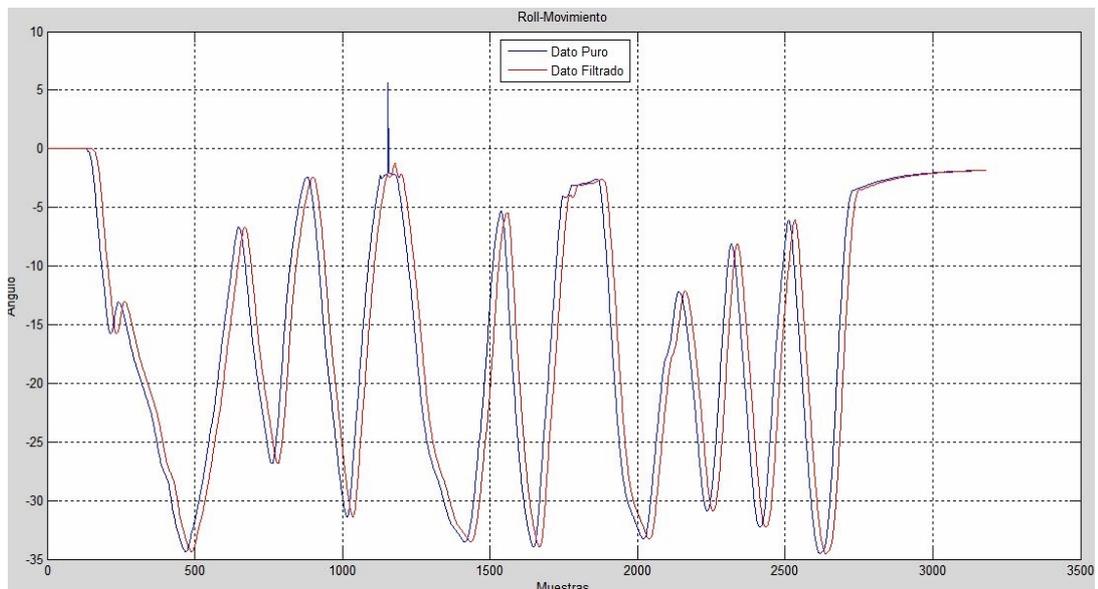


Figura3.3: Prueba de filtrado con un solo giroscopio

Para la siguiente prueba se tomó un solo integrado ATMEGA 328P-PU cargado el sketch del esclavo y conectado dos giroscopios, los datos recolectados fueron enviados al puerto serial de un ordenador para ser almacenados y luego ser graficados en *matlab*. Durante esta prueba al giroscopio se lo mantuvo en movimiento.

En la figura 3.4 la línea azul representa los datos tomados de forma pura mientras que la línea roja son las muestras puras pasadas por un filtro pasa bajo butterworth con frecuencia de corte 10Hz.

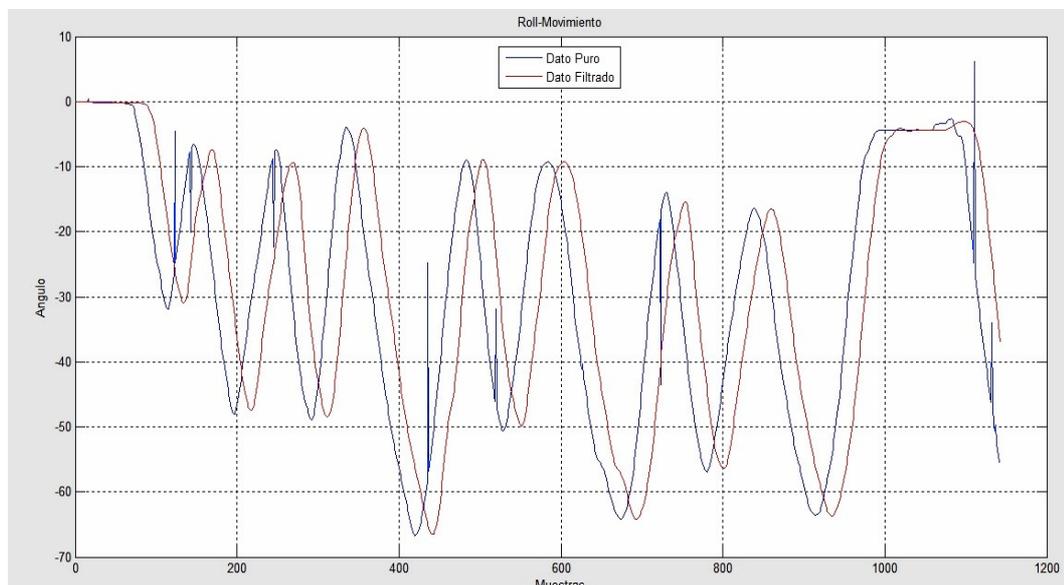


Figura 3.4: Prueba de filtrado con dos giroscopios

3.1.4 PRUEBA PARA EL CÁLCULO DEL ERROR

Para poder precisar el error del sistema procedimos a generar un experimento donde vamos a recolectar los datos calculados y reales de un solo MPU6050 puesto que el algoritmo y el sensor son los mismos el error será el mismo para los demás sensores.

Se construyó un sistema que permita medir con exactitud el ángulo mediante un instrumento patrón. En la figura 3.5 se puede observar el sistema creado para la toma de datos el cual consta de un instrumento patrón basado en una bisagra con un grado de libertad, para poder girarlo se sujeta a un extremo de la bisagra una cuerda la cual

atraviesa un punto fijo. Mientras se aplica una mayor tensión en la cuerda esta moverá la bisagra en un ángulo de rotación

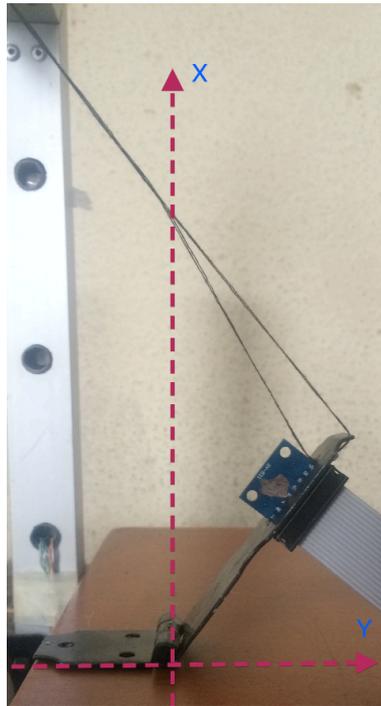


Figura 3.5: Instrumento patrón

Las pruebas se las realizó sobre los tres ejes rotando el sensor.

En la figura 3.6 se observa cómo se realizó el experimento y mediante geometría se pudo sacar el ángulo en ese instante.

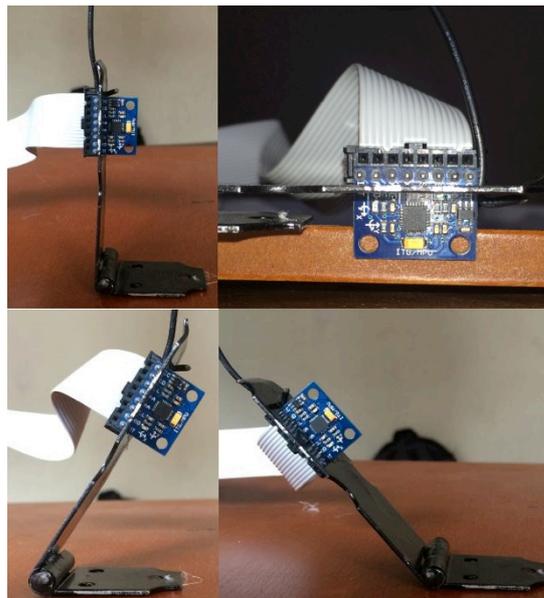


Figura 3.6: Captura de datos mediante el instrumento patrón

En los anexos 2, 3 y 4 podemos apreciar los resultados de cada uno de los experimentos realizados en los diferentes ejes junto al error, la distribución normal, la media, la distribución normal acumulada y la desviación estándar.

El cálculo del error se lo obtuvo mediante la ecuación (7). La media del error se calculó mediante la siguiente ecuación:

$$\mu = \frac{x_0 + x_1 + x_2 + \dots + x_n}{n} \quad (8)[33]$$

Donde n es el número de muestras.

La desviación estándar se las calculó mediante la siguiente ecuación.

$$\sigma = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \quad (9)[33]$$

Con los datos del error obtenidos en los anexos 2, 3,4 se pudo calcular los siguientes valores de la media y de la desviación estándar.

$$\begin{array}{ll} u_{\varphi} = 0.599 & \sigma_{\varphi} = 4.802 \\ u_{\theta} = 0.271 & \sigma_{\theta} = 0.722 \\ u_{\psi} = -3.62 & \sigma_{\psi} = 2.35 \end{array}$$

Finalmente la distribución normal se calcula a través de los errores, la media calculada y la desviación estándar mediante la siguiente ecuación

$$Z = \frac{x - \mu}{\sigma} \quad (10)[33]$$

Mediante el anexo 5 y la distribución normal se obtuvo las siguientes probabilidades:

La probabilidad de que yaw tenga un error entre ± 4 es del 59%.

La probabilidad de que pitch tenga un error entre ± 1 es del 80%.

La probabilidad de que roll tenga un error entre -4 y 0 es del 52%.

A continuación se muestra las gráficas de distribución normal, distribución normal acumulada, frecuencia de datos y ángulo para cada eje.

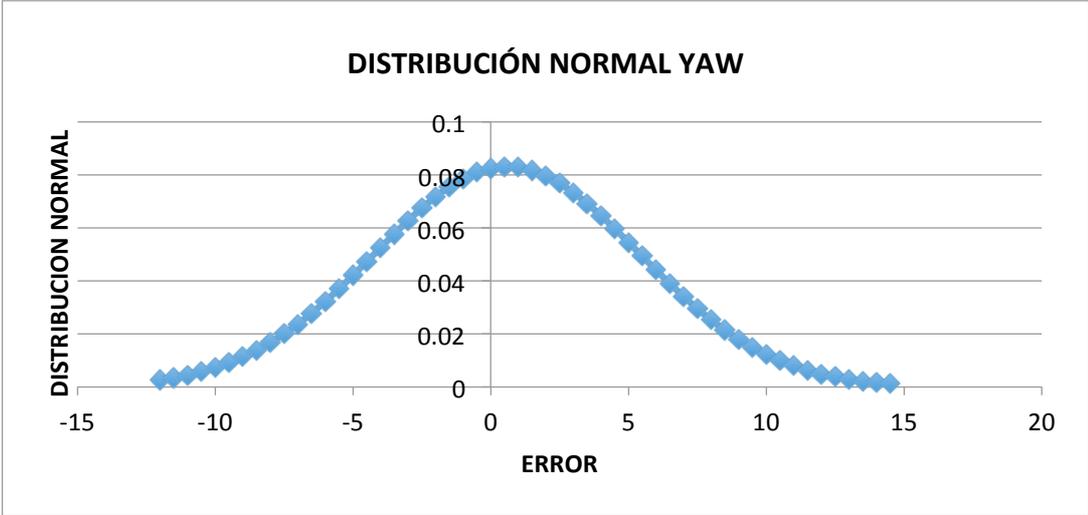


Figura 3.7: Distribución normal del ángulo yaw

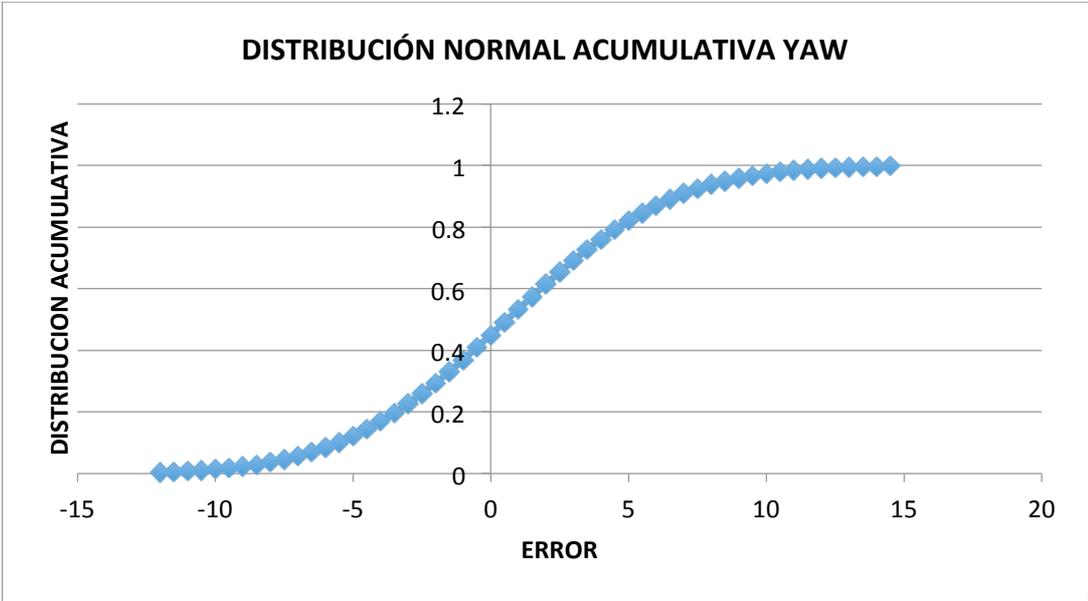


Figura 3.8: Distribución normal acumulada del ángulo yaw

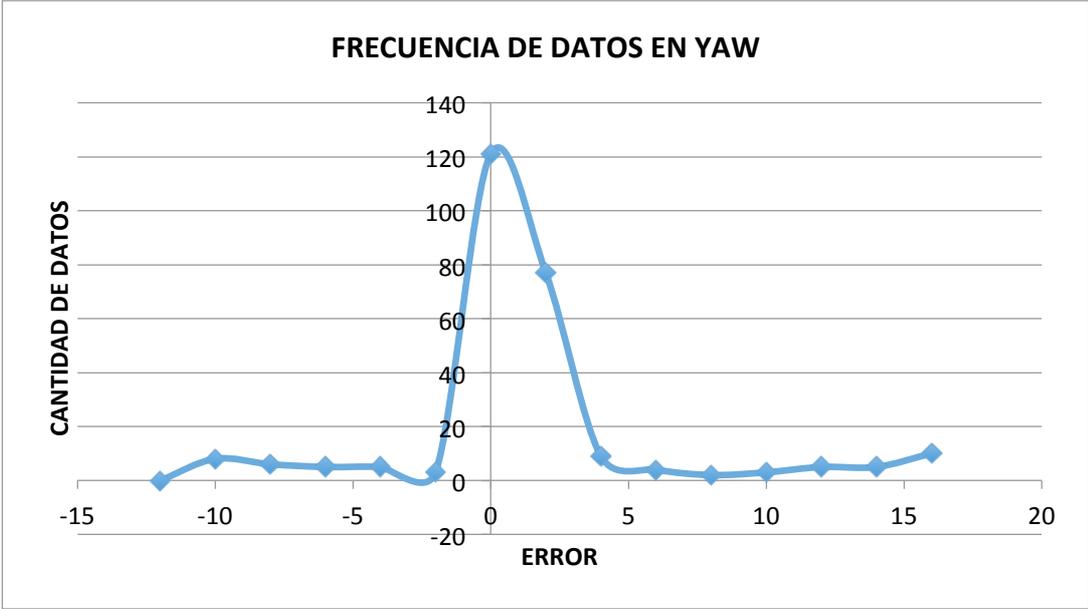


Figura 3.9: Frecuencia de datos del ángulo yaw

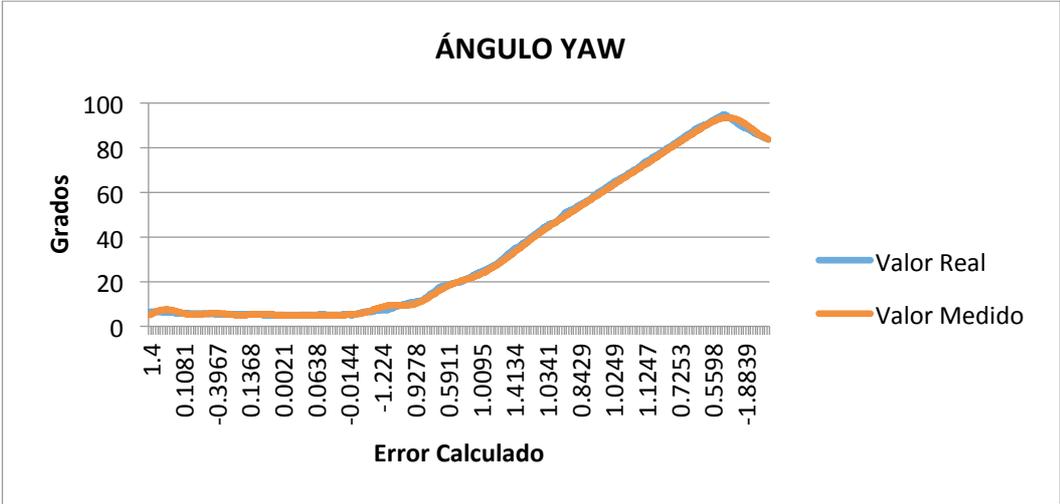


Figura 3.10: Ángulos reales y medidos del ángulo yaw

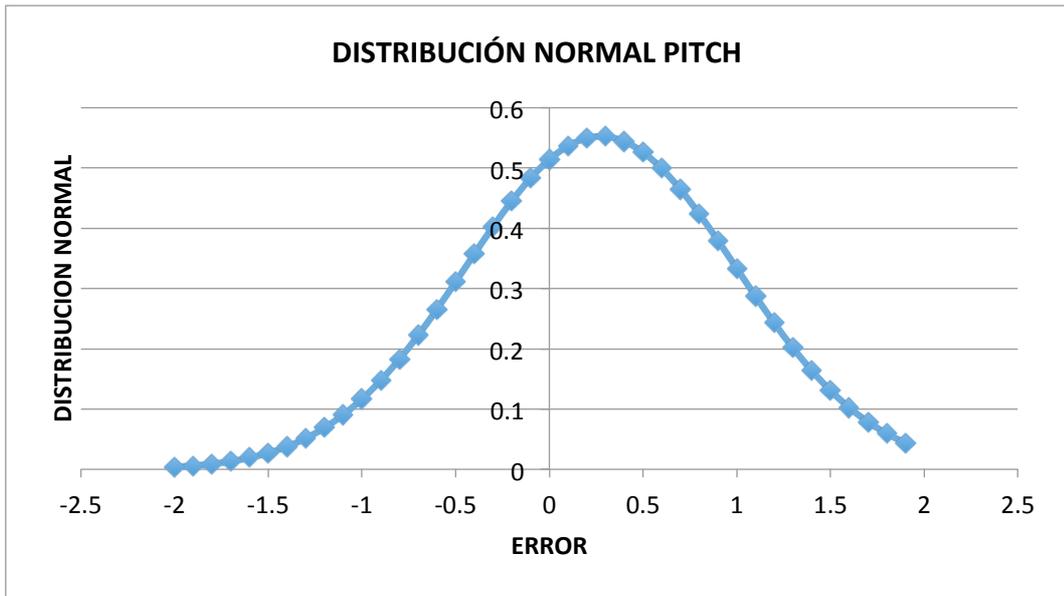


Figura 3.11: Distribución normal del ángulo pitch.

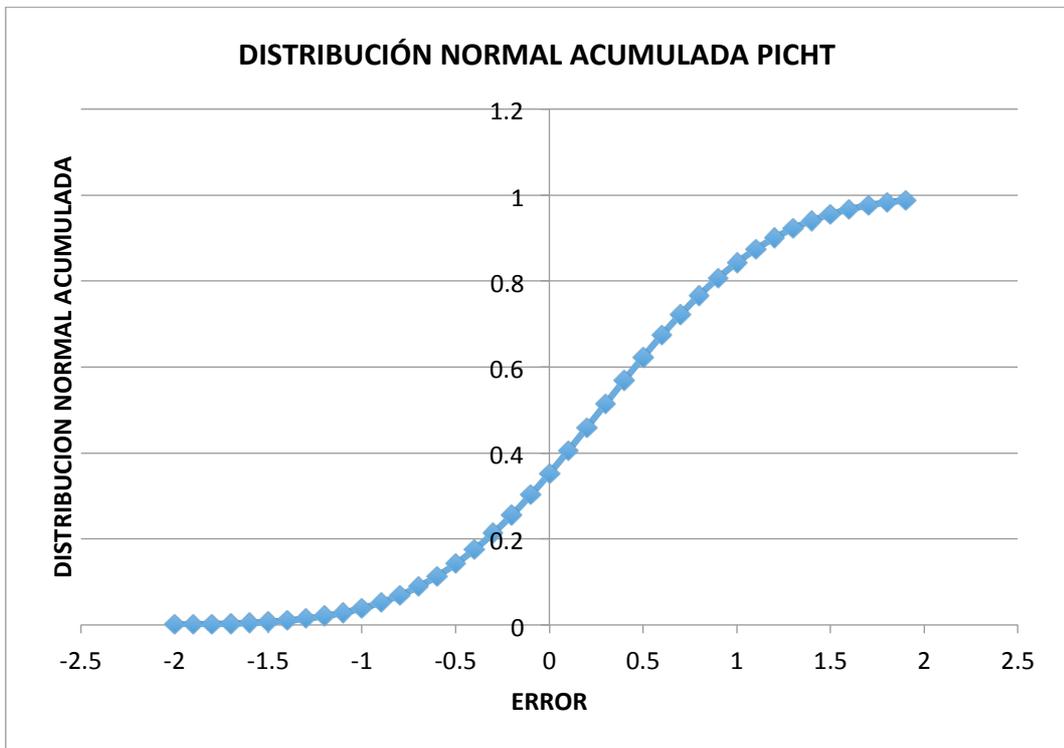


Figura 3.12: Distribución normal acumulada del ángulo pitch

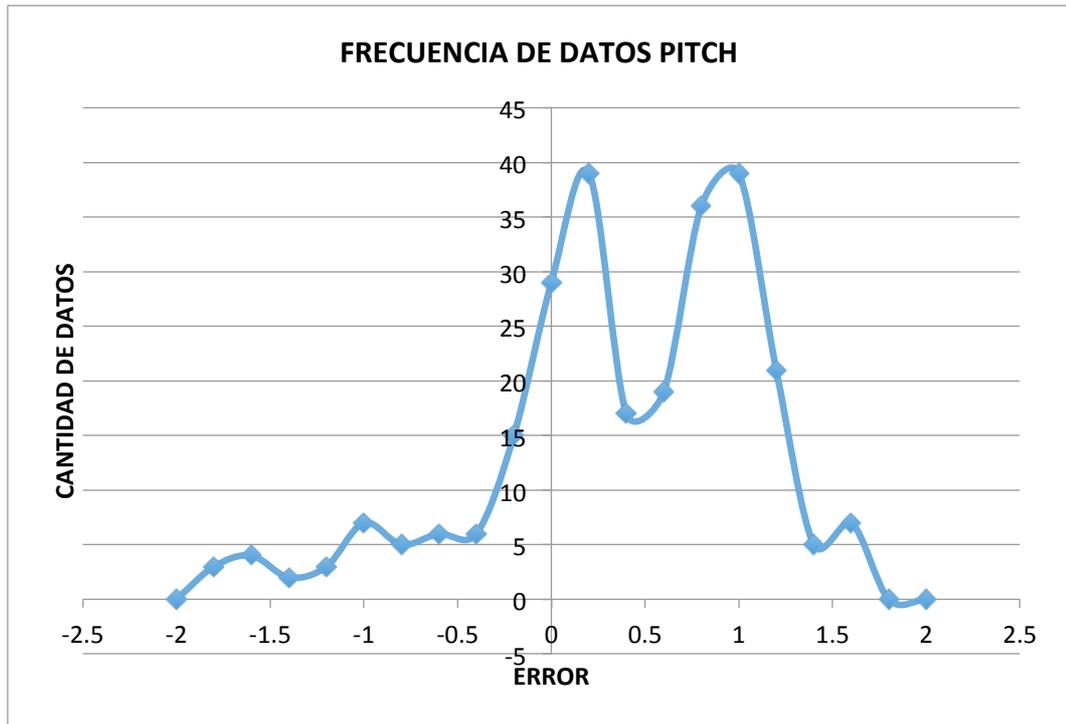


Figura 3.13: Frecuencia de datos del ángulo pitch.

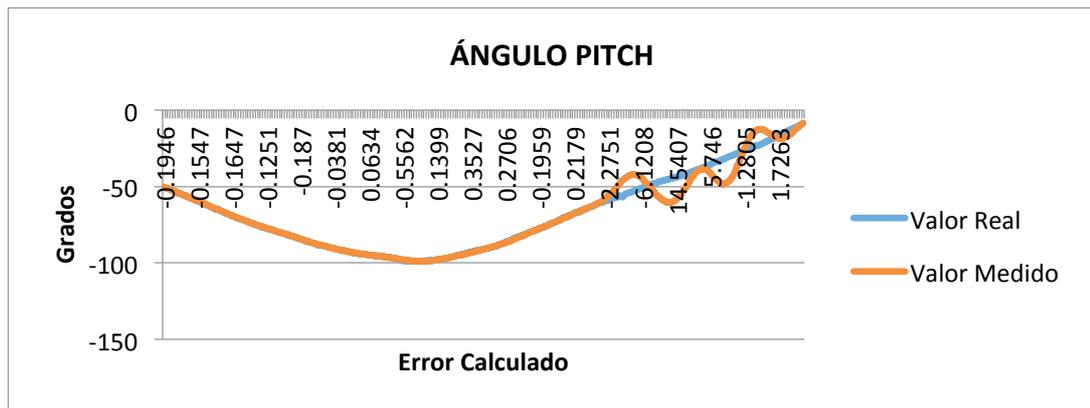


Figura 3.14: Ángulos reales y medidos del ángulo pitch

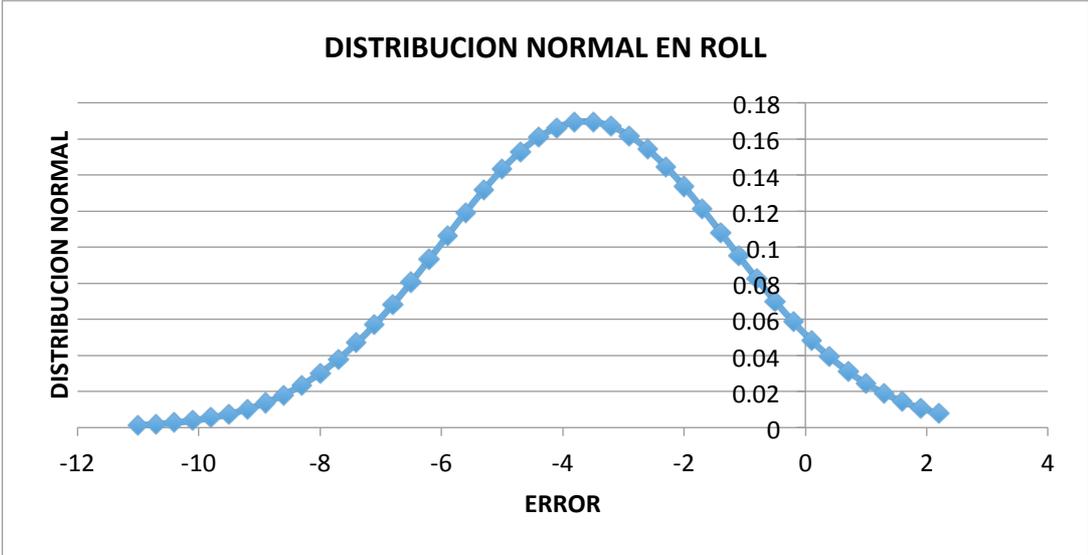


Figura 3.15: Distribución normal del ángulo roll

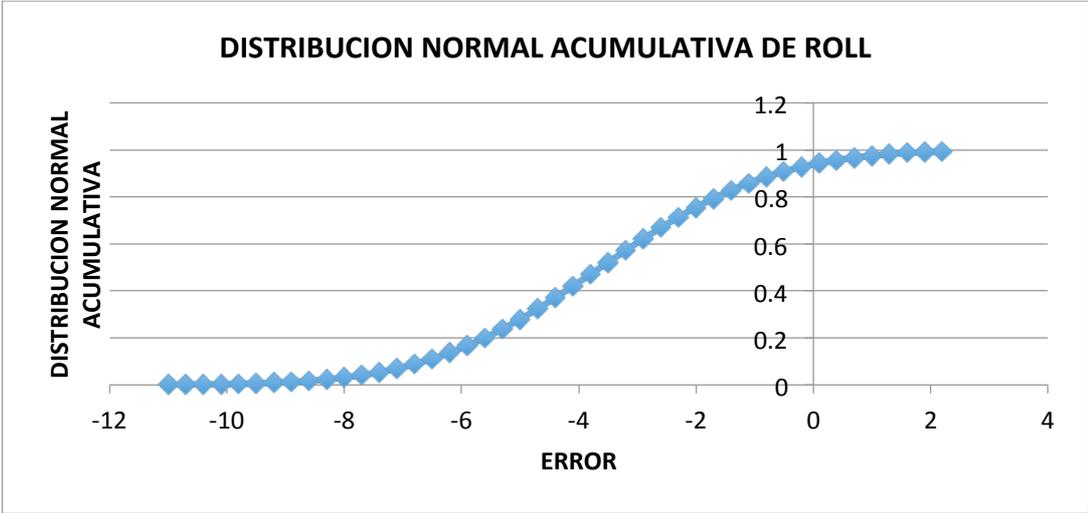


Figura 3.16: Distribución acumulada del ángulo roll.

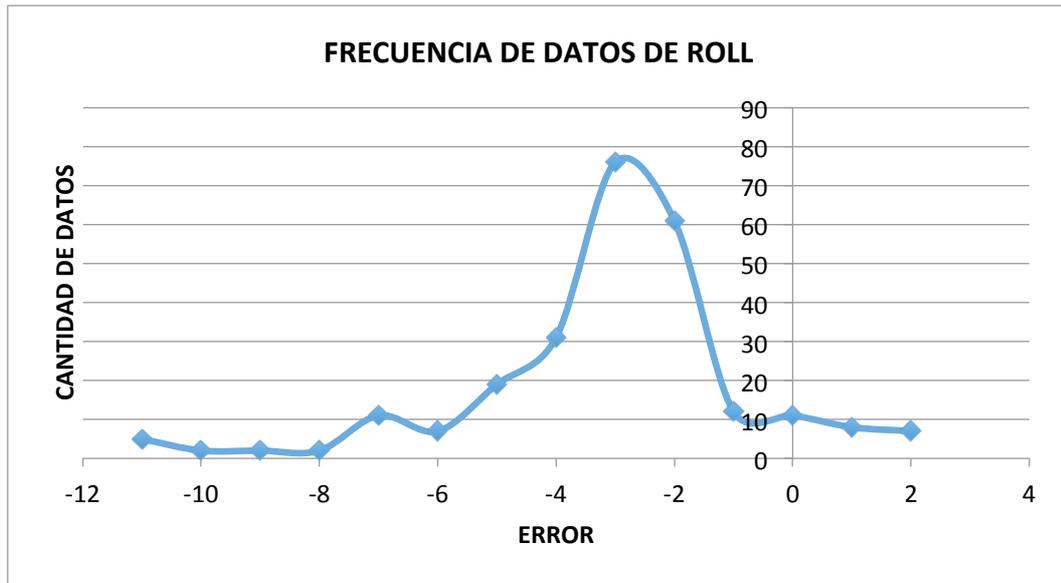


Figura 3.17: Frecuencia de datos del ángulo roll.

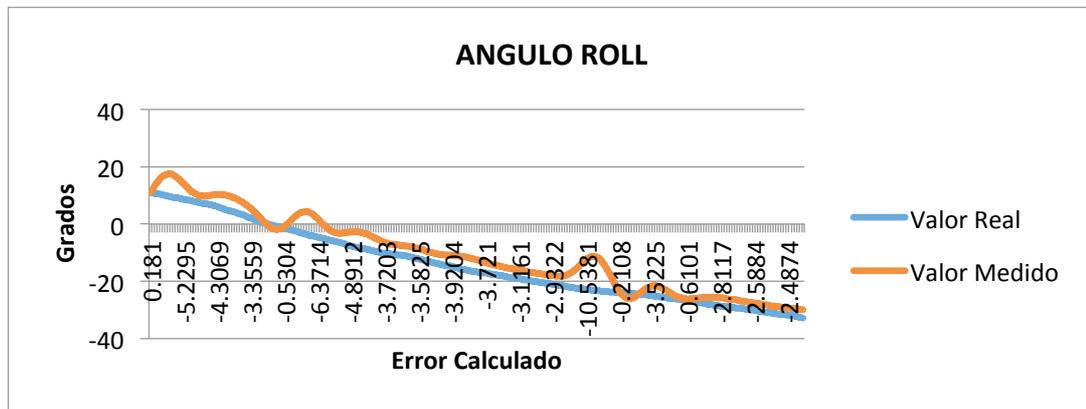


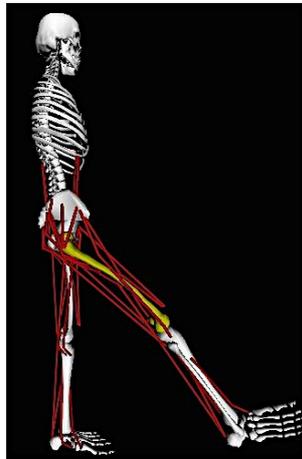
Figura 3.18: Ángulos reales y medidos del ángulo roll

3.1.5 PRUEBAS DE FUNCIONAMIENTO

Prueba 1: Movimiento de flexión en la cadera. Los ángulos obtenidos fueron de 50 grados y en su posición normal cero grados. En la figura 3.18 se muestran cómo se movió el avatar de *OpenSim*, la foto de la persona a quien se realizó la prueba y la curva que describe el movimiento.



a)



b)



c)

Figura 3.19 a) Sujeto de prueba b) Avatar de OpenSim c) Curva que describe el movimiento de flexión en la cadera

Prueba 2: Movimientos de abducción de 21 grados y aducción de 0 grados.

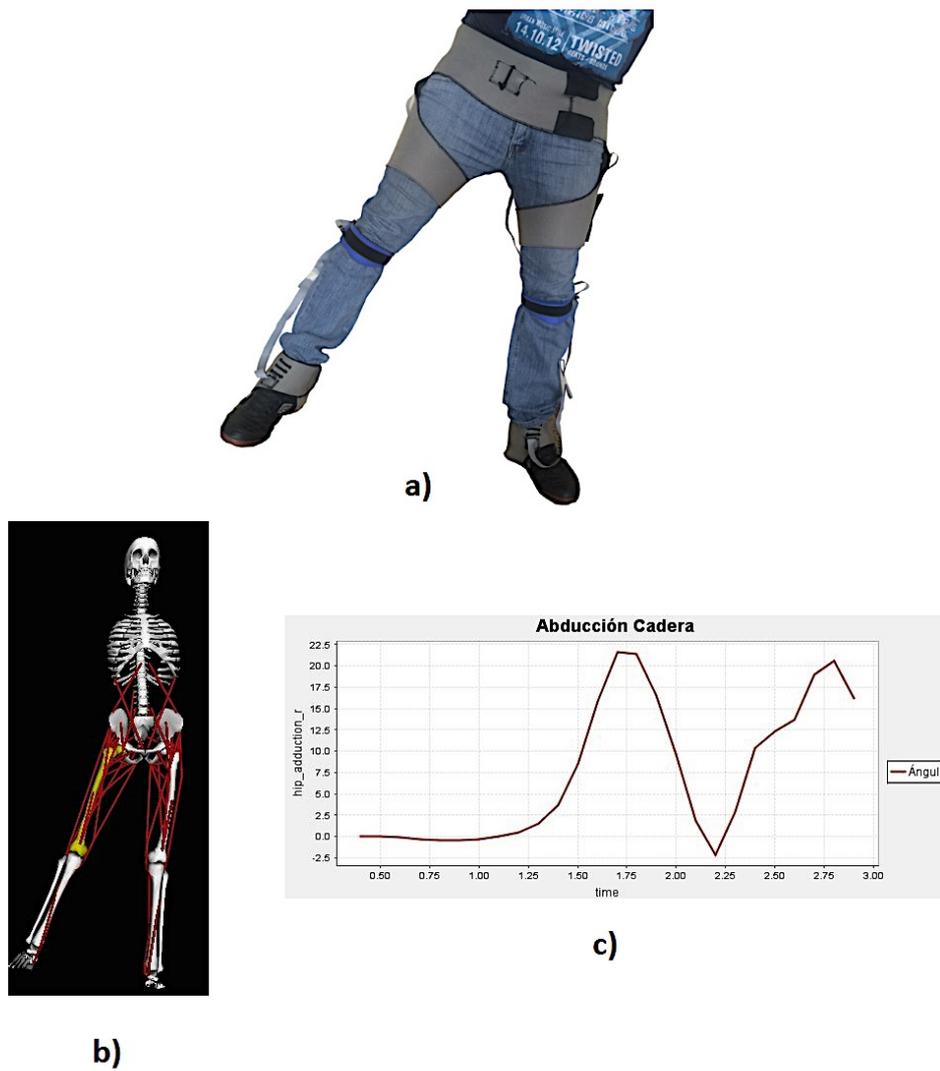


Figura 3.20: a) Sujeto de prueba b) Avatar de OpenSim c) Curva que describe el movimiento de abducción y aducción

Prueba 3: Rotación de la cadera, en OpenSim la rotación externa toma valores negativos es por ello que en la figura 3.20 se observa un valor aproximado a los -25 grados.

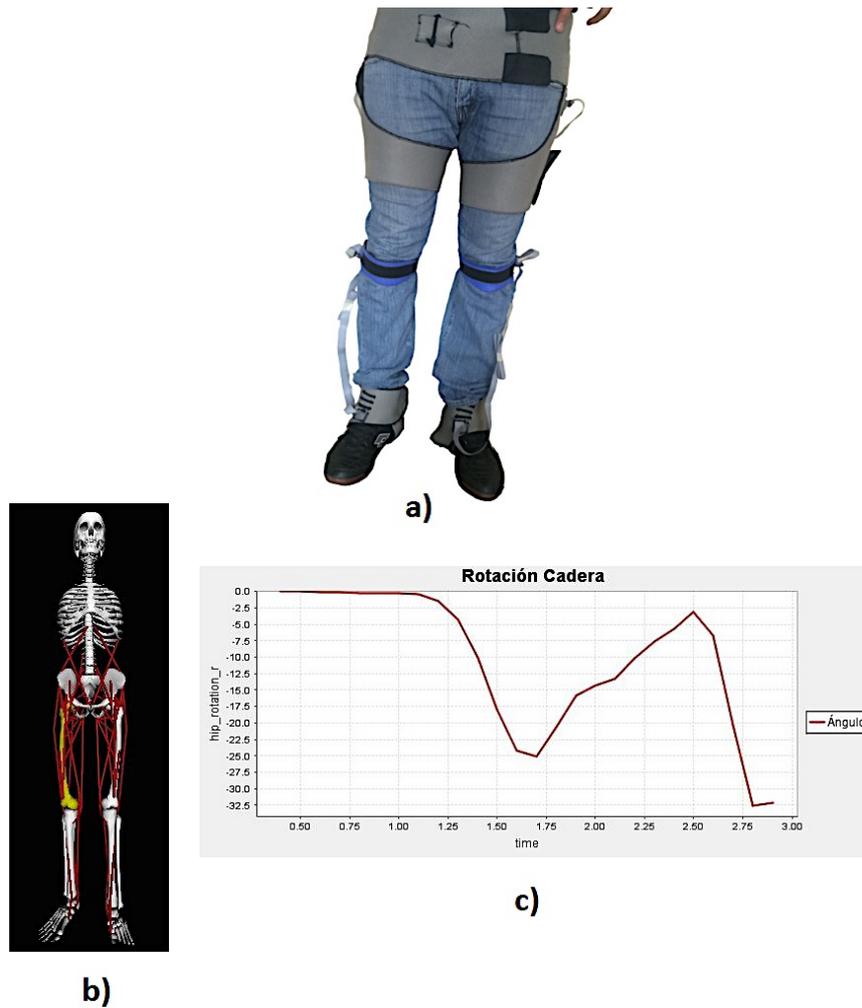


Figura 3.21 a) Sujeto de prueba b) Avatar de OpenSim c) Curva que describe el movimiento rotación en la cadera

Prueba 4: Movimientos flexión en la rodilla se alcanzó un ángulo de -75 grados y 0 grados en su posición normal.

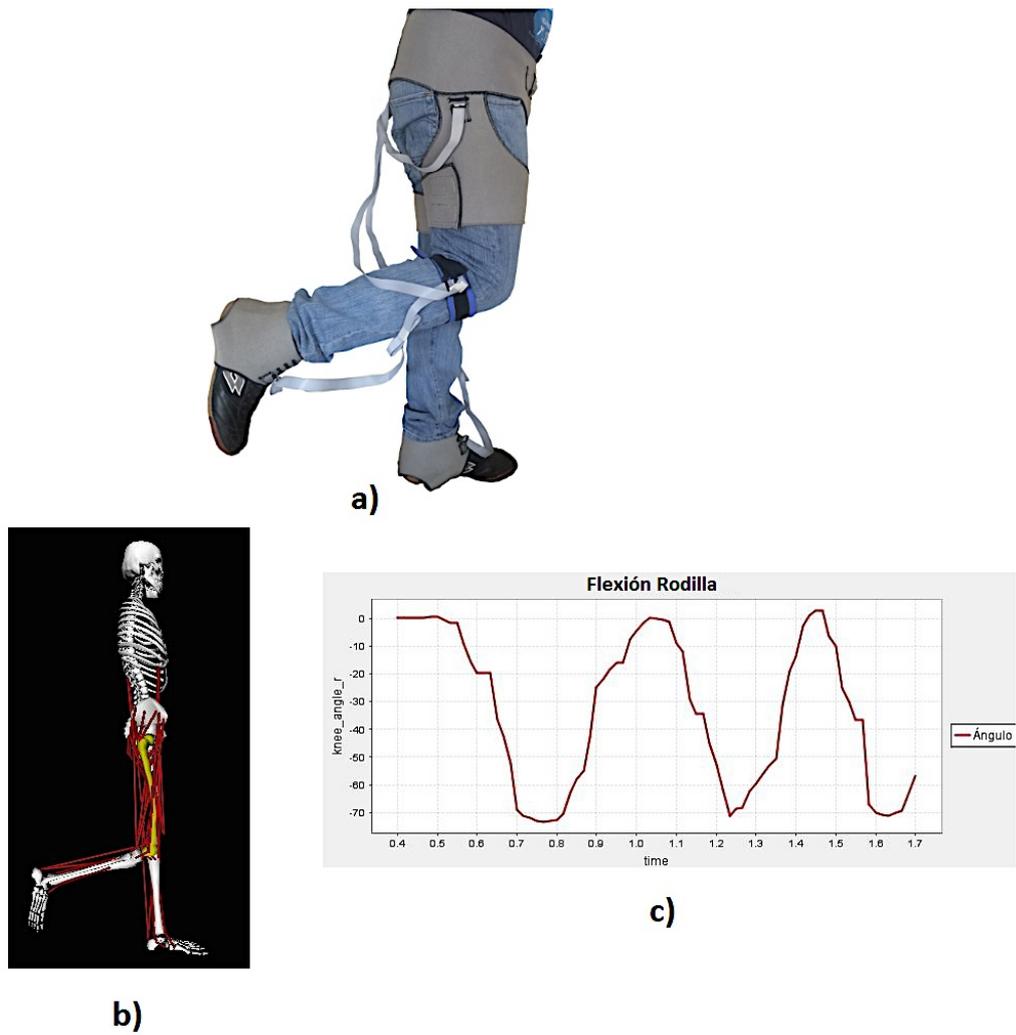


Figura 3.22: a) Sujeto de prueba b) Avatar de OpenSim c) Curva que describe el movimiento flexión de la rodilla.

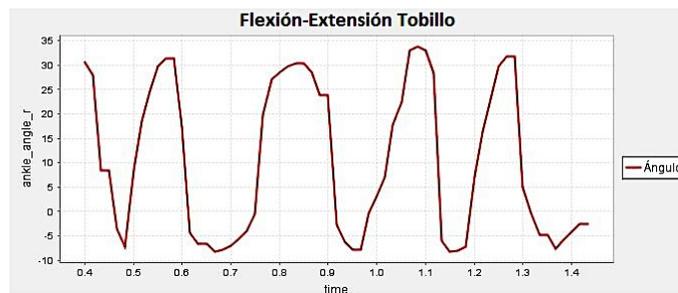
Prueba 5: Movimientos de flexión dorsal y flexión plantar para el primer movimiento los ángulos son positivos es por ello que se alcanzó los 35 grados aproximadamente mientras que para el segundo movimiento los ángulos son negativos por lo que se alcanzó ángulos cercanos a los - 10 grados.



a)



b)



c)

Figura 3.23: a) Sujeto de prueba b) Avatar de OpenSim c) Curva que describe de flexión-extensión del tobillo

Prueba 6: Ciclo de marcha , registrado en la figura 3.24. Donde se muestran las gráficas correspondientes a la cadera, rodilla y tobillo de la pierna derecha.

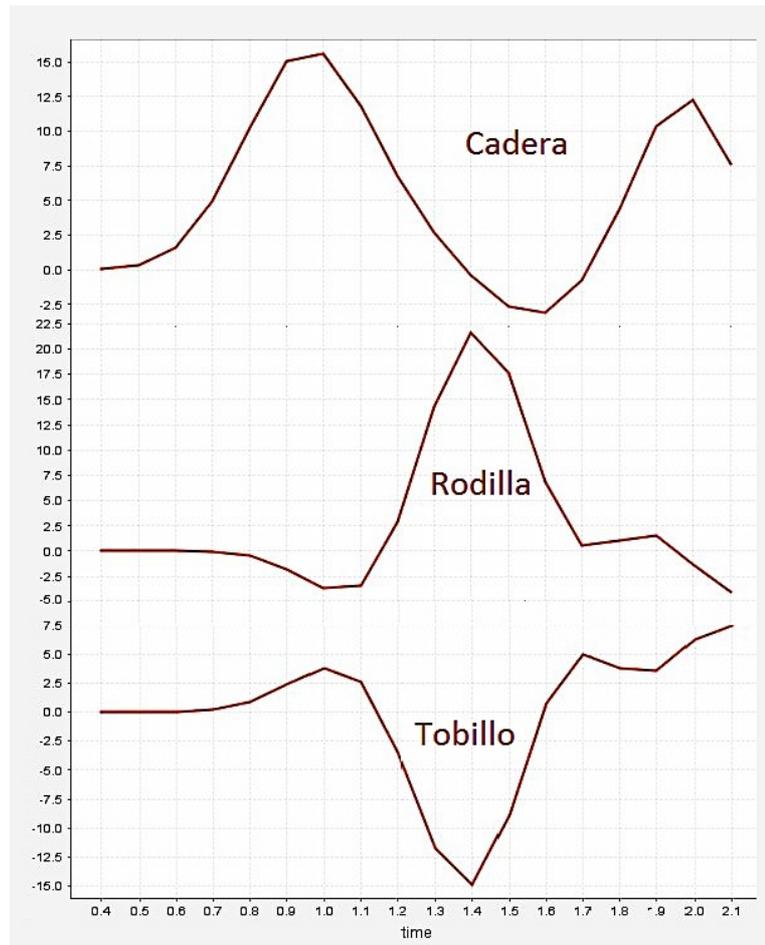


Figura 3.24: Ciclo de marcha

3.2 FALLAS DEL SISTEMA

PÉRDIDA DE DATOS.

En la configuración del envío y recepción en el puerto serial se trató de optimizar lo máximo posible, para el envío de datos se lo hace mediante un torrente de bits, confiando en que no haya pérdida, pero si las hubiese el programa del maestro siempre lee la trama de los datos entrantes la cual debería coincidir con lo trama de datos enviados si no es así los datos son descartados en ese instante y no serán grabados.

Si se realizara la petición para el reenvío de datos cuando una trama haya llegado mal, ocasionamos cuellos de botella en la trasmisión serial y nuestra prioridad es recolectar la mayor cantidad de datos posibles en el menor tiempo.

Si el algoritmo sufre retrasos en la lectura de datos en la red I2C estos se acumularían en forma masiva en la memoria *FIFO* haciendo que los datos sean imposibles de identificar por lo cual se descartan generando pérdida.

MOVIMIENTOS FALSOS.

En ocasiones el sistema recolecta ángulos falsos o erróneos que pueden variar del dato anterior y posterior hasta en 180 grados para eliminar estos datos falsos se implementó un filtro butterworth pasa bajo con frecuencia de corte 10Hz. Pero el filtrado no puede eliminar de forma absoluta este pico sino tan solo puede atenuarlo en forma de senoide lo que ocasiona que la curva que describe el movimiento en ese punto se vea alterada y esto se verá reflejado en la simulación como un movimiento.

ALGORITMOS INCOMPLETOS

A pesar que el sistema fue creado con gran fiabilidad en diferentes aspectos, todavía no puede representar los movimientos de los miembros inferiores al 100% debido a que las piernas son extremidades que pueden moverse a grandes velocidades angulares, lineales y de una forma muy compleja y el sensor no recolecta dichos datos.

El sistema puede mostrar falencias al momento de representar movimientos realizados a gran velocidad debido a los sensores inerciales empleados y la matemática empleada en este sistema. Es posible eliminar algunas fases del movimiento de las extremidades y el filtrado empleado no es del todo adecuado.

SENSORES INERCIALES POCO CONFIABLES.

Los sensores utilizados no son aptos para este tipo de proyecto puesto que se ven muy afectados por el ruido provocado en su propia red I2C a causa de otro giroscopio o por fuentes externas.

La carencia de un magnetómetro dentro del integrado MPU6050 impide la creación de un algoritmo más completo para la corrección de errores.

Debemos notar que el integrado trabaja en condiciones ideales a temperaturas de 25 grados pero durante la realización de una actividad física el cuerpo humano puede alcanzar temperaturas mayores.

El error del drift causa lecturas incorrectas en los sensores inerciales debido a la calidad del sensor por lo cual produce datos incorrectos con la realidad.

3.3 DISCUSIÓN

Mediante las pruebas de estabilización realizadas sobre uno de los giroscopios MPU 6050 se determinó la cantidad de tiempo necesario y mínimo para el momento de capturar los datos, estableciendo una pausa en el algoritmo creado para evitar la toma de datos falsos.

Las pruebas de estabilidad inercial nos ayudaron a reconocer la estabilidad del giroscopio cuando este se encuentra en reposo en donde se pudo apreciar falsos ángulos creando picos bruscos en la curva que describe su movimiento.

Mediante estas pruebas se logró establecer la necesidad de un filtro pasa bajo butterworth con frecuencia de corte 10Hz. La frecuencia de corte se la determinó estableciendo las pruebas de movimiento de una persona, la velocidad máxima a la cual una persona se puede mover. Con la aplicación del filtro se redujo de manera muy notable el pico generado en la señal, sin embargo esto provoca una onda senoidal la cual es leída como un movimiento falso oscilatorio.

Luego en las pruebas de filtrado con la realización de los dos tipos de experimento el uno con un giroscopio y el otro con dos giroscopios dieron resultados desalentadores en la estabilidad del sistema, el incremento de otro giroscopio en la red I2C provoca mayor cantidad de falsos ángulos tanto en el primer y segundo giroscopio. Pero la utilización del filtrado reduce estos picos de manera significativa. Cabe recalcar que mientras más grande es el ángulo falso más grande será el pico de la sinusoide creado por el filtrado de la señal.

Los datos calculados como el error medio y la distribución normal nos indica que nuestro sistema tiene un 59% probabilidad de error entre ± 4 en yaw, un 80% probabilidad de error entre ± 1 en pitch, un 52% probabilidad de error entre -4 y 0 en

roll. El margen de error al igual que la media se vio incrementado debido a los datos falsos ocasionados por el ruido, la moda en los tres ángulos oscila entre los -3 y 0 grados indicándonos que la mayor cantidad de datos se encuentran en ese rango esto se lo puede ver en las gráficas de frecuencia de datos.

En las pruebas de funcionamiento obtuvimos ligeros inconvenientes en la simulación puesto que el filtro utilizado amortiguaba el pico del ángulo falso convirtiéndolo en senoide el cual se veía reflejado en movimiento en las extremidades pero se debe resaltar que el avatar se movía en función a como la persona se había movido.

La utilización del filtro ocasionó que se produzca un desplazamiento de la curva de movimiento sobre el eje del tiempo lo que es interpretado como un retardo en el tiempo. Como el filtro es el mismo en todas las señales el retardo en el tiempo es el mismo para todos. Esto demuestra que el retardo es irrelevante en esta aplicación puesto que no se está visualizando en tiempo real.

Si bien es cierto, los giroscopios nos entregan el valor de los ángulos en los tres ejes, no todos son utilizados para la simulación en *OpenSim* esto es debido a los grados de libertad de cada articulación. Otro aspecto a tomar en cuenta es que para algunas articulaciones es necesario cambiar el signo de los datos en las mediciones, debido al sentido de ubicación de los giroscopios en el traje.

Para la creación del archivo .mot se debe considerar los encabezados, que la longitud de los vectores con los datos tanto de las articulaciones como del tiempo sean iguales y finalmente la creación de un vector de ceros para las articulaciones que no van a ser utilizadas caso contrario el archivo no podrá ser cargado en *OpenSim*.

Para el tiempo de simulación nos basamos en el número de muestras por segundo que en nuestro caso son 10, para ello se creó un vector de tiempo de igual longitud que los vectores que contienen los datos de cada articulación con un incremento de 0.1 segundos.

CAPÍTULO 4

4. COSTOS DEL PROYECTO

En éste punto se incluye todo los materiales necesarios que hicieron falta para construir el prototipo incluyendo la mano de obra de los tesistas así como los diferentes elementos dañados y descartados durante la fase de diseño y construcción.

COSTO DEL PROTOTIPO			
	Cantidad	P. unitario	P Total
Tarjetas Arduinos	3	30	90
Traje	1	82	82
PCBs	2	35	70
Materiales Electrónicos	1	80	80
Set de Instrumentación	1	30	30
Mano de obra de dos personas por mes	6	770	4620
Modulo SD	1	5	5
Batería	1	10	10
Giroscopios MPU 6050	9	10	90
Computador	1	580	580
Bus de datos	3	5	15
Varios	1	30	30
TOTAL PROTOTIPO			5702

Tabla 4.1: Costos del prototipo.

Una vez hecha la inversión del prototipo se puede sacar el costo de producción del sistema en donde la mano de obra se reduce drásticamente al igual que otros elementos innecesarios para construir el traje. Cabe recalcar que el cálculo de la mano de obra para conectar el traje y probarlo es de una hora.

Todos los gastos necesarios se los puede observar en la siguiente tabla.

Costo de producción del sistema por unidad			
	Cantidad	P. unitario	P Total
Traje	1	82	82
PCBs	2	35	70
Materiales Electrónicos	1	80	80
Mano de obra por ensamble	1	2.4	2.4
Modulo SD	1	5	5
Batería	1	10	10
Giroscopios MPU 6050	6	10	60
Bus de datos	3	5	15
TOTAL			324.4

Tabla 4.2: Costos de producción del sistema.

CONCLUSIONES Y RECOMENDACIONES

Se logró desarrollar un sistema recolector de datos y representarlos gráficamente en un entorno donde es posible un análisis detallado de la biomecánica humana este prototipo puede ser un complemento para el análisis de la biomecánica.

El sistema puede tener falencias en ciertas condiciones de movimientos y velocidad, demandando una gran labor en el diseño en hardware debido a que se descartó muchas formas de conexiones y en el desarrollo del software se vio de cierto modo limitado al procesador y al lenguaje de programación.

Los errores generados en el cálculo de los ángulos son despreciables debido al rango que maneja el giroscopio de los 360 grados y que son reflejados en la moda y mediana. El incremento de la media del error depende de la cantidad de picos generados por el ruido más no por el cálculo generado en el software del arduino.

Los movimientos falsos del avatar en la simulación de *OpenSim* son generados por la utilización del filtro pasa bajo butterworth que atenúa los picos de ruido generado en el sistema en forma de senoidal.

Se logró representar los movimientos de la persona en base a los sensores inerciales pero estos datos no fueron del todo confiables en circunstancia de un movimiento prolongado y de alta velocidad.

Una gran falencia en el sistema se da debido a la utilización de sensores inerciales poco confiables y a la vez incompletos como fue el caso del MPU6050 que pese a tener un acelerómetro no bastó para la corrección de los errores que se pueden producir debido a la carencia de un magnetómetro.

El traje confeccionado específicamente para esta aplicación es cómodo, maleable, con una fijación estable al cuerpo y ajustable de acuerdo a la contextura de la persona. Pero limitante en un 5% al movimiento articular de la persona debido a la sujeción muscular.

Para conseguir mayor velocidad en la captura de datos se debería conectar un solo giroscopio a un solo integrado. Y el cálculo se lo debería hacer en un procesador

de mayor capacidad por ejemplo un intel i5 o i7 debido a que el procesamiento directamente en el integrado le quita tiempo de recolección de datos.

El uso de un magnetómetro externo para el MPU 6050 puede ser una solución para corregir el ángulo yaw pero en el mercado ya existe otro tipos de giroscopios que tienen incluido en su integrado un magnetómetro.

Para reducir el tamaño de la placa se lo puede hacer con un montaje superficial debido a que el prototipo tenía que reprogramarse continuamente se lo tuvo que hacer con sócalos para retirar los integrados.

Lo ideal sería crear un traje de algodón y elástico y a este incluir los giroscopios pero como el tamaño de las personas es variante el traje servirá tan solo a una sola persona.

El sistema no está enfocado a competencias de contacto como son el futbol, básquet etc. Debido a que los giroscopios pueden sufrir golpes o caídas.

Para la eliminación de buses de datos se los puede implementar módulos wifi pero hesitaríamos un mejor procesador portátil para la recolección y almacenamiento, el hacerlo con buses de datos facilito la comunicación el procesamiento y por su puesto la economía.

Si bien es cierto él .mot es un archivo de texto delimitado por tabulaciones, no es recomendable trabajar directamente sobre este archivo debido a la cantidad de datos que se requiere ordenar los cuales pueden desplazarse ocasionando que el archivo resulte erróneo. Trabajar sobre una hoja de Excel facilita la tarea de ubicar los datos correspondientes a cada articulación de forma ordenada.

ANEXOS

ANEXO 1

TABLA DE ERROR EN ESTADO INERCIAL

ángulo Puro	ángulo Filtrado	ángulo real	Error de datos puros	Error de datos filtrados
0.04	0	0	-0.04	0
0.04	0	0	-0.04	0
0.04	0	0	-0.04	0
0.04	0	0	-0.04	0
0.03	0	0	-0.03	0
0.03	0	0	-0.03	0
0.04	0	0	-0.04	0
0.03	0	0	-0.03	0
0.03	0.0001	0	-0.03	-0.0001
0.03	0.0001	0	-0.03	-0.0001
0.03	0.0003	0	-0.03	-0.0003
0.03	0.0005	0	-0.03	-0.0005
0.03	0.0009	0	-0.03	-0.0009
0.03	0.0015	0	-0.03	-0.0015
0.03	0.0023	0	-0.03	-0.0023
0.03	0.0035	0	-0.03	-0.0035
0.03	0.005	0	-0.03	-0.005
0.03	0.0069	0	-0.03	-0.0069
0.03	0.0093	0	-0.03	-0.0093
0.03	0.012	0	-0.03	-0.012
0.03	0.015	0	-0.03	-0.015
0.03	0.0183	0	-0.03	-0.0183
0.03	0.0217	0	-0.03	-0.0217
0.03	0.0251	0	-0.03	-0.0251
0.03	0.0284	0	-0.03	-0.0284
0.03	0.0313	0	-0.03	-0.0313
0.03	0.0339	0	-0.03	-0.0339
0.03	0.0359	0	-0.03	-0.0359
0.03	0.0373	0	-0.03	-0.0373
0.03	0.0381	0	-0.03	-0.0381
0.03	0.0382	0	-0.03	-0.0382
0.03	0.0377	0	-0.03	-0.0377
0.03	0.0367	0	-0.03	-0.0367
0.03	0.0354	0	-0.03	-0.0354
0.03	0.0338	0	-0.03	-0.0338
0.03	0.0321	0	-0.03	-0.0321
0.03	0.0304	0	-0.03	-0.0304
0.03	0.0289	0	-0.03	-0.0289
-2.3	0.0276	0	2.3	-0.0276
0.03	0.0267	0	-0.03	-0.0267

0.03	0.0262	0	-0.03	-0.0262
0.03	0.0261	0	-0.03	-0.0261
0.03	0.0263	0	-0.03	-0.0263
0.03	0.0267	0	-0.03	-0.0267
0.03	0.0272	0	-0.03	-0.0272
0.03	0.0275	0	-0.03	-0.0275
0.03	0.0273	0	-0.03	-0.0273
0.03	0.026	0	-0.03	-0.026
0.04	0.023	0	-0.04	-0.023
0.04	0.0176	0	-0.04	-0.0176
0.04	0.0092	0	-0.04	-0.0092
-46.9	-0.0029	0	46.9	0.0029
0.04	-0.0189	0	-0.04	0.0189
0.04	-0.0388	0	-0.04	0.0388
0.04	-0.0624	0	-0.04	0.0624
0.04	-0.0889	0	-0.04	0.0889
0.04	-0.118	0	-0.04	0.118
0.04	-0.1496	0	-0.04	0.1496
0.04	-0.1854	0	-0.04	0.1854
0.04	-0.229	0	-0.04	0.229
0.04	-0.287	0	-0.04	0.287
0.04	-0.3689	0	-0.04	0.3689
0.04	-0.4861	0	-0.04	0.4861
0.04	-0.6514	0	-0.04	0.6514
0.04	-0.8759	0	-0.04	0.8759
0.04	-1.1677	0	-0.04	1.1677
0.04	-1.5285	0	-0.04	1.5285
0.04	-1.9527	0	-0.04	1.9527
0.03	-2.4257	0	-0.03	2.4257
0.03	-2.924	0	-0.03	2.924
0.03	-3.4166	0	-0.03	3.4166
0.03	-3.8676	0	-0.03	3.8676
0.03	-4.2388	0	-0.03	4.2388
0.03	-4.4945	0	-0.03	4.4945
0.03	-4.6043	0	-0.03	4.6043
0.03	-4.5477	0	-0.03	4.5477
0.04	-4.3158	0	-0.04	4.3158
0.04	-3.9134	0	-0.04	3.9134
0.03	-3.359	0	-0.03	3.359
0.04	-2.6835	0	-0.04	2.6835
0.04	-1.9275	0	-0.04	1.9275
0.04	-1.1385	0	-0.04	1.1385
0.04	-0.3661	0	-0.04	0.3661
0.04	0.3419	0	-0.04	-0.3419
0.04	0.9436	0	-0.04	-0.9436

0.04	1.4065	0	-0.04	-1.4065
0.04	1.7097	0	-0.04	-1.7097
0.05	1.8457	0	-0.05	-1.8457
0.05	1.8201	0	-0.05	-1.8201
0.04	1.6505	0	-0.04	-1.6505
0.05	1.3646	0	-0.05	-1.3646
0.05	0.9972	0	-0.05	-0.9972
0.04	0.587	0	-0.04	-0.587
0.04	0.1726	0	-0.04	-0.1726
0.04	-0.2102	0	-0.04	0.2102
0.04	-0.5315	0	-0.04	0.5315
0.04	-0.7694	0	-0.04	0.7694
0.04	-0.9113	0	-0.04	0.9113
0.04	-0.954	0	-0.04	0.954
0.04	-0.9034	0	-0.04	0.9034
0.04	-0.7729	0	-0.04	0.7729
0.04	-0.582	0	-0.04	0.582
0.04	-0.3536	0	-0.04	0.3536
0.04	-0.112	0	-0.04	0.112
0.04	0.1197	0	-0.04	-0.1197
0.04	0.3216	0	-0.04	-0.3216
0.04	0.478	0	-0.04	-0.478
0.04	0.5792	0	-0.04	-0.5792
0.04	0.6211	0	-0.04	-0.6211
0.04	0.6056	0	-0.04	-0.6056
0.04	0.5395	0	-0.04	-0.5395
0.04	0.4339	0	-0.04	-0.4339
0.04	0.3022	0	-0.04	-0.3022
0.04	0.1589	0	-0.04	-0.1589
0.04	0.0185	0	-0.04	-0.0185
0.04	-0.1064	0	-0.04	0.1064
0.04	-0.2057	0	-0.04	0.2057
0.04	-0.2726	0	-0.04	0.2726
0.04	-0.304	0	-0.04	0.304
0.04	-0.3004	0	-0.04	0.3004
0.04	-0.2653	0	-0.04	0.2653
0.04	-0.2051	0	-0.04	0.2051
0.04	-0.1276	0	-0.04	0.1276
0.03	-0.0418	0	-0.03	0.0418
0.03	0.0436	0	-0.03	-0.0436
0.03	0.1207	0	-0.03	-0.1207
0.03	0.1831	0	-0.03	-0.1831
0.03	0.2262	0	-0.03	-0.2262
0.03	0.248	0	-0.03	-0.248
0.03	0.2483	0	-0.03	-0.2483

0.03	0.2291	0	-0.03	-0.2291
0.03	0.194	0	-0.03	-0.194
0.03	0.1478	0	-0.03	-0.1478
0.04	0.0958	0	-0.04	-0.0958
0.04	0.0434	0	-0.04	-0.0434
0.03	-0.0046	0	-0.03	0.0046
0.03	-0.044	0	-0.03	0.044
0.03	-0.072	0	-0.03	0.072
0.04	-0.0871	0	-0.04	0.0871
0.04	-0.0892	0	-0.04	0.0892
0.04	-0.0793	0	-0.04	0.0793
0.04	-0.0595	0	-0.04	0.0595
0.04	-0.0326	0	-0.04	0.0326
0.04	-0.0018	0	-0.04	0.0018
0.04	0.0295	0	-0.04	-0.0295
0.04	0.0583	0	-0.04	-0.0583
0.04	0.0821	0	-0.04	-0.0821
0.04	0.0991	0	-0.04	-0.0991
	Media		0.297027027	0.303039189
	Desviación Estándar		3.861580421	1.167651715

ANEXO 2

TABLA DE ERROR DE YAW.

Valor Real	Valor Medido	Error	Distribución Normal	Dist. Nor. Acumulad	rango	frecuencia
-50.43	-50.2354	-0.1946	0.081984751	0.434340812	-12	0
-51.15	-50.9081	-0.2419	0.081847315	0.430466153	-10	8
-51.5	-51.5766	0.0766	0.082622206	0.456667342	-8	6
-52.22	-52.2404	0.0204	0.082511335	0.452027038	-6	5
-52.94	-52.8994	-0.0406	0.082378382	0.446997835	-4	5
-53.62	-53.5541	-0.0659	0.082319401	0.444914403	-2	3
-54.29	-54.2057	-0.0843	0.082275097	0.443400132	0	121
-54.95	-54.8553	-0.0947	0.082249531	0.442544603	2	77
-55.62	-55.5047	-0.1153	0.082197776	0.440850793	4	9
-56.26	-56.1554	-0.1046	0.082224844	0.441730455	6	4
-56.93	-56.8091	-0.1209	0.082183451	0.440390526	8	2
-57.59	-57.4671	-0.1229	0.082178308	0.440226164	10	3
-58.28	-58.1304	-0.1496	0.082108318	0.438032932	12	5
-58.95	-58.7998	-0.1502	0.082106717	0.437983668	14	5
-59.63	-59.4753	-0.1547	0.082094667	0.437614215	16	10
-60.3	-60.1568	-0.1432	0.082125321	0.43855848		
-60.97	-60.8437	-0.1263	0.082169533	0.439946773		
-61.67	-61.5347	-0.1353	0.082146112	0.439207352		
-62.4	-62.2288	-0.1712	0.082049883	0.436260021		
-63.13	-62.9245	-0.2055	0.081953765	0.433447347		
-63.87	-63.62	-0.25	0.081823006	0.429803288		
-64.61	-64.3141	-0.2959	0.081680995	0.426050844		
-64.97	-65.005	0.035	0.082541212	0.453231922		
-65.71	-65.6916	-0.0184	0.082428284	0.448827192		
-66.44	-66.3726	-0.0674	0.082315834	0.444790926		
-67.18	-67.0472	-0.1328	0.082152646	0.439412726		
-67.88	-67.7145	-0.1655	0.082065461	0.43672775		
-68.56	-68.374	-0.186	0.082008909	0.435045985		
-69.19	-69.0253	-0.1647	0.082067638	0.436793403		
-69.81	-69.6681	-0.1419	0.082128757	0.438665245		
-70.42	-70.3023	-0.1177	0.08219165	0.440653526		
-71.04	-70.9277	-0.1123	0.082205406	0.441097398		
-71.65	-71.5441	-0.1059	0.082221576	0.441623565		
-72.25	-72.1516	-0.0984	0.082240345	0.442240297		
-72.85	-72.7499	-0.1001	0.082236108	0.442100492		
-73.43	-73.3387	-0.0913	0.082257931	0.442824266		
-74.02	-73.9179	-0.1021	0.08223111	0.441936025		
-74.6	-74.4873	-0.1127	0.08220439	0.441064516		
-75.19	-75.0466	-0.1434	0.082124792	0.438542055		

-75.74	-75.5958	-0.1442	0.082122673	0.438476356
-76.27	-76.135	-0.135	0.082146897	0.439231996
-76.79	-76.6646	-0.1254	0.08217186	0.440020726
-77.31	-77.1849	-0.1251	0.082172635	0.440045378
-77.58	-77.697	0.117	0.082694995	0.460006769
-78.13	-78.202	0.072	0.082613551	0.4562873
-78.69	-78.7012	0.0112	0.082492123	0.451268021
-79.23	-79.1963	-0.0337	0.082394078	0.4475663
-79.75	-79.689	-0.061	0.082331	0.445317796
-80.25	-80.1812	-0.0688	0.082312497	0.444675687
-80.74	-80.6748	-0.0652	0.082321063	0.444972027
-81.23	-81.1712	-0.0588	0.08233618	0.44549893
-81.73	-81.6718	-0.0582	0.08233759	0.445548332
-82.25	-82.1777	-0.0723	0.082304125	0.444387607
-82.8	-82.6892	-0.1108	0.082209208	0.441220709
-83.35	-83.2061	-0.1439	0.082123468	0.438500993
-83.9	-83.7279	-0.1721	0.082047413	0.436186177
-84.44	-84.253	-0.187	0.082006113	0.434963977
-84.99	-84.7796	-0.2104	0.081939702	0.433045807
-85.52	-85.3052	-0.2148	0.081927003	0.432685301
-86.04	-85.8272	-0.2128	0.081932783	0.432849161
-86.55	-86.3427	-0.2073	0.081948609	0.433299834
-87.04	-86.8488	-0.1912	0.081994333	0.434619576
-87.51	-87.3429	-0.1671	0.082061099	0.436596448
-87.97	-87.8226	-0.1474	0.082114179	0.438213577
-88.4	-88.2861	-0.1139	0.08220134	0.440965873
-88.63	-88.7325	0.1025	0.082669536	0.458807876
-89.07	-89.1612	0.0912	0.082649179	0.457873824
-89.51	-89.5724	0.0624	0.082595247	0.455494298
-89.94	-89.967	0.027	0.082524934	0.452571657
-90.37	-90.3463	-0.0237	0.082416528	0.448390353
-90.75	-90.7119	-0.0381	0.082384088	0.447203788
-91.11	-91.0657	-0.0443	0.082369896	0.44669305
-91.44	-91.4092	-0.0308	0.082400624	0.447805252
-91.78	-91.7441	-0.0359	0.082389091	0.447385038
-92.14	-92.0714	-0.0686	0.082312974	0.444692149
-92.5	-92.3916	-0.1084	0.082215277	0.441418019
-92.87	-92.7048	-0.1652	0.082066277	0.436752369
-93.21	-93.0104	-0.1996	0.081970589	0.433930923
-93.51	-93.3073	-0.2027	0.081961764	0.433676828
-93.76	-93.5941	-0.1659	0.082064371	0.436694924
-93.98	-93.8695	-0.1105	0.082209968	0.441245372
-94.2	-94.1318	-0.0682	0.082313928	0.444725075
-94.4	-94.3802	-0.0198	0.082425188	0.448711794
-94.58	-94.6139	0.0339	0.082538987	0.453141128

-94.77	-94.8334	0.0634	0.082597169	0.455576894
-94.96	-95.0396	0.0796	0.08262781	0.456915217
-95.13	-95.2348	0.1048	0.082673624	0.45899802
-95.22	-95.4219	0.2019	0.082829044	0.467033446
-95.4	-95.6046	0.2046	0.082832885	0.46725709
-95.59	-95.7872	0.1972	0.082822295	0.466644166
-95.82	-95.9742	0.1542	0.08275689	0.46308419
-96.06	-96.1696	0.1096	0.082682095	0.459394874
-96.3	-96.377	0.077	0.082622955	0.456700391
-96.6	-96.5986	-0.0014	0.082465325	0.450228789
-96.96	-96.8352	-0.1248	0.08217341	0.44007003
-97.36	-97.0859	-0.2741	0.081749344	0.427832237
-97.77	-97.3477	-0.4223	0.081252786	0.415752857
-98.13	-97.6158	-0.5142	0.080907632	0.40830137
-98.44	-97.8838	-0.5562	0.080740524	0.404906738
-98.67	-98.1436	-0.5264	0.080859693	0.407314588
-98.84	-98.3868	-0.4532	0.081139888	0.413243882
-98.94	-98.6042	-0.3358	0.08155169	0.422794333
-98.97	-98.7875	-0.1825	0.082018667	0.435333033
-98.95	-98.9289	-0.0211	0.082422308	0.448604643
-98.9	-99.0224	0.1224	0.082704285	0.460453347
-98.83	-99.0639	0.2339	0.082872897	0.469684687
-98.72	-99.0514	0.3314	0.082983915	0.477770484
-98.6	-98.9853	0.3853	0.083030647	0.482244623
-98.46	-98.868	0.408	0.083047202	0.48412961
-98.39	-98.7038	0.3138	0.082966396	0.476310119
-98.22	-98.4983	0.2783	0.082927678	0.473365486
-98.04	-98.258	0.218	0.082851565	0.468367176
-97.85	-97.9899	0.1399	0.082733679	0.461900931
-97.64	-97.7004	0.0604	0.082591393	0.455329111
-97.41	-97.3953	-0.0147	0.082436433	0.449132191
-97.17	-97.0793	-0.0907	0.082259409	0.442873621
-96.9	-96.7558	-0.1442	0.082122673	0.438476356
-96.59	-96.4267	-0.1633	0.082071443	0.4369083
-96.22	-96.0926	-0.1274	0.082166686	0.439856388
-95.84	-95.7531	-0.0869	0.082268741	0.443186225
-95.41	-95.4071	-0.0029	0.082462098	0.450105093
-94.98	-95.0532	0.0732	0.082615816	0.456386438
-94.53	-94.6899	0.1599	0.082765939	0.46355593
-94.07	-94.3163	0.2463	0.082888906	0.470712411
-93.61	-93.9319	0.3219	0.082974597	0.47698218
-93.17	-93.537	0.367	0.083015951	0.480725295
-92.78	-93.1327	0.3527	0.08300363	0.479538254
-92.41	-92.7206	0.3106	0.082963091	0.476044632
-92.06	-92.3023	0.2423	0.082883802	0.470380865

-91.72	-91.8798	0.1598	0.082765781	0.463547653
-91.55	-91.4543	-0.0957	0.082247053	0.442462355
-91.19	-91.0265	-0.1635	0.0820709	0.436891886
-90.81	-90.5962	-0.2138	0.081929895	0.432767229
-90.4	-90.1619	-0.2381	0.081858641	0.430777194
-89.97	-89.7211	-0.2489	0.08182632	0.429893295
-89.52	-89.2707	-0.2493	0.081825115	0.429860565
-88.99	-88.8064	-0.1836	0.082015605	0.435242814
-88.39	-88.3241	-0.0659	0.082319401	0.444914403
-87.77	-87.8196	0.0496	0.082570335	0.454437237
-87.13	-87.2893	0.1593	0.082764992	0.46350627
-86.46	-86.7306	0.2706	0.082918683	0.472726977
-85.79	-86.1425	0.3525	0.083003452	0.479521653
-85.1	-85.5252	0.4252	0.083058511	0.485558121
-84.44	-84.8809	0.4409	0.083067904	0.486862214
-83.77	-84.213	0.443	0.083069093	0.487036658
-83.13	-83.5264	0.3964	0.083038973	0.48316631
-82.49	-82.827	0.337	0.082989256	0.478235209
-81.82	-82.121	0.301	0.082952956	0.475248235
-81.15	-81.4148	0.2648	0.082911768	0.472246069
-80.5	-80.7138	0.2138	0.082845779	0.468019212
-79.84	-80.0228	0.1828	0.082801126	0.465451676
-79.51	-79.3447	-0.1653	0.082066005	0.436744163
-78.85	-78.6811	-0.1689	0.082056182	0.436448743
-78.21	-78.0316	-0.1784	0.082030045	0.435669333
-77.59	-77.3941	-0.1959	0.081981077	0.434234234
-76.95	-76.7653	-0.1847	0.082012539	0.435152599
-76.28	-76.1406	-0.1394	0.082135349	0.438870575
-75.56	-75.5149	-0.0451	0.082368055	0.446627155
-74.83	-74.8834	0.0534	0.082577791	0.454751019
-74.09	-74.2413	0.1513	0.082752242	0.462844201
-73.34	-73.5853	0.2453	0.082887636	0.470629523
-72.57	-72.9132	0.3432	0.082995038	0.47874976
-71.82	-72.2244	0.4044	0.0830447	0.483830645
-71.09	-71.5199	0.4299	0.083061416	0.485948503
-70.39	-70.8024	0.4124	0.083050196	0.484495024
-69.69	-70.0756	0.3856	0.083030877	0.482269532
-69	-69.3442	0.3442	0.082995958	0.478832755
-68.33	-68.6133	0.2833	0.082933404	0.473780139
-67.67	-67.8879	0.2179	0.082851428	0.468358891
-67	-67.1727	0.1727	0.082785837	0.464615462
-66.66	-66.471	-0.189	0.082000511	0.434799971
-65.96	-65.7854	-0.1746	0.082040537	0.435981067
-65.27	-65.1164	-0.1536	0.082097619	0.43770452
-64.57	-64.462	-0.108	0.082216286	0.441450905

-63.88	-63.8168	-0.0632	0.082325803	0.445136674
-63.18	-63.1704	-0.0096	0.082447586	0.449552646
-62.5	-62.5066	0.0066	0.082482404	0.45088858
-61.83	-61.8024	-0.0276	0.082407814	0.448068945
-61.19	-61.0288	-0.1612	0.082077138	0.437080656
-60.59	-60.153	-0.437	0.081199478	0.414558832
-60.01	-59.1421	-0.8679	0.079321076	0.379953024
-59.44	-57.9684	-1.4716	0.075728807	0.333097862
-58.89	-56.6149	-2.2751	0.069472957	0.274662857
-58.36	-55.0809	-3.2791	0.059968494	0.209563281
-57.82	-53.3857	-4.4343	0.047962309	0.147180169
-57.27	-51.5711	-5.6989	0.035143751	0.094749524
-56.69	-49.7001	-6.9899	0.023816709	0.056936637
-56.08	-47.8536	-8.2264	0.015331812	0.032985181
-57	-46.1238	-10.8762	0.004771006	0.008408907
-55.14	-44.6057	-10.5343	0.005642378	0.010185663
-54.51	-43.3879	-11.1221	0.004215521	0.007305181
-53.91	-42.5438	-11.3662	0.003718423	0.006337903
-53.32	-42.124	-11.196	0.004059499	0.006999449
-52.71	-42.1515	-10.5585	0.005576711	0.010049913
-52.09	-42.6213	-9.4687	0.009212662	0.017977432
-51.46	-43.5016	-7.9584	0.016962886	0.037310111
-50.86	-44.7392	-6.1208	0.031195042	0.080762898
-50.28	-46.265	-4.015	0.05236299	0.168213241
-49.68	-48.001	-1.679	0.074261018	0.317541994
-49.11	-49.8655	0.7555	0.083068811	0.513004876
-48.55	-51.777	3.227	0.07154472	0.707982365
-47.99	-53.657	5.667	0.04759886	0.854477817
-47.46	-55.4308	7.9708	0.025556017	0.937705258
-46.95	-57.0278	10.0778	0.011826964	0.975851737
-46.48	-58.3814	11.9014	0.005196437	0.990730336
-46.01	-59.4295	13.4195	0.002347257	0.996218014
-45.56	-60.1157	14.5557	0.001212881	0.998179197
-45	-60.3922	15.3922	0.000719712	0.998971638
-44.89	-60.2226	15.3326	0.000747729	0.998927913
-44.43	-59.5858	15.1558	0.000836664	0.998787984
-43.94	-58.4807	14.5407	0.001223946	0.998160921
-43.43	-56.9296	13.4996	0.002244621	0.996401892
-42.92	-54.9813	12.0613	0.004801741	0.991529349
-42.41	-52.7131	10.3031	0.010768128	0.978395677
-41.87	-50.23	8.36	0.022489723	0.947047367
-41.31	-47.6617	6.3517	0.04052928	0.884634736
-40.7	-45.1554	4.4554	0.060187318	0.789133027
-40.09	-42.8662	2.7762	0.074988285	0.674935624
-39.48	-40.9435	1.4635	0.081775859	0.571464604

-38.85	-39.5166	0.6666	0.083104733	0.505618251
-38.25	-38.6804	0.4304	0.08306172	0.485990033
-37.64	-38.4823	0.8423	0.083006276	0.520212731
-36.99	-38.9135	1.9235	0.080008276	0.608701965
-36.3	-39.9062	3.6062	0.068302728	0.734506482
-35.59	-41.336	5.746	0.046772533	0.858205476
-34.85	-43.0308	8.1808	0.023872446	0.942894066
-34	-44.7858	10.7858	0.008742781	0.983091625
-33.75	-46.382	12.632	0.003589417	0.993909811
-33.03	-47.6062	14.5762	0.001197902	0.998203908
-32.33	-48.2718	15.9418	0.00050241	0.99930427
-31.63	-48.2364	16.6064	0.000319661	0.999573311
-30.95	-47.4145	16.4645	0.000352627	0.999525647
-30.29	-45.7853	15.4953	0.000673456	0.999043432
-29.66	-43.3936	13.7336	0.001966641	0.99689399
-29.04	-40.3449	11.3049	0.006909298	0.987139412
-28.41	-36.7946	8.3846	0.022303839	0.947598326
-27.77	-32.9337	5.1637	0.052879733	0.829192524
-27.09	-28.9719	1.8819	0.080196829	0.605369679
-26.4	-25.1195	-1.2805	0.076979583	0.34769075
-25.71	-21.5708	-4.1392	0.051059568	0.161790682
-25.04	-18.4892	-6.5508	0.027407964	0.068172395
-24.37	-15.9963	-8.3737	0.014483879	0.030789675
-23.7	-14.1642	-9.5358	0.00894559	0.017368257
-23.03	-13.0135	-10.0165	0.007204433	0.013498454
-23	-12.5143	-10.4857	0.005776158	0.010463123
-22	-12.5923	-9.4077	0.009460762	0.018546946
-21.3	-13.1374	-8.1626	0.015709725	0.033975371
-20.57	-14.0148	-6.5552	0.027370555	0.068051882
-19.82	-15.077	-4.743	0.04474184	0.132872111
-19.01	-16.1765	-2.8335	0.064361385	0.23727234
-18.22	-17.1769	-1.0431	0.078388968	0.366136251
-17.47	-17.9626	0.4926	0.083092558	0.491157504
-16.72	-18.4463	1.7263	0.080852183	0.592839038
-15.96	-18.5727	2.6127	0.076111697	0.662582234
-15.14	-18.3205	3.1805	0.071921819	0.704646749
-14.3	-17.7008	3.4008	0.070094415	0.72029172
-13.49	-16.7529	3.2629	0.071250361	0.710545545
-12.71	-15.5385	2.8285	0.074614166	0.678847759
-11.93	-14.1348	2.2048	0.078589788	0.631014573
-11.14	-12.6258	1.4858	0.081706581	0.573287437
-10.35	-11.0953	0.7453	0.083074379	0.512157546
-9.51	-9.6196	0.1096	0.082682095	0.459394874
-8.67	-8.2618	-0.4082	0.081303236	0.416898878
	Media	0.599336		

	604	
Desviación estándar	4.802126741	

ANEXO 3

TABLA DE ERROR DE PITCH

Valor Real	Valor Medido	Error	Distribución Normal	Drist. Norm. Acumulada	Rango	Frecuencia
6.49	5.09	1.4	0.162706736	0.94105738	-2	0
6.46	5.6947	0.7653	0.437112448	0.753210034	-1.8	3
6.43	6.2403	0.1897	0.54905964	0.455172305	-1.6	4
6.41	6.7049	-0.2949	0.406416938	0.2165802	-1.4	2
6.38	7.0707	-0.6907	0.227566446	0.091431005	-1.2	3
6.35	7.326	-0.976	0.124341129	0.04207065	-1	7
6.34	7.4658	-1.1258	0.085043078	0.026518101	-0.8	5
6.3	7.4923	-1.1923	0.070861481	0.02134495	-0.6	6
6.23	7.4146	-1.1846	0.072405688	0.021896512	-0.4	6
6.15	7.2478	-1.0978	0.091600157	0.02899028	-0.2	15
6.08	7.0118	-0.9318	0.137949665	0.04786388	-2.E-16	29
6.04	6.7295	-0.6895	0.228070485	0.091704387	0.2	39
6.02	6.4252	-0.4052	0.356370099	0.174491156	0.4	17
5.99	6.1222	-0.1322	0.472772538	0.288268995	0.6	19
5.95	5.8419	0.1081	0.538665054	0.410747221	0.8	36
5.87	5.6014	0.2686	0.552548582	0.498673879	1	39
5.78	5.4134	0.3666	0.547729025	0.552669986	1.2	21
5.71	5.2847	0.4253	0.540076338	0.584614139	1.4	5
5.68	5.2169	0.4631	0.533335776	0.604905996	1.6	7
5.68	5.2062	0.4738	0.531178591	0.610601244	1.8	0
5.69	5.2445	0.4455	0.536646676	0.595489714	2	0
5.7	5.3204	0.3796	0.546336165	0.559781598		0
5.69	5.4203	0.2697	0.552550739	0.499281683		
5.66	5.53	0.13	0.542114713	0.422582626		
5.64	5.636	0.004	0.516031914	0.355763487		
5.6	5.7262	-0.1262	0.474955302	0.291112189		
5.56	5.7915	-0.2315	0.433698874	0.243219619		
5.51	5.8259	-0.3159	0.397088516	0.208143177		
5.43	5.8267	-0.3967	0.360296224	0.177536993		
5.37	5.7949	-0.4249	0.347249338	0.167560463		
5.33	5.7344	-0.4044	0.356739894	0.1747764		
5.32	5.6514	-0.3314	0.390129109	0.202042165		
5.32	5.554	-0.234	0.432652361	0.24213668		
5.33	5.4509	-0.1209	0.476864398	0.29363452		
5.33	5.3508	-0.0208	0.509217994	0.343049325		
5.33	5.2615	0.0685	0.531240543	0.389558119		
5.33	5.1893	0.1407	0.543626272	0.428391443		
5.36	5.1385	0.2215	0.551254549	0.472670106		
5.36	5.1111	0.2489	0.552292843	0.487790515		
5.33	5.1071	0.2229	0.551326802	0.473441913		

5.31	5.124	0.186	0.548735688	0.453141379
5.32	5.1579	0.1621	0.546301973	0.440054506
5.34	5.2032	0.1368	0.543088657	0.426272344
5.39	5.254	0.136	0.542976484	0.425837918
5.41	5.304	0.106	0.538309396	0.409616397
5.4	5.3475	0.0525	0.527819279	0.381085323
5.34	5.3794	-0.0394	0.503776449	0.333628047
5.26	5.3963	-0.1363	0.471268036	0.286333708
5.17	5.3959	-0.2259	0.436033285	0.245654876
5.09	5.3779	-0.2879	0.409497867	0.219435911
5.07	5.3434	-0.2734	0.41582993	0.225419621
5.07	5.2946	-0.2246	0.436573241	0.24622207
5.07	5.2353	-0.1653	0.460338225	0.272824221
5.05	5.1695	-0.1195	0.477365672	0.294302481
5.03	5.1019	-0.0719	0.493620636	0.317418167
5.01	5.0367	-0.0267	0.507522052	0.340049928
4.98	4.9779	0.0021	0.515528184	0.354783504
4.95	4.9288	0.0212	0.520450424	0.364677603
4.95	4.8916	0.0584	0.529108539	0.384203276
4.98	4.8674	0.1126	0.539412601	0.413172903
4.99	4.8565	0.1335	0.5426218	0.424480919
4.97	4.8582	0.1118	0.539281159	0.412741425
4.94	4.8709	0.0691	0.531364195	0.3898769
4.93	4.8926	0.0374	0.524374421	0.373140999
4.93	4.9209	0.0091	0.517368749	0.358398668
4.95	4.9533	-0.0033	0.514079777	0.352003552
4.96	4.9875	-0.0275	0.507289921	0.339644004
4.98	5.0215	-0.0415	0.503144767	0.332570779
5.03	5.0533	-0.0233	0.508502829	0.341777173
5.11	5.0819	0.0281	0.522150291	0.368274599
5.17	5.1062	0.0638	0.530260264	0.387063584
5.2	5.126	0.074	0.532361333	0.392483037
5.21	5.1411	0.0689	0.531323015	0.389770631
5.23	5.1517	0.0783	0.53321768	0.394774038
5.21	5.1584	0.0516	0.527619791	0.380610375
5.19	5.1618	0.0282	0.522174617	0.368326816
5.15	5.1628	-0.0128	0.511472075	0.347132121
5.09	5.1622	-0.0722	0.493523192	0.317270096
5.06	5.1612	-0.1012	0.483799389	0.303097481
5.03	5.161	-0.131	0.473210901	0.288836585
5.08	5.163	-0.083	0.48997165	0.311959148
5.15	5.169	-0.019	0.509729751	0.343966379
5.21	5.181	0.029	0.522368904	0.368744633
5.22	5.2017	0.0183	0.519723472	0.363169349
5.22	5.2344	-0.0144	0.511025481	0.346314122

5.15	5.2831	-0.1331	0.472443176	0.287843647
5.15	5.3524	-0.2024	0.445674839	0.256015448
5.3	5.4478	-0.1478	0.466993226	0.280938629
5.53	5.5746	-0.0446	0.502205966	0.331012483
5.91	5.7376	0.1724	0.547423033	0.445687283
6.29	5.9403	0.3497	0.549278766	0.543399853
6.47	6.1842	0.2858	0.552435558	0.508177192
6.47	6.4676	0.0024	0.515607925	0.354938175
6.44	6.7859	-0.3459	0.383569086	0.196432799
6.52	7.1307	-0.6107	0.262143743	0.111100683
6.72	7.491	-0.771	0.19502243	0.074480198
7	7.853	-0.853	0.164474323	0.059760568
7.14	8.2022	-1.0622	0.100453678	0.032407107
7.3	8.524	-1.224	0.06476581	0.01919636
7.31	8.8056	-1.4956	0.027690697	0.007206252
7.32	9.0374	-1.7174	0.012456507	0.002943439
7.48	9.2138	-1.7338	0.011698124	0.002745428
7.8	9.3341	-1.5341	0.024269019	0.006207238
8.24	9.4027	-1.1627	0.076936237	0.02353143
8.65	9.4289	-0.7789	0.191955452	0.072951652
8.98	9.4259	-0.4459	0.337506797	0.160370502
9.27	9.4101	-0.1401	0.469864365	0.284545553
9.56	9.3992	0.1608	0.546152743	0.43934441
9.88	9.4113	0.4687	0.532220271	0.607889567
10.2	9.4629	0.7371	0.448616297	0.740720374
10.48	9.5686	0.9114	0.372851264	0.81245533
10.71	9.7393	0.9707	0.345487453	0.833755737
10.91	9.9822	0.9278	0.365320162	0.818508386
11.07	10.3004	0.7696	0.435326068	0.75508578
11.3	10.6932	0.6068	0.495907441	0.679068986
11.49	11.1562	0.3338	0.550465384	0.534656537
12.04	11.6821	0.3579	0.54856381	0.547901056
12.75	12.2612	0.4888	0.527973921	0.618545149
13.65	12.8821	0.7679	0.436033285	0.754345124
14.48	13.5325	0.9475	0.356231412	0.825615734
15.19	14.1998	0.9902	0.336439211	0.840404536
16	14.8715	1.1285	0.272943047	0.882518893
16.9	15.5357	1.3643	0.175571477	0.935020797
17.59	16.1818	1.4082	0.159832327	0.94237977
18.02	16.8006	1.2194	0.233179685	0.905505099
18.28	17.3847	0.8953	0.380204737	0.806393171
18.52	17.9289	0.5911	0.500829877	0.671244353
18.78	18.4302	0.3498	0.549270469	0.543454781
19.08	18.8882	0.1918	0.549237173	0.456325518
19.31	19.3049	0.0051	0.516322137	0.356331282

19.51	19.6847	-0.1747	0.456692	0.26851414
19.73	20.0343	-0.3043	0.402256638	0.212779414
20.05	20.3618	-0.3118	0.398919314	0.209774995
20.49	20.6767	-0.1867	0.451967843	0.263062105
21.03	20.9891	0.0409	0.525191343	0.374977743
21.7	21.3089	0.3911	0.544959678	0.566056679
22.05	21.6454	0.4046	0.543172364	0.573401777
22.75	22.0065	0.7435	0.446038904	0.743583281
23.39	22.3985	0.9915	0.33583578	0.840841515
23.9	22.8257	1.0743	0.297556223	0.867060703
24.3	23.2905	1.0095	0.327481828	0.846811367
24.68	23.7931	0.8869	0.384022911	0.803183405
25.11	24.3325	0.7775	0.43202318	0.758511827
25.59	24.9064	0.6836	0.469307858	0.716158826
26.14	25.5118	0.6282	0.488903249	0.689607054
26.75	26.1454	0.6046	0.49660843	0.677977218
27.31	26.8041	0.5059	0.52406818	0.627540489
27.89	27.4852	0.4048	0.543144502	0.573510409
28.68	28.1867	0.4933	0.526971939	0.620918785
29.57	28.9069	0.6631	0.476792684	0.706460846
30.54	29.6447	0.8953	0.380204737	0.806393171
31.56	30.3995	1.1605	0.258692917	0.891024423
32.46	31.1704	1.2896	0.20425391	0.920848693
33.28	31.9566	1.3234	0.190990209	0.92752703
34.17	32.7566	1.4134	0.158025338	0.943206194
35.06	33.5688	1.4912	0.132482435	0.954488672
35.47	34.3906	1.0794	0.295219495	0.868572279
36.24	35.2193	1.0207	0.322287909	0.850450073
37	36.0516	0.9484	0.355815307	0.825936155
37.78	36.8839	0.8961	0.379840406	0.806697189
38.54	37.7125	0.8275	0.41055067	0.779580031
39.34	38.534	0.806	0.419896578	0.77065244
40.24	39.3452	0.8948	0.380432384	0.806203011
41.04	40.1432	0.8968	0.37952152	0.806962965
41.71	40.9259	0.7841	0.429243623	0.761354018
42.49	41.6917	0.7983	0.423203932	0.767406489
43.39	42.4401	0.9499	0.355121652	0.826469358
44.22	43.1711	1.0489	0.309242525	0.859354517
44.92	43.8859	1.0341	0.316081926	0.854727139
45.57	44.5862	0.9838	0.339409752	0.838241819
46.08	45.2746	0.8054	0.42015508	0.770400425
46.35	45.9539	0.3961	0.544319214	0.568779887
46.73	46.6274	0.1026	0.537724423	0.407787136
47.46	47.2983	0.1617	0.54625624	0.439835994
48.51	47.9692	0.5408	0.515288502	0.645680363

49.68	48.6426	1.0374	0.314555387	0.85576769
50.74	49.3199	1.4201	0.155715278	0.944257214
51.48	50.0018	1.4782	0.136553685	0.952740024
51.96	50.6883	1.2717	0.211459508	0.91712824
52.18	51.3785	0.8015	0.421832122	0.768758548
52.75	52.0711	0.6789	0.471046993	0.713948987
53.56	52.7645	0.7955	0.424401086	0.766219841
54.3	53.4571	0.8429	0.403764389	0.785850349
54.89	54.1475	0.7425	0.446442956	0.74313704
55.38	54.8349	0.5451	0.514133862	0.647893627
55.88	55.5188	0.3612	0.548256389	0.549710812
56.61	56.1995	0.4105	0.542333539	0.576604037
57.49	56.8779	0.6121	0.494203912	0.68169279
58.36	57.5552	0.8048	0.42041345	0.770148254
59.19	58.2328	0.9572	0.351743446	0.829049418
59.99	58.9124	1.0776	0.296043806	0.868040143
60.65	59.595	1.055	0.306429361	0.861232314
61.2	60.2818	0.9182	0.369733102	0.814980121
61.84	60.9728	0.8672	0.392919845	0.795530379
62.59	61.6677	0.9223	0.367849881	0.816492166
63.36	62.3657	0.9943	0.334536072	0.841780035
64.09	63.0651	1.0249	0.320341623	0.851799594
64.72	63.7642	0.9558	0.352391613	0.828556523
65.27	64.461	0.809	0.418602118	0.771910189
65.83	65.1537	0.6763	0.472003242	0.71272302
66.45	65.8408	0.6092	0.495138611	0.680258242
67.11	66.5216	0.5884	0.501657416	0.669890994
67.78	67.1959	0.5841	0.50296365	0.667731054
68.52	67.8643	0.6557	0.479428796	0.702922805
69.23	68.5282	0.7018	0.462448787	0.724638107
69.92	69.1895	0.7305	0.451252695	0.737750794
70.31	69.8505	0.4595	0.534037155	0.602984721
71.18	70.5135	0.6665	0.475569611	0.708079864
72.08	71.1809	0.8991	0.378473132	0.807834659
72.91	71.8544	1.0556	0.306152862	0.861416089
73.66	72.5353	1.1247	0.274650729	0.881478466
74.33	73.2239	1.1061	0.283051641	0.876291939
74.96	73.9201	1.0399	0.313399469	0.856552633
75.56	74.6228	0.9372	0.360988353	0.821922044
76.17	75.3303	0.8397	0.405180402	0.784556037
76.86	76.0405	0.8195	0.414046555	0.776281628
77.51	76.7512	0.7588	0.4397971	0.750360068
78.2	77.4602	0.7398	0.447531437	0.741930174
78.92	78.1657	0.7543	0.44164439	0.748376821
79.6	78.8663	0.7337	0.449977208	0.739192763

80.27	79.5613	0.7087	0.45979827	0.727819875
80.93	80.2508	0.6792	0.470936388	0.714090284
81.59	80.9353	0.6547	0.479782278	0.702443199
82.3	81.6161	0.6839	0.469196392	0.716299601
83.02	82.2947	0.7253	0.453314084	0.735398914
83.78	82.9726	0.8074	0.4192929	0.771239873
84.55	83.6511	0.8989	0.378564333	0.807758956
85.28	84.331	0.949	0.355537867	0.826149561
85.97	85.0127	0.9573	0.351697143	0.82908459
86.66	85.6953	0.9647	0.34826907	0.831674467
87.38	86.3774	1.0026	0.330683621	0.844540696
88.14	87.0568	1.0832	0.293480821	0.869690809
88.85	87.7302	1.1198	0.2768572	0.880127273
89.38	88.3942	0.9858	0.338481508	0.83891971
89.8	89.0447	0.7553	0.441234693	0.748818261
90.31	89.6769	0.6331	0.487253223	0.691998644
90.59	90.2858	0.3042	0.551967766	0.518338251
91.27	90.8657	0.4043	0.543214081	0.573238819
91.97	91.4102	0.5598	0.510069446	0.655421742
92.59	91.9118	0.6782	0.471304855	0.713619164
93.13	92.3622	0.7678	0.436074847	0.754301519
93.68	92.7518	0.9282	0.365136036	0.818654477
94.26	93.0705	1.1895	0.246004667	0.89834198
94.76	93.3079	1.4521	0.144967176	0.949066871
94.62	93.4535	1.1665	0.256049043	0.892568644
93.84	93.4981	0.3419	0.549893883	0.539113039
93.01	93.4344	-0.4244	0.347481116	0.167734146
92.27	93.2574	-0.9874	0.120980991	0.040672372
91.57	92.9661	-1.3961	0.038428524	0.010471714
90.83	92.5631	-1.7331	0.011729654	0.002753627
90.11	92.0553	-1.9453	0.004968497	0.001071486
89.46	91.4539	-1.9939	0.004031841	0.000853487
88.89	90.7739	-1.8839	0.006427275	0.001419626
88.43	90.0332	-1.6032	0.019017157	0.004717933
87.92	89.2518	-1.3318	0.047015149	0.013211264
87.33	88.4505	-1.1205	0.086257114	0.026972041
86.69	87.6494	-0.9594	0.129343873	0.044176055
86.07	86.867	-0.797	0.185025658	0.069540178
85.58	86.1188	-0.5388	0.294578685	0.131014862
85.16	85.4163	-0.2563	0.423203932	0.232593511
84.69	84.7666	-0.0766	0.492086461	0.315101749
84.2	84.1718	0.0282	0.522174617	0.368326816
83.62	83.6297	-0.0097	0.512331302	0.348719018
	Media	0.271885 66		

Desviación estándar	0.722017 295
------------------------	-----------------

ANEXO 4.
TABLA DE ERROR DE ROLL

Valor Real	Valor Medido	Error	Distribucion Normal	Distr. Nor. Acumulada	rango	frecuencia
10.9	12.4514	-1.5514	0.115235367	0.810639494	-11	5
10.69	13.8599	-3.1699	0.166677229	0.575945561	-10	2
10.5	15.1025	-4.6025	0.155555696	0.337942447	-9	2
10.3	16.125	-5.825	0.109311321	0.174045676	-8	2
10.1	16.8851	-6.7851	0.06853863	0.089014102	-7	11
9.99	17.3558	-7.3658	0.047658035	0.055472783	-6	7
9.78	17.5277	-7.7477	0.036299768	0.039504258	-5	19
9.56	17.4098	-7.8498	0.033600959	0.035937039	-4	31
9.33	17.0284	-7.6984	0.037654021	0.041327082	-3	76
9.14	16.4248	-7.2848	0.050319762	0.0594403	-2	61
8.97	15.6518	-6.6818	0.072648702	0.096305542	-1	12
8.79	14.7692	-5.9792	0.102563166	0.157709819	0	11
8.61	13.8395	-5.2295	0.13427122	0.246705527	1	7
8.45	12.9225	-4.4725	0.158951933	0.358389862	2	7
8.27	12.0711	-3.8011	0.169259324	0.469286384		0
8.07	11.3282	-3.2582	0.167762614	0.561178356		
7.87	10.7235	-2.8535	0.160968374	0.627852212		
7.67	10.2728	-2.6028	0.154581489	0.667438513		
7.49	9.9778	-2.4878	0.151160464	0.685021451		
7.3	9.8271	-2.5271	0.152361995	0.679057126		
7.12	9.7986	-2.6786	0.156673331	0.65564111		
6.92	9.8622	-2.9422	0.162846294	0.613489487		
6.71	9.9833	-3.2733	0.16792519	0.558643905		
6.47	10.1262	-3.6562	0.169742532	0.493854834		
6.3	10.257	-3.957	0.16802605	0.442985461		
6.04	10.3469	-4.3069	0.162663296	0.385029443		
5.71	10.3736	-4.6636	0.153821929	0.328490527		
5.37	10.3233	-4.9533	0.144524894	0.285234232		
5.05	10.19	-5.14	0.137719746	0.258877942		
4.75	9.9753	-5.2253	0.134435463	0.247269811		
4.48	9.6863	-5.2063	0.135175584	0.249831124		
4.25	9.3329	-5.0829	0.139859965	0.266803075		
4.03	8.9249	-4.8949	0.146531808	0.29373339		
3.77	8.4693	-4.6993	0.152770061	0.323017776		
3.47	7.9684	-4.4984	0.158308074	0.354281309		
3.15	7.4194	-4.2694	0.163402977	0.391143305		
2.81	6.8151	-4.0051	0.167498492	0.434915821		
2.44	6.1464	-3.7064	0.169647974	0.485335809		
2.05	5.4059	-3.3559	0.168694005	0.544740125		

1.64	4.591	-2.951	0.16302113	0.612055669
1.25	3.708	-2.458	0.15022769	0.68951218
0.91	2.7738	-1.8638	0.128401036	0.772564532
0.62	1.8173	-1.1973	0.099781892	0.848714623
0.36	0.878	-0.518	0.071037039	0.906582491
0.23	0.0033	0.2267	0.044464497	0.94917365
-0.03	-0.7556	0.7256	0.030711797	0.967784916
-0.27	-1.3492	1.0792	0.022990526	0.977231482
-0.49	-1.7356	1.2456	0.019905227	0.980795734
-0.71	-1.885	1.175	0.021173137	0.979346019
-0.94	-1.7846	0.8446	0.027930654	0.97127227
-1.17	-1.4408	0.2708	0.043111816	0.951104611
-1.41	-0.8796	-0.5304	0.071532551	0.905698561
-1.64	-0.1455	-1.4945	0.112772234	0.817126375
-1.9	0.7021	-2.6021	0.154561553	0.667546713
-2.15	1.5941	-3.7441	0.169526126	0.47894224
-2.4	2.4571	-4.8571	0.147796968	0.29929629
-2.66	3.2204	-5.8804	0.106890208	0.168056867
-2.94	3.822	-6.762	0.069448708	0.090607846
-3.23	4.2133	-7.4433	0.045192935	0.051875327
-3.52	4.3639	-7.8839	0.032731287	0.034806119
-3.79	4.2627	-8.0527	0.028657691	0.029630279
-4.02	3.919	-7.939	0.0313594	0.033040609
-4.14	3.3605	-7.5005	0.043425387	0.049341055
-4.36	2.63	-6.99	0.060713276	0.07578006
-4.59	1.7814	-6.3714	0.085541799	0.120837878
-4.81	0.8741	-5.6841	0.115429559	0.189879502
-5.04	-0.032	-5.008	0.142590177	0.277381399
-5.26	-0.882	-4.378	0.161157337	0.373516824
-5.49	-1.6302	-3.8598	0.168881127	0.459361449
-5.74	-2.2433	-3.4967	0.169529163	0.520922138
-5.98	-2.7027	-3.2773	0.167967121	0.55797212
-6.21	-3.0043	-3.2057	0.167144874	0.569970033
-6.42	-3.1586	-3.2614	0.167797633	0.56064146
-6.64	-3.1878	-3.4522	0.16933045	0.528461988
-6.88	-3.1232	-3.7568	0.169475276	0.476789576
-7.14	-3.0015	-4.1385	0.165680426	0.412687039
-7.4	-2.8613	-4.5387	0.157273428	0.347922209
-7.63	-2.7388	-4.8912	0.146656842	0.29427579
-7.86	-2.6657	-5.1943	0.135640559	0.251456022
-8.09	-2.6658	-5.4242	0.12642958	0.221319443
-8.33	-2.7539	-5.5761	0.120057489	0.202596403
-8.57	-2.9355	-5.6345	0.11756323	0.195657776
-8.68	-3.2067	-5.4733	0.124390553	0.21516172
-8.91	-3.5558	-5.3542	0.129296837	0.230270154

-9.13	-3.9646	-5.1654	0.136752312	0.255392127
-9.36	-4.4113	-4.9487	0.144685212	0.285899415
-9.58	-4.8725	-4.7075	0.152524502	0.321766067
-9.76	-5.3255	-4.4345	0.159866184	0.364447522
-9.92	-5.7503	-4.1697	0.165181246	0.407525525
-10.06	-6.1317	-3.9283	0.16830803	0.447811914
-10.18	-6.4597	-3.7203	0.169608118	0.482977972
-10.28	-6.7302	-3.5498	0.169686945	0.511915567
-10.38	-6.9448	-3.4352	0.169238579	0.531339838
-10.49	-7.1102	-3.3798	0.168878191	0.540706102
-10.62	-7.2371	-3.3829	0.168900816	0.540182545
-10.74	-7.3384	-3.4016	0.169031122	0.537022865
-10.86	-7.4286	-3.4314	0.169216838	0.531982903
-10.97	-7.5212	-3.4488	0.169312781	0.529037682
-11.09	-7.6285	-3.4615	0.16937698	0.526886997
-11.23	-7.7597	-3.4703	0.169418577	0.525396295
-11.42	-7.9208	-3.4992	0.16953853	0.520498303
-11.64	-8.114	-3.526	0.169626917	0.515953437
-11.89	-8.3378	-3.5522	0.169692033	0.511508313
-12.17	-8.5875	-3.5825	0.16974106	0.50636583
-12.31	-8.8558	-3.4542	0.16934068	0.528123317
-12.59	-9.1337	-3.4563	0.169351289	0.527767691
-12.85	-9.4114	-3.4386	0.169257658	0.530764394
-13.08	-9.6794	-3.4006	0.169024422	0.537191893
-13.32	-9.9292	-3.3908	0.168957158	0.538848005
-13.57	-10.1542	-3.4158	0.169122984	0.534621963
-13.79	-10.35	-3.44	0.169265411	0.530527428
-13.98	-10.515	-3.465	0.169393807	0.526294148
-14.16	-10.6501	-3.5099	0.169576458	0.518684035
-14.34	-10.7589	-3.5811	0.169739416	0.506603466
-14.53	-10.8469	-3.6831	0.169701486	0.489289262
-14.71	-10.9211	-3.7889	0.169324772	0.471351751
-14.91	-10.9896	-3.9204	0.168381324	0.449141838
-15.13	-11.0606	-4.0694	0.166686731	0.424171117
-15.36	-11.1419	-4.2181	0.164352512	0.399550543
-15.61	-11.2404	-4.3696	0.161342221	0.374871323
-15.83	-11.3612	-4.4688	0.159042549	0.358978152
-16.03	-11.5079	-4.5221	0.157704406	0.350536534
-16.22	-11.682	-4.538	0.157291737	0.348032306
-16.38	-11.883	-4.497	0.158343302	0.354502965
-16.52	-12.1086	-4.4114	0.160404021	0.368146669
-16.66	-12.3549	-4.3051	0.16269967	0.385322269
-16.79	-12.6168	-4.1732	0.165123526	0.406947492
-16.86	-12.8888	-3.9712	0.167877449	0.440600539
-17	-13.1647	-3.8353	0.169051699	0.463501164

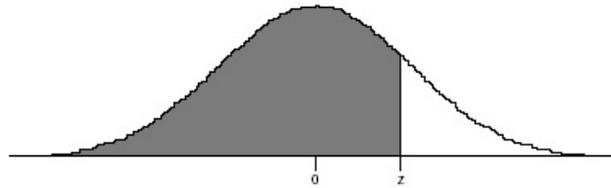
-17.16	-13.439	-3.721	0.169605955	0.482859247
-17.33	-13.7066	-3.6234	0.169762495	0.499422807
-17.49	-13.9636	-3.5264	0.169628069	0.515885586
-17.63	-14.2071	-3.4229	0.169166618	0.533421034
-17.75	-14.4356	-3.3144	0.16833329	0.55173362
-17.87	-14.6491	-3.2209	0.16733208	0.567428
-18.01	-14.8485	-3.1615	0.166562092	0.577345167
-18.18	-15.0357	-3.1443	0.166319954	0.580207965
-18.36	-15.213	-3.147	0.166358531	0.579758849
-18.55	-15.3834	-3.1666	0.166632241	0.576495521
-18.73	-15.5496	-3.1804	0.166818264	0.574194706
-18.89	-15.7139	-3.1761	0.166760895	0.574911902
-19.03	-15.8783	-3.1517	0.16642518	0.578976807
-19.16	-16.0439	-3.1161	0.16590449	0.584892383
-19.3	-16.2114	-3.0886	0.165477388	0.589448934
-19.46	-16.3805	-3.0795	0.165331313	0.590954115
-19.61	-16.5504	-3.0596	0.165003701	0.594240967
-19.79	-16.72	-3.07	0.165176312	0.592524028
-19.97	-16.8879	-3.0821	0.165373289	0.590524199
-20	-17.0527	-2.9473	0.162947875	0.612658712
-20.2	-17.2132	-2.9868	0.163710664	0.606207065
-20.32	-17.3686	-2.9514	0.163029027	0.611990459
-20.45	-17.5183	-2.9317	0.162634944	0.615198266
-20.56	-17.6619	-2.8981	0.161938739	0.620651188
-20.69	-17.7986	-2.8914	0.161796314	0.621735702
-20.82	-17.9257	-2.8943	0.161858107	0.621266403
-20.97	-18.0378	-2.9322	0.162645076	0.615116946
-21.12	-18.1253	-2.9947	0.163858094	0.604913167
-21.28	-18.1732	-3.1068	0.165762469	0.586434637
-21.43	-18.1614	-3.2686	0.167875313	0.559433036
-21.59	-18.0661	-3.5239	0.169620786	0.516309647
-21.75	-17.8626	-3.8874	0.168667218	0.454703229
-21.91	-17.5296	-4.3804	0.161104174	0.37313011
-22.07	-17.0533	-5.0167	0.142277752	0.276142223
-22.21	-16.4323	-5.7777	0.111372802	0.17926488
-22.33	-15.6812	-6.6488	0.073982819	0.098724935
-22.45	-14.8322	-7.6178	0.039939874	0.044453518
-22.58	-13.9348	-8.6452	0.017254285	0.016242836
-22.69	-13.0535	-9.6365	0.006404858	0.005230405
-22.8	-12.2619	-10.5381	0.002228234	0.001620668
-22.86	-11.637	-11.223	0.000905522	0.000607534
-22.98	-11.2509	-11.7291	0.000440785	0.000279587
-23.11	-11.1633	-11.9467	0.00031886	0.000197594
-23.23	-11.4148	-11.8152	0.000388176	0.000243944
-23.35	-12.0213	-11.3287	0.000782098	0.000518488

-23.46	-12.9716	-10.4884	0.002370842	0.001734923
-23.54	-14.2273	-9.3127	0.00902798	0.007708643
-23.62	-15.7251	-7.8949	0.03245412	0.034447601
-23.69	-17.3822	-6.3078	0.088263395	0.126364769
-23.75	-19.1032	-4.6468	0.154307106	0.33107882
-23.81	-20.7882	-3.0218	0.164350732	0.600465892
-23.86	-22.3414	-1.5186	0.113817146	0.814395968
-23.89	-23.6792	-0.2108	0.059269937	0.926571552
-23.94	-24.7367	0.7967	0.029027431	0.969908245
-24.01	-25.473	1.463	0.016365295	0.984728592
-24.08	-25.8731	1.7931	0.011958805	0.989373142
-24.16	-25.9486	1.7886	0.012011648	0.989319208
-24.23	-25.7349	1.5049	0.015743672	0.985401209
-24.33	-25.2875	0.9575	0.025464825	0.97428488
-24.43	-24.6756	0.2456	0.043881548	0.950008513
-24.4	-23.9759	-0.4241	0.067333595	0.913078305
-24.56	-23.2653	-1.2947	0.104048467	0.838788114
-24.66	-22.6144	-2.0456	0.135636692	0.748557541
-24.77	-22.0821	-2.6879	0.156920679	0.654182897
-24.9	-21.7115	-3.1885	0.166924867	0.572843045
-25.05	-21.5275	-3.5225	0.169616623	0.516547113
-25.19	-21.5362	-3.6538	0.169745114	0.494262219
-25.34	-21.7268	-3.6132	0.169761962	0.501154385
-25.48	-22.0735	-3.4065	0.169063513	0.536194533
-25.62	-22.5402	-3.0798	0.165336166	0.590904515
-25.75	-23.0845	-2.6655	0.156321424	0.657691231
-25.87	-23.6624	-2.2076	0.141710767	0.726087092
-25.95	-24.2324	-1.7176	0.122330996	0.790895374
-26.05	-24.7586	-1.2914	0.10390389	0.839131235
-26.16	-25.2136	-0.9464	0.088873886	0.872377558
-26.28	-25.5793	-0.7007	0.07847674	0.892928812
-26.41	-25.8473	-0.5627	0.072830047	0.903367131
-26.55	-26.0181	-0.5319	0.071592591	0.905591217
-26.71	-26.0999	-0.6101	0.07475127	0.899869533
-26.87	-26.1066	-0.7634	0.081092524	0.887926467
-27.02	-26.0558	-0.9642	0.089640493	0.870788782
-27.09	-25.9665	-1.1235	0.096555486	0.855959416
-27.24	-25.8577	-1.3823	0.10788295	0.829505456
-27.38	-25.7463	-1.6337	0.11877027	0.801009911
-27.53	-25.6465	-1.8835	0.129203423	0.770027121
-27.7	-25.5689	-2.1311	0.138891509	0.736820716
-27.88	-25.5205	-2.3595	0.147016977	0.704153054
-28.04	-25.5046	-2.5354	0.152611513	0.677791485
-28.18	-25.5215	-2.6585	0.156131717	0.658784817
-28.31	-25.5688	-2.7412	0.158298	0.645782012

-28.44	-25.6428	-2.7972	0.159669608	0.636878553
-28.55	-25.7383	-2.8117	0.160011879	0.634560856
-28.66	-25.8502	-2.8098	0.159967334	0.634864836
-28.78	-25.9735	-2.8065	0.159889749	0.6353926
-28.9	-26.1039	-2.7961	0.159643424	0.637054175
-29.01	-26.2382	-2.7718	0.159057211	0.640926418
-29.12	-26.374	-2.746	0.158418628	0.645021892
-29.21	-26.5101	-2.6999	0.157236776	0.652297948
-29.3	-26.6461	-2.6539	0.156006424	0.659502735
-29.4	-26.7823	-2.6177	0.155003199	0.665132101
-29.51	-26.9195	-2.5905	0.154229559	0.669337705
-29.65	-27.0583	-2.5917	0.154264045	0.669152609
-29.81	-27.1997	-2.6103	0.154794393	0.666278353
-29.9	-27.3441	-2.5559	0.153221356	0.674656683
-30.08	-27.4916	-2.5884	0.154169132	0.669661523
-30.24	-27.6418	-2.5982	0.154450275	0.668149287
-30.4	-27.794	-2.606	0.154672485	0.666943707
-30.54	-27.9469	-2.5931	0.154304237	0.668936611
-30.66	-28.0993	-2.5607	0.153362813	0.67392088
-30.77	-28.2495	-2.5205	0.152162519	0.680062058
-30.9	-28.3959	-2.5041	0.151662802	0.682553433
-31.03	-28.5372	-2.4928	0.151315152	0.684265261
-31.17	-28.6725	-2.4975	0.151460079	0.683553739
-31.32	-28.801	-2.519	0.152117053	0.680290267
-31.46	-28.9226	-2.5374	0.152671415	0.677486202
-31.57	-29.0377	-2.5323	0.152518494	0.678264437
-31.65	-29.1473	-2.5027	0.151619878	0.682765731
-31.74	-29.2526	-2.4874	0.151148066	0.685081912
-31.86	-29.3553	-2.5047	0.151681186	0.68246243
-32.01	-29.4571	-2.5529	0.153132686	0.675116214
-32.2	-29.56	-2.64	0.15562481	0.661668577
-32.4	-29.6658	-2.7342	0.158121066	0.64688948
-32.6	-29.7759	-2.8241	0.16030032	0.632574916
-32.81	-29.8915	-2.9185	0.162365037	0.617343271
		-		
	Media	3.642723		
	Desviación estándar	715		
		2.344107		
		74		

ANEXO 5

TABLA PARA EL CÁLCULO DE PROBABILIDAD MEDIANTE LA DISTRIBUCIÓN NORMAL [33]



Normal Deviate z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
-4.0	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
-3.9	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
-3.8	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
-3.7	.0001	.0001	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
-3.6	.0002	.0002	.0001	.0001	.0001	.0001	.0001	.0001	.0001	.0001
-3.5	.0002	.0002	.0002	.0002	.0002	.0002	.0002	.0002	.0002	.0002
-3.4	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0002
-3.3	.0005	.0005	.0005	.0004	.0004	.0004	.0004	.0004	.0004	.0003
-3.2	.0007	.0007	.0006	.0006	.0006	.0006	.0006	.0005	.0005	.0005
-3.1	.0010	.0009	.0009	.0009	.0008	.0008	.0008	.0008	.0007	.0007
-3.0	.0013	.0013	.0013	.0012	.0012	.0011	.0011	.0011	.0010	.0010
-2.9	.0019	.0018	.0018	.0017	.0016	.0016	.0015	.0015	.0014	.0014
-2.8	.0026	.0025	.0024	.0023	.0023	.0022	.0021	.0021	.0020	.0019
-2.7	.0035	.0034	.0033	.0032	.0031	.0030	.0029	.0028	.0027	.0026
-2.6	.0047	.0045	.0044	.0043	.0041	.0040	.0039	.0038	.0037	.0036
-2.5	.0062	.0060	.0059	.0057	.0055	.0054	.0052	.0051	.0049	.0048
-2.4	.0082	.0080	.0078	.0075	.0073	.0071	.0069	.0068	.0066	.0064
-2.3	.0107	.0104	.0102	.0099	.0096	.0094	.0091	.0089	.0087	.0084
-2.2	.0139	.0136	.0132	.0129	.0125	.0122	.0119	.0116	.0113	.0110
-2.1	.0179	.0174	.0170	.0166	.0162	.0158	.0154	.0150	.0146	.0143
-2.0	.0228	.0222	.0217	.0212	.0207	.0202	.0197	.0192	.0188	.0183
-1.9	.0287	.0281	.0274	.0268	.0262	.0256	.0250	.0244	.0239	.0233
-1.8	.0359	.0351	.0344	.0336	.0329	.0322	.0314	.0307	.0301	.0294
-1.7	.0446	.0436	.0427	.0418	.0409	.0401	.0392	.0384	.0375	.0367
-1.6	.0548	.0537	.0526	.0516	.0505	.0495	.0485	.0475	.0465	.0455
-1.5	.0668	.0655	.0643	.0630	.0618	.0606	.0594	.0582	.0571	.0559
-1.4	.0808	.0793	.0778	.0764	.0749	.0735	.0721	.0708	.0694	.0681
-1.3	.0968	.0951	.0934	.0918	.0901	.0885	.0869	.0853	.0838	.0823
-1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.0985
-1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
-1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
-.9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
-.8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
-.7	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
-.6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
-.5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
-.4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
-.3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483

ANEXO 6

MANUAL DE FUNCIONAMIENTO

PULSANTE 1: Es utilizado para resetear el sistema

PULSANTE 2: Setea los giroscopios

SW: Graba los datos en la memoria SD

LED 1: Indica cuando el pulsante de reset ha sido presionado.

LED 2: Indica cuando el pulsante de seteo ha sido presionado.

LED 3: Indica cuando el Switch ha sido presionado para empezar a grabar.

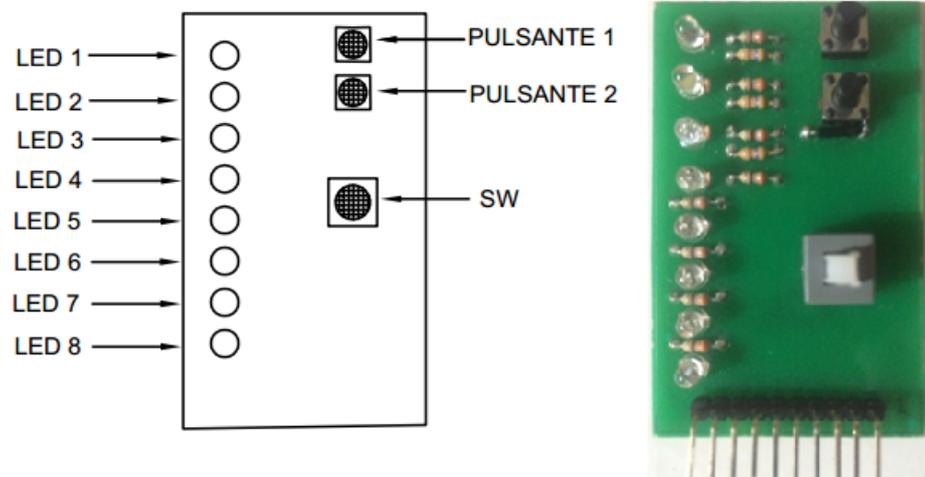
LED 4: Indica cuando el Integrado maestro está funcionando correctamente.

LED 5: Indica cuando los giroscopios de la rodilla y tobillo de la pierna derecha están funcionando correctamente.

LED 6: Indica cuando los giroscopios de las caderas están funcionando correctamente.

LED 7: Indica cuando los giroscopios de la rodilla y tobillo de la pierna izquierda están funcionando correctamente.

LED 8: Indica si la grabación de datos se está ejecutando con normalidad.



BIBLIOGRAFÍA

- [1] LOSADA BAREDECI, P., "Alteraciones de los miembros inferiores: deformidades angulares, torsionales, alteraciones de la marcha y disimetrías", Zaragoza, 2002.
- [2] WHITTLE, Michael, W. *Gait Analysis an Introduction*, Fourth Edition, Editorial Elseiver, USA, 2007.
- [3] NORDIN, Margareta y FRANKEL, Victor, H., *Biomecánica Básica del Sistema Musculoesquelético*, Tercera Edición, McGraw Hill, Madrid-España, 2004.
- [4] Archery Australia Inc Coaching and Standards Committee, *Shooting Technique Biomechanics*, 3 de septiembre de 2007.
- [5] "Applying Biomechanics to Sport", Chapter 3.
- [6] SAUCEDO ROMERO, Miguel Ángel, *Valoración de la Marcha Humana*, Universidad Nacional Autónoma de México, División de Ingeniería Mecánica e Industrial, México D.F, 2009.
- [7] BARTLETT, Roger, *Introduction to Sport Biomechanics*, Second Edition, Routledge, 2007.
- [8] KAPANDJI, A. I., *Fisiología Articular*, Quinta Edición, Editorial Panamericana, Volumen 2.
- [9] TAO, Weijun, LIU, Tao, ZHENG, Rencheng, FENG, Hutian, *Gait Analysis Using Wearable Sensors*, 2012.
- [10] RODRIGUEZ MARTIN, Daniel M., *Contribución al Análisis del Movimiento Humano Aplicado a la Identificación de Posturas y Bloqueos de la Marcha en Pacientes con Parkinson*, Universidad Politécnica de Catalunya, España, 2014.
- [11] AN, Qi, et al, *Evaluation of Wearable Gyroscope and Accelerometer Sensor (PocketIMU2) during Walking and Sit-to-Stand Motions*.
- [12] LATTES, M. Fernanda, et al, *Assessment of Micro Sensor (SM-MAE) in Monitoring of Cycling*, Universidad Estatal de Campinas (UNICAMP), Brazil, 2013.
- [13] DADASHI P. et al, *Inertial Measurement unit and biomechanical analysis of swimming: an update*, 2013.
- [14] DADASHI, F. et al, "Frontcrawl Propulsive Phase Detection Using Inertial Sensors", *Portuguese Journal of Sport Sciences*, Porto-Portugal, 2011.
- [15] LEE, J. B. et al, *Inertial Sensors 3D and 2D Assessment of Stroke Phases in Freestyle Swimming*, Procedia Engineering, 2011.

- [16] SHI, Guangyi, et al, “Mobile Human System for Fall Protection Using MEMS Sensors and Embedded SVM Classifier”, *IEEE sensors journal*, No 5, Mayo, 2009.
- [17] XSENS, MVN Biomech Awinda, 3D Kinematic Measurement System with Wireless Trackers.
- [18] ROETENBERG, Daniel, et al, *Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors*, Xsens Technologies, 2013.
- [19] RODRÍGUEZ, Ana Cristina. Gerente de Casa Ortopédica. “Entrevista personal”. 15 de diciembre de 2014.
- [20] INVENSENSE, *MPU-6000 and MPU-6050 Product Specification*, Revisión 3.4, Invensense Inc., California, EEUU, 19 de agosto de 2013, 52 paginas.
- [21] WIKIPEDIA, Ángulos de Navegación, Fecha de consulta 3 de junio de 2014, Disponible en:
http://es.wikipedia.org/wiki/Ángulos_de_navegación
- [22] HIBBELER, R. C., Mecánica Para Ingenieros, *Dinámica*, Tercera Edición, Editorial Continental, México, 2006.
- [23] WOLFRAM MATHWORLD, Euler Angles, Fecha de consulta 27 de abril de 2014, Disponible en:
<http://mathworld.wolfram.com/EulerAngles.html>
- [24] CPROGRAMMING, Using Quaternion to Perform 3D Rotations, Fecha de consulta 2 de mayo de 2014, Disponible en:
<http://www.cprogramming.com/tutorial/3d/quaternions.html>
- [25] EUCLIDEAN SPACE, Maths-Transformations Using Quaternions, Fecha de consulta 2 de mayo de 2014, Disponible en:
<http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/forms/index.htm>
- [26] NASA, *Euler Angles, Quaternions, and Transformation Matrices*, Julio, 1977.
- [27] GAVILAN, María Elena y MUÑOZ, José Daniel, “Simulación por Dinámica Molecular del Movimiento de un Trompo Pesado”, *Revista colombiana de física*, No. 1, Bogotá, 5 de abril de 2006.
- [28] ROWBERG, Jeff, Librerías MPU 6050, I2C, fecha de consulta 19 marzo de 2014 disponible en:
<https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>
- [29] ARDUINO, MPU-6050 Accelerometer + Gyro, fecha de consulta 25 de noviembre de 2014, disponible en:
<http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>

[30] ARDUINO, Libraries, fecha de consulta 2 de abril de marzo de 2014, disponible en:
<http://arduino.cc/en/pmwiki.php?n=Reference/Libraries>

[31] ARDUINO, Define, fecha de consulta 2 de abril de marzo de 2014, disponible en:
<http://arduino.cc/en/Reference/Define>

[32] ARDUINO, Volatile keyword, fecha de consulta 2 de abril de marzo de 2014, disponible en:
<http://arduino.cc/en/pmwiki.php?n=Reference/Volatile>

[33] DEPOOL RIVERO, Ramón y MONASTERIO, Dióscoro, *Probabilidad y Estadística. Aplicaciones a la Ingeniería*, Universidad Nacional Experimental Politécnica Antonio José de Sucre, 2013.