



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**INTRUMENTACIÓN DE UN ROBOT BÍPEDO
DE 12 GDL: SENSORES DE POSICIÓN, PRESIÓN E
INGERCIAL**

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECATRÓNICO

PRESENTA:

ERNESTO VILLALOBOS GUERRERO

Director de tesis:

Dr. Edmundo Gabriel Rocha Cózatl



CIUDAD UNIVERSITARIA, MÉXICO, D.F.

2013

Instrumentación de un Robot Bípedo de 12 GDL: Sensores de Posición, Presión e Inercial

Ernesto Villalobos Guerrero

miércoles, 15 de mayo de 2013

Agradecimientos

A mis padres por el apoyo que me han dado durante toda mi vida, pero principalmente a Josefina Guerrero Chávez por todo su amor y compañía

A mis hermanos Victoria y Fernando por su inmensa alegría y cariño, por tantos momentos juntos

A Edmundo y Octavio quien me brindaron la oportunidad de continuar en este proyecto

A mis amigos Víctor, Pedro, Daniela y Javier que han estado a mi lado compartiendo

A Konny y Rafael que me ayudaron a terminar mi trabajo

Esta trabajo se lo dedico a Pepe y Chaya

Índice general

1. Introducción	5
2. Robots bípedos y criterios de estabilidad	11
2.1. Escenario Histórico	11
2.2. Escenario Actual	13
2.3. Criterios importantes sobre estabilidad en marcha bípeda	14
2.3.1. Ground Projection of the Center of Mass (GCOM)	15
2.3.2. Zero Moment Point (ZMP)	16
2.3.3. Foot Rotation Indicator (FRI)	17
2.3.4. Diferencia entre ZMP y CoP	18
2.3.5. Zero Moment Point (ZMP) y Fictitious Zero Moment Point (FZMP)	19
3. Sensores relevantes para una marcha bípeda	21
3.1. Receptores naturales	21
3.1.1. Oído interno	22
3.1.2. Ojos	22
3.1.3. Tacto	23
3.2. Sensores artificiales	24
3.2.1. Sensores inerciales	24
3.2.1.1. Acelerómetros	24
3.2.1.2. Principio de funcionamiento de un acelerómetro	26
3.2.1.3. Giroscopio	27
3.2.2. Sensores de fuerza	29
3.2.3. Sensores de posición	30
3.2.3.1. Encoders	30
3.2.3.2. Potenciómetro angular	31
3.2.3.3. Sensor ultrasónico	31
3.3. Ejemplo de instrumentación bípeda: Robot Nao de Aldebaran	31
3.3.1. Sensado de Nao	32
3.3.1.1. Sensor inercial usado en el proyecto Nao	32
3.3.1.2. Nao y sus sensores de fuerza	32
3.3.1.3. Las articulaciones de Nao y sus sensores de posición	34
4. Sensores utilizados en el robot bípedo Scout	35
4.1. Sensores resistivos de fuerza	35
4.1.1. Propiedades y descripción de los sensores Flexiforce®	35
4.1.2. Caracterización de los sensores Flexiforce®	36
4.1.3. Medición del ZMP a partir de los sensores de fuerza	38
4.2. El potenciómetro como sensor de posición	40
4.2.1. Descripción de los servomotores	42

<i>ÍNDICE GENERAL</i>	4
4.2.2. Obtención de la posición angular de los servomotores	42
4.3. Sensor Inercial	42
4.3.1. Implementación y programación del sensor inercial	45
4.3.1.1. Filtro de Kalman	45
4.3.2. Tarea que desempeña el sensor inercial	46
5. Pruebas en cada sensor y evaluación de resultados	48
5.1. Resultados de las juntas rotacionales	48
5.1.1. Especificación de la prueba realizada	48
5.1.2. Gráficas de las juntas rotacionales	48
5.1.2.1. Resultados de experimento que resalta aquellos motores con mayor actividad de carga	49
5.2. Resultados del ZMP calculado a través de los sensores de fuerza.	60
5.2.1. Especificación de la prueba realizada	60
5.2.2. Restricciones de la prueba realizada	60
5.2.3. Gráficas del ZMP	61
5.3. Resultados del sensor inercial	64
5.3.1. Descripción de la prueba	64
5.3.2. Gráficas del sensor ArduIMU	64
6. Conclusiones y Trabajo a futuro	68
6.1. Conclusiones	68
6.2. Trabajo a futuro	69
Bibliografía	70
Apéndice A	72
A. Comportamiento de los potenciómetros y sensores de presión	72
A.1. Funciones que describen el comportamiento de las curvas obtenidas a partir de la caracterización de los sensores de presión	72
A.2. Funciones que describen el comportamiento de las curvas obtenidas a partir de la caracterización de los potenciómetros de los servomotores	76
Apéndice B	83
B. Programas utilizados para la instrumentación	83
B.1. Programa para sensor ArduIMU V2. Controlador PID y filtro de Kalman	83
B.1.1. Programa principal	83
B.1.2. Filtro de Kalman	86
B.1.2.1. Kalman.h	86
B.1.2.2. Kalman	86
B.1.3. Control PID	92
B.2. Programa en Matlab para cálculo de ZMP	93
B.3. Programa para obtener los ángulos rotacionales de los potenciómetros	97

Capítulo 1

Introducción

Interés por la Locomoción Bípeda

El interés que ha provocado la locomoción, tanto animal como humana, ha causado que éste sea un tema recurrente de investigación durante años. Se puede apreciar en el trabajo realizado por Eadweard Muybridge y su interés por la locomoción humana y animal [1].

Si observamos el trabajo del fotógrafo y todas las investigaciones acerca de la robótica bípeda, podemos argüir entonces, que el interés de la locomoción se basa tanto en la necesidad del hombre para comprender su movilidad como en el menester de construir máquinas que le sirvan de apoyo.

Marc Raibert, profesor del Massachusetts Institute of Technology (MIT) y fundador del CMU Leg Lab, señala que existen dos razones importantes que despiertan el interés por la locomoción con piernas. Siendo una de ellas la enorme movilidad que presenta una máquina con piernas y la segunda, el hecho de que construir una máquina con piernas nos ayuda a entender la movilidad humana y animal.

Desde un punto de vista comercial, la capacidad y ventajas que ofrecen los robots que utilizan piernas para desplazarse es muy atractivo, si se compara su movilidad contra la de un vehículo de ruedas o la de un tractor oruga. Los robots con piernas, ya sea que cuenten con 8 o sólo dos, tienen una movilidad más versátil y no se ven afectados por terrenos imperfectos u obstáculos menores.

Las máquinas construidas por el hombre generalmente tienen una base natural; esto quiere decir que muchas veces el hombre toma en cuenta la manera en cómo esta estructurada la naturaleza para poder partir de ahí. De esta forma, diseñadores de robots han encontrado inspiración en la biodiversidad terrestre.

Una manera sencilla de generar una clasificación de los robots con piernas se basa en el número de piernas que tienen. Éstas pueden ser descritas de la siguiente manera [2]:

- Octópodos: Sistemas con ocho piernas, como los arácnidos y crustáceos
- Hexápodos: Máquinas con 6 piernas basadas principalmente en insectos
- Cuadrúpedos: Robots con cuatro piernas, como la mayoría de los mamíferos
- Bípedos: Robots que cuentan con dos piernas simétricas, como los humanos, canguros y pájaros. En el caso particular de un bípedo que tiene la estructura similar a la de un humano, se mueve de la misma manera y trata de imitar su comportamiento, se le conoce como humanoide.
- Monópodos. Sistemas que cuentan solamente con una pierna. No existen, por naturaleza sistemas biológicos con una sola pierna; sin embargo, pueden ser representados por casos de discapacidad humana o animal. Éstos robots han sido estudiados con propósitos teóricos. Una ventaja de éstos sistemas es que permiten concentrar una variedad de características conceptuales de la locomoción bípeda. Sin la necesidad de interacción con múltiples piernas.

La razón por la que investigadores han tomado tanto interés en los robots, es debido a sus amplias posibilidades de aplicación. Una aplicación muy común es el uso de robots para realizar tareas de mantenimiento o inspección en entornos agresivos para el ser humano.

Dentro de la industria nuclear y química donde la inspección y tareas de reparación en zonas contaminadas son muy comunes, son algunos ejemplos. También entran en la industria del entretenimiento que resulta ser muy explotada debido a las bajas exigencias de estabilidad y confiabilidad que requieren estos robots. Ya que no es lo mismo manipular un pelota de ping pong que una herramienta dentro de un sector nuclear.

Debemos destacar además, la importancia que tiene el desarrollo de robots bípedos. La finalidad de un robot es interactuar con humanos o desarrollar tareas ejecutadas normalmente por humanos, esto nos obliga a elaborar un sistema capaz de moverse dentro de un ambiente común para los seres humanos, y la movilidad bípeda es potencialmente útil en éste sentido.

Al igual que los robots hexápodos o con más piernas, los robots bípedos se han utilizado dentro de la industria de entretenimiento así como en la de mantenimiento o servicial; sin embargo, existe una ventaja estructural muy importante que hace sobresalir a los robot bípedos como tema central de investigación y es que además de todas las aplicaciones mencionadas los robots bípedos nos brindan el apoyo necesario para estudiar y desarrollar temas de morfología humana.

Proyecto Robot Bípedo

Trabajo Previo

Ésta tesis forma parte de un proyecto desarrollado dentro de la Facultad de Ingeniería, UNAM.

Un grupo pequeño conformado por investigadores, alumnos de posgrado y de licenciatura han desarrollado lo necesario para llevar a cabo una marcha dinámicamente estable de un robot bípedo de doce grados de libertad internos.

El proyecto comenzó junto con el desarrollo de la tesis “Modelos cinemático y dinámico de un robot bípedo de doce grados de libertad internos” [3] utilizando un robot comercial como banco de pruebas, el robot bípedo Scout de Lynxmotion®. En el trabajo mencionado se desarrollaron los modelos cinemáticos y dinámicos del robot bípedo; se realizó un programa en el software Mathematica® que determina las coordenadas articulares del robot calculadas a través de la cinemática inversa; el cálculo del *Zero Moment Point* (ZMP); la simulación de una primera trayectoria y finalmente se acondicionó al robot para dar inicios a las primeras pruebas.

El modelo gráfico del robot diseñado en Solid Edge® es presentado en la figura 1.1 y su simplificación en la figura 1.2. La representación computarizada del modelo ayudó a comprender mejor la estructura del robot gracias a que los giros y manipulación dentro de la simulación facilitan el trabajo en la elaboración de trayectorias y sirve de referencia para las pruebas físicas, tomando en cuenta que el estado ideal (representación computarizada) no presenta ningún tipo de fuerzas y su desplazamiento es mucho más libre. La nomenclatura señala al eslabón con la letra B como el torso del robot y los eslabones de cada pierna con las etiquetas ni ; donde $1 \leq n \leq 6$ y la letra i corresponde $i=1$ a los eslabones de la pierna izquierda e $i=2$ a los de la pierna derecha.

Las juntas rotacionales mecánicamente representadas por servomotores están de color rojo mientras que los puntos de apoyo (pies) de color morado.

El segundo trabajo de interés realizado en torno al proyecto se presenta en la tesis “Planificación y optimización de la caminata de un robot bípedo” [4]. El trabajo presenta una interfaz gráfica de usuario que facilita la interacción con el robot bípedo Scout; se llevó a cabo una planificación de marcha y finalmente se optimiza dicha marcha a través de un algoritmo genético que determina la trayectoria óptima de la cadera dadas las trayectorias que definen la posición espacial y orientación de ambos pies durante un ciclo de marcha.

Alcances

El alcance primordial del proyecto es hacer caminar de manera estable un robot bípedo de 12 GDL y durante este proceso aprender todo lo relevante al estudio de robots bípedos, esto implica analizar numerosos temas. La relevancia que tiene la investigación permite al proyecto desarrollarse en diversas etapas.

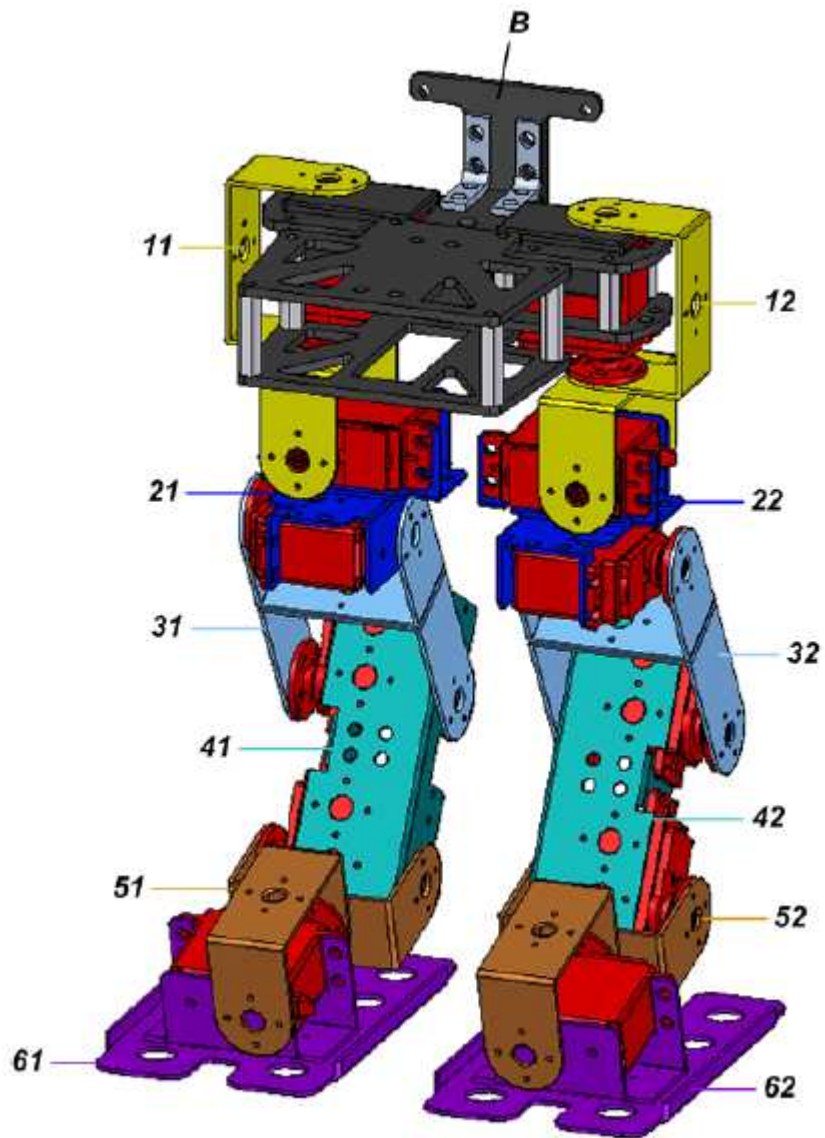


Figura 1.1: Diseño computarizado del robot bípedo Scout elaborado en Solid Edge. Figura tomada de [3]

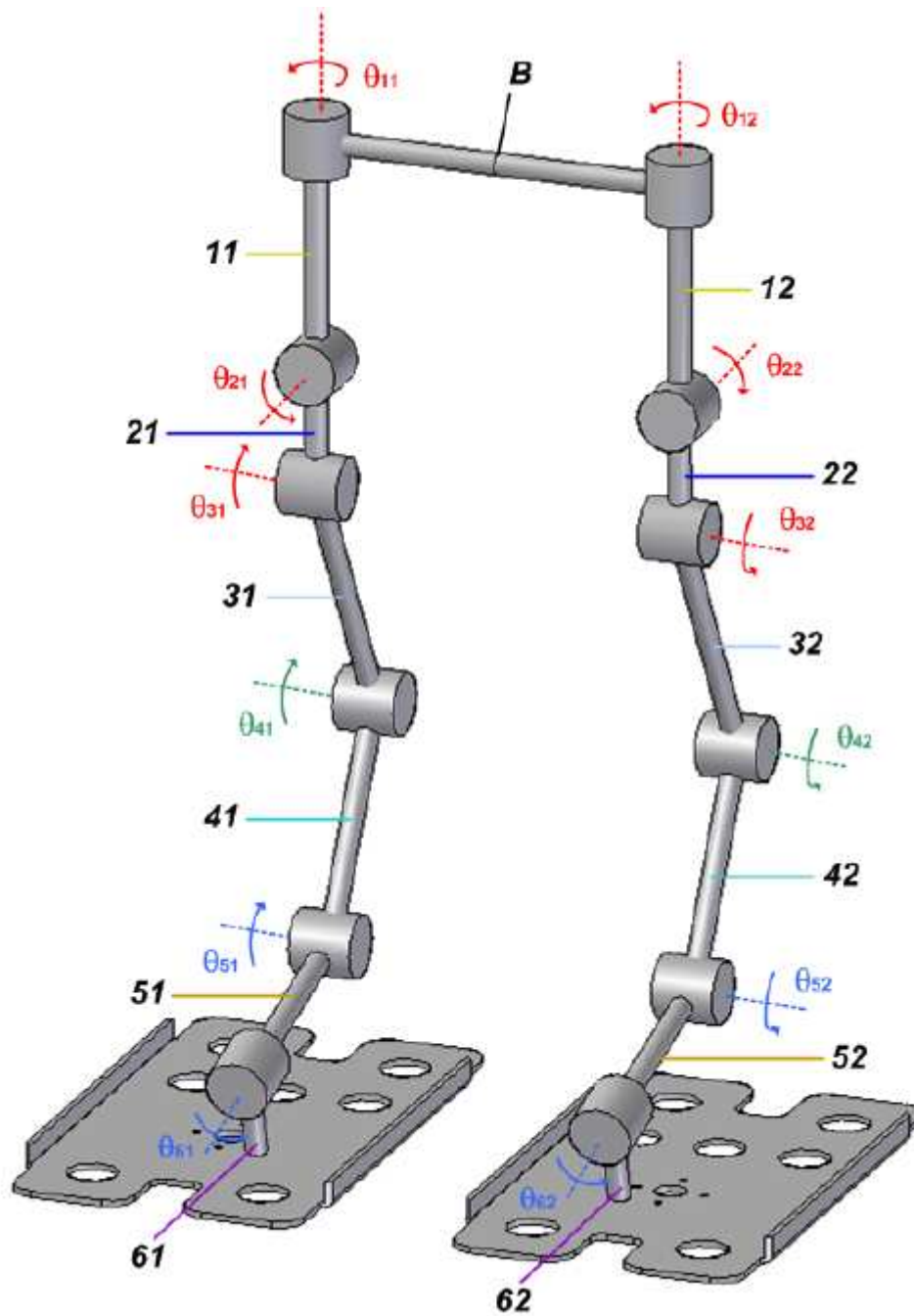


Figura 1.2: Diseño simplificado del robot bípido Scout. El giro de los servomotores esta representado mediante los ángulos θ_n marcados al extremo de los segmentos cilíndricos. Figura tomada de [3]

- Estudio de caminata bípeda
- Modelado cinemático y dinámico de sistemas
- Planificación y optimización de trayectorias
- Instrumentación sensorial
- Control
- Aplicaciones de la robótica bípeda

El último punto remarca el alcance final como conjunto de todo este estudio. Dicha aplicación esta centrada en la rehabilitación de pacientes que han sufrido lesiones de la medula espinal, traumatismos cráneo-encefálicos, apoplejías, pacientes con síndrome de dawn o que padecen esclerosis múltiple, parálisis cerebral o cualquier tipo de daño o síndrome que afecte su movilidad requiere de un número conveniente de personal y mucho esfuerzo continuo, especialmente con paciente con sobrepeso o espásticos.

Una ventaja que ofrece la rehabilitación automatizada es reducir el esfuerzo físico de los terapeutas, prolongar las sesiones de rehabilitación y hacerlas más eficientes. Tomando en cuenta la plasticidad del cerebro humano y las repeticiones continuas, las terapias robotizadas logran cumplir las expectativas de una rehabilitación motora funcional.

Debido a los alcances que ha tenido la medición y el control automático dentro de la mecatrónica es posible usar la robótica bípeda como apoyo para rehabilitación motora. Las aplicaciones existentes dentro de la rama de la medicina han impulsado a investigadores a tratar de mejorar las capacidades de muchos robots tanto de estabilidad como de balanceo.

Ésto es lo que se busca con el desarrollo de las teorías de control aplicadas al robot bípedo Scout. Garantizar su equilibrio dinámico durante la marcha para llevar el conjunto de conocimientos adheridos al proyecto a una implementación de rehabilitación motora en humanos.

Objetivos de la Tesis

Los resultados obtenidos hasta ahora en torno al proyecto han generado grandes avances. El robot bípedo es capaz de realizar una caminata estable y natural. Sin embargo, es necesario que el robot cuente con sensores para seguir avanzando en el estudio de la robótica bípeda, proporcionando bases para resolver problemas de control, estabilidad, ajustes en tiempo real, entre otros. El objetivo principal de esta tesis se enfoca en mejorar la marcha bípeda a través del sensado y la retroalimentación de señales obtenidas del robot Scout, así como el análisis de los resultados obtenidos. Para lograrlo se han planteado los siguientes objetivos específicos:

1. Hacer un análisis de las doce juntas rotacionales (servomotores) del robot a partir de la medición de la señal de posición de cada una de ellas. El análisis consistirá en una comparación de la trayectoria ideal contra la trayectoria desarrollada por el robot bípedo. La señal que se obtenga de los servomotores será acondicionada y se aplicará una función que sea capaz de codificar o transformar las señales eléctricas de cada motor en ángulos.
2. Medir el ZMP del robot a través de sensores de fuerza en las plantas de los pies. La medición se hará a través de un acoplamiento mecánico-eléctrico de los sensores de fuerza al robot. El ZMP se calculará a través de ecuaciones que ofrecerán el punto experimental y no analítico de la trayectoria del bípedo.
3. Instrumentar un sensor inercial que permita sentar las bases para el diseño de controladores. Se programará un sensor ArduIMU para controlar las juntas rotacionales 51 y 52 del robot bípedo con la finalidad de compensar el ángulo de un plano cuya pendiente variará a modo de prueba en el plano sagital a través de un PID básico retroalimentado por la señal de un sensor inercial.

Organización de la Tesis

En el segundo capítulo se desarrolla un marco teórico tanto de escenarios históricos como actuales y se presentan los criterios que se consideran más importantes para el desarrollo de una marcha bípeda.

El tercer capítulo presenta una reseña de los receptores sensoriales tanto naturales como artificiales. La finalidad de éste capítulo es remarcar la importancia de usar sensores en un robot, comparando la manera en cómo los humanos perciben el mundo y lo que necesitamos para desplazarnos. Así también, se ofrece una breve explicación del funcionamiento básico de los sensores. Al final se presenta un robot comercial que incorpora los mismos tipos de sensores utilizados en éste trabajo, con la finalidad de ejemplificar la instrumentación sensorial en un bípedo.

En el cuarto capítulo se presenta el desarrollo teórico y práctico de la instrumentación desarrollada durante la tesis. Se intenta introducir al lector a la instrumentación empleada mostrando características importantes de cada sensor implementado, y la teoría con la que se basa su trabajo.

En el capítulo quinto se ofrecen los resultados obtenidos durante éste trabajo. Ya que es un trabajo práctico de instrumentación y control, se presentan las gráficas obtenidas de cada sensor y algunas discusiones desarrolladas en torno a las mismas.

El capítulo sexto presenta las conclusiones del trabajo y da a conocer algunas propuestas para la continuación del proyecto.

Capítulo 2

Robots bípedos y criterios de estabilidad

En este capítulo se presentarán algunos robots bípedos que destacan por sus aportaciones en el estudio de la robótica. Así como algunos criterios importantes, usados para asegurar una marcha estable. Se estudian y ejemplifican algunos conceptos importantes con el fin de justificar el término usado en los próximos capítulos.

No se pretende profundizar demasiado en la definición y aplicación tanto de los robots presentados como de los criterios usados para alcanzar una marcha bípeda natural y robusta; ya que éste trabajo se limita a su instrumentación como fin de medición y no de desarrollo teórico.

2.1. Escenario Histórico

En el año de 1921, el dramaturgo y novelista checo, Karel Capek estrenó la obra: '*Robots Universales Rossum* (R.U.R)', en Praga. La obra tuvo gran éxito y una gran repercusión por hacer uso, por primera vez, del término *robot*. La trama de la obra recurre al abuso inconmensurable del maquinismo en donde los robots terminan exterminando a sus creadores. Es así como el término robot es usado por primera vez y aunque la palabra puede tener múltiples definiciones generalmente la robot es acuñado para todos aquellos sistemas electromecánico cuya programación le permite realizar decisiones independientes.

Los robots han tomado diferentes formas durante décadas como los robots bípedos. En 1888, George Fallis patentó un juguete capaz de realizar una caminata pasiva [5]. Fallis comenta que su patente es una combinación de un péndulo y un balancín de construcción, que al colocarlo en un plano inclinado éste comenzará a moverse debido a la fuerza de gravedad. Una característica importante del juguete de Fallis es que su centro de gravedad estaba localizado siempre en su base de soporte. En 1990 Tad McGeer se basó en el juguete de Fallis para demostrar que un mecanismo pasivo, no necesita de ningún control de realimentación para lograr una marcha dinámica. Éste tipo de mecanismos, cuya marcha no necesita de sistemas de control, actuadores o de algún tipo de sensor para alcanzar su cometido han sido objeto de estudio durante mucho tiempo. Las investigaciones de McGeer han ofrecido un apoyo importante dentro de la documentación en marcha pasiva y se han tomado en cuenta con la finalidad de mejorar los estudios en la caminata bípeda.

En 1967 en la Universidad de Waseda, Ichiro Kato profesor y pionero en el desarrollo y construcción de robots en Japón y un grupo de colaboradores, construyeron el WL-1. Un robot bípedo que sirvió para el estudio de la locomoción bípeda. En los siguientes años Waseda desarrolló una variedad importante de robots bípedo, figura 2.2. El WL-3 (1969), con ayuda de actuadores electro-hidráulicos era capaz de levantarse y sentarse.

En 1969, se desarrolló el robot WAP-1 usando músculos artificiales hechos de caucho, como actuadores. EL WAP-2 a diferencia de su antecesor fue controlado a través de un control de postura basado en las lecturas de sensores de presión ubicadas en las patas. En 1971, nace el robot bípedo WAP-3 con la capacidad de balancear su centro de gravedad sobre el plano frontal. En 1972, un robot controlado con una minicomputadora se desarrolla bajo el nombre de WL-5.

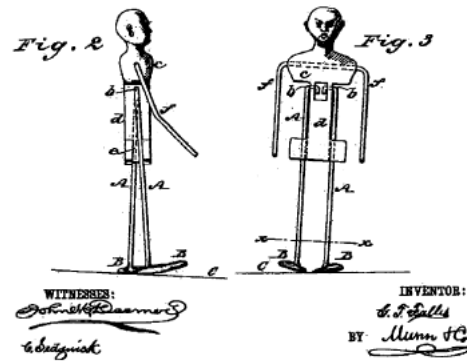


Figura 2.1: Esquema de la patente de Fallis. Las primeras investigaciones acerca del desarrollo de una marcha pasiva fueron basados en esta patente fabricada en 1888 [5].

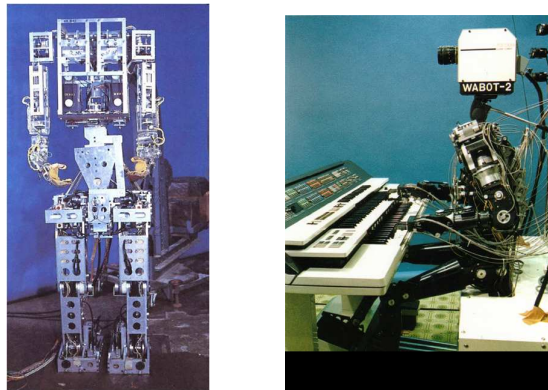


Figura 2.3: Primer robot humanoide a gran escala WABOT-1 y su sucesor WABOT-2, respectivamente

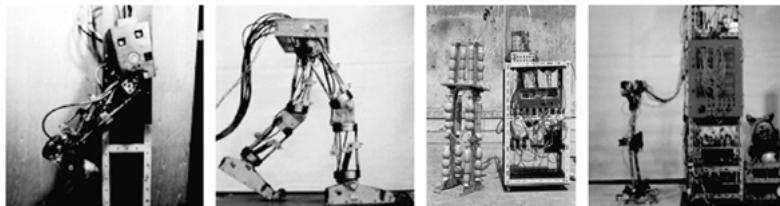


Figura 2.2: Robots bípedos de la Universidad de Waseda de izquierda a derecha: WL-1, WL-3, WAP-1, WAP-2

La evolución de éstos robots remarca la importancia que tuvo el estudio de la robótica móvil y que desemboca en uno de lo más nombrados proyectos en la historia de la robótica bípeda. En 1973, se crea el primer robot antropomórfico a gran escala, WABOT-1 (WASeda roBOT), un robot bípedo capaz de comunicarse en japonés, medir distancias, recoger objetos, entre otras cosas.

En 1985, doce años después, nuevamente el grupo de Ichiro Kato construye a WASUBOT (WASeda SUMitomo roBOT), un robot humanoide que es capaz de tocar un piano. WASUBOT era capaz de leer música y tocar un repertorio de 16 canciones.

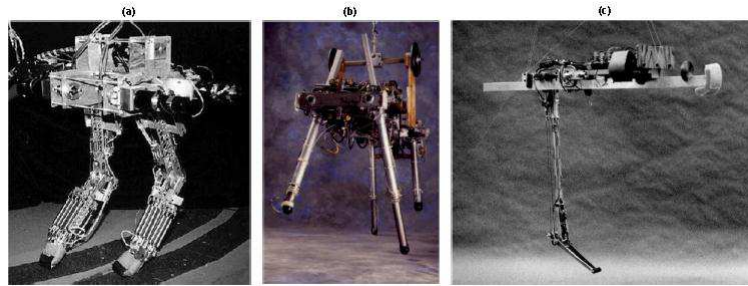


Figura 2.4: Robots del laboratorio Leg-Lab del MIT: (a) Spring Turkey, (b) Quadruped, (c) Monopod

En 1979 Ivan Sutherland le otorga la posibilidad y los fondos a Mark Raibert para construir robots con la capacidad de saltar. Un año después, en 1980, se financió al MIT Leg Laboratory en Estados Unidos. Desde entonces los investigadores pertenecientes al laboratorio han desarrollado robots con piernas, incluyendo robots de una sola pierna (Monopod, 1988), robots cuadrúpedos (Quadruped, 1984), robots bípedos (Spring Turkey, 1994) , entre otros.

2.2. Escenario Actual

Los estudios acerca de la locomoción bípeda han incrementado y los avances se pueden apreciar en las capacidades que tienen robots como ASIMO, un robot creado por Honda capaz de subir y bajar escaleras, saltar, cargar con objetos (tareas serviciales), comunicarse con humanos y caminar a una velocidad de 1.6 km/h.

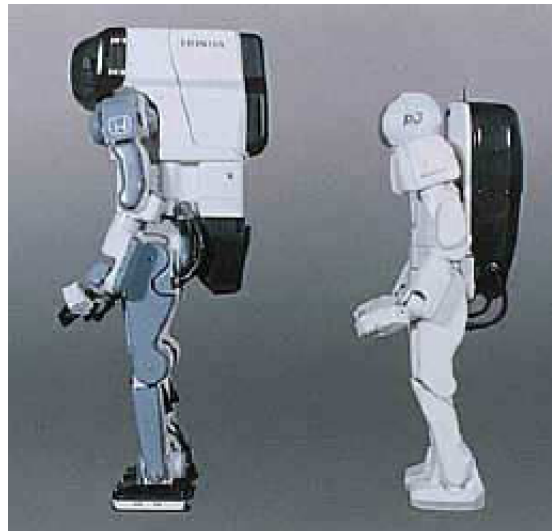


Figura 2.5: Robots humanoides de Honda P1 y P2, respectivamente. Robots bípedos antecesores del robot ASIMO

Tuvieron que pasar varios años para que ASIMO alcanzara una estatura aproximada de 120 cm y un peso de 43 kg. Sus antecesores, el robot bípedo P2 desarrollado en 1996, tenía un peso de más de 200 kg y una estatura de 182 cm. El robot P3, fue creado en 1997 con una estatura de 160 cm, un peso de 130 kg y brazos extensos para evitar que doblara el torso en tareas específicas ya que no contaba con una cadera versátil como la que presentaría ASIMO años después.

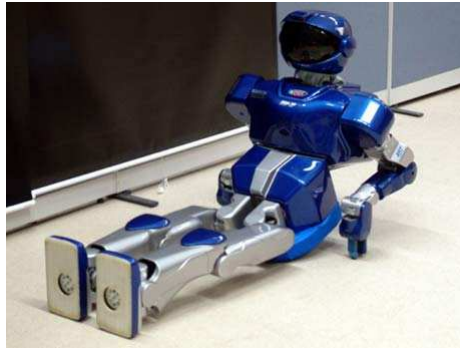


Figura 2.7: HRP-2P, el robot humanoide de Kawada. Destacado por su capacidad de incorporarse por si mismo



Figura 2.6: ASIMO (Advanced Step in Innovative Mobility), el robot humanoide más avanzado de Honda y uno de los robots bípedos más versátiles en el mercado

Industria Kawada en Japón desarrolló el robot HRP-2P (véase figura 2.7) con una estatura de 154 cm y una peso aproximado de 58 kg. Su sistema de navegación y la suavidad de sus movimientos hacen de éste robot un gran ejemplar en la gama inmensa de robots humanoides. La compañía SONY participó con el desarrollo de QRIO un robot semejante a ASIMO capaz de correr, subir escaleras e incluso realizar bailes sincronizados como los presentados en la figura 2.8.

2.3. Criterios importantes sobre estabilidad en marcha bípeda

La generación de trayectorias de un robot bípedo involucra, además del desarrollo de puntos significativos que describen una ruta, mantener el equilibrio durante la marcha y para ésto se toman criterios que ayudan a cuantificar el nivel de estabilidad que tiene el desplazamiento del mecanismo. En la siguiente sección se describirá de manera simple algunos conceptos como Ground projection of the *Center of Mass* (GCOM), *Zero-Moment Point* (ZMP), *Center of Pressure* (CoP), *Fiction Zero-Moment Point* (FZMP) y *Foot Rotation Indicator* (FRI) a modo de introducción para los siguientes capítulos.



Figura 2.8: Rutina de baile sincronizado realizado por el robot QRIO de Sony

2.3.1. Ground Projection of the Center of Mass (GCOM)

Como su nombre lo indica, el GCOM (*Ground projection of the Center of Mass*) es la proyección sobre el suelo del centro de masa del mecanismo y es el criterio sobre el que se basa la caminata bípeda estáticamente estable. Partiendo del enunciado anterior podemos decir que un criterio de estabilidad para la caminata estática se basa en la proyección del centro de gravedad del mecanismo o bien del GCOM sobre el área de soporte.

El área de soporte es el conjunto convexo formado por los puntos en contacto entre el robot y su entorno. En fase simple es aproximadamente inferior al área de la planta del pie, el área de soporte en fase doble es mucho más amplia, como se muestra en la figura 2.9.

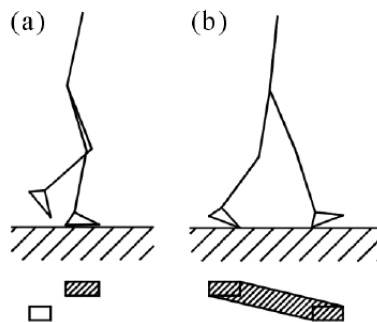


Figura 2.9: Área de soporte generada en la (a) fase de soporte simple y en la (b) fase de doble soporte. Figura tomada de [6]

Para que una marcha bípeda sea estáticamente estable el sistema debe balancear la línea que forma la proyección de su centro de gravedad de manera que caiga dentro del área de soporte. Si el punto GCOM está más allá del polígono de soporte, esto equivaldría a la presencia de un momento no compensado sobre el pie, el cual causaría una rotación alrededor del punto sobre el límite del polígono [6].

2.3.2. Zero Moment Point (ZMP)

Para que la locomoción bípeda logre una caminata estable; es decir, que alcance un balance dinámico, el pie de apoyo debe estar totalmente en contacto con el suelo. Esto no quiere decir que la presión en todos los puntos del pie deba ser la misma, al contrario. La presión dentro de la planta del pie debe variar ya que es una acción progresiva y además el entorno del robot casi nunca cuenta con un suelo idealmente plano.

Para que un robot sea capaz de ponerse en marcha sobre un plano horizontal, es necesario que sus juntas estén actuadas y que puedan ser controladas dentro de un ciclo lógico para desarrollar una caminata. Si el robot bípedo no está en contacto con el suelo, entonces se puede decir que tiene tantas articulaciones actuadas como grados de libertad; sin embargo, si entra en contacto con el suelo un grado de libertad pasivo entra en juego. Este contacto es esencial para la realización de la marcha bípeda, ya que la postura del mecanismo completo depende de la reacción que resulte de las fuerzas que ejerce el suelo contra el robot.

La postura del robot durante la marcha se puede controlar asegurando una estabilidad dinámica por encima del pie. Esto quiere decir que es posible resumir todas las fuerzas que actúan sobre el robot y colocarlas en punto de acción, una sola fuerza actuando en un sólo punto. Éste punto fue propuesto en 1970 por Miomir Vukobratovic y fue nombrado como ZMP por sus siglas en inglés *Zero-Moment Point* [7].

Éste método ha sido utilizado a lo largo de todos éstos años como punto clave para el desarrollo de la caminata bípeda. Un claro ejemplo es el presentado por el robot WL-10RD cuya caminata fue la primera desarrollada por un robot bípedo basada en el cálculo del criterio. El criterio del ZMP también es indispensable en el control del bípedo, es útil para establecer una señal de retroalimentación entre la fuerza ejercida por la tierra y el pie en contacto.

Las fuerzas que acciona el robot al entrar en contacto con su entorno son: fuerza de fricción, y una fuerza de reacción del piso. Las demás fuerzas que actúan sobre el robot pueden ser remplazadas por una fuerza y un momento aplicadas en un punto A, como lo muestra la figura 2.10. Se ha resumido todo a un punto A, con la finalidad de explicar el criterio.

El punto A es un punto importante, porque es la representación geométrica del resto del mecanismo, además de éste punto la figura 2.10 presenta: un vector G que simboliza la fuerza de gravedad; F_A y M_A la fuerza y el momento aplicados en el eje de giro A respectivamente; la fuerza de reacción del piso se representa mediante una fuerza R (R_x, R_y, R_z); también se representa un momento M (M_x, M_y, M_z) que es el momento generado por la fuerza R. Y ya que el pie se encuentra en reposo, sus componentes horizontales estarán en equilibrio debido a la fricción.

Por lo tanto, la fuerza de reacción horizontal (R_x, R_y) representa la fuerza de fricción que está equilibrando las componentes horizontales de la fuerza F_A , mientras que el momento de reacción vertical M_z representa el momento de la fuerza de reacción de fricción que equilibra la componente vertical del momento M_A y el momento inducido por la fuerza F_A . Si se supone que no hay deslizamiento entre el pie y el piso, la fricción estática compensará las componentes horizontales R_x y R_y y el par de reacción vertical M_z .

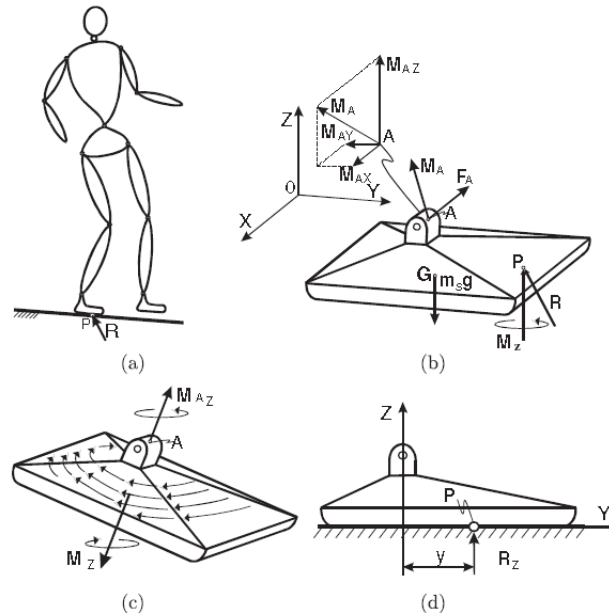


Figura 2.10: Fuerzas que actúan sobre la planta del pie del mecanismo. Tomada de [7]

La componente vertical R_z representa la reacción de la tierra que equilibra las fuerzas verticales restantes. Falta considerar el equilibrio de la componente horizontal del momento de carga del pie. Debido a la naturaleza de la relación entre el pie y el piso (siempre hacia arriba), las componentes horizontales de cualquier momento activo puede ser compensado cambiando la fuerza de reacción R dentro del polígono de soporte, o planta del pie. Esto quiere decir que si el robot da un mal paso, inclinando su cuerpo hacia un lado, se generará un momento provocado por dichas componentes horizontales, y la única solución inmediata, sería mediante una acción de corrección que compense dicho momento. Esto sería mediante la fuerza de reacción R que empuja al pie hacia arriba.

Lo anterior se ilustra en Figura 2.10 (d), donde el momento M_{Ax} esta en equilibrio debido al punto de acción de la fuerza R_z y evidentemente a su intensidad, cuya magnitud necesaria se determina mediante la ecuación de balance de las fuerzas que actúan sobre el pie, multiplicada por la distancia y .

Se debe recordar que la fuerza de reacción esta dentro del área formada por los puntos del pie del robot que entran en contacto con el piso y que además, el incremento del momento del tobillo será compensado por los cambios de posición de dicha fuerza, y las componentes M_x y M_y no existirán. Sin embargo, si el polígono de soporte no es lo suficientemente largo como para guardar la posición necesaria de la fuerza R_z que balancea los momentos activos, la fuerza R actuará en la orilla del pie y la cantidad del momento que no logra ser compensada causará que el mecanismo vuelque. Por lo que podemos afirmar que la condición suficiente y necesaria para que la locomoción del mecanismo esté en equilibrio dinámico es que para el punto P en la suela del pie, donde la fuerza de reacción esta actuando,

$$\begin{aligned} M_x &= 0, \\ M_y &= 0 \end{aligned}$$

Ya que ambos componentes son relevantes para la realización de un equilibrio dinámico y son iguales a cero, resulta natural nombrar a éste punto como: *Zero-Moment Point*.

2.3.3. Foot Rotation Indicator (FRI)

Ambarish Goswami presenta un nuevo criterio para evaluar lo que llama una “estabilidad postural”. Este criterio llamado FRI es definido como un punto sobre la superficie en contacto (pie/piso), dentro o fuera de

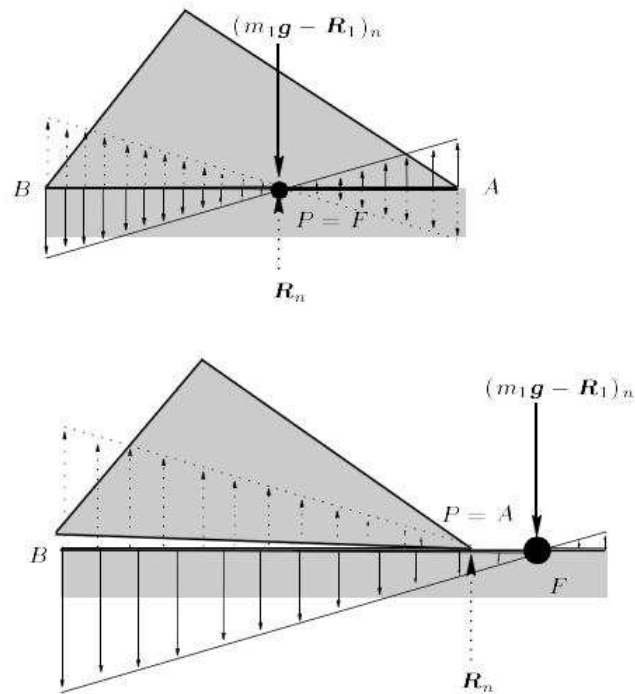


Figura 2.11: CoP (P) y FRI (F) .Magnitud de momentos compensados (en la parte superior) donde $P=F$. Perturbación no compensada, inestabilidad (en la parte inferior) donde $P \neq F$. Tomada de [8]

polígono de soporte, en el cual los momentos resultantes de las fuerzas y pares de fuerzas impresas sobre el pie son normales a la superficie [8].

Podemos inferir que dentro de la definición escrita la única diferencia entre el FRI y el ZMP se encuentra en la distancia que pueden tener uno del otro del polígono de soporte. Por otra parte, la definición de un ZMP ficticio complementa el concepto que desea abarcar el FRI.

La consideración que Godswami hace para implementar un nuevo punto como análisis de estabilidad, es la de igualar los términos ZMP y CoP; así entonces, encuentra un espacio en el que cabe un punto indicador fuera del polígono de soporte. Sin embargo, el criterio del ZMP difiere por esa razón del CoP. En la Figura 2.11 se muestra la ubicación del FRI dentro y fuera del polígono de soporte y explica de manera similar que la magnitud del momento originado por una perturbación será linealmente proporcional a la distancia que existe entre la orilla de la planta del pie y la ubicación del FRI.

Podemos concluir que el FRI es otra manera de nombrar a un ZMP fuera del polígono de soporte. Más adelante se hablará de un ZMP que se encuentra lejos del área convexa formada por los puntos en contacto entre el robot y el suelo.

2.3.4. Diferencia entre ZMP y CoP

Es muy importante saber la diferencia que existe entre la connotación que tiene el CoP dentro del análisis de estabilidad de la marcha de robots bípedos y el concepto del ZMP. Una posible definición del CoP como fuerza que actúa sobre el bípedo es la de una presión que existe entre el piso y el pie del robot, que puede ser remplazada por una fuerza que actúa en el centro de presión (CoP). Ahora, si esta fuerza equilibra a las fuerzas que actúan sobre el mecanismo durante su marcha (inercia, gravitación, fuerzas de Coriolis y centrífugas y momentos) su punto de acción tendrá que ser el ZMP [7]. De esta forma, en el caso de una caminata dinámicamente estable, el CoP y el ZMP coinciden. Cuando la marcha no esta dinámicamente balanceada, el ZMP no existe

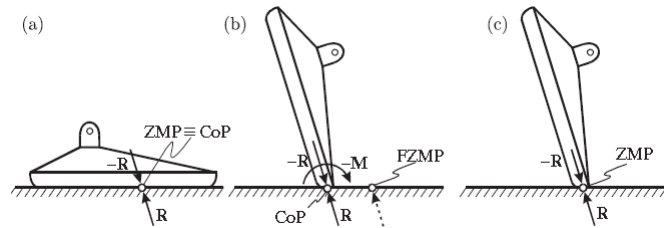


Figura 2.12: Tres ejemplos que relacionan el punto ZMP con el CoP: (a) marcha estable el CoP y ZMP coinciden, (b) marcha inestable CoP y ZMP no coinciden y (c) corrección de posible inestabilidad con movimientos muy particulares hechos por el robot. Tomada de [7]

y el mecanismo colapsa.

Otra definición, es la descrita en [9], donde se describe al CoP como un campo de fuerzas de presión (normal a la planta del pie en contacto) equivalente a una sola fuerza resultante, ejercida en el punto dónde el momento resultante es cero.

Para hacer una distinción más clara entre estos dos conceptos, se resumirá la discusión anterior del ZMP en tres casos característicos:

- **Marcha estable.** En una marcha estable, el ZMP coincide con el CoP (ver figura 2.12 (a)).
- **Marcha inestable.** En el caso que una perturbación lleve el punto de acción de la fuerza de reacción a la orilla del pie, el momento generado por la perturbación ocasionará una rotación del sistema alrededor de la orilla del pie y el bípodo caerá. En ése caso estamos hablando de un ZMP ficticio, cuya distancia desde la orilla del pie representa la intensidad del momento generado por la perturbación (véase figura 2.12 (b)).
- **Marcha inestable corregida.** Sin embargo, es posible hacer que el bípodo se mantenga de pie realizando movimientos precisos como en la figura 2.12 (c), sobre las puntas del pie, quizás con zapatos especiales de tal manera que tenga un área determinada y una posición del ZMP dentro de ésa área .

Con todo esto podemos concluir que el concepto del ZMP y el CoP no es el mismo, ya que el ZMP puede existir no sólo en la planta de apoyo del pie (en soporte simple) sino en otras partes del robot e incluso fuera de él.

En resumen, el ZMP siempre coincide con el CoP dentro de una caminata dinámicamente estable; sin embargo, el CoP no siempre coincide con el ZMP. Y finalmente, un ZMP ficticio (FZMP) nunca coincidirá con el CoP ya que el CoP no puede existir fuera del polígono de soporte.

2.3.5. Zero Moment Point (ZMP) y Fictitious Zero Moment Point (FZMP)

Ya que el concepto de un ZMP fuera del polígono de soporte fue mencionado en las secciones anteriores, es necesario explicar, de manera breve, lo que ésto significa. El FZMP por sus siglas en inglés *Fictitious Zero-Moment Point*, es el punto en cual los momentos de inercia y fuerzas de gravedad no tienen componentes a lo largo del eje horizontal, o como se describió anteriormente, $M_x = 0$ y $M_y = 0$ y que además no se encuentra dentro del polígono de soporte.

Su estudio parecería no importar en la caminata bípoda, ya que si el ZMP no se encuentra dentro del polígono de soporte el robot colapsará. Si se observa desde otro punto de vista, por ejemplo, como un indicador, el FZMP resulta un concepto muy importante.

La inestabilidad es un estado que se pretende evitar en una marcha, por eso es importante que el robot sea capaz de tomar decisiones para su "rehabilitación" en caso de ocurrir un posible vuelco; es decir, se deben prediseñar estrategias de emergencia como volver inmediatamente a una postura inicial estable, apagar ciertos actuadores o realizar movimientos drásticos para generar un nuevo ZMP, con el fin de evitar una caída. Claro que

para realizar alguna de estas tareas delicadas se requiere de sensores especiales como giroscopios, acelerómetros, unidades de control capaces de actualizar el estado de los servomotores en microsegundos, entre otras cosas. La información que proporciona el FZMP nos puede ayudar a reducir gastos en la instrumentación del robot y es potencialmente útil en el área de control .

Imaginemos que podemos anticiparnos al cálculo del FZMP, ésto implicaría anticiparnos a la caída y la magnitud del momento de perturbación; realizar una retroalimentación y controlar la postura del robot de manera exitosa.

Capítulo 3

Sensores relevantes para una marcha bípeda

En ingeniería los sensores son definidos como sistemas que son capaces de medir señales del entorno de interés. La finalidad varía dependiendo del objetivo del sistema en general, no obstante todos los sensores son parte de sistemas retroalimentados.

La naturaleza esta poblada de ejemplos sensoriales y su capacidad de procesar información es aún inigualable. Es arriesgado pretender igualar los procesos biológicos que dan marcha a la vida, aunque estudiar su funcionamiento y finalidad puede proporcionar a la ingeniería una herramienta útil en el desarrollo de sus propios sensores. Los robots bípedos son un ejemplo de sinergia de ciencias aplicadas que sirven tanto al estudio como al desarrollo y mejoramiento tecnológico.

3.1. Receptores naturales

Para trasladarse de un lugar a otro de manera natural el ser humano realiza múltiples procesos de manera inconsciente, hace trabajar los músculos de su cuerpo y procesa la información a través de su sistema nervioso central. El estudio de los procesos biológicos del ser humano ayuda a comprender la función que desarrolla dentro de la naturaleza, el entorno en el que convive. Y nos brinda las herramientas necesarias para poder crear sistemas en beneficio del hombre a partir de analogías.

Estudiar el desarrollo de la caminata bípeda humana involucra procesos tanto psicológicos (procesos cognitivos) como físicos; a través de su estudio podemos comprender cómo percibe el ser humano su entorno, cómo se orienta, en qué momento desarrolla el sentido de la ubicación, cómo detecta su medio ambiente y evita tropiezos, entre muchas otras cosas.

El ser humano desarrolla una marcha independiente entre los 13 y 15 meses de edad, y no es sino hasta los 5 o 7 años que es capaz de realizar una marcha parecida a la de un adulto promedio [10]. Esto quiere decir que el ser humano requiere de mucho tiempo (a comparación con otros seres vivos) para controlar su caminata. Una pieza clave para el éxito es manejar a la perfección todos sus sentidos, y esto lo logra gracias a que cuenta con sensores naturales.

En biología la palabra receptor suele usarse en varios contextos y es más usada que el término sensor. En el sentido más general, un receptor es una estructura que cambia cuando un estímulo de su entorno actúa sobre ella y ese cambio genera una señal. Todos los receptores sensoriales producen señales eléctricas, pero cada tipo de receptor se especializa en producir su señal únicamente en respuesta a un tipo dado de estímulo del entorno [11].

La excitación de un receptor sensorial produce una señal eléctrica llamada potencial de receptor. La amplitud del potencial de receptor varía según la intensidad del estímulo: cuanto más fuerte sea el estímulo, mayor será el potencial de receptor. En algunos receptores sensoriales neuronales, el potencial de receptor hace que la célula

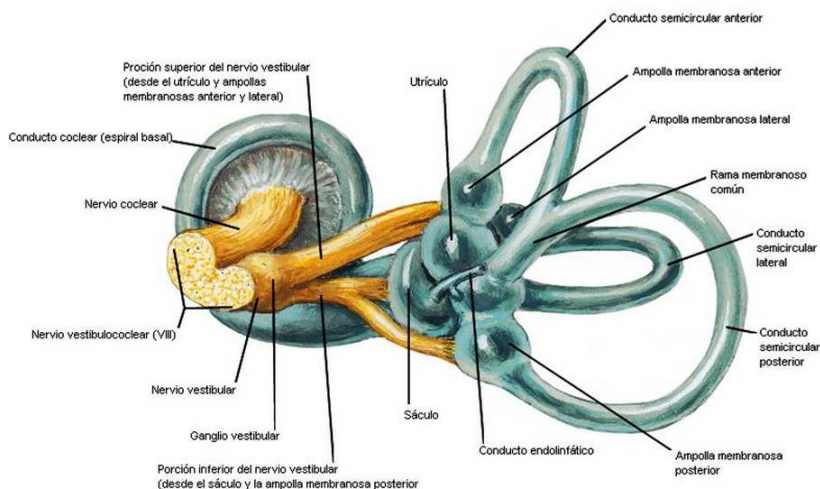


Figura 3.1: Laberinto membranoso del oído interno, organismo sensorial importante para el equilibrio humano. Imagen obtenida de [13]

rebasa un umbral y genera potenciales de acción que viajan hasta el sistema nervioso central, y es así cómo el cerebro interpreta intensidad.

Por lo tanto nos referiremos a receptor u órgano sensorial cuando estén involucrados términos biológicos y sensor cuando estemos hablando sobre sistemas mecatrónicos.

3.1.1. Oído interno

El oído interno, por ejemplo, es un receptor sensorial que nos ayuda, además de percibir sonidos, a mantener el equilibrio a través del *órgano vestibulococlear*. Enterrado en la porción petrosa del hueso temporal, el oído interno consta de los sacos y conductos del laberinto membranoso. El **laberinto membranoso** (figura 3.1), con *endolinfa*, queda suspendido dentro del laberinto óseo por la *perilinf*a; los dos líquidos transportan las ondas sónicas a los órganos terminales para la audición y el equilibrio.

El laberinto óseo del oído interno es una cripta formada por tres elementos: la cóclea, el vestíbulo y los conductos semicirculares óseos.

La cóclea es la porción del laberinto óseo, con forma de caparazón, que contiene el conducto coclear, una parte del oído interno relacionada con la audición.

El **vestíbulo** es una pequeña cámara ovalada (de unos 5 cm de longitud) que contiene el *utrículo* y el *sáculo* elementos del aparato del equilibrio (laberinto membranoso).

Los **conductos semicirculares óseos**, son conductos (anterior, posterior y lateral) que se comunican con el vestíbulo del laberinto óseo. Los conductos se sitúan posterosuperior al vestíbulo, donde desembocan; se disponen de manera perpendicular y ocupan los tres planos del espacio [12].

3.1.2. Ojos

Además del oído el ser humano cuenta con sensores que le permiten a ubicarse y dimensionar su entorno. Los ojos son instrumentos perfectos de visión que están ubicados en una cavidad llamada **órbita**. La órbita además de contener al globo ocular protege al nervio óptico, los músculos oculares, fascia, vasos sanguíneos, grasa y la glándula y el saco lacrimales.

La forma en que el ser humano detecta la luz se produce a través de una reflexión de luz; es decir, la luz reflejada por cualquier objeto incide primero en la **córnea**, una cubierta transparente en la parte frontal del globo ocular.

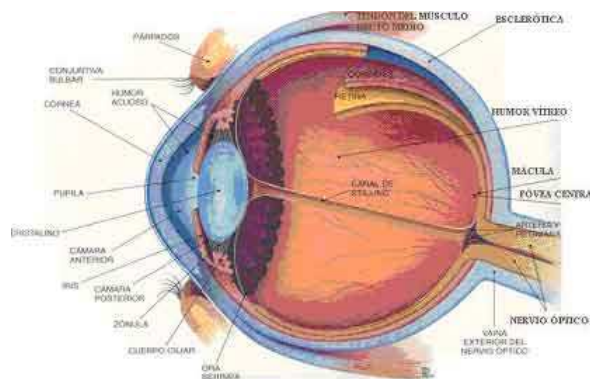


Figura 3.2: Anatomía del ojo humano

Después de la córnea, la luz atraviesa una cámara llena con un líquido acuoso, llamado **humor acuoso**, que nutre tanto a la lente como a la cornea. El **iris**, formado por tejido muscular pigmentado, ajusta la cantidad de luz que ingresa en el ojo. El iris regula el tamaño de la **pupila**, una abertura circular en el centro del iris. La luz que atraviesa la pupila incide en el **crystalino**, una estructura que semeja una esfera aplastada y está compuesta por fibras proteicas transparentes. El cristalino está suspendido detrás de la pupila por músculos que regulan su forma (véase la figura 3.2).

Detrás del cristalino hay una cámara mucho mayor, llena con el **humor vítreo**: una sustancia gelatinosa transparente que ayuda a mantener la forma del ojo. Después de pasar por el humor vítreo, la luz llega a la **retina**, una membrana de fotorreceptores y neuronas que tiene varias capas. Ahí, la energía luminosa se convierte en impulsos eléctricos nerviosos que se transmiten al encéfalo.

El sistema nervioso central se ocupa de procesar y almacenar toda la información transmitida, es así como el ser humano y los ojos de los mamíferos en general captan, enfocan y convierten las ondas luminosas. Además, la visión binocular nos permite percibir la profundidad y calcular con exactitud la distancia a la que se encuentra un objeto.

3.1.3. Tacto

Es conveniente estudiar los diferentes tipos de receptores que se encuentran en nuestro organismo, puesto que varias investigaciones alrededor de la instrumentación bípeda juegan con los conceptos declarados en esta sección. Otro ejemplo de ello, es el tacto; las células nerviosas especializadas que conforman el sentido del tacto captan una serie de estímulos específicos que permiten diferenciar distintas sensaciones.

La mayoría de estas estructuras (corpúsculos, discos de Merkel y terminaciones libres) se encuentran a lo largo de la superficie del cuerpo (repartidas en mayor o menor cantidad según la zona) y perciben el frío, calor, presión y dolor; de modo que el sistema nervioso recibe información de lo que ocurre en el interior y el exterior del organismo.

Los receptores cutáneos que se encuentran en la piel, cuya importancia radica en la información (exploración física) que obtiene de su medio ambiente, cubren el cuerpo y presentan una estructura (a través de la piel) que al ser estimulada producen impulsos que se transmiten al sistema nervioso central. De esta manera, las terminaciones libres perciben las sensaciones doloras y los discos de Merkel, se estimulan por la presión [14].

Los corpúsculos de la dermis detectan estímulos táctiles y térmicos: los de Krause captan el frío; los de Ruffini reciben el calor; los de Paciniy Meissner son sensibles a la presión; y los receptores pilosos sienten el movimiento del pelo (ver figura 3.3).

En la tabla 3.1 se resumen algunos receptores que ocupan la mayoría de los vertebrados para su intercambio continuo con el medio ambiente. Estudiando las funciones de los receptores podemos concluir que son de vital importancia para el desenvolvimiento natural del hombre y que son pieza clave para desarrollar actividades continuas, es por eso que la robótica ha tomado conceptos naturales y desarrollado analogías artificiales.

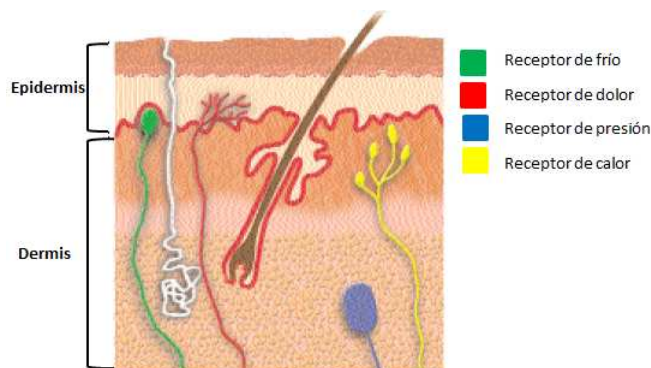


Figura 3.3: Tacto. Receptores cutáneos. La piel es el órgano sensorial más grande en el ser humano y un importante puente de comunicación entre éste y su entorno [14] .

3.2. Sensores artificiales

Según [15] un sensor es un dispositivo el cual, una vez sujeto a la acción de un fenómeno físico genera una señal eléctrica. Los sensores proporcionan una señal vinculada directamente con la magnitud que ha sido medida. Esas señales pueden ser analógicas, numéricas, Booleanas o digitales.

En el laboratorio de robótica, Humanoid Lab del IRI (Institut de Robòtica i Informàtica Industrial) de la Universidad de Catalunya (CSIC-UPC) diseñan robots bípedos antropomórficos cuyos sensores representan los sentidos humanos. Así entonces, el sistema visual del robot se consigue a través de una cámara en la parte superior del robot más un sensor de distancia que permite al robot ubicarse en el espacio y evadir obstáculos.

El equilibrio del mismo se consigue utilizando sensores de inclinación, y sensores de presión en los pies para conseguir una estimación de la posición del cuerpo. Los sensores de fuerza fungen también como sentido del tacto del robot, permitiéndole detectar presencia directa de obstáculos en el camino. Toda esta retroalimentación ayudará al robot a tener un mejor desempeño.

La medición de estos parámetros dependen principalmente del tipo de control o de los objetivos de sus aplicaciones. Cada robot puede hacer uso del sensado que desee; sin embargo, dentro de un proyecto, incorporar sensores es una tarea que requiere del estudio de las características que intervienen, por ejemplo, la accesibilidad al sensor, característica de temporalidad (si la medida puede ser leída directamente o requiere de tiempo para acondicionarla o transmitirla) y precisión.

De la misma manera en que se abordó a los receptores sensoriales, se abordará a los sensores mecatrónicos, tratando de plantear una comparación somera, mencionando sólo a aquellos que se consideran más importantes ó aquellos que son usados frecuentemente en la robótica bípeda.

3.2.1. Sensores inerciales

Los sensores inerciales se componen de sensores que miden la aceleración (acelerómetros) y velocidad angular (giroscopios). Estos sensores contienen dichos componentes dentro de un solo instrumento, haciendo más útil y práctica su implementación.

3.2.1.1. Acelerómetros

Los acelerómetros miden fuerzas que son aplicadas a una masa en movimiento. Y su funcionamiento se basa en la segunda ley de Newton, ya que describe la fuerza inercial en función de la masa y la aceleración. Existen varios tipos de acelerómetros pero todos trabajan bajo ese principio.

- Mecánicos
- Capacitivos

Tipo de receptor	Tipo de célula sensorial	Estímulo	Ubicación
Termorreceptor	Terminación nerviosa libre	Calor, frío	Piel
Mecanorreceptor	Célula pilosa	Vibración, gravedad movimiento	Oído interno
	Terminación nerviosa especializadas y terminaciones nerviosas libres en la piel (corpúsculo de Pacini, disco de Merkel) Terminaciones nerviosas especializadas en músculos o articulaciones (huso muscular, órgano tendinoso de Golgi)	Vibración, presión, tacto Estiramiento	Piel Músculos, tendones
Fotorreceptor	Bastón, cono	Luz	Retina del ojo
Quimiorreceptor	Receptor olfatorio Receptor del gusto	Olor (moléculas en el aire)	Cavidad nasal
		Sabor (moléculas en agua)	Lengua
Receptor del dolor	Terminación nerviosa libre	Sustancias liberadas por tejidos lesionados	Ubicación en el cuerpo

Tabla 3.1: Algunos tipos de receptores. Tomado de [12]

- Piezoeléctricos
- Piezoresistivos

El avance tecnológico ha permitido la miniaturización de las herramientas o equipos para desarrollar o manufacturar productos de menor tamaño. La microtecnología es el resultado de este avance, definida como la tecnología que está caracterizada por tener dimensiones cerca de un micrómetro, es decir, una millonésima parte de un metro [16]. Las aplicaciones de los sistemas MEMS (Micro-Electro Mechanical Systems) son un ejemplo de la importancia que tiene la microtecnología para el desarrollo tecnológico, y para este trabajo las ventajas que ofrecen los sensores de tamaño reducido; algunos de éstos ejemplos son: en robótica bípeda el uso de sensores inerciales basados en éste tipo de tecnología; en la industria del entretenimiento el control remoto WIIMOTE; frenos ABS o bolsas de aire (Airbag) en la industria automotriz; en telefonía, sus múltiples aplicaciones que ocupan acelerómetros y giroscopios, entre otras. Algunas características que comparten este tipo de aplicaciones es que no requieren frecuencias altas (ancho de banda), su diseño no requiere de baja sensibilidad y no necesitan gran resistencia de temperatura por lo que representan una oportunidad dentro de la división de estudio de tecnología MEMS.

3.2.1.2. Principio de funcionamiento de un acelerómetro

Se explicará a grandes rasgos, sólo con fines didácticos, el funcionamiento de los acelerómetros mecánicos y capacitivos.

Los acelerómetros mecánicos emplean una masa de kilaje conocida que se encuentra suspendida dentro de un cilindro o entorno que la limita a un movimiento lineal, con un resorte de propiedades previamente establecidas y un sistema de amortiguación que evita oscilaciones injustificadas. Su funcionamiento es igual a la de un dinamómetro.

El modelo matemático del sistema mecánico descrito es el siguiente:

$$F_a = ma$$

$$F_a - b\dot{y} - ky = m\ddot{y}$$

$$ma = m\ddot{y} + b\dot{y} + ky$$

De las ecuaciones anteriores se puede deducir que para conocer la aceleración de la masa tenemos que obtener la F_a , que a su vez se obtiene del sistema resorte-amortiguador.

$$H(s) = \frac{y(s)}{a(s)} = \frac{m}{ms^2 + bs + k} \implies \text{Función de transferencia}$$

$$y(0) = \frac{ma}{k}$$

$$\omega_0 = \sqrt{\frac{k}{m}} \implies \text{Frecuencia natural}$$

$$y = \frac{a}{\omega_0^2}$$

La aceleración dependerá entonces de la posición del objeto, de la constante de elasticidad del resorte o constante de deformación de una viga (depende del sistema de suspensión) y de su masa. Ahora, si el análisis del sistema se hace en tres direcciones ortogonales entre sí como se muestra en la figura 3.4, tendremos la aceleración de un objeto en el espacio.

Los acelerómetros capacitivos trabajan bajo el mismo principio que los acelerómetros mecánicos, con la diferencia que éstos miden el desplazamiento de la masa (posición) utilizando un condensador diferencial, que

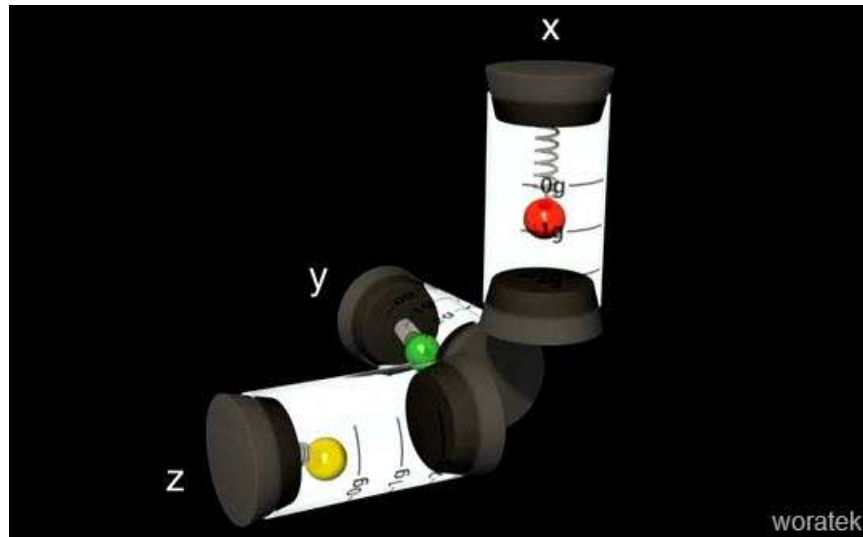


Figura 3.4: Esquema virtual representativo de un acelerómetro mecánico, cuyo arreglo de masas y resortes facilitan la comprensión de su principio de funcionamiento

consta de dos condensadores variables colocados de tal manera que su capacitancia cambie del mismo modo pero con signos contrarios. Usando un circuito adicional para acondicionar la señal de salida, se logra obtener una medición lineal y sensible con respecto a la configuración de un condensador simple (ver figura 3.5).

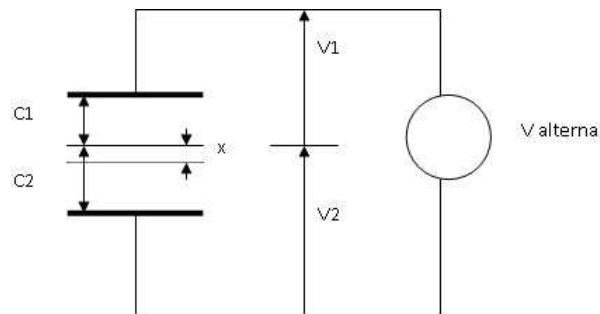


Figura 3.5: Circuito con la configuración típica para medir desplazamientos o distancia en un acelerómetro capacitivo.

La estructura de un condensador diferencial está formada por tres placas planas paralelas, que generalmente están fijas, y una placa central móvil que actúa como la masa conocida dentro de un acelerómetro mecánico. La salida acondicionada (lineal y sensible) se logra a través de una medición diferencial.

3.2.1.3. Giroscopio

El giroscopio es un mecanismo simétrico de rotación que gira alrededor de su eje de simetría y que nos permite averiguar su orientación o variación angular a través de un comportamiento poco intuitivo, ya que éste tiende a girar en dirección al par de fuerzas generado al intentar cambiar la orientación espacial de su eje de giro.

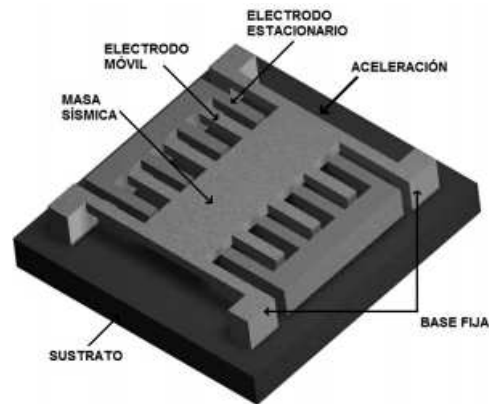


Figura 3.6: Estructura de un condensador diferencial con placa sísmica sustituyendo la masa en los acelerómetros mecánicos. Imagen tomada de [17]

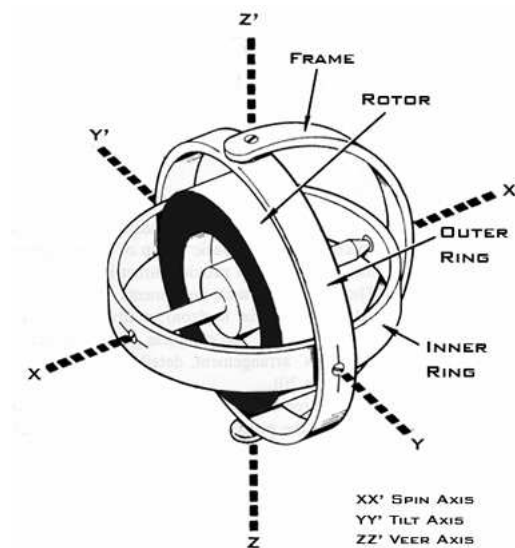


Figura 3.7: Vectores que actúan sobre un Giroscopio

El efecto giroscópico es básicamente una fuerza de reacción del mecanismo al aplicarse un par de fuerzas perpendicular a la superficie del disco rotando. La figura 3.8 muestra un disco con una barra como eje de giro; si se le aplica un par de fuerzas en los extremos el mecanismo tenderá a girar en el sentido de la resultante del mismo par de fuerza (ver figura 3.8), a este movimiento poco intuitivo y natural de los cuerpos que rotan se le llama precesión. Si el extremo A del giroscopio se fija a través de una cuerda y se presenta el movimiento de precesión el giroscopio comenzará a girar en sentido contrario a las manecillas del reloj, y su aceleración será proporcional a la fuerza externa ejercida en el punto B, es decir al par de fuerzas F.

Ahora bien si se tiene un giroscopio montado sobre una estructura fija como en un soporte Cardano (Giroscopio de Foucault), la energía de la resultante del par de fuerza descrita anteriormente que llamamos precesión se transmitirá a través de los ejes del giroscopio. De esta manera se puede calcular los ángulos con los que se inclina un giroscopio de Foucault.

Los giroscopios de tecnología MEMS están basados en vibraciones. El principio de funcionamiento de un giroscopio de estructura vibrante se basa en un cuerpo capaz de vibrar libremente en el mismo sentido con el que su plano de apoyo gira. Estos giroscopios también son conocidos como Giroscopios de vibración de Coriolis,

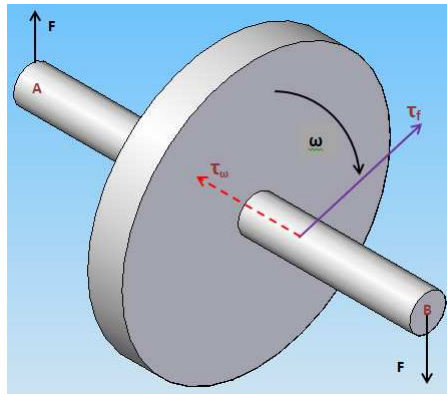


Figura 3.8: Fuerzas que influyen en un disco rotatorio. Donde F son las fuerzas externas que excitan al sistema; ω la velocidad angular del disco; τ_F y τ_ω son los pares de reacción tanto de la fuerza F como del momento inercial respectivamente.

ya que al ser rotados generan una fuerza de Coriolis que origina una segunda vibración ortogonal a la dirección de la vibración original.

3.2.2. Sensores de fuerza

Los sensores de fuerza son dispositivos que se destinan para medir las posibles fuerzas de reacción ejercidas entre el sistema y su entorno. En la robótica bípeda son colocados en las extremidades inferiores del robot para medir las fuerzas de reacción del piso. Normalmente, estos sensores poseen un diafragma que funciona como transductor de la fuerza actuando como elementos resistivos. Bajo una fuerza de presión ejercida sobre el área de medición, los valores resistivos cambian.

En el libro "Bipedal Robots Modeling, Design and Walking Synthesis" [15] se encuentra una breve descripción acerca de los sensores de fuerza, mostrando como ejemplo el diagrama de un sensor (ver figura 3.9) usado en la Universidad de Moscú que permite controlar la fuerza del bípido durante la etapa de doble soporte, basándose en la deformación axial.

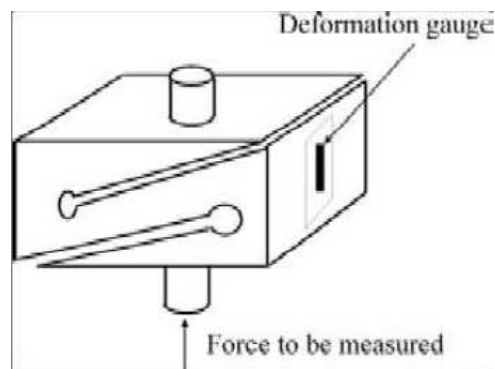


Figura 3.9: Sensor de fuerza cuya medición es perpendicular a su base. Tomada de [15]

La construcción de un sensor el cual es capaz de medir las tres componentes de un esfuerzo es muy delicada, especialmente con respecto a impactos. Si consideramos la intensidad de impactos durante una caminata dinámicamente estable la medida de dos componentes simples es un buen compromiso.

Un sensor de rango bajo mono-dimensional puede también ser útil para determinar los instantes de impacto. La ventaja de estos sensores son un buen paso de ancho de banda y su resistencia contra impacto, robustos.

Como se puede ver en la figura 3.10, medidores de esfuerzo axial fueron pegados a una parte del robot SemiQuad con la finalidad de medir el impacto de las piernas con el piso.

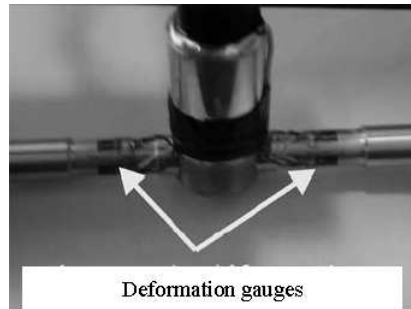


Figura 3.10: Ensamble para medición de deformación sobre la pierna del robot SemiQuad para detectar impactos. Tomada de [15]

3.2.3. Sensores de posición

Existe una amplia gama de sensores de posición: angular, lineal, relativos o absolutos; sin embargo sólo se presentará una descripción breve de aquellos que se consideran relevantes en robótica bípeda para entender el rol que desempeñan en la instrumentación sensorial.

3.2.3.1. Encoders

Los encoders son dispositivos formados por un rotor con uno o varios grupos de bandas opacas y translúcidas alternadas y un estator con una serie de captadores ópticos que detectan la presencia o no de banda opaca. Existen dos tipos de encoders, los encoders absolutos y los incrementales. Los encoders incrementales tienen un determinado número de impulsos grabados en el rotor que son contados para obtener, a partir de una referencia inicial, la posición por vuelta. Los encoders absolutos contienen bandas ordenadas según un código binario en el rotor y el circuito detector decodifica el código que es único para cada posición del rotor.

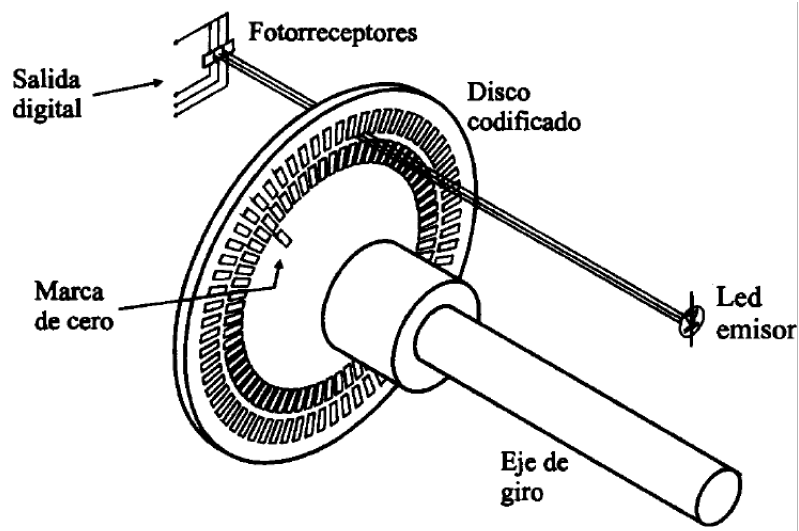


Figura 3.11: Encoder. La resolución depende del número de marcas que se pueden tener físicamente sobre el disco. Figura tomada de [18]

3.2.3.2. Potenciómetro angular

Los potenciómetros angulares son dispositivos que convierten una posición angular en una salida analógica. Es una bobina sobre un material conductor, distribuida a lo largo de un arco que funge como soporte y un apuntador unido a un eje de salida que se desliza sobre dicho conductor. El movimiento angular del eje provoca cambios de resistencia de tal forma que al aplicar una diferencia de potencial constante en los extremos, una tensión proporcional al ángulo de giro a partir del origen podrá ser medida en la salida del potenciómetro.

$$\frac{V_{BC}}{V_{AC}} = \frac{R_{BC}}{R_{AC}}$$

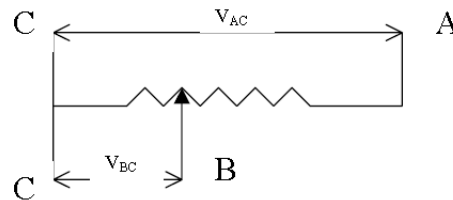


Figura 3.12: Potenciómetro. Determinan el desplazamiento lineal o angular según ésta relación eléctrica. Figura tomada de [18]

3.2.3.3. Sensor ultrasónico

Los sensores ultrasónicos constan básicamente de dos componentes un emisor y un receptor. El circuito inicial emite una señal que está dirigida al detector que se encarga de medir el tiempo que tarda en llegar dicha señal a su diafragma transductor (receptor), que puede ser cualquier tipo de sistema electromecánico. Los ejemplos más conocidos son los sensores ultrasónicos basados en sonidos o los sistemas de ecografía.

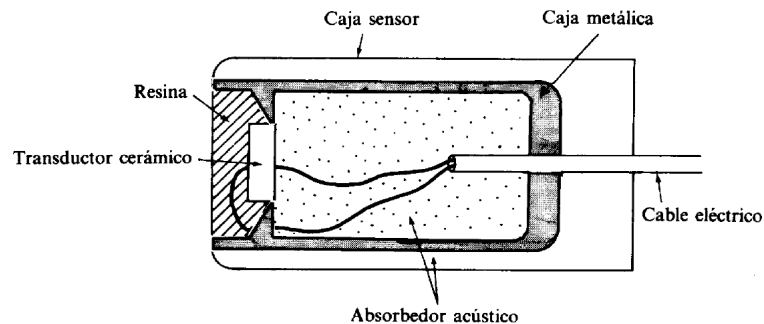


Figura 3.13: Sensor Ultrasónico. Es utilizado para determinar la distancia al objeto de interés. Figura tomada de [18]

3.3. Ejemplo de instrumentación bípeda: Robot Nao de Aldebaran

El robot humanoide Nao es de sumo interés ya que es uno de los robots bípedos con mejor desempeño en el mercado, dedicado también a la investigación. Nao sirve como ejemplo en este trabajo en el área de instrumentación, cuenta con sensores de fuerza ubicados en la planta de los pies para mejorar su marcha, con sensores inerciales para su orientación y equilibrio así como encoders para retroalimentar su posición. Se puede deducir que Nao es una referencia para la elaboración de este trabajo.

Nao es un robot humanoide programable de 57 cm de altura desarrollado por la empresa Francesa Aldebaran, con sede en París. Aldebaran Robotics es una empresa creada en el año 2005 a cargo de Bruno Maisonier, siendo la primera compañía francesa enfocada en el desarrollo de robots humanoides.

Nao es uno de los robot humanoides más populares de los últimos tiempos por su desarrollo tecnológico, Nao cuenta con :

- Nao tiene 25 grados de libertad y esta actuado con motores eléctricos y actuadores.
- Una red de sensores que incluye 2 cámaras, 4 micrófonos, un sensor sonar para telemetría y 8 sensores de fuerza
- Varios dispositivos de comunicación, incluyendo sintetizador de voz, LED's, y dos altavoces de alta definición
- Intel ATOM, 1.6 ghz CPU (localizado en la cabeza) con un *kernel* programado en Linux y apoya a a un *middleware* propiedad de Aldebaran (NAOqi)
- Una segunda CPU (localizada en el torso)
- Una batería de 27.6 watt/h que proporciona a Nao con 1.5 o más horas de autonomía, dependiendo del uso

3.3.1. Sensado de Nao

Los robots en cualquier industria, desde dispositivos sencillos para el entretenimiento o consumidores no especializados hasta para grandes empresas o áreas de investigación, tienen que ser capaces de sensar cualquier tipo de contacto, o medir la fuerza durante la acción.

3.3.1.1. Sensor inercial usado en el proyecto Nao

Nao cuenta con un sensor inercial con su propio procesador ubicado en el torso. El sensor cuenta con

- Girómetros con 5 % de precisión y una velocidad angular de $\sim 500^\circ/\text{s}$
- Un acelerómetro con 1 % de precisión y una aceleración de $\sim 2G$

La salida permite estimar la velocidad y posición del torso respecto a sus tres ejes fijos Cabeceo (**Yaw**), Alabeo (**Pitch**) y Guiñada (**Roll**).

Un algoritmo diseñado por Aldebaran Robotics calcula el ángulo del torso a partir de los acelerómetros y los giroscopios que han sido programados en la tarjeta inercial.

El acelerómetro tiene la única referencia absoluta y otorga el ángulo del torso en modo estático. Cuando algún movimiento es detectado la salida del ángulo es detectada con el giroscopio.

3.3.1.2. Nao y sus sensores de fuerza

Además del sensor inercial, Nao cuenta con sensores de presión en la planta de los pies que proveen a Nao de una marcha natural y estable mientras camina, corre o baila.

Los sensores miden una resistencia que cambia de acuerdo a la presión aplicada. Los FSR (Force Sensitive Resistors) esta ubicados en la planta de los pies del robot y trabajan en un rango de 0 N a 25 N, ver figura 3.15.

El balance del robot, como se ha mencionado a lo largo de la tesis, es importante en muchos aspectos, incluso en el cuidado del robot y Aldebaran ha introducido delgados y flexibles sensores de fuerza FlexiForce dentro de la planta del pie del robot para monitorear el balance. Los sensores aseguran que el robot balancee su peso apropiadamente de tal manera que no caiga mientras camina o baila.

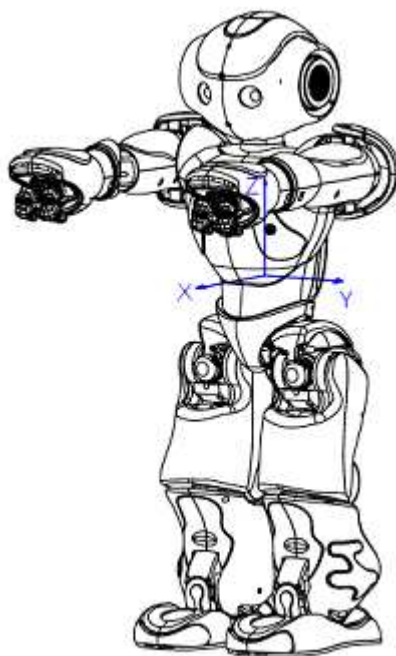


Figura 3.14: Vista dimétrica del robot Nao señalando el sensor inercial. Imagen obtenida de [19]

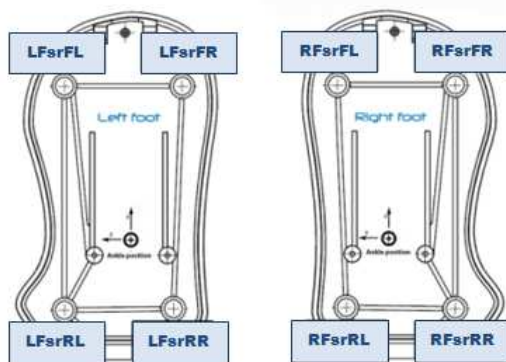


Figura 3.15: Ubicación de sensores de fuerza en los pies del robot bípido Nao. Imagen obtenida de [19]

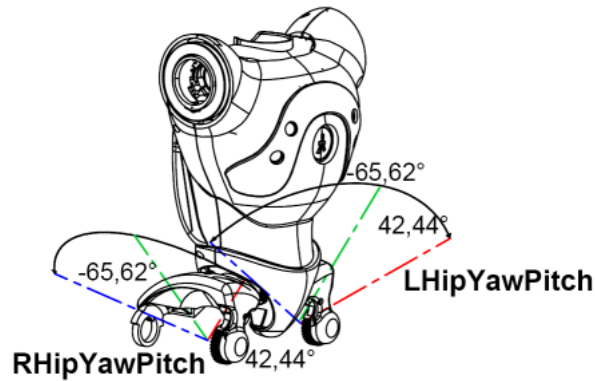


Figura 3.16: Perspectiva dimétrica de la pelvis del robot bípedo Nao. Imagen obtenida de [19]

3.3.1.3. Las articulaciones de Nao y sus sensores de posición

Nao tiene articulaciones en la cabeza, con rangos limitados en los ángulos yaw y pitch debido a las colisiones con su armazón; ambos brazos están articulados en hombro, codo y muñeca, que le permiten un mayor alcance; la pelvis está articulada y tiene movimiento en dirección de los ángulos yaw y pitch; cada una de sus piernas están articuladas en la cadera, rodilla y tobillo lo que le permite tener una gran versatilidad de movimientos como: sentarse, pararse, caminar, correr, o bailar.

Nao usa sensores de posición, encoders rotatorios magnéticos, en cada una de sus juntas rotatoria. Tienen una precisión de 12 bit, es decir, 4096 valores por vuelta que corresponden a una precisión de 0.1° . Para lograr una caminata omnidireccional en lazo cerrado es necesario contar con información sensorial. El proceso de la caminata de Nao está basado en la posición real del tronco del robot, proporcionada por los sensores de posición.

Capítulo 4

Sensores utilizados en el robot bípedo Scout

Este capítulo ofrece una descripción de los sensores utilizados en el robot bípedo Scout; se revisan sus propiedades físicas; se estudian las ventajas que ofrecen cada uno de éstos sensores y se describen las razones en las que se basó la decisión de su integración dentro del proyecto. Se usa un marco conceptual de cada sensor, describiendo las tareas para las cuales estarán destinados cada uno de ellos.

4.1. Sensores resistivos de fuerza

Los sensores de fuerza son empleados para medir las fuerzas de reacción del suelo ejercidas sobre la planta del robot, como complemento del estudio de la marcha bípeda. El sensado de presión o fuerza dentro de ingeniería es una tarea muy frecuente, por lo que ha generado una amplia variedad de sensores dedicados a ésta tarea.

4.1.1. Propiedades y descripción de los sensores Flexiforce®

Se optó por sensores de fuerza Flexiforce® por su espesor, que permite al robot plantarse sobre ellos sin generar momentos de fuerza que desequilibren al robot durante su marcha; por su flexibilidad que permitió adaptarlos con facilidad sobre la planta del pie del robot; por el tamaño de su área de sensado; por su precisión y repetibilidad y por su simplicidad de uso.

Los sensores Flexiforce® tienen un espesor de 0.2032 mm (0.008 in), cuentan con un circuito impreso flexible que detecta fuerzas en contacto y además dispone de diferentes modelos que permiten medir diversos rangos de fuerza. Los sensores de la marca Tekscan ofrecen [20]:

- Una linealidad y exactitud mayor ($\pm 3\%$)
- Amplio rango de fuerzas
- Gran flexibilidad
- Hay una gran variedad en el tamaño de áreas de sensado
- Soporta altas temperaturas (arriba de los 400°F)

El peso total del robot es de 1258.06 gr (12.34N) con un área en cada pie de 90.17 cm². Considerando estos puntos, el modelo del sensor empleado es un modelo estándar denominado A201 con un rango de fuerza de 110 N.

En la tabla 4.1 se muestran tanto las propiedades físicas como de rendimiento de los sensores Flexiforce® A201.

4.1.2. Caracterización de los sensores Flexiforce®

Debido a que su rango de fuerza es nueve veces más de lo necesario, se caracterizó a los sensores con pesos definidos, dentro de un rango limitado por el peso del robot. Se obtuvieron curvas y se realizó una regresión polinomial de tercer grado con la finalidad de obtener la conversión de voltaje a peso en kg (véase el Apéndice A.1) .

La compañía Tekscan ofrece un manual para conocer las características de sus sensores, como consideraciones que se deben de tener en cada lectura, rendimiento del sensor, propiedades físicas y puntos claves para su calibración. En la figura 4.3 se muestra que la resistencia de los sensores de fuerza disminuye a medida que la fuerza aumenta; es decir, que la resistencia eléctrica es inversamente proporcional a la magnitud de la fuerza actuando sobre los sensores. La calibración de los sensores se apoyó en las recomendaciones que Tekscan ofrece.

La caracterización de los sensores se realizó con ayuda de 3 pesos constantes (dos cargas de 550 gr y una de 175 gr) y la combinación de éstos. Los resultados se adquirieron con ayuda de un circuito eléctrico mostrado en la figura 4.1, un Arduino Mega y el software de desarrollo LabVIEW®. Los datos leídos a través de los sensores se almacenaron en un archivo .xlsx con el objetivo de dibujar una gráfica y obtener una función que describa su funcionamiento al termino del proceso.

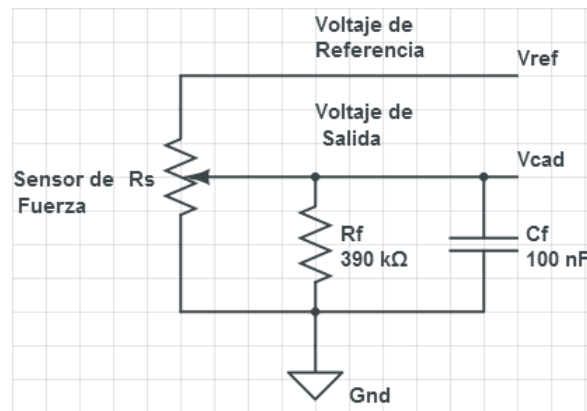


Figura 4.1: Circuito eléctrico usado para acondicionar la señal de los Sensores de Fuerza ubicados en cada una de las plantas del pie del robot Scout.

Para asegurar el contacto entre el suelo, el sensor y el peso o planta del pie del robot (que posteriormente será el peso mismo del robot) los sensores se adaptaron mecánicamente utilizando una configuración similar a la presentada en [21] (ver figura 4.2).

PROPIEDADES FÍSICAS				
Espesor	Longitud	Anchura	Área de sensado	Conector
0.203 mm	197 mm	14 mm	9.53 mm	3-pin cuadrado
RENDIMIENTO				
Error de linealidad	Repetibilidad	Temperatura de operación	Rangos de fuerza	Sensibilidad a la temperatura
<±3	<±2.5 de la escala completa	-9°C a 60°	4.4 N 110 N 440 N	Variación en la salida hasta de un 0.2% por °F
Histéresis	Drift	Tiempo de respuesta		
<4.5 % de la escala completa	<5 % según escala de tiempo logarítmica	<5 microsegundos		

Tabla 4.1: Especificaciones del sensor Flexiforce® A201

Pesos (gr)	Experimento1	Experimento2	Experimento3	Experimento4	Experimento5	Promedio
0	0	0	0	0	0	0
175	1.24	1.78	1.34	1.25	1.03	1.328
550	2.33	2.39	2.2	2.41	2.15	2.296
725	2.71	2.55	2.61	2.58	2.5	2.59
1100	2.91	3.0	2.99	3.0	3.03	2.986

Tabla 4.2: Valores registrados de los 5 experimentos y su promedio de uno de los ocho sensores.

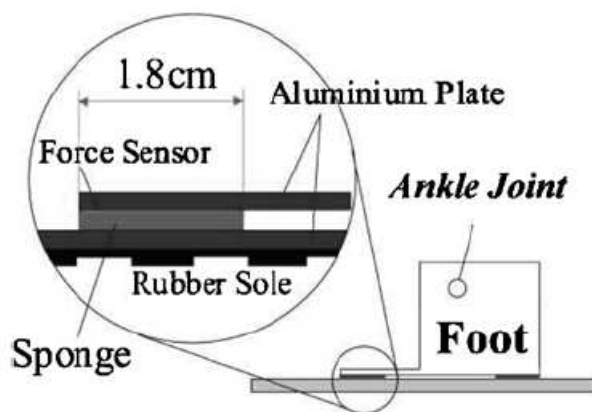


Figura 4.2: Adaptación mecánica de sensores de fuerza.

Se colocó el peso -o alguna combinación de ellos- sobre cada sensor, comenzando con el de menor masa, dejando pequeños intervalos de tiempo entre cada carga para obtener un mayor rango de datos y un mejor análisis. Se repitió este proceso 5 veces para cada sensor obteniendo el promedio de las 5 lecturas (ver tabla 4.2). Al final se realizó una gráfica con cada punto promediado y se describió su comportamiento a través de una regresión polinomial. Las tablas 4.3 y 4.4 muestran los datos que se usan para realizar dicha regresión.

La salida de los sensores es acondicionada a través de un circuito eléctrico (divisor de voltaje) que convierte la fuerza registrada en los sensores a voltaje, a través de la ecuación (4.1). Ésta señal analógica es digitalizada

Pesos (gr)	Sensor1	Sensor2	Sensor3	Sensor4
0	0	0	0	0
175	0.505	0.196	0.316	0.466
550	1.646	0.792	1.5	2.06
725	2.018	1.202	1.86	2.48
1100	2.43	1.66	2.35	3.032

Tabla 4.3: Voltajes registrados en los sensores de fuerza del pie derecho del robot con pesos definidos.

Pesos (gr)	Sensor5	Sensor6	Sensor7	Sensor8
0	0	0	0	0
175	0.488	0.548	1.13	1.328
550	1.648	1.802	2.414	2.296
725	1.956	2.168	2.812	2.59
1100	2.392	2.702	3.246	2.986

Tabla 4.4: Voltajes registrados en los sensores de fuerza del pie izquierdo del robot con pesos definidos.

con el convertidor analógico-digital de Arduino Mega y acondicionada con las curvas correspondientes a su caracterización.

$$V_{cad} = \left(\frac{390k\Omega}{R_s + 390k\Omega} \right) x V_{ref} \quad (4.1)$$

4.1.3. Medición del ZMP a partir de los sensores de fuerza

El robot bípedo camina sobre un plano irregular y con fuerzas externas que lo sujetan a cambios imprevistos, es por eso que es necesario medir las fuerzas de reacción involucradas entre los pies y el suelo.

En [3] se calcula el ZMP analíticamente con base en los momentos inerciales de cada una de las juntas del robot bípedo y su centro de masa total. En [4] se realiza un cálculo teórico del ZMP para la optimización de la marcha basado en la simplificación del péndulo invertido y el peso total del bípedo.

Estos cálculos ayudaron a generar trayectorias asegurando su estabilidad dinámica, los resultados mostrados en ambas tesis son satisfactorias; sin embargo, es necesario cuantificar el ZMP de manera experimental para analizar el desempeño del robot.

Uno de los temas tratados en la sección 2.3.4 de este trabajo habla sobre las diferencias y semejanzas entre el ZMP y el CoP. Si el ZMP cae dentro del polígono de soporte, éste coincide con el CoP. Esto quiere decir, que la magnitud del ZMP depende de la cantidad de puntos en contacto del robot con el suelo y de la variación de fuerzas involucradas.

En [22] se modela el contacto del robot con el plano de desplazamiento como un conjunto de puntos infinitos. La fuerza ejercida en cada unidad de área del pie es distribuida y la fuerza que se registró en cada punto es medida a través de un circuito eléctrico. Modelar dicho contacto como un conjunto de puntos infinitos implica un gasto mayor tanto en cálculos como en implementación, por otra parte ofrece una mayor exactitud.

Durante la fase de soporte simple, cuando una de las piernas se esta balanceando, el comportamiento del robot es similar al de un péndulo invertido. La masa total del robot puede ser reemplazada por una masa concentrada unida a través de la pierna de apoyo del robot. El modelo es representado en la figura 4.4 en donde \mathbf{c} representa la masa concentrada del robot y \mathbf{a} es la articulación del pie de apoyo en el momento de la fase de soporte simple.

Las coordenadas \mathbf{x}, \mathbf{y} corresponden a las posiciones de los sensores con respecto a la articulación del pie en las direcciones correspondientes (\mathbf{i}, \mathbf{j}) .

Dichas direcciones mantienen la nomenclatura propuesta en [3] en donde el bípedo camina sobre el plano sagital en dirección \mathbf{j} y el plano frontal esta en dirección \mathbf{i} .

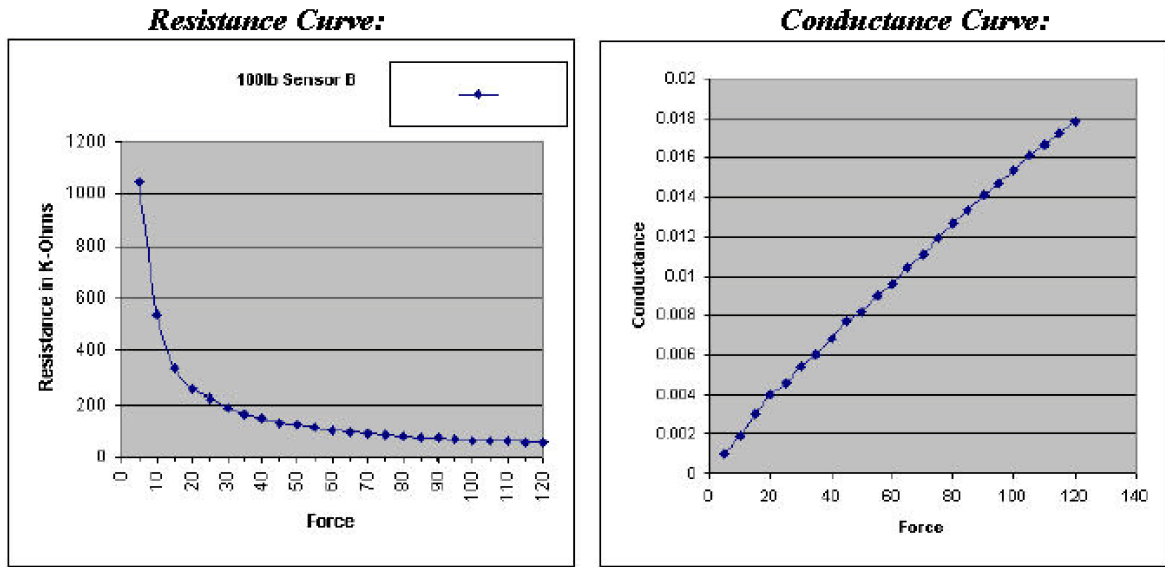


Figura 4.3: Gráficas de calibración de los sensores de fuerza Flexiforce®

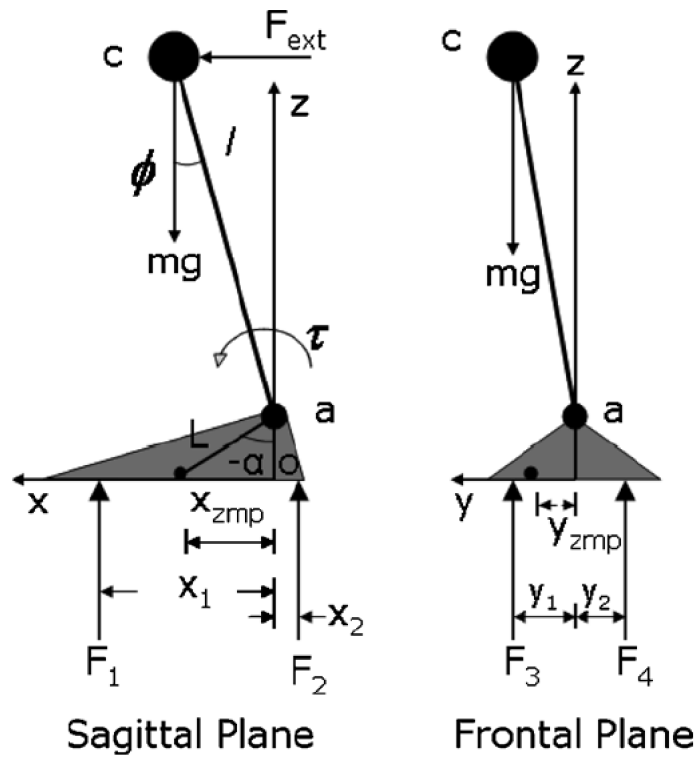


Figura 4.4: Modelo simplificado. Plano sagital/Plano Coronal o Frontal. Imagen tomada de[21]

Los vectores que van de la articulación hacia cada uno de los sensores es constante, esto causará que la variación de las coordenadas del ZMP en los planos sagital y frontal y en torno al polígono de soporte dependa de la magnitud de las fuerzas de reacción medidas por los sensores. El arreglo de los sensores se propuso con la finalidad de medir las fuerzas de reacción en cuatro puntos claves del pie de apoyo del robot; estos puntos claves forman la zona de seguridad dentro del polígono de soporte. Los siguientes cálculos se basaron en la ubicación de dichos puntos mostrados en las figuras 4.5 y 4.6.

El ZMP en dirección \mathbf{j} , se calcula desde el movimiento en plano sagital por la ecuación (4.2).

$$F_1(y_1 - y_{zmp}) - F_2(y_2 + y_{zmp}) = 0$$

$$\implies y_{zmp} = \frac{F_1 y_1 - F_2 y_2}{F_1 + F_2} \quad (4.2)$$

donde,

$$F_1 = f_2 + f_3 \text{ cuando la pierna derecha es el pie de apoyo}$$

$$F_1 = f_6 + f_7 \text{ cuando la pierna izquierda es el pie de apoyo}$$

$$F_2 = f_1 + f_4 \text{ cuando la pierna derecha es el pie de apoyo}$$

$$F_2 = f_5 + f_8 \text{ cuando la pierna izquierda es el pie de apoyo}$$

$$y_1 = 54 \text{ mm}$$

$$y_2 = 13 \text{ mm}$$

El ZMP en dirección \mathbf{i} , se calcula desde el movimiento en plano frontal por la ecuación 4.3

$$F_3(x_1 - y_{zmp}) - F_4(x_2 + y_{zmp}) = 0$$

$$\implies x_{zmp} = \frac{F_3 x_1 - F_4 x_2}{F_3 + F_4} \quad (4.3)$$

donde,

$$F_3 = f_1 + f_2 \text{ cuando la pierna derecha es el pie de apoyo}$$

$$F_3 = f_7 + f_8 \text{ cuando la pierna izquierda es el pie de apoyo}$$

$$F_4 = f_3 + f_4 \text{ cuando la pierna derecha es el pie de apoyo}$$

$$F_4 = f_5 + f_6 \text{ cuando la pierna izquierda es el pie de apoyo}$$

$$x_1 = 13.5 \text{ mm}$$

$$x_2 = 28.5 \text{ mm}$$

4.2. El potenciómetro como sensor de posición

Tal como se menciona en la sección 3.2.3, los potenciómetros forman parte de los sensores usados como instrumentos de medición. Son dispositivos mecánicos ampliamente usados en la industria debido a la simplicidad de su diseño. Una desventaja del potenciómetro como sensor, al ser un sistema mecánico, es el ruido eléctrico que pueden generar sus componentes; sin embargo, la adaptación de éstos como instrumentos de precisión es posible utilizando filtros para eliminar el ruido y acondicionando la señal.

Ya que el robot tiene una configuración establecida que no contempla el uso de sensores de posición en cada articulación, los encoders y cualquier otro dispositivo externo queda descartado como instrumento de medición. Se decidió aprovechar al servomotor que cuenta con un sistema de control y cuya realimentación usa un potenciómetro como lector. Se adaptó su señal a través de una placa que filtra el ruido generado y una conversión de datos, de la señal convertida en volts a grados descritos por el motor.

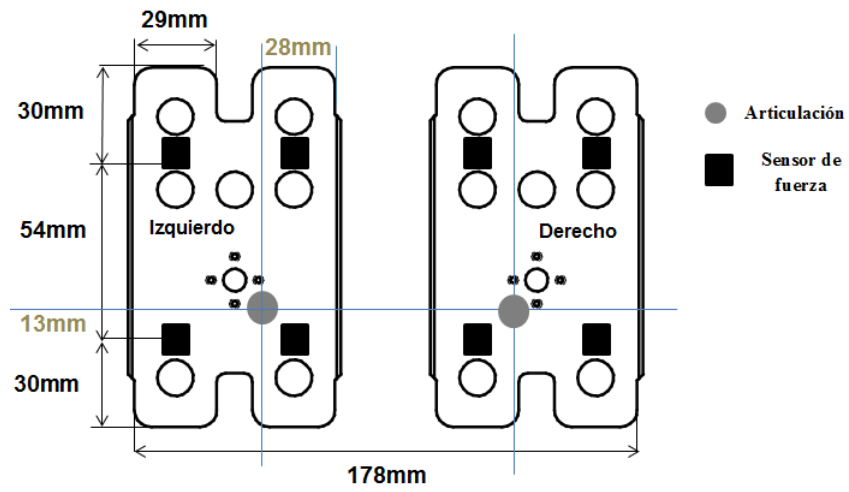


Figura 4.5: Arreglo de sensores sobre la planta de los pies del robot Scout

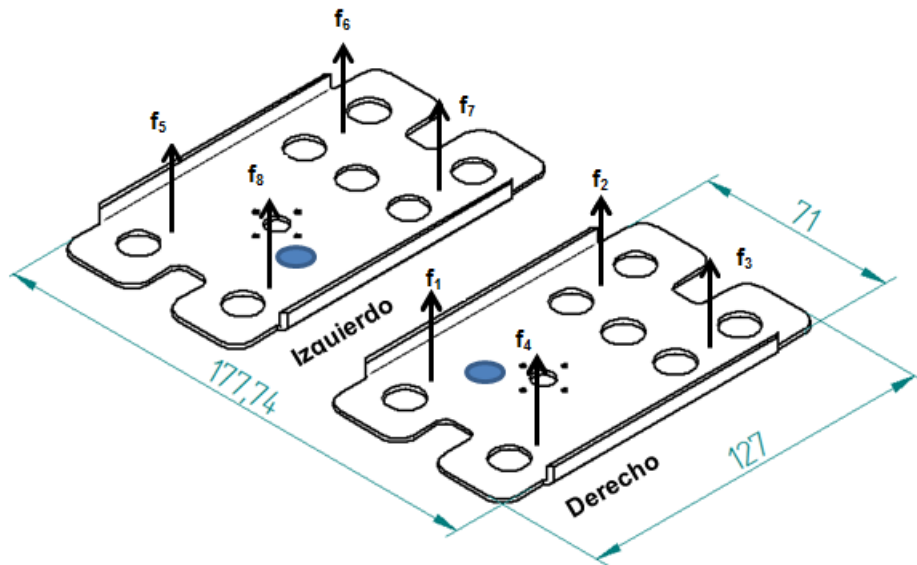


Figura 4.6: Vista dimétrica de la lectura de los sensores de fuerza

4.2.1. Descripción de los servomotores

El robot bípedo Scout cuenta con 12 servomotores Hitec HS-5475 en cada una de sus articulaciones. Su modulación es de tipo digital, cuentan con un par de fuerza de 4.39 kg-cm si son alimentados con 4.8 v y con 5.47 kg-cm si se alimentan con 6.0 v. Tienen una velocidad angular que depende de la diferencia de potencial de entrada con 0.23 sec/60° con 4.8 v y 0.18 sec/60° con 6.0 v. Su peso es de 39.7 g y sus dimensiones son 38.1 x 20.3 x 35.6 mm. El tipo del motor es de 3 polos y sus engranes están hechos de plástico.

Los servomotores fueron programados para tener un rango de giro más amplio que el ofrecido por fábrica (60°); es importante mencionar que con estos servomotores el bípedo es capaz de caminar sólo con el peso mínimo (recomendación de la compañía Lynxmotion®).

Debido a los circuitos que se le han agregado, a los pequeños cambios que pudiesen tener sus juntas debido al uso como banco de pruebas y a la necesidad de tener una retroalimentación de sus juntas rotacionales, este trabajo se dedica a obtener y a comparar los ángulos rotacionales de cada uno de sus servomotores con los calculados teóricamente, a partir de los trabajos anteriores.

4.2.2. Obtención de la posición angular de los servomotores

Un servomotor esta compuesto básicamente de un motor de cd y un circuito de control. Éste motor esta conectado a una resistencia variable, generalmente un potenciómetro. El motor y el potenciómetro están conectados a través de un juego de engranes que además le otorga el par de fuerza señalado. Cuando el motor gira, la resistencia del potenciómetro cambia y le indica al circuito de control cuándo detenerse, ya que el circuito calcula el ángulo y dirección exacta donde se encuentra el eje del motor.

Cuando el eje del motor esta colocado en la posición deseada, la fuente de alimentación del motor se apaga, de lo contrario sigue girando hacia la dirección apropiada. Dicha señal deseada es enviada a través de un cable que hará girar al motor a una velocidad y una cantidad proporcional a la diferencia que exista entre la posición deseada y la posición real.

El esquema de la figura 4.7 muestra el funcionamiento básico de un servomotor. La señal deseada es enviada a través de un pulso codificado y recibido por un amplificador operacional, que se encarga de alimentar al motor con la diferencia que existe entre la salida sensada por el potenciómetro y la deseada. La caja de engranes conecta al motor con el sensor y aplica el par necesario para mover la flecha de salida.

Entre los pasos de sensado y recepción de la señal a través del amplificador operacional, se puede adquirir una señal analógica proveniente del potenciómetro. Dicha señal es una señal de interés debido a que a través de ella se puede conseguir un sensor de posición, según las características que ofrecen los potenciómetros en el subtema 3.2.3. A esta señal se le realizará un acondicionamiento para convertir la diferencia de potencial recibida a los grados exactos marcados por la flecha del motor.

La señal del potenciómetro es enviada a un microcontrolador Arduino Mega programado para fungir como tarjeta de adquisición de datos y procesador, es decir, comunicar al ordenador con Arduino y guardar los datos del servomotor con LabVIEW®. Los datos obtenidos durante el proceso mencionado se guardan en un archivo de texto en dos columnas de datos, una contiene los ángulos deseados y la otra su respuesta, que contiene la diferencia de voltaje obtenida directa del potenciómetro.

Para lograr obtener la imagen de cada dato, se hace uso de la interfaz gráfica de usuario que ya tiene programado el rango de ángulos que puede alcanzar cada servomotor. Se realiza motor por motor y se trabaja cada 20°, de esta manera se obtienen los datos que corresponden en grados y en voltaje.

Finalmente, se realiza una conversión de los datos adquiridos, con ayuda de un programa para hojas de cálculo (Excel), se organizan los datos y se realiza una regresión lineal hasta obtener una función que describa la relación que existe entre el movimiento del bípedo — rotación de los servomotores— y el voltaje adquirido (ver figura 4.8).

4.3. Sensor Inercial

Como se describió en la sección 3.2.1 un sensor inercial cuenta con acelerómetros y giroscopios para saber la orientación y la aceleración de un objeto en el espacio. Como parte de la instrumentación del robot se utilizó

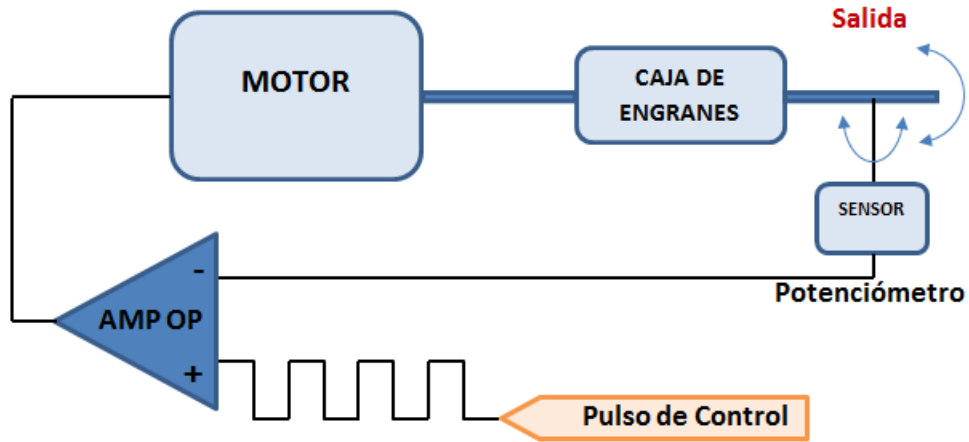


Figura 4.7: Esquema representativo de las partes que conforman un potenciómetro

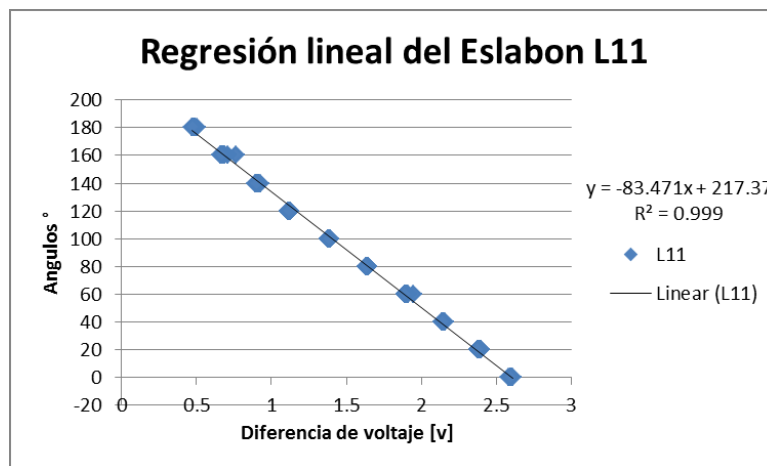


Figura 4.8: Conversión de los datos adquiridos del bípedo. Regresión lineal cuya función entrega la relación entre los ángulos y la señal analógica de los potenciómetros

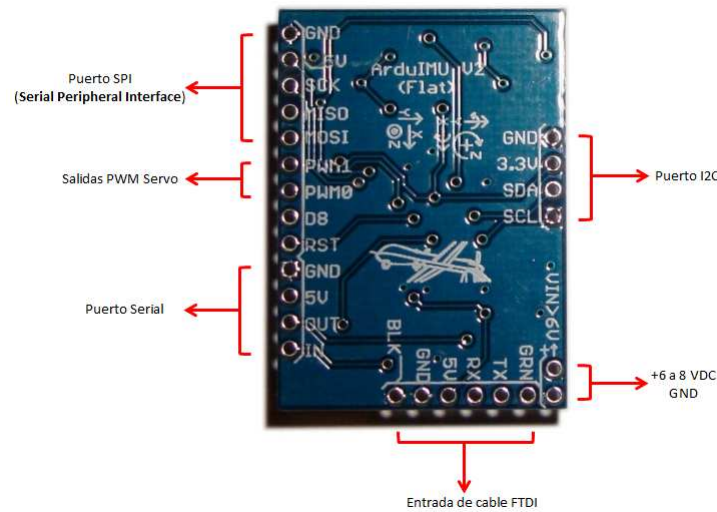


Figura 4.9: Vista inferior del sensor ArduIMU. Descripción de las salidas y entradas

un sensor inercial ArduIMU.

El sensor ArduIMU es una *Inertial Measure Unit* (IMU) compatible con un procesador Arduino. Existen varios tipos de IMU's en el mercado; sin embargo los 6 grados de libertad del ArduIMU+v2, la compatibilidad con el entorno de programación de Arduino (IDE que se ha ocupado dentro del proyecto), su costo y diseño permiten cubrir en su totalidad los objetivos planteados en este trabajo. Sus características se pueden resumir en:

- Diseño pequeño y delgado (27.7x38.6 mm)
- Bajo costo (aprox. \$ 99.90)
- Acelerómetro de 3 ejes (XYZ 3g Acel. ADXL335)
- Giroscopio de 3 ejes (XY giros. LPR530AL, Z giro. LY530ALH)
- Compatible con Arduino
- Código fuente incluido y código abierto (Códigos disponibles en [23])
- LED's indicadores
- Frecuencia de reloj de 16MHz
- Puerto SPI
- Puerto I2C (permite expansión con magnetómetros)
- Puerto GPS
- Dos salidas PWM para servomotores
- Diodo de protección
- Puerto serial de salida con conector de servomotor standard para fácil interfaz con cualquier dispositivo (tierra, 5V, TX-OUT)

4.3.1. Implementación y programación del sensor inercial

Para adquirir los datos de un sensor inercial es necesario realizar un programa que procese los datos tanto del giróscopo como del acelerómetro, ya sea a través de un filtro de Kalman, un PID, una matriz de rotación y dirección o un algoritmo de compensación. Para procesar los datos del sensor ArduIMU se utilizó un archivo que forma parte de el paquete de código de Ardupilot.

El software libre Ardupilot hace uso de un filtro de Kalman para procesar los datos provenientes del acelerómetro y giroscopio. El filtro hace uso de dos estados, siendo el ángulo de inclinación uno de ellos y el otro las *bias* del giroscopio. Se puede traducir la palabra *bia* como **error sistemático**; sin embargo en mecánica el término *bia* puede traducirse como **desplazamiento** o **desviación**, para evitar confusión en la lectura, el término *bias* tendrá la siguiente definición.

Bias Las *bias*, o error de *bias*, de un giroscopio es la señal de salida de dicho componente cuando éste no esta experimentando ninguna rotación. Incluso el giroscopio con mejor desempeño en el mundo tiene alguna fuente de error y las *bias* es una de ellas. Esencialmente, éstas representan una velocidad rotacional [24].

El programa que se encarga de procesar la información del sensor IMU tiene suficientes comentarios para entender someramente el proceso. Así también, ArduIMU y Ardupilot ofrecen la documentación necesaria para su comprensión. Con la finalidad de asociar la descripción del programa con conceptos teóricos, este trabajo ofrece una descripción breve acerca del filtro de Kalman .

4.3.1.1. Filtro de Kalman

El filtro de Kalman surge como solución a las técnicas de mínimos cuadrados usadas para analizar ruido en datos y búsqueda de coeficientes óptimos para un modelo de un sistema de particular interés. Uno de los principales problemas con los filtros de mínimos cuadrados empleados para estimar estados en control está en el procesamiento por bloques; es decir, que el análisis requiere que todos los datos medidos estén disponibles antes de ajustar los datos que pueden ser generados. Por lo tanto, los filtros por mínimos cuadrados no podían ocurrir en tiempo real.

La solución a esto la dio Rudolph Kalman en 1960 y fue un filtro recursivo que permite la estimación de cada dato usando el método de mínimos cuadrados tan pronto como la medición sea tomada.

El filtro de Kalman estima estados basado en un modelo matemático representado por ecuaciones diferenciales lineales; esencialmente, estos estados estimados son generados basados en el modelo estimado y en una escala residual que muestra las diferencias entre las mediciones previstas y las actuales.

El factor de escala, el cual es calculado en cada paso, es referido como ganancia de Kalman e indica la precisión de la medición comparada con la del modelo. Una ganancia pequeña indica baja confianza en la medición, mientras que una ganancia alta otorga prioridad en las medidas al estimar los datos.

Es posible presentar al filtro de Kalman como dos pasos, uno de predicción y uno de corrección. Dentro del paso de predicción se estima un estado x^* basado en el modelo matemático lineal de un sistema representado como

$$x^* = A\hat{x}_{k-1} + Bu_{k-1}$$

Donde \hat{x}_{k-1} es el estado estimado y u representa las entradas del sistema. La matriz de covarianza de error anticipado P^* puede entonces ser calculada como una función de la matriz del modelo A , la matriz de covarianza de error actual P^*_{k-1} , y la matriz de covarianza del ruido del proceso Q que refleja el ruido aleatorio en las entradas de control:

$$P_k^* = AP_{k-1}A^T + Q$$

En el paso de corrección, la ganancia de Kalman es calculada en cada paso k como

$$K_k = P_k^* H^T [H P_k^* H^T + R]^{-1}$$

donde H es una matriz de dimensiones $\mathbf{m} \times \mathbf{n}$ donde \mathbf{m} son las mediciones y \mathbf{n} los estados; el Jacobiano de las mediciones con respecto a los estados y R es la covarianza de ruido de las mediciones y refleja el ruido proveniente de los sensores. El estado estimado entonces se calcula como una función del modelo de predicción, medición y ganancia de Kalman:

$$\hat{x}_k = x_k^* + K_k (z_k - H x_k^*)$$

La covarianza del error P es también corregida con base en la ganancia de Kalman

$$P_k = (I - K_k H) P_k^*$$

Este proceso de predicción y corrección se repite para todo término k . Esta descripción del filtro de Kalman asume que el modelo del sistema es lineal; además, es tomada de un trabajo de grado del MIT [25].

4.3.2. Tarea que desempeña el sensor inercial

El sensor inercial estará ubicado en la parte media de la cadera -eslabón B, según la nomenclatura en [3] - y a través de él se sensorará la orientación del robot, figura 4.10. La tarea que desempeña el sensor en este trabajo es un control de balance en el plano sagital, las características del mismo facilitaron el trabajo debido a su fácil programación y tamaño, entre otras.

Para el control de balance se medirá la orientación en el eje x , su valor se guardará en una variable correspondiente a la señal de error en un control PID y los servomotores se encargarán de compensar la pendiente añadida.

Para el control PID se consideró lo siguiente

- El robot permanecerá en su posición inicial sin movimiento. Debido a la carga computacional que implica almacenar todos los valores en tiempo real, así como las entradas y salidas limitadas por los microcontroladores utilizados en el proyecto
- El sensor inercial medirá todos los valores, tanto de aceleración como de orientación pero sólo ocupará los ángulos registrados alrededor de x , ya que el control PID no hace uso de ningún otro valor
- El movimiento, entonces, estará restringido en el plano sagital dependiente de las articulaciones L51 y L52, ver figura 1.1
- A modo de prueba, el robot estará parado sobre un plano que se inclinará gradualmente y esta pendiente es la que será compensada

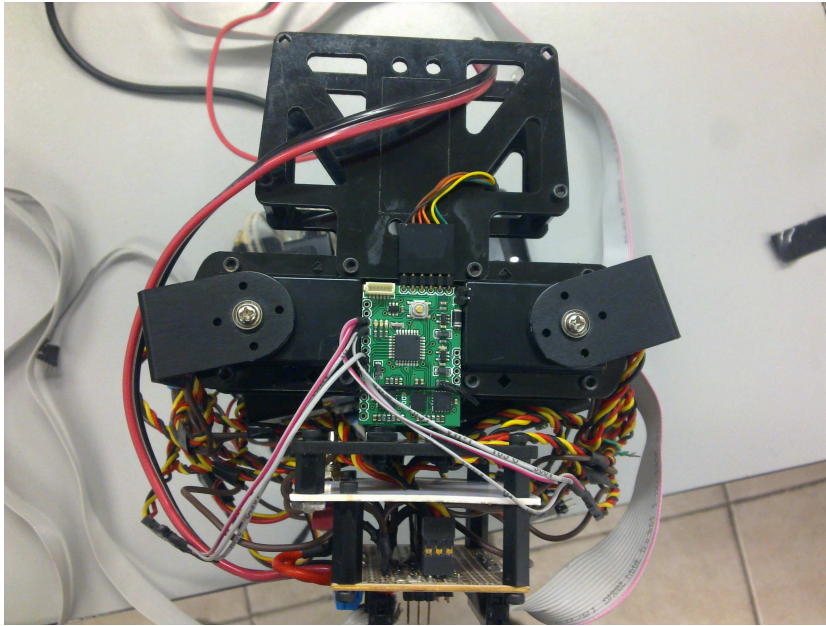


Figura 4.10: Sensor ArduIMU colocado encima de la cadera del robot bípedo

Ardupilot es un software libre que permite su redistribución y/o modificación bajo los términos de la GNU (General Public License). El programa que permite la implementación del sensor se modificó agregando una función llamada `Servo_control` que permite controlar a los servomotores de las articulaciones L51 y L52. Dentro de esa misma sección se programó el control PID, para mayor información referirse al Apéndice B.1.

Capítulo 5

Pruebas en cada sensor y evaluación de resultados

Se implementaron tres diferentes sensores sobre el bípedo con la finalidad de sentar bases para una re-almimentación sensorial en la marcha bípeda. La implementación es un proceso que involucra diseño, acondicionamiento y finalmente pruebas. Se hicieron pruebas a cada uno de los sensores en condiciones relacionadas con sus propósitos. Se trabajó y reportó su funcionalidad.

5.1. Resultados de las juntas rotacionales

Los resultados de los sensores de posición y su evaluación se presentan a través de gráficas que comparan los valores obtenidos contra los valores deseados; las tablas y figuras que indican los errores relativos ayudan a señalar el rango de datos en dónde se originan los mayores márgenes de inexactitud con mayor precisión. Con esta información, se plantean algunas razones que pudieran explicar las diferencias significativas presentadas en los resultados.

5.1.1. Especificación de la prueba realizada

Los servomotores cuentan con un potenciómetro que sirve como sensor dentro del circuito de control proporcional. Como se describe en la sección 4.2.2, se aprovechó dicha señal para medir el voltaje de salida traducido en ángulo en cada una de las articulaciones.

Las señales de los potenciómetros obtenidas a través de la modificación de los servomotores, se utilizaron para comparar y analizar de forma experimental la marcha del robot bípedo. Para adquirir y guardar las señales enviadas desde los motores, se añadió un arreglo de doce filas en la interfaz gráfica utilizada para comunicar al robot con el usuario.

Se hizo la prueba con el robot en marcha - trayectoria previamente calculada basada en algoritmos genéticos [4], que en este trabajo se les denomina valores teóricos deseados - adquiriendo datos con el mismo micro-controlador usado para controlar al robot y se diseñó una pequeña placa basada en capacitores para evitar el ruido generado por los servomotores. Los datos fueron almacenados en un archivo de texto y procesados con un programa en Matlab®. De esta forma se integraron las siguientes gráficas.

5.1.2. Gráficas de las juntas rotacionales

Este apartado se divide en dos subsecciones por el tipo de pruebas que se desarrollaron con los sensores de posición. Los resultados de los motores con una perturbación mayor se muestran en el apartado de: **“Resultados de experimento que resalta aquellos motores con mayor actividad de carga”**, mientras que los ángulos

de los 12 servomotores durante una marcha cíclica se encuentran en “**Resultados de los ángulos de los 12 servomotores durante una marcha cíclica**”.

A pesar del peso extra que se le ha agregado al robot (cables, sensores) los motores responden bien a la señal enviada, con un margen de error debido a los cálculo de la regresión lineal, que son aceptables.

5.1.2.1. Resultados de experimento que resalta aquellos motores con mayor actividad de carga

Para detectar aquellos motores que sufren un mayor esfuerzo, y por lo tanto más vibraciones se realizó un pequeño experimento que se compone de tres etapas, las tres con el robot en posición inicial y estático. Las siguientes gráficas muestran tres líneas de colores diferentes, que representan esas etapas.

En la primera prueba, representada con el color amarillo, se sostuvo al robot de forma que los motores no cargaron con todo el peso; se debe mencionar que algunos motores por su ubicación reciben cargas mayores si el robot se encuentra parado y apoyado sobre el suelo, que al no estarlo. La segunda prueba se muestra en color azul y en esta etapa el robot bípedo es apoyado sobre el suelo y por lo tanto los motores sufren un tipo de esfuerzo distinto a la etapa anterior debido a la fuerza que la tierra ejerce sobre el mismo. Por último, la línea roja representa los valores teóricos deseados. Las figuras de 5.1 a 5.4 muestran los resultados sólo, como se menciona, de los motores que presentaron mayores vibraciones según sus resultados.

Al observar las gráficas hay dos puntos visiblemente importantes, el primero de ellos es la perturbación inicial en la prueba sin esfuerzos (color amarillo) que probablemente sea generada por una reacción del servomotor al querer elevar los eslabones o parte de la estructura que dependen de él o bien es posible que exista algún problema de comunicación entre Arduino y el robot que generan esta gran variación de ángulos al inicio de la prueba. El siguiente punto a remarcar, es el desfase que existe entre los valores teóricos y las otras pruebas. Las figuras 5.4 y 5.2 hacen evidente el desfase de aproximadamente 2 grados cuyo origen se encuentra en la regresión no lineal de los datos, que por definición no es exacta y cuyo residuo se muestra en el Apéndice A.2.

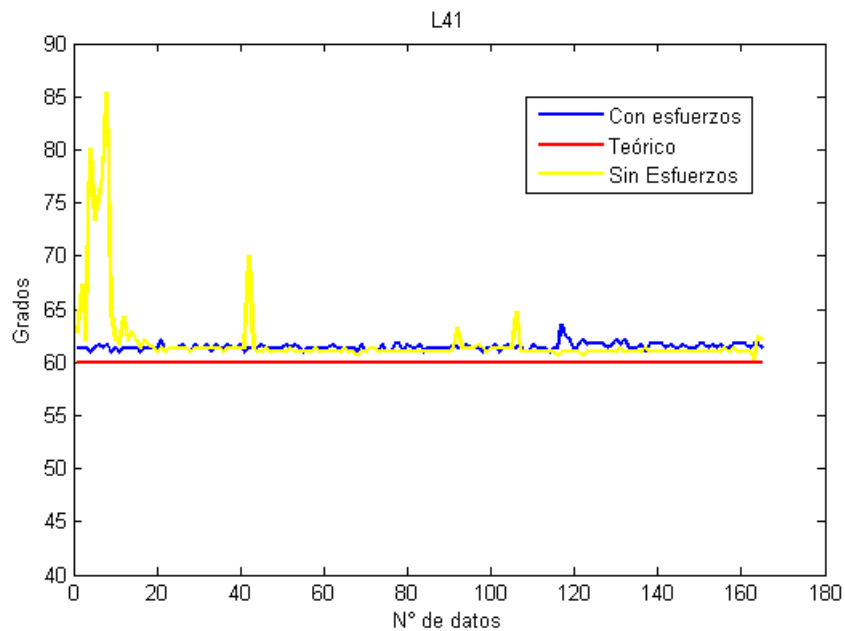


Figura 5.1: Resultado que muestra la variación de grados en posición inicial del motor en el eslabón L41

Una ventaja que brinda la información gráfica, y que se aprovechó en la evaluación de resultados, es la ayuda visual. Si comparamos los cuatro resultados podemos decir que el motor del eslabón 41 es el que menos esfuerzo recibe por la poca variación que tienen sus ángulos (ver figura 5.1). Tiene el pequeño desfase que se menciona y las pruebas con esfuerzo y sin esfuerzo son muy similares.

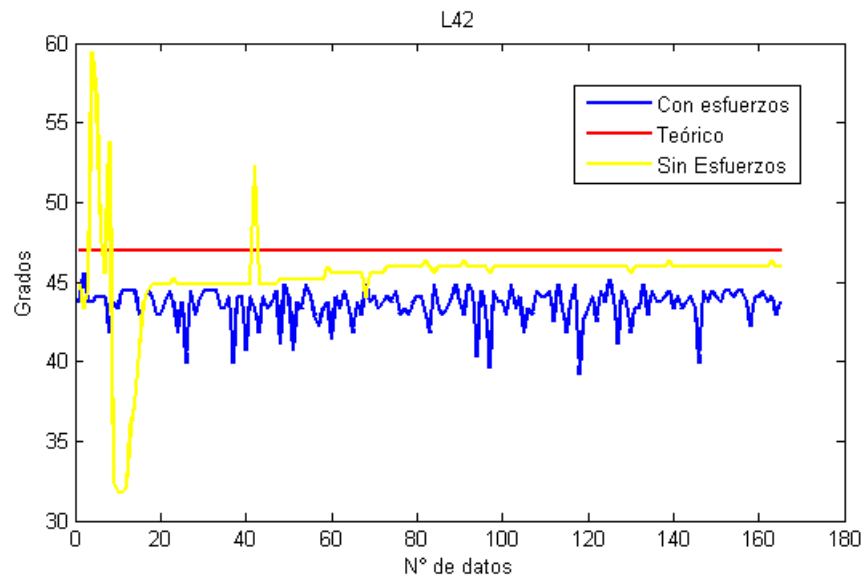


Figura 5.2: Resultado que muestra la variación de grados en posición inicial del motor en el eslabón L42

Aunque se espera que dos motores ubicados en un mismo nivel tengan una respuesta similar la gráfica 5.2 que representa al motor del eslabón 42 no es parecida al del eslabón 41 (figura 5.1). Las razones pueden ser varias y muy diversas, desde las diferencias mecánicas de cada motor (no son el mismo motor) hasta el nivel del piso que no es totalmente plano. La diferencia puede ser ocasionada también, por la posición inicial que probablemente tenga el robot, inclinada ligeramente más hacia la derecha. Lo que se debe destacar es que esa articulación recibe más esfuerzo que su par del lado izquierdo.

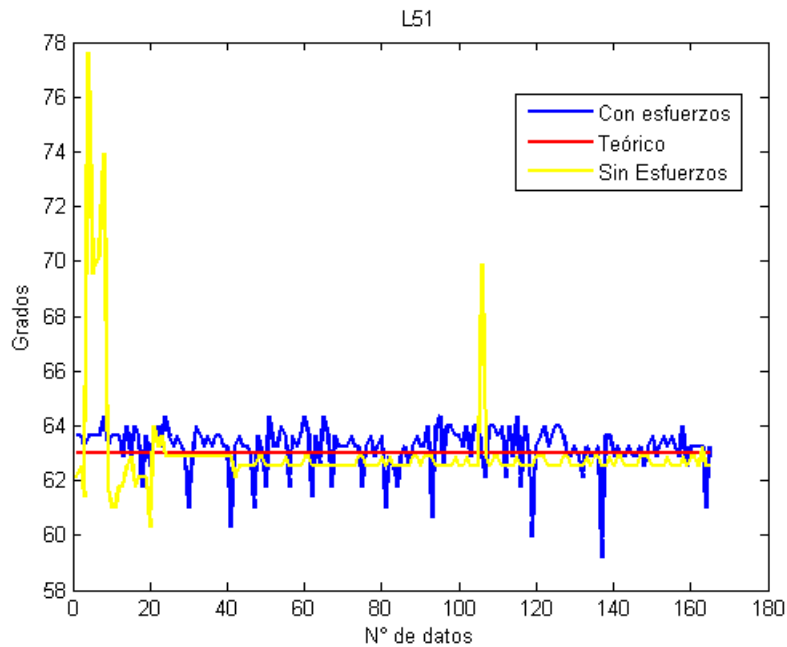


Figura 5.3: Resultado que muestra la variación de grados en posición inicial del motor en el eslabón L51

Las figuras 5.3 y 5.4 muestran las variaciones angulares de los motores ubicados en las penúltimas articulaciones del robot y como se espera también reciben gran esfuerzo debido a su configuración (su eje de giro es perpendicular al peso) y ubicación.

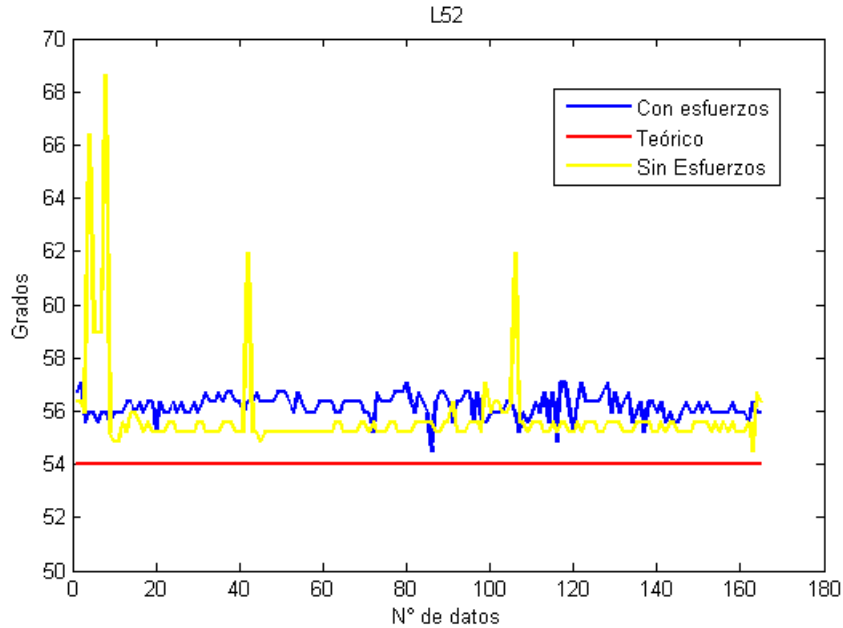


Figura 5.4: Resultado que muestra la variación de grados en posición inicial del motor en el eslabón L52

Una forma de evaluar los resultados numéricamente, es analizando los errores relativos entre ambos (hablar de porcentaje de error), tomando como valor exacto los valores deseados y las cantidades a evaluar las pruebas restantes. A continuación se presenta una tabla que muestra los errores máximos y mínimos de cada junta para cada prueba y dos figuras que representan el porcentaje de error para el resto de los datos.

Prueba	Sin Esfuerzo		Con Esfuerzo	
	Error mínimo (%)	Error máximo (%)	Error mínimo (%)	Error máximo (%)
L42	1.4586	32.4812	3.0495	16.5722
L41	0.3298	42.3928	1.5854	5.9801
L52	0.8707	27.1211	0.8707	5.7063
L51	0.1821	23.2383	0.1821	6.0372

Tabla 5.1: Valores máximos y mínimos del error relativo entre los valores deseados y los prácticos

La tabla 5.1 muestra que el error máximo encontrado es de 42.29% para el eslabón L41 en la prueba sin esfuerzo y el error mínimo encontrado es de 0.1821% para el eslabón L51 en la misma prueba; sin embargo, si se analizan los resultados en conjunto, el valor máximo encontrado corresponde al intervalo de datos asociados a problemas de conexión y no a un comportamiento propio del motor, además de pertenecer al eslabón que gráficamente se consideró con menos carga (L41). Al descartar estos picos de error que se muestran en la figura 5.6 (ya que se pueden considerar no representativos) observamos que el porcentaje de error máximo en esa prueba corresponde al eslabón L42 y es de 5% dejando como error máximo alcanzado al eslabón L42 en la prueba con esfuerzo con un 16.57%.

La figura 5.6 muestra que los picos de error se presentan en los mismos datos para los 4 motores, esto

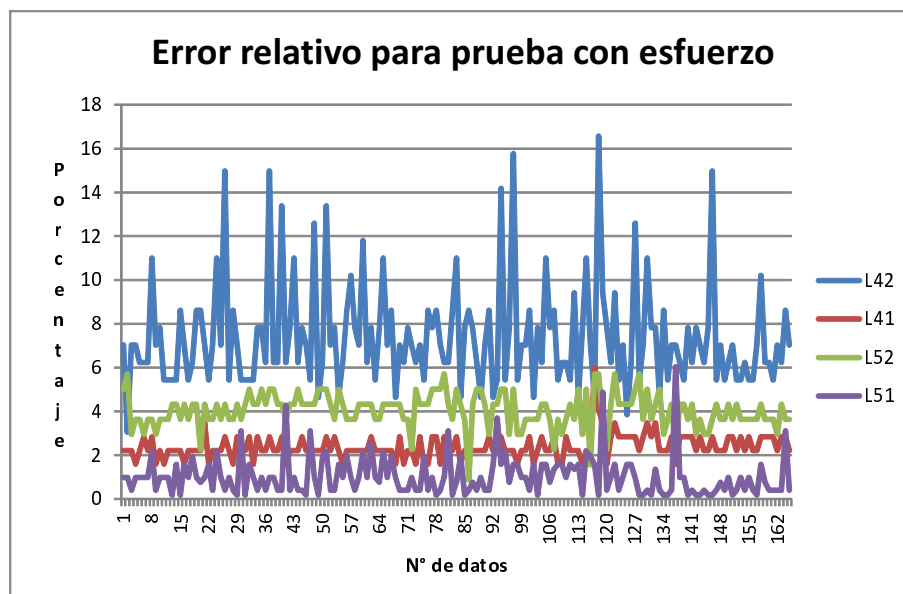


Figura 5.5: Gráfica del error relativo de la prueba con esfuerzo

quiere decir, que probablemente el problema de comunicación se deba a la trayectoria seleccionada, enviando datos que los servomotores no pueden realizar con facilidad. La figura 5.5 muestra, por otra parte, un patrón más desordenado que se puede asociar al comportamiento de los servomotores durante la caminata.

Los valores de error máximo no coinciden en ambas pruebas debido a que el peso amortigua los saltos de valores que se reflejan en la gráfica 5.6, y se concluye que las vibraciones que presenta el robot son un problema ocasionado no sólo por la excitación que reciben por la carga que llevan (continuo intento por alcanzar un valor) sino también por otro agente que genera pulsos indeseados.

Resultados de los ángulos de los 12 servomotores durante una marcha cíclica

Las siguientes gráficas muestran el comportamiento de una marcha de 15 a 20 segundos aproximadamente. Se menciona el tiempo de la marcha para que el lector tome en cuenta que las curvas se repiten en más de dos ciclos. Es importante analizar la repetibilidad a través del sensado de más de un ciclo, así como mostrar las 12 gráficas correspondientes a cada junta para verificar que los ángulos recibidos por el robot sean ejecutados de la forma correcta. El objetivo de medir el voltaje de salida (ángulos de cada motor) y de sensar cada articulación, es identificar problemas o comportamientos que idealmente no se consideran, ya sea por una falla mecánica, de programación o por cualquier otro aspecto.

Como se menciona en otros apartados el robot presenta vibraciones en ciertos puntos de la estructura, y las siguientes curvas muestran esos pequeños cambios bruscos que sufren los servomotores debido a que el par de fuerzas le es insuficiente para seguir el ángulo.

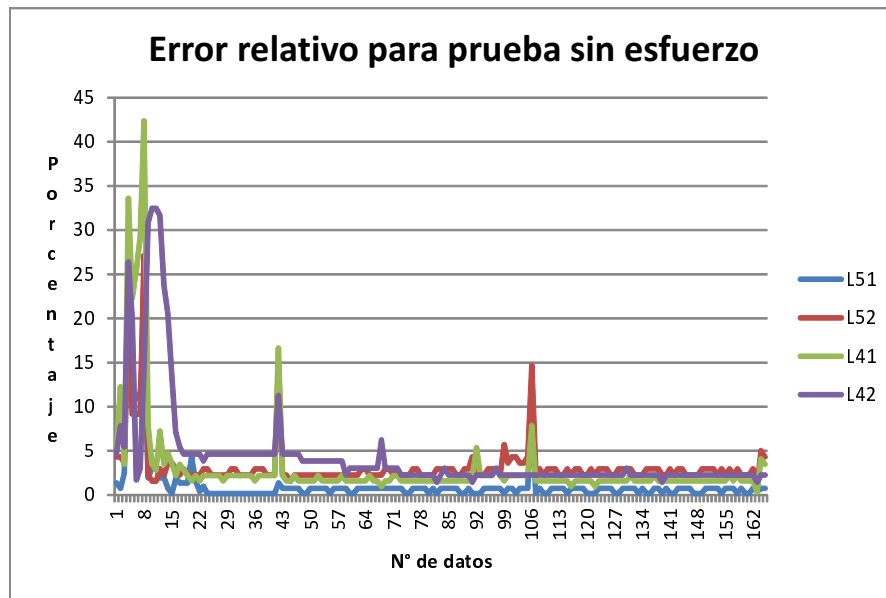


Figura 5.6: Gráfica del error relativo de la prueba sin esfuerzo

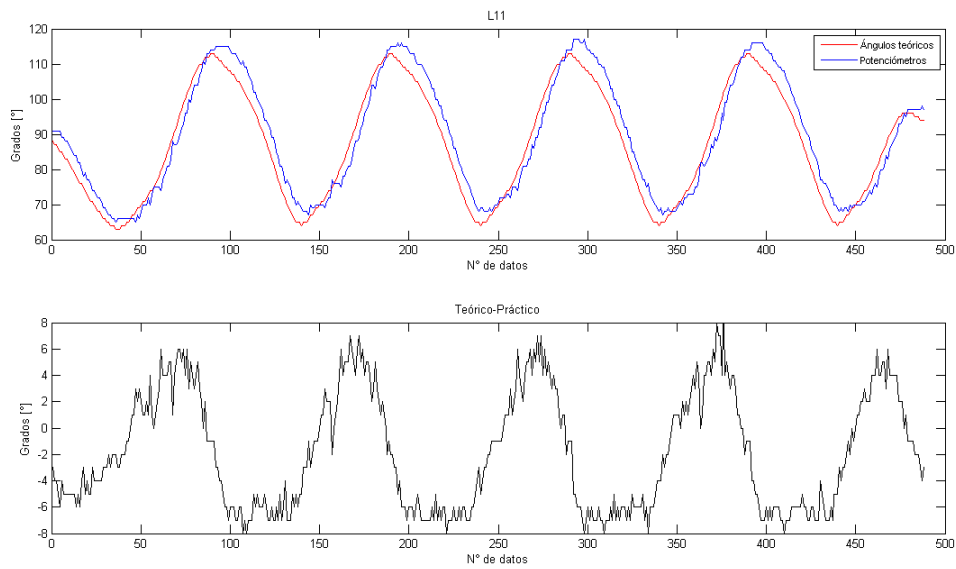


Figura 5.7: Gráfica de los ángulos de las articulaciones θ_{11}

Todas las gráficas de este apartado se dividen en dos figuras: en la parte superior se encuentran las curvas que describen las articulaciones a través de la trayectoria trazada para valores teóricos deseados y prácticos; en la parte inferior se encuentra la diferencia entre los valores deseados y los prácticos (error absoluto).

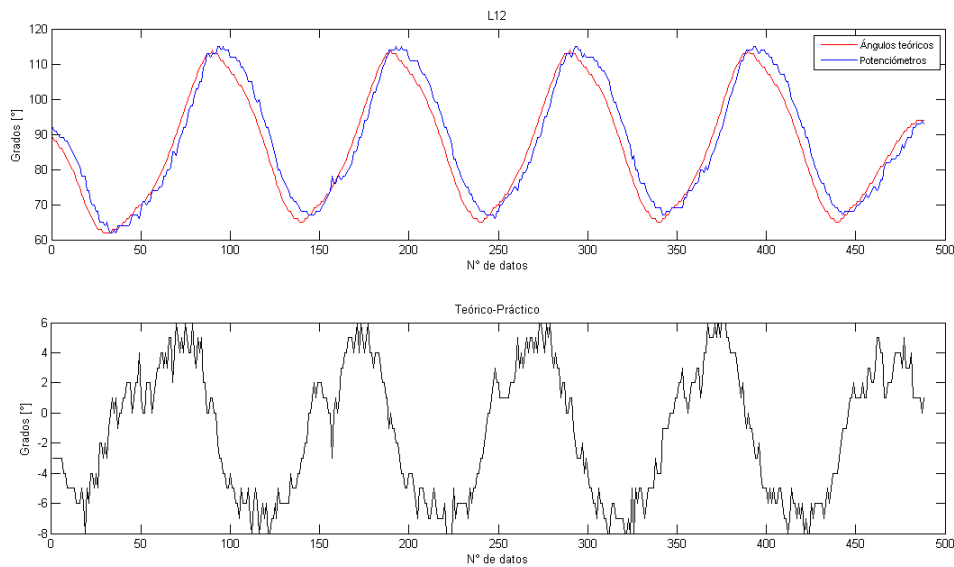


Figura 5.8: Gráfica de los ángulos de las articulaciones θ_{12}

Las gráficas presentadas en las figuras 5.7 y 5.8 corresponden a los primeros servomotores del robot, encargados del movimiento de rotación de la cadera. La diferencia entre las dos curvas, que llamaremos error absoluto o simplemente error, no es mayor a 8 grados (todos los demás motores tienen un error de más de 8 grados). Si consideramos que todos los servomotores están casi en las mismas condiciones, es decir, mismas propiedades y tiempo de uso, la diferencia de errores puede deberse a su ubicación y función (movimientos que generan sobre el robot: rotación, flexión, extensión, abducción y aducción) que desempeña cada motor.

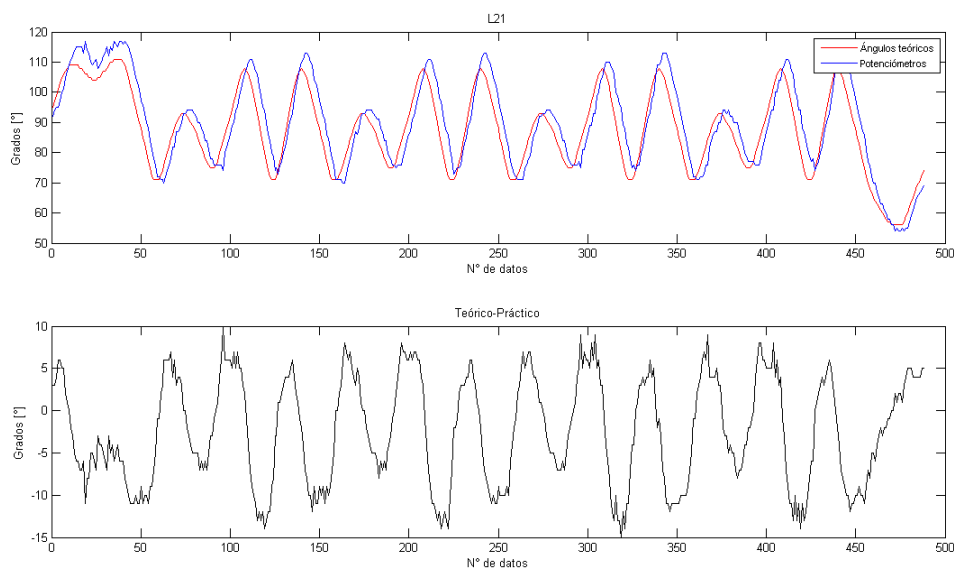


Figura 5.9: Gráfica de los ángulos de las articulaciones θ_{21}

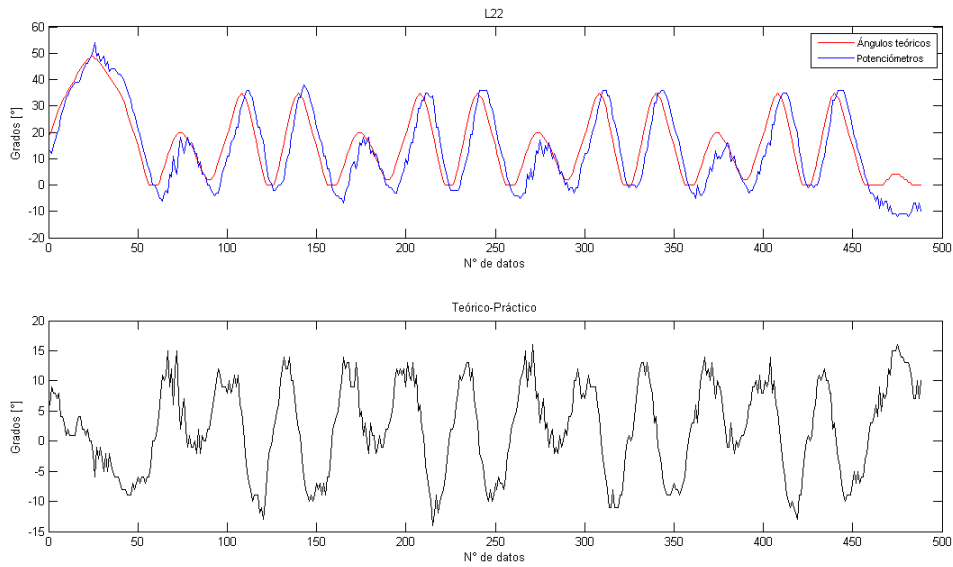


Figura 5.10: Gráfica de los ángulos de las articulaciones θ_{22}

Los ángulos de las articulaciones de los eslabones L21 y L22 representados en las figuras 5.9 y 5.10 respectivamente, tienen un error máximo de 15 grados aproximadamente, 7 grados más que en la articulación inmediata anterior. La diferencia es que estos motores llevan a cabo los movimientos de abducción y aducción, y requieren de mayor energía debido a que el movimiento implica elevar el peso de la pierna.

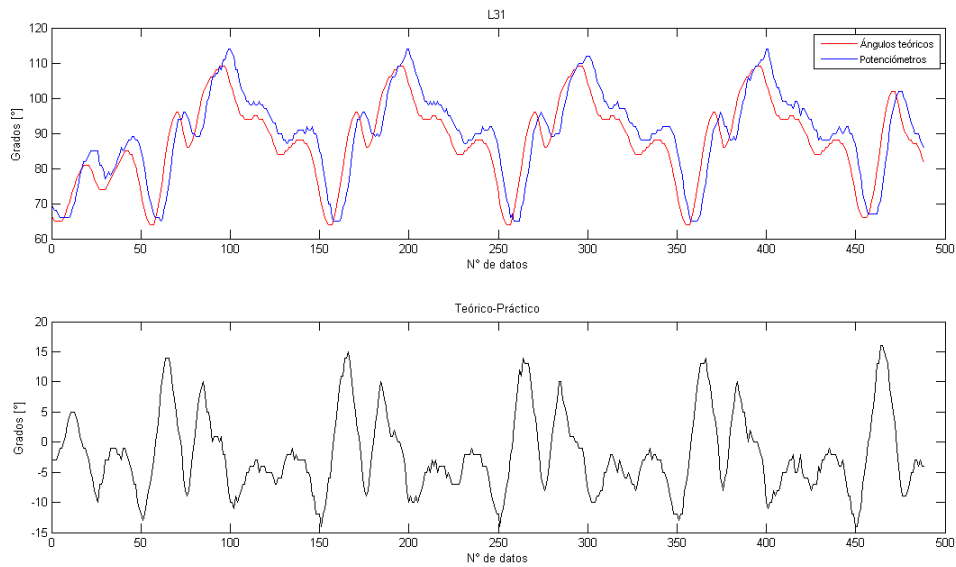


Figura 5.11: Gráfica de los ángulos de las articulaciones θ_{31}

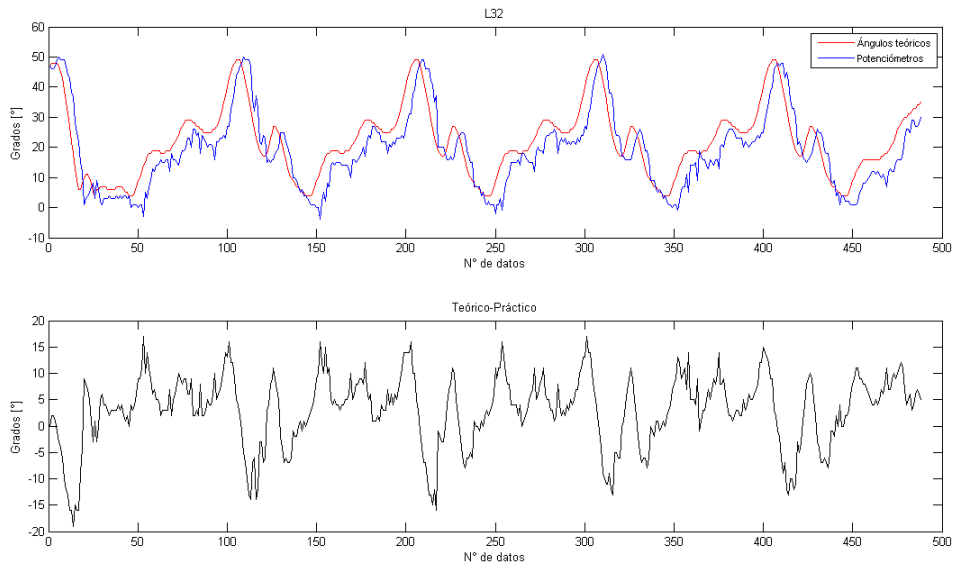


Figura 5.12: Gráfica de los ángulos de las articulaciones θ_{32}

El movimiento que desarrollan los motores 32 y 31 es el de extensión y flexión, y debido a que la marcha es hacia el frente con poca variación hacia los lados (ligero balanceo para compensar el CoP) las figuras 5.11 y 5.12 presentan una movilidad mayor (variación de ángulos) que las figuras 5.9 y 5.10, motores anteriores.

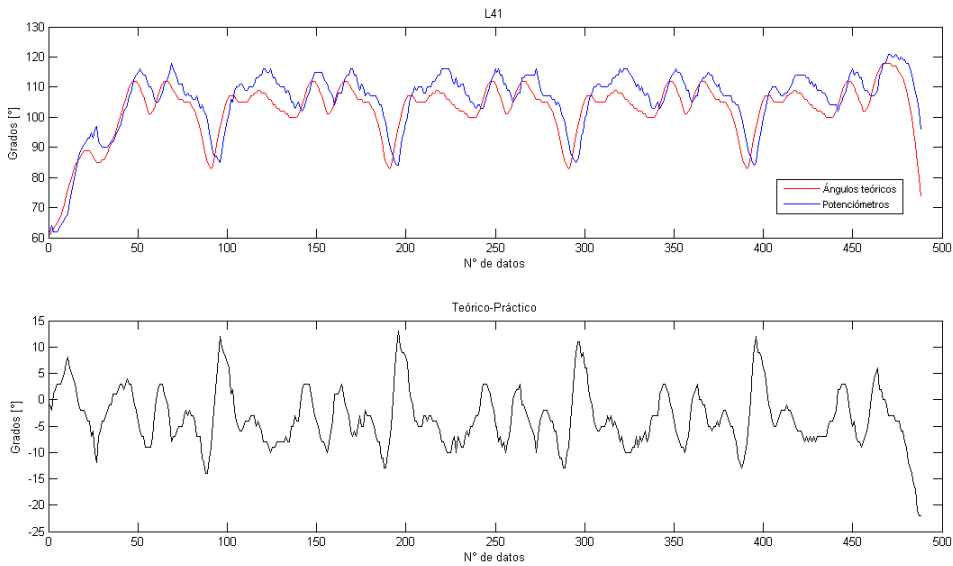


Figura 5.13: Gráfica de los ángulos de las articulaciones θ_{41}

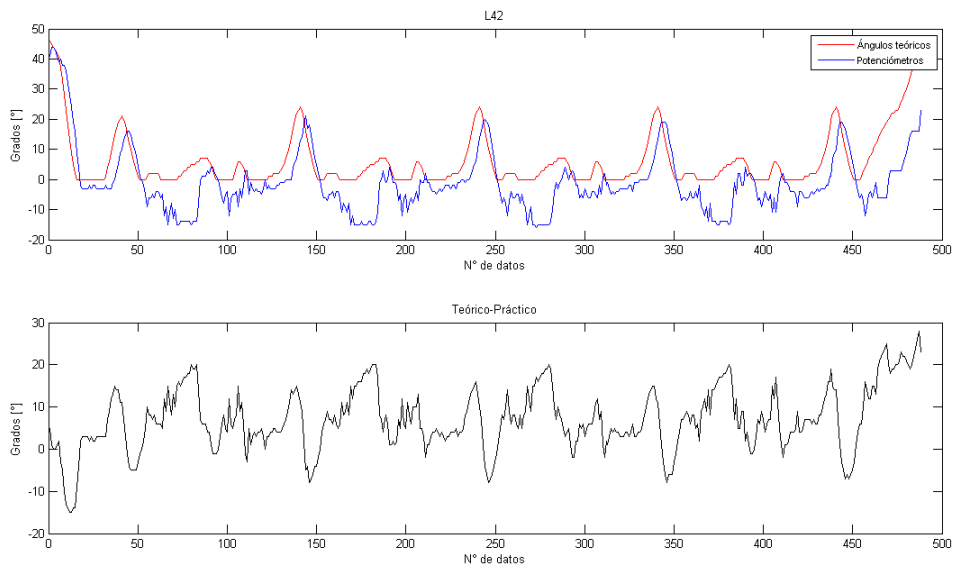


Figura 5.14: Gráfica de los ángulos de las articulaciones θ_{42}

Las curvas de las articulaciones θ_{41} y θ_{42} son los ángulos generados por el movimiento de las rodillas. En la gráfica 5.14 se observa que los ángulos enviados al servomotor no son alcanzados por el robot como debería. Esto se traduce, evidentemente, en una falla en la caminata física del robot. Si se analiza el patrón de ángulos sugerido por el algoritmo se puede observar que existe una limitación forzada en los espacios donde los ángulos medidos por el potenciómetro marcan negativos. Ésto sugiere que existe un filtro que trunca los ángulos negativos (ya que la programación del servomotor no lo permite) y que evita que la marcha sugerida siga un curso previamente calculado o propuesto.

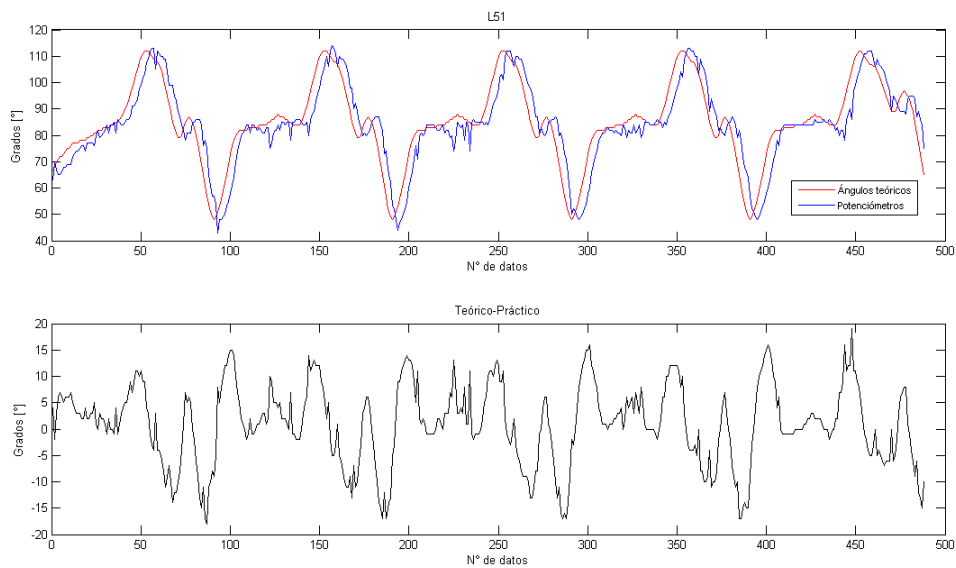


Figura 5.15: Gráfica de los ángulos de las articulaciones θ_{51}

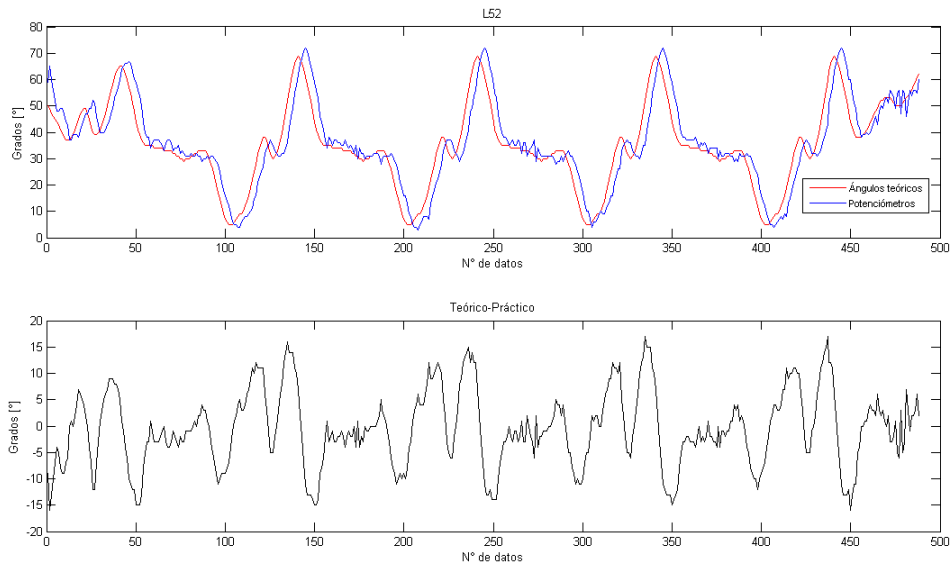


Figura 5.16: Gráfica de los ángulos de las articulaciones θ_{52}

En la sección 5.1.2.1 se mencionan a las articulaciones que desarrollan mayor esfuerzo. Se comprueba que también los motores de los eslabones L51 y L52 tienen diferencias mayores a 15 grados (ver 5.15 y 5.16).

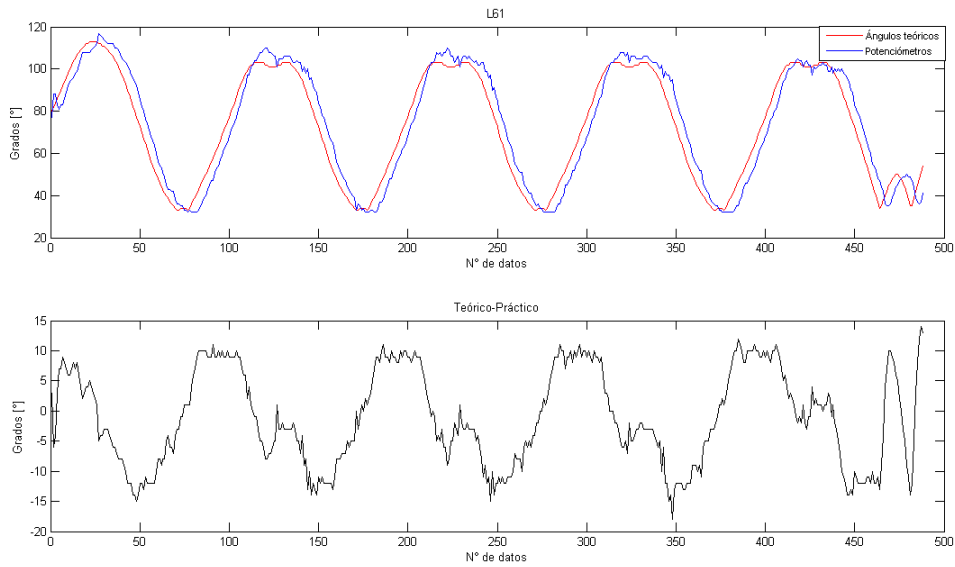


Figura 5.17: Gráfica de los ángulos de las articulaciones θ_{61}

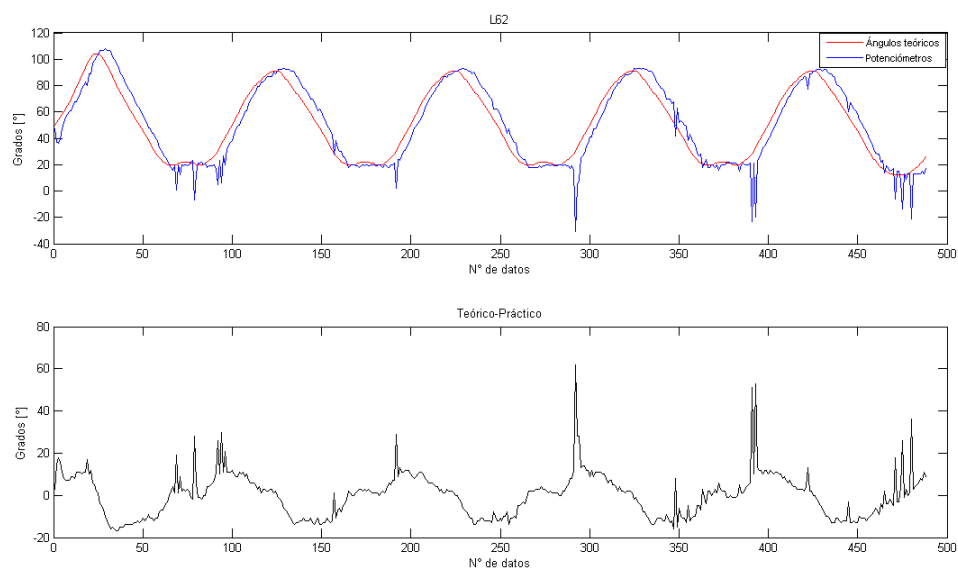


Figura 5.18: Gráfica de los ángulos de las articulaciones θ_{62}

A diferencia de los errores mostrados anteriormente las gráficas de los errores de las figuras 5.17 y 5.18 muestran comportamientos que no se asocian con un movimiento continuo (presenta grandes variaciones aleatoriamente), muestran esos picos que se asocian con ruido o problemas de comunicación.

5.2. Resultados del ZMP calculado a través de los sensores de fuerza.

A continuación se presentan las especificaciones y restricciones de las pruebas realizadas al robot, los resultados que se obtuvieron de los sensores de fuerza, comentarios acerca de la instrumentación y su rendimiento, así como una evaluación estadística que servirá como referencia para evaluar los resultados.

5.2.1. Especificación de la prueba realizada

La magnitud de cada uno de los ocho sensores de fuerza es registrada durante la marcha y guardada en un archivo de texto; un programa realizado en Matlab® utiliza esta magnitud para calcular el ZMP (ver Apéndice B.2). El robot siguió una trayectoria calculada en [4] que no toma en cuenta el ZMP y dio cuatro pasos con los sensores registrando la presión ejercida, el robot no cayó durante la prueba.

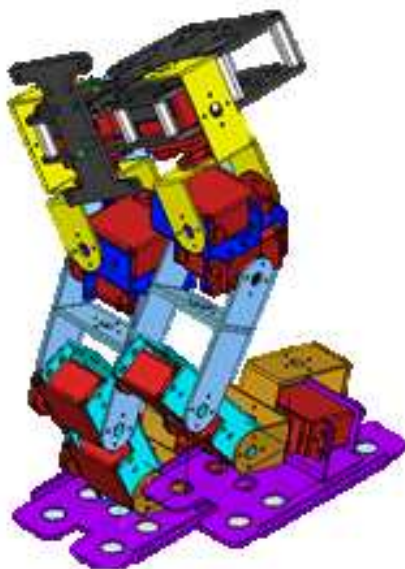


Figura 5.19: Representación de los pasos con un pie fijo y otro libre

5.2.2. Restricciones de la prueba realizada

Éste proceso no se puede realizar en tiempo real debido a que la programación del bípodo se ve afectada en su rendimiento si se pretende realizar más cálculos de los establecidos, así que el primer paso para obtener las siguientes gráficas es poner en marcha al robot junto con los sensores de presión para poder continuar con el cálculo del ZMP y obtener las coordenadas como lo describe el apartado anterior.

Otra restricción ocurre en la fase de doble soporte. La fase de doble soporte no es calculada ya que el comportamiento del bípodo difiere a causa del centro de masa. La fase de soporte simple, tal y como se describe en los capítulos anteriores, describe el comportamiento de un péndulo invertido, por lo que es posible concentrar la masa total del robot en un sólo punto.

A pesar de la última restricción, los datos se guardan incluso en fase de doble soporte, sin embargo, sólo se procesan aquellos que no entran dentro de ésta fase. Cuando los sensores de un pie no están siendo apoyados (excitados por una fuerza externa) los sensores marcan cero "0", esto se traduce en una pierna balanceándose. Se realiza un filtro para aislar cada paso y se realizan los cálculos.

5.2.3. Gráficas del ZMP

Cada figura representa la suela del pie del robot en fase de soporte simple, con un pie fijo y uno libre como se muestra en la figura 5.19. Las coordenadas del ZMP como resultados de los sensores de fuerza son representados con puntos dibujados sobre la planta del pie de apoyo del robot, ambos trazados con el software Matlab®. Cada figura contiene la suela del pie del robot en color azul, la zona de seguridad en color verde delimitada sin un cálculo específico, además tres diferentes marcas que representan: círculos en color rojo al ZMP, el diamante en color azul al valor esperado (ideal) del ZMP y finalmente el pentágono color magenta la media de los valores obtenidos (el valor promedio de los círculos rojos).

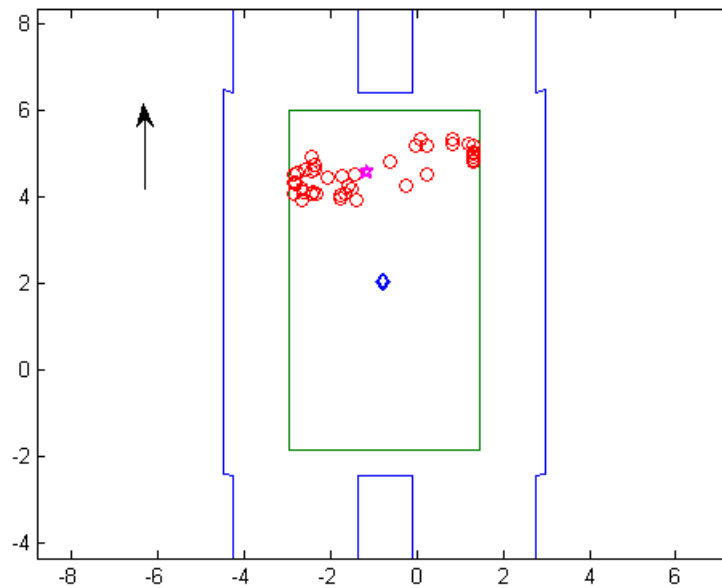


Figura 5.20: Silueta del primer paso del pie de apoyo del robot. ZMP en rojo, zona de seguridad en verde

La figura 5.20 muestra que la mayor parte de la presión en el primer paso se ejerce en la punta de pie. Esta tendencia se genera posiblemente porque la marcha comienza con una posición inicial que implica piernas a la misma altura, el ascenso de uno de los pies provoca que el centro de masa ubicado en la zona media del robot genere, probablemente, una inclinación hacia la dirección del pie en ascenso.

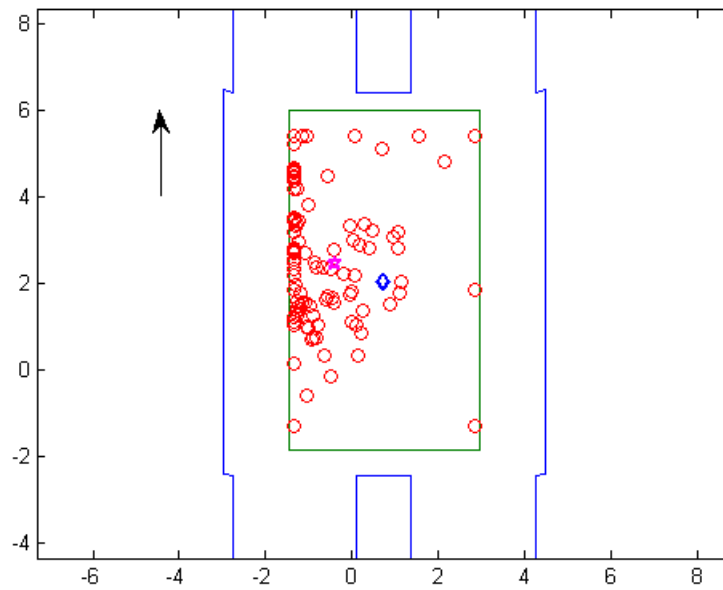


Figura 5.21: Silueta del segundo paso del pie de apoyo del robot. ZMP en rojo, zona de seguridad en verde

La figura 5.21 muestra la huella del segundo paso, más estable que el primero, con una mejor distribución de puntos. La acumulación de puntos del lado izquierdo se debe a la inclinación del robot en su marcha.

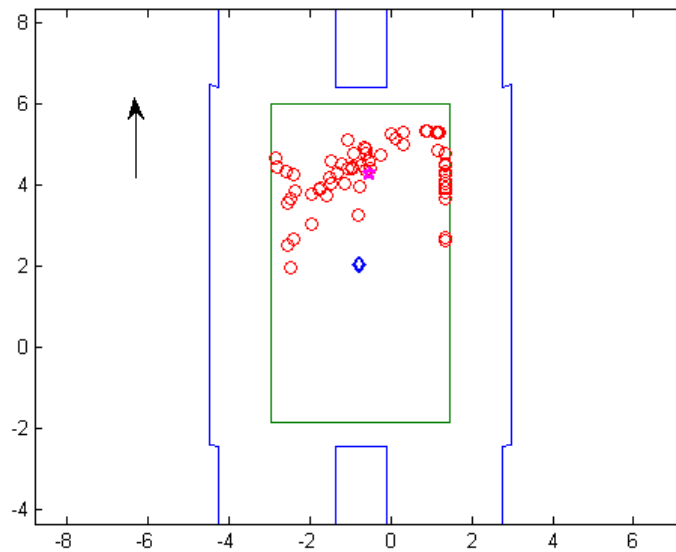


Figura 5.22: Silueta del tercer paso del pie de apoyo del robot. ZMP en rojo, zona de seguridad en verde

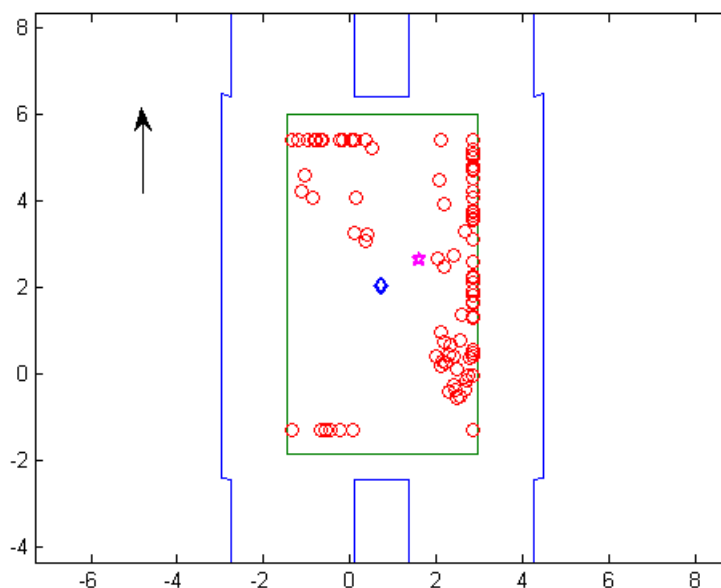


Figura 5.23: Silueta del cuarto paso del pie de apoyo del robot. ZMP en rojo, zona de seguridad en verde

Para evaluar los resultados obtenidos se comparó la media aritmética de los puntos del ZMP contra el valor esperado. Se eligió el valor esperado como el punto medio de la zona de seguridad únicamente basado en la experiencia del observador. La media aritmética o valor promedio ayuda a saber la tendencia que tienen los sensores en cada paso y la evaluación de los puntos en conjunto resulta más fácil si se tiene una representación única de ellos.

Numéricamente la tabla 5.2 muestra el error que tiene cada coordenada (x,y) del ZMP comparada con el punto más céntrico de la silueta del pie del robot. Gráficamente podemos concluir que las figuras 5.21 y 5.23 que muestran el segundo y cuarto paso, muestran una inclinación que es el resultado de la planeación de trayectorias, mientras que los pasos restantes (figuras 5.20 y 5.22) concentran la mayoría de puntos (presión) en la punta del pie. Podemos decir, con ayuda de la tabla de error y las figuras, que la trayectoria y por lo tanto la caminata podría mejorar si considera que la presión del pie debe distribuirse con mayor uniformidad y constancia en cada paso.

Pasos	X_{ZMP}			Y_{ZMP}		
	Media	Esperado	Error (%)	Media	Esperado	Error (%)
PRIMERO	-1.1355	0.907	0.0022	4.5701	2.04	1.241
SEGUNDO	-0.3789	0.907	0.001418	2.4480	2.04	0.20
TERCERO	-0.5164	0.907	0.001569	4.2788	2.04	1.097
CUARTO	1.6357	0.907	0.000804	2.6284	2.04	0.288

Tabla 5.2: Porcentajes de error evaluada para la media aritmética de cada coordenada del ZMP y su valor esperado

Como se mencionó, la elaboración de trayectoria hasta el momento, no toma en cuenta el ZMP y el robot no cuenta con unidad de control, esto ocasiona que el rendimiento de su caminata varíe como lo muestra la evolución del ZMP a través de estas figuras.

El criterio del ZMP concede que si el punto está dentro del polígono de soporte el robot tendrá una caminata estable (se aclara que dicho criterio no basta para asegurar una marcha dinámicamente estable).

5.3. Resultados del sensor inercial

Las siguientes gráficas presentan la respuesta de los componentes involucrados en el control de estabilidad implementado con el sensor inercial ArduIMU como parte de su instrumentación. Los componentes a considerar son los servomotores que compensarán el ángulo desviado por la fuerza externa, la señal realimentada proporcionada por el sensor, captada y procesada por el controlador y la respuesta analógica de los elementos principales del sensor inercial: acelerómetro y giroscopio.

5.3.1. Descripción de la prueba

La instrumentación del sensor inercial ArduIMU se aprovechó para desarrollar tareas de compensación; es decir, el trabajo del sensor es medir la inclinación del robot con respecto a la horizontal y entonces compensar dicha desviación. En una de las pruebas el robot se coloca en una posición donde todos sus motores están apagados excepto aquellos que se encuentran en la dirección debida para compensar la fuerza ejercida sobre el robot. En otra prueba, el robot esta en su posición inicial y la superficie de contacto entre sus pies y el suelo varía cierto ángulo, formando una pendiente; la pendiente será positiva y negativa para abarcar ambas direcciones. En ninguna de las dos pruebas existe otro factor más que el ángulo de inclinación del eje del plano involucrado (sagital).

5.3.2. Gráficas del sensor ArduIMU

La gráfica 5.24 muestra los ángulos de los servomotores ubicados en los eslabones L51 y L52. Ésta indica la evolución del movimiento durante una prueba en la cual el robot es sometido a movimientos constantes y arbitrarios con la finalidad de que los motores coloquen al robot en la posición deseada.

En la gráfica 5.25 se muestra la señal de realimentación que sirve al controlador para corregir el error o en este caso el ángulo medido por la IMU. Se debe considerar que el error y la magnitud del ángulo con el que es perturbado el robot es lo mismo, ya que la magnitud de la perturbación externa a la que se somete el bípedo no tiene otro actor que intervenga en el lazo cerrado.

La programación del controlador PID fue hecha dentro del sensor inercial, en la IDE de Arduino. Las constantes fueron seleccionadas sin algún proceso computarizado o método de control usando función de transferencia, sino se utilizó la experiencia y la experimentación.

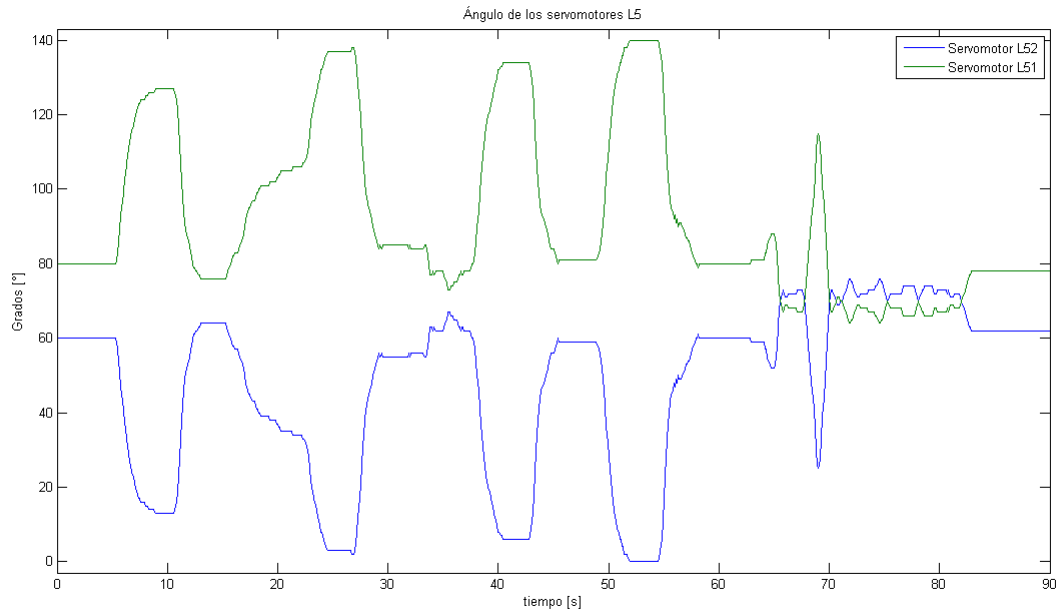


Figura 5.24: Respuesta de los ángulos rotacionales de las juntas L51 y L52 a una perturbación arbitraria

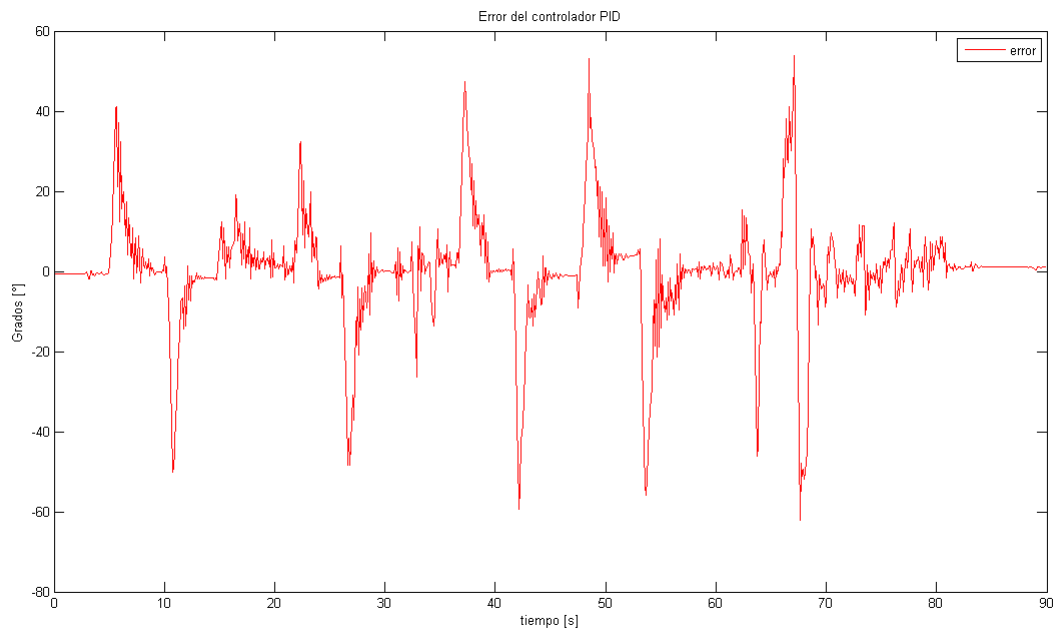


Figura 5.25: Error del controlador PID, representa la magnitud de perturbación añadida

Finalmente, las señales medidas por el giroscopio y por el acelerómetro son presentadas en las figuras 5.26 y 5.27 respectivamente. No se precisa tener una interpretación individual de cada una de estas señales, lo que

se pretende es remarcar la importancia que tienen éstos dos componentes en conjunto. Interpretar las señales aisladamente como un sensor único, ya sea giróscopo o acelerómetro, nos llevaría a cometer un error conceptual de acuerdo a lo apuntado en las secciones 4.3 y 3.2.1; sin embargo, podemos apreciar el comportamiento en cada uno de los ejes y confirmar que su funcionamiento sea el apropiado.

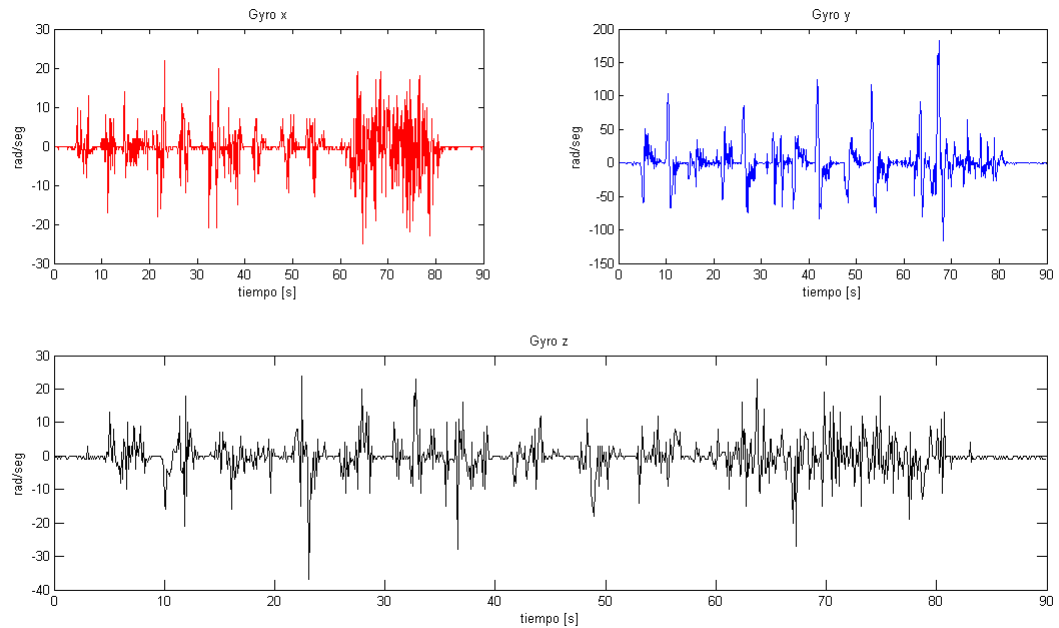


Figura 5.26: Mediciones del giroscopio

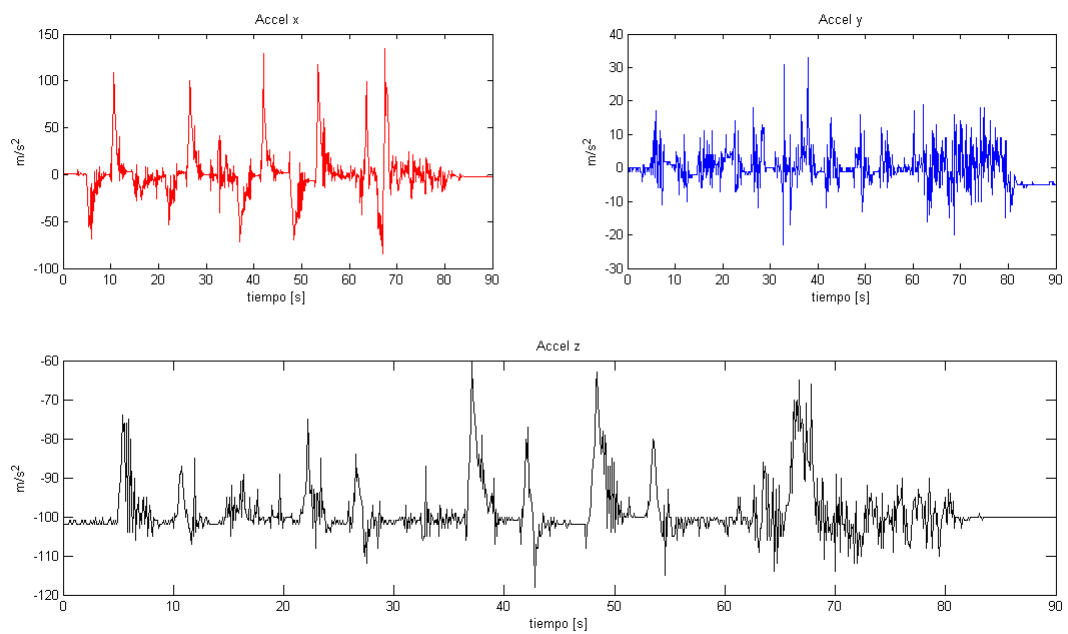


Figura 5.27: Mediciones del acelerómetro

Capítulo 6

Conclusiones y Trabajo a futuro

6.1. Conclusiones

Este trabajo presentó la instrumentación de un robot bípedo de doce grados de libertad que forma parte de una investigación extensa, con el objetivo de brindarle una herramienta más para mejorar el desempeño de la marcha del robot. La instrumentación de los sensores, como muestran los resultados, fueron los esperados, se midió y se pudo comparar el modelo teórico contra el modelo experimental. La disposición de los sensores fue un reto debido a que se partió de un robot bípedo comercial y de una interfaz gráfica ya programada; sin embargo, ya que el proyecto es un trabajo en conjunto se consiguió el apoyo de todos los estudiantes involucrados con el robot, permitiendo hacer adaptaciones tanto en el software como en el banco de pruebas.

El primer paso en el estudio de la robótica bípeda es desarrollar una caminata estable y es uno de los objetivos más importantes que presentan la mayoría de proyectos de robots bípedos. Éste proyecto comenzó con el análisis cinemático y dinámico del robot; esto implica dar los primeros pasos para una caminata bípeda, sin el cálculo de los ángulos rotacionales de cada junta es imposible avanzar. El siguiente paso fue mejorar el trazo de trayectorias del robot, el interés se basa en que el robot siga una trayectoria más estilizada, una trayectoria más natural y que el robot no se vea forzado a movimientos indeseados.

Este trabajo se encargó de dar pie al siguiente paso en un camino sistemático esperado, es decir “aprender a caminar”, saber hacia dónde ir y conocer su entorno. La instrumentación sensorial en un robot es muy importante debido a la carga de responsabilidad que tiene por asumir al robot en un espacio de trabajo. Los sensores de posición brindan información del ángulo generado en cada una de sus juntas, con esto, el usuario puede analizar el comportamiento real que tiene el robot yendo más allá de una simple observación metódica, saber cuánto vale cada ángulo en un movimiento común nos ayuda a determinar si el peso adicional o el propio peso del robot está influyendo sobre sus articulaciones, si los servomotores están trabajando como se espera, si existe alguna fuerza externa que molesta al robot en alguna rotación, entre otras observaciones.

Los sensores de presión ayudaron a calcular el ZMP de una manera distinta a la presentada en [3] y [4], lo que representa un cálculo computacional mucho menor. Además de servir como herramienta para el diseño de control del robot bípedo. La realimentación en sistemas dinámicos como los robots, ayuda a mantener su comportamiento durante su ejecución de tareas. Es un recurso en el área de control debido al cual la señal de salida del sistema completo o de una parte del mismo se compensa a través de una señal de entrada deseada y se corrige cualquier error de comportamiento. Un ejemplo claro es la corrección de ángulo de inclinación que se llevó a cabo en este trabajo con un sensor inercial colocado en la parte alta del robot y que compensaba, con ayuda de un control PID, cualquier perturbación en un plano sagital.

Por último, cabe mencionar que el proyecto ha padecido pausas en su desarrollo y se considera que las razones no recaen en el personal de la facultad sino en razones que se encuentran fuera de su alcance, como el presupuesto destinado a la educación y tecnología en el país. El alumno de ingeniería compite contra grandes países cuya inversión en ciencia y tecnología supera el 3% de su PIB (Finlandia 3.96) esto no quiere decir que la calidad del alumno de ingeniería sea menor o en la misma proporción de su inversión sino que se ve limitado tanto en infraestructura como en el alcance tecnológico que le pueda brindar. Y es que resulta alarmante saber

que otros países invierten 8 veces más en sus investigaciones científicas que México (México 0.47 % de su PIB) [26].

6.2. Trabajo a futuro

Ya que este trabajo es el principio de la instrumentación electrónica del robot, deja la bases para comenzar un enfoque en diversas líneas de investigación que darán continuidad al proyecto.

Como propuesta, desarrollar un sistema a través de redes neuronales o utilizando el algoritmo genético empleado con anterioridad que sea capaz de aprender en línea. Es posible usando el cálculo del ZMP realizado con los sensores de presión y las características del paso del robot o mediante un aprendizaje artificial (redes neuronales).

El sensor inercial solamente controla dos de los doce servomotores del robot en posición estática. Una propuesta es establecer el sensor inercial durante la marcha del robot. Esto brindará la oportunidad de caminar sobre pendientes o subir escaleras.

Desarrollar trayectorias futuras óptimas utilizando los sensores proporcionados.

Dentro del área de control como trabajo a futuro se propone:

- Control de aterrizaje del pie: Controlar la reacción del suelo, utilizando la fuerza de reacción como indicador de irregularidades, perturbaciones o cambios de nivel; ejecutar medidas de control como corrección.
- Control del ZMP: Teniendo la medida del ZMP emitir una señal de corrección hacia las articulaciones lo más rápido posible con el objetivo de cambiar la posición del ZMP a una ideal.
- Aprendizaje como medida de control: El objetivo es realizar los ajustes necesarios a la trayectoria planificada de tal forma que el siguiente paso contemple las perturbaciones presentadas y realice una compensación.

En [21] se presenta un cálculo que permite compensar una perturbación mediante un par de fuerza. Dicha compensación se obtiene con el vector resultante de los sensores de fuerza, el intervalo de muestreo del microcontrolador y el momento de inercia.

Aparte de las tareas propuestas existen muchos otros puntos dentro del área de la robótica bípeda que pueden ayudar al proyecto a ir hacia adelante. Aquí se propusieron, bajo cierta consideración, solamente aquellos que están directamente relacionados con la instrumentación sensorial desarrollada en este trabajo.

Bibliografía

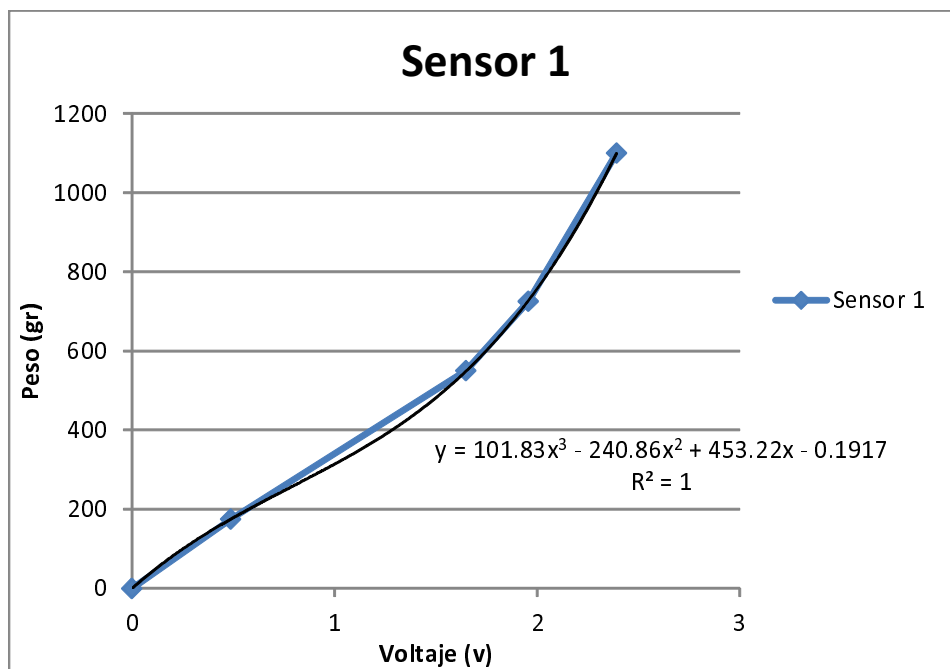
- [1] E. Muybridge, "Freeze frame." Disponible en http://americanhistory.si.edu/muybridge/htm/htm_sec1/sec1.htm. 5
- [2] J. Vermeulen, "Trajectory generation for planar hopping and walking robots: an objective parameter and angular momentum approach," *Vrije Universiteit Brussel, Faculteit Toegepaste Wetenschappen*, 2004. 5
- [3] N. A. Octavio, "Modelo cinemático y dinámico de un robot bípedo de doce grados de libertad internos," Master's thesis, Universidad Nacional Autónoma de México, 2010. 6, 7, 8, 38, 46, 68
- [4] L. G. Rafael, "Planificación y optimización de la caminata de un robot bípedo," Master's thesis, Universidad Nacional Autónoma de México, 2012. 6, 38, 48, 60, 68
- [5] G. PALLIS, "Fallis," Jan. 17 1888. US Patent 376,588. 11, 12
- [6] J. Mrozowski, J. Awrejcewicz, and P. Bamberski, "Analysis of stability of the human gait," *JOURNAL OF THEORETICAL AND APPLIED MECHANICS-WARSAW*, vol. 45, no. 1, p. 91, 2007. 15
- [7] M. Vukobratovic and B. Borovac, "Zero-moment point-thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004. 16, 17, 18, 19
- [8] A. Goswami, "Postural stability of biped robots and the foot-rotation indicator (fri) point," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, 1999. 18
- [9] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure-zero moment point," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 34, no. 5, pp. 630–637, 2004. 19
- [10] S. Vázquez, "Desarrollo de la marcha," 2005. 21
- [11] T. Audesirk, G. Audesirk, and B. Byers, *Biología: La vida en la Tierra*. Pearson educación, 2003. 21
- [12] K. Moore, A. Agur, and M. Moore, *Anatomía con orientación clínica*. Editorial Medica Panamericana Sa de, 2007. 22, 25
- [13] "Anatomia oido." Disponible en <http://anomiadeoido.blogspot.mx/>. 22
- [14] M. Irusta, "El tacto." Disponible en http://usuarios.multimania.es/neurofyk/docs/neurokinesio/3_SENSIBILIDAD.pdf. 23, 24
- [15] G. A. Chevallereau Christine, Guy Bessonnet and Y. Aoustin, *Bipedal robots: modeling, design and walking synthesis*. Wiley-ISTE, 2009. 24, 29, 30
- [16] "Mems." Disponible en http://www.mecatronica.ith.mx/Publicacion_MEMS.pdf. 26
- [17] "Acelerómetro." Disponible en <http://www.uv.mx/cienciahombre/revistae/vol22num2/articulos/microace/index.html>. 28

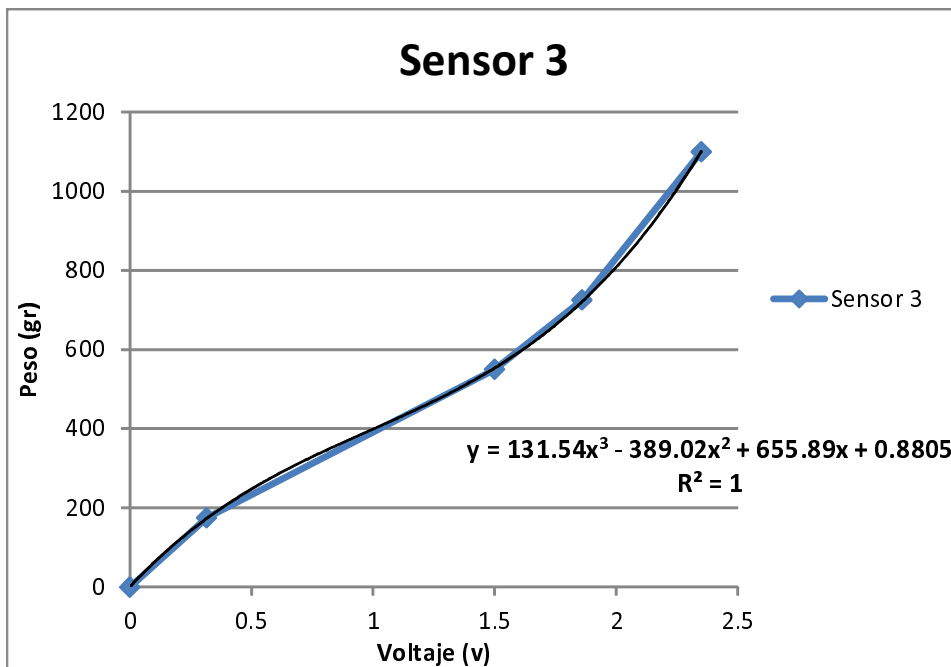
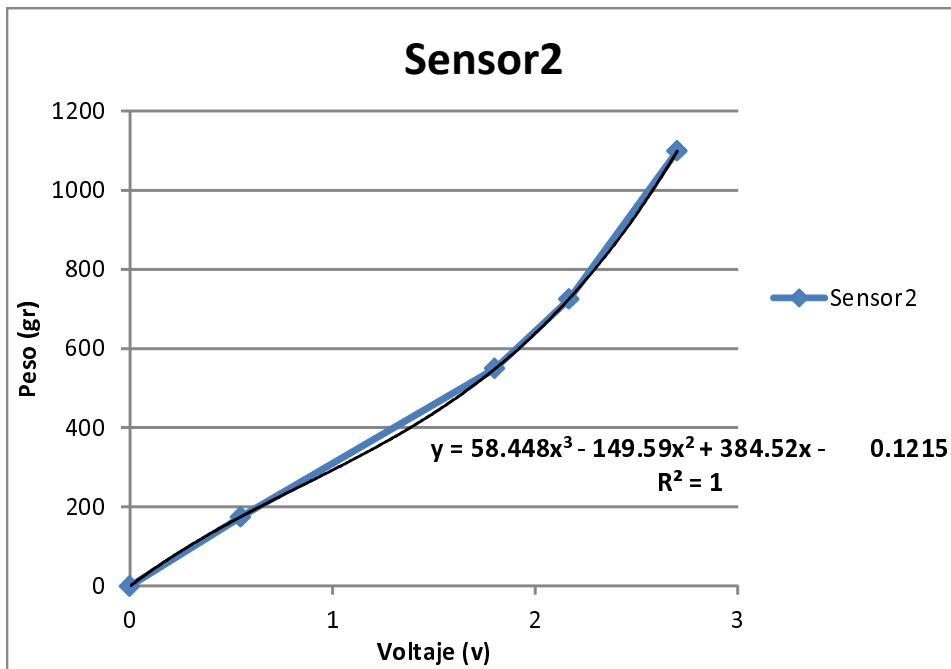
- [18] "Sensores." Disponible en <http://www.isa.cie.uva.es/~maria/sensores.pdf>. 30, 31
- [19] "Nao de aldebaran." Disponible en <http://www.aldebaran-robotics.com>. 33, 34
- [20] Tekscan, "Flexiforce® sensors." Disponible en <http://www.tekscan.com/flexible-force-sensors>. 35
- [21] V. Prahlad, G. Dip, and C. Meng-Hwee, "Disturbance rejection by online zmp compensation," *Robotica*, vol. 26, no. 1, p. 9, 2008. 36, 39, 69
- [22] M. Shimojo, T. Araki, A. Ming, and M. Ishikawa, "A zmp sensor for a biped robot," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1200–1205, IEEE, 2006. 38
- [23] "Arduimu." Disponible en <https://code.google.com/p/ardu-imu/downloads/list>. 44
- [24] Sensorwiki, "Bias." Disponible en <http://www.sensorwiki.org/doku.php/sensors/gyroscope>. 45
- [25] G. Angelosanto, "Kalman filtering of imu sensor for robot balance control," Master's thesis, Massachusetts Institute of Technology, 2008. 46
- [26] "Pib." Disponible en http://www.mexicoylacuencadelpacifico.cucsh.udg.mx/noticia/mayor_inversion_del_pib_para_ciencia_y_tecnologia. 69

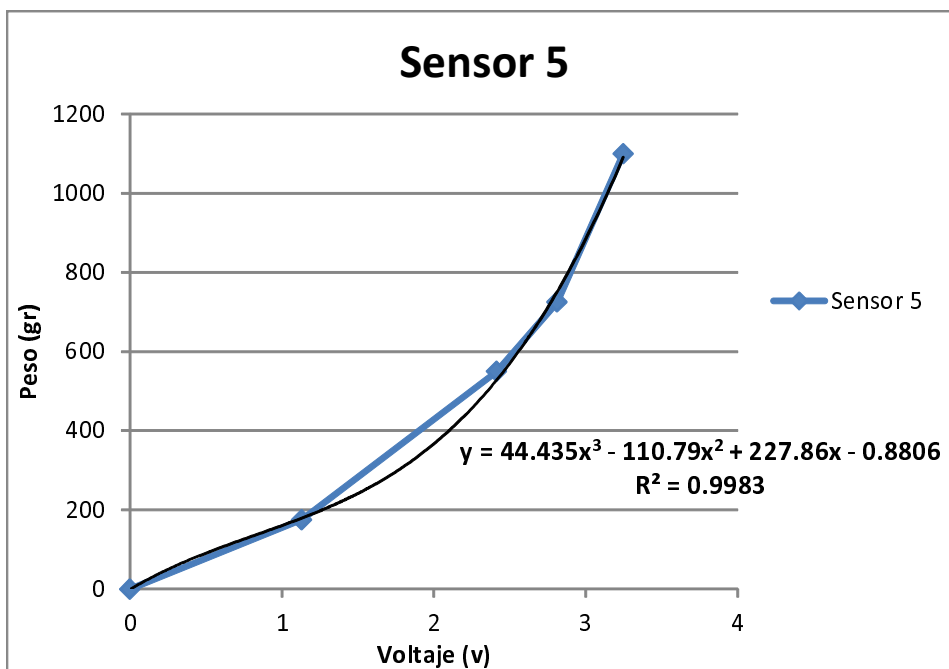
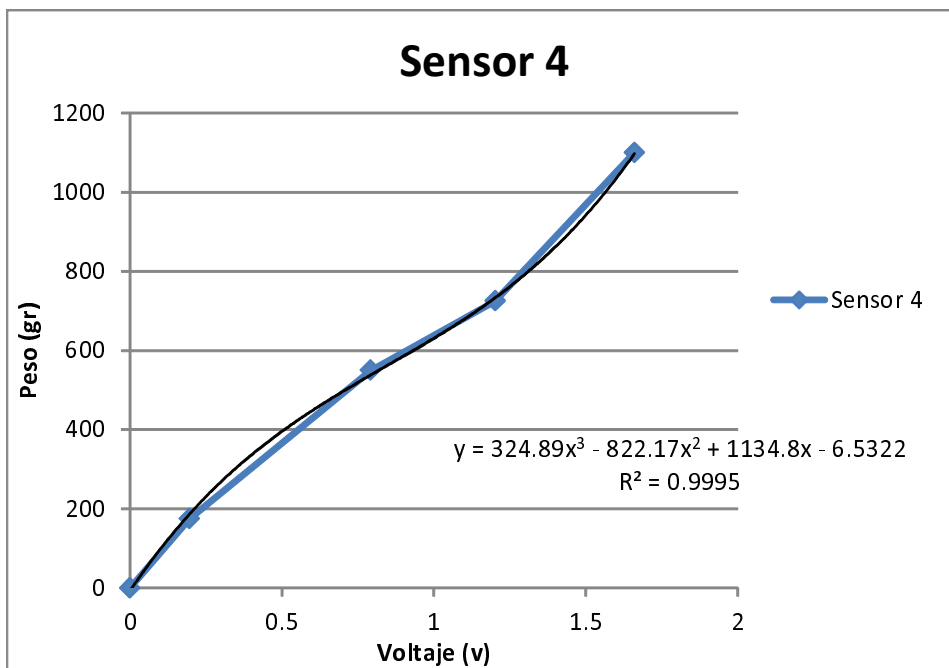
Apéndice A

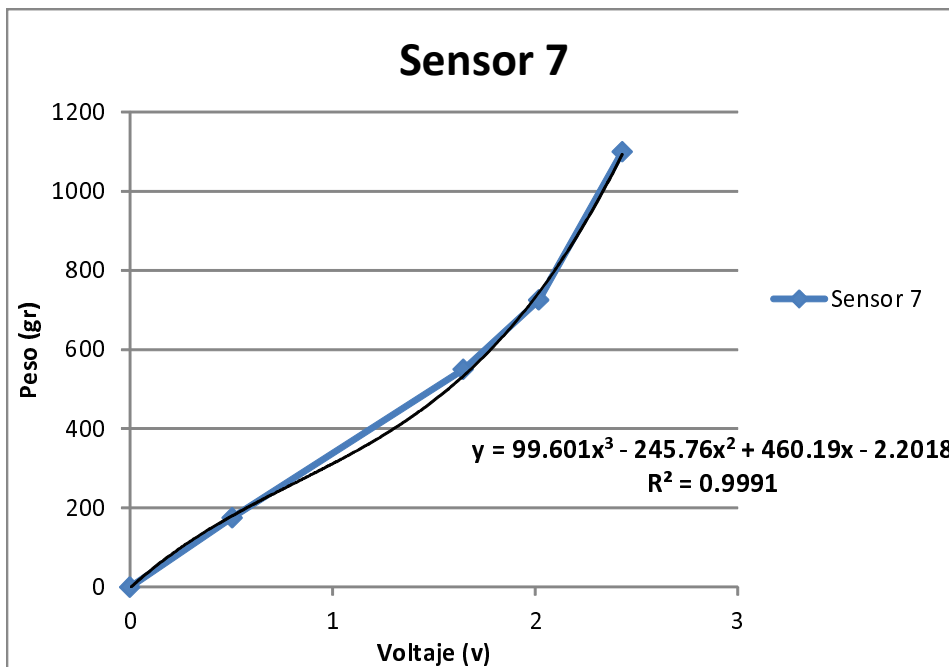
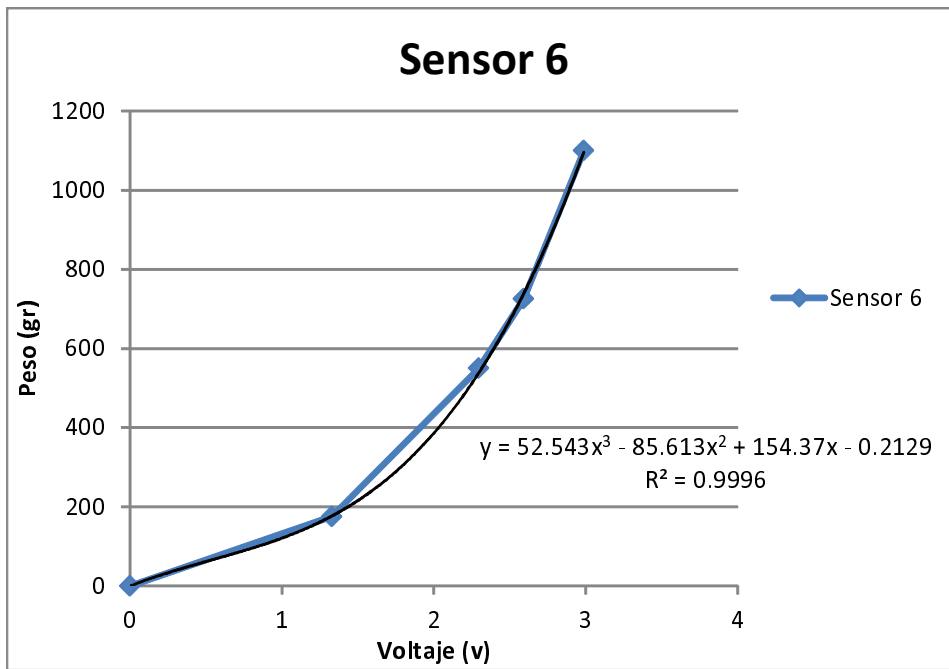
Comportamiento de los potenciómetros y sensores de presión

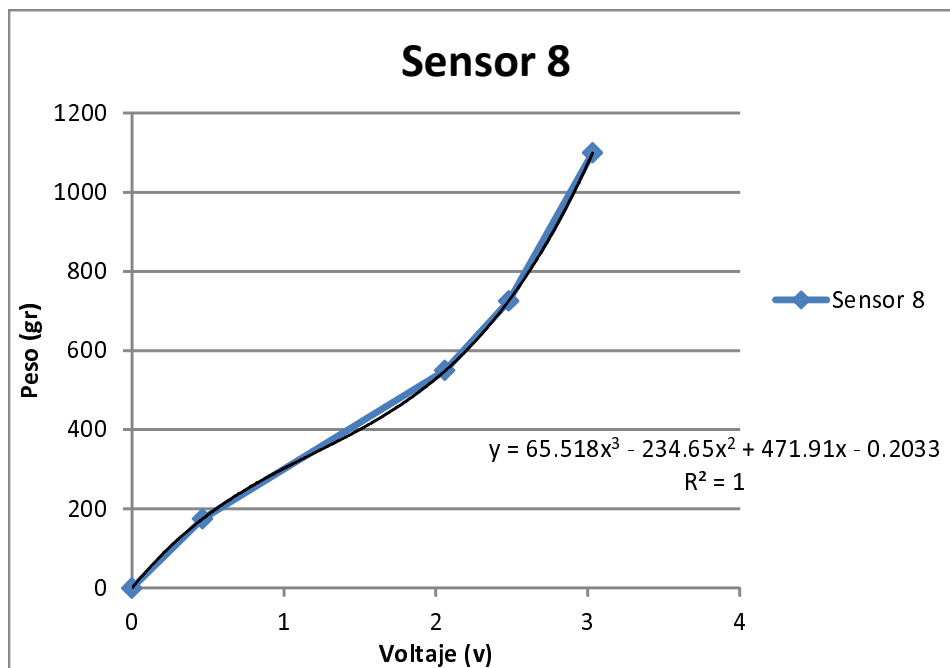
A.1. Funciones que describen el comportamiento de la curvas obtenidas a partir de la caracterización de los sensores de presión



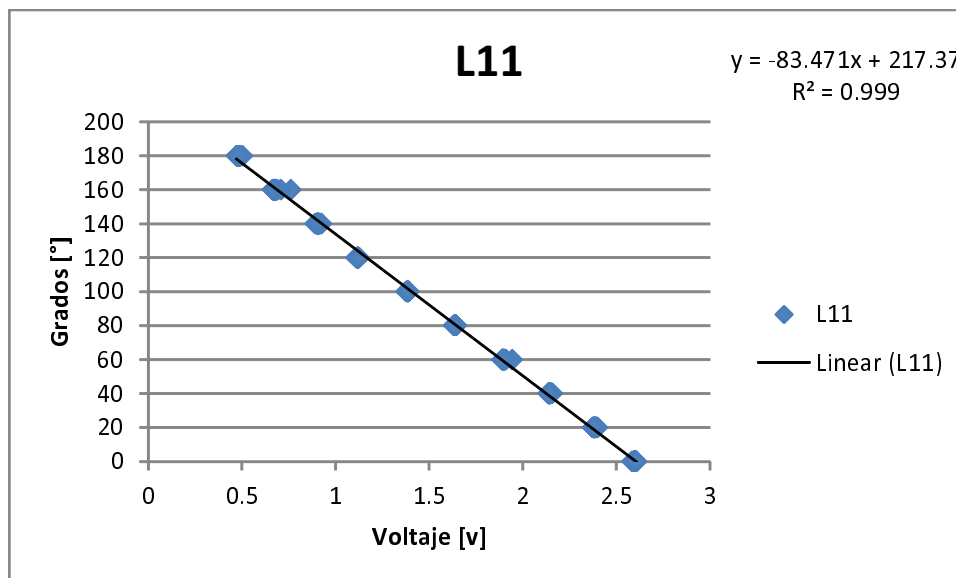


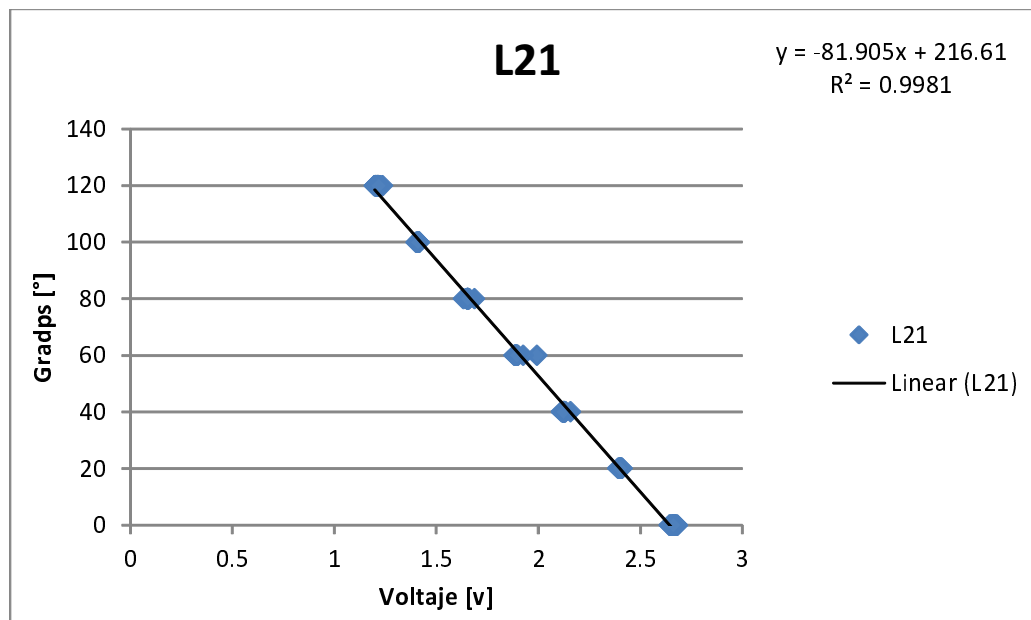
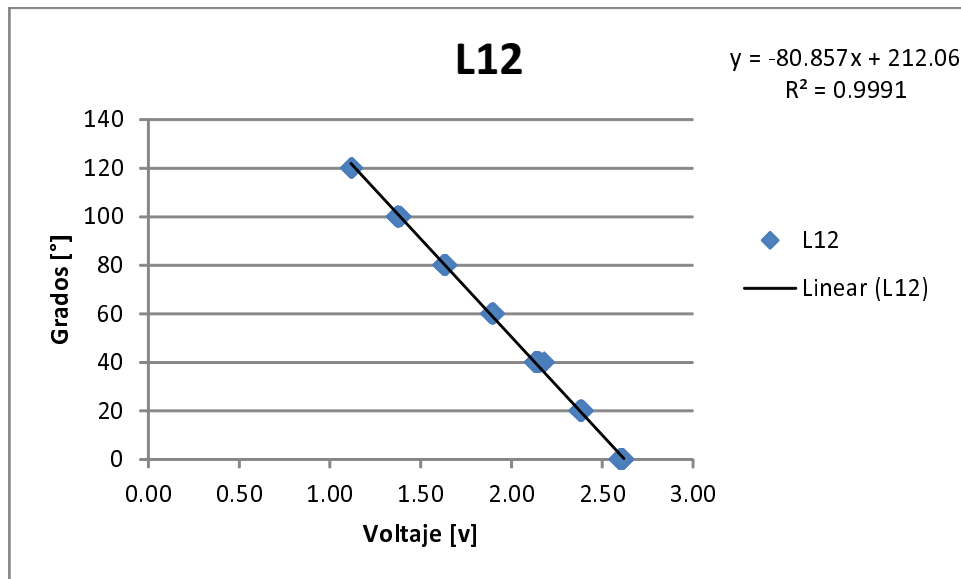


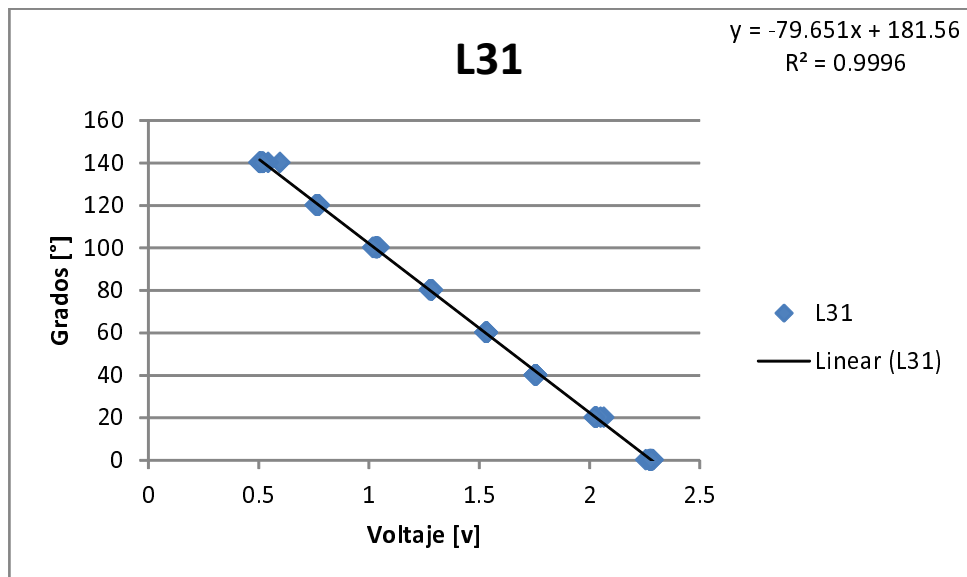
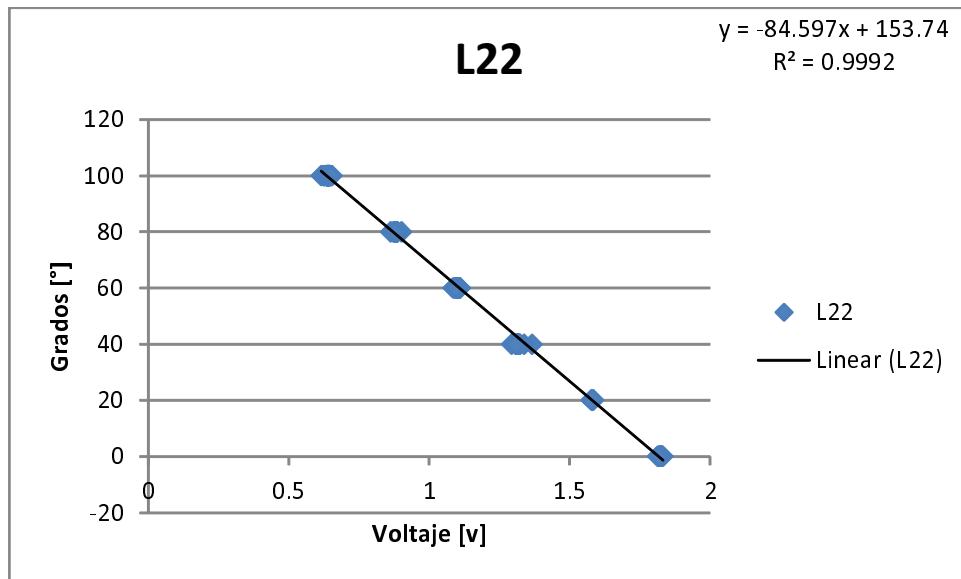


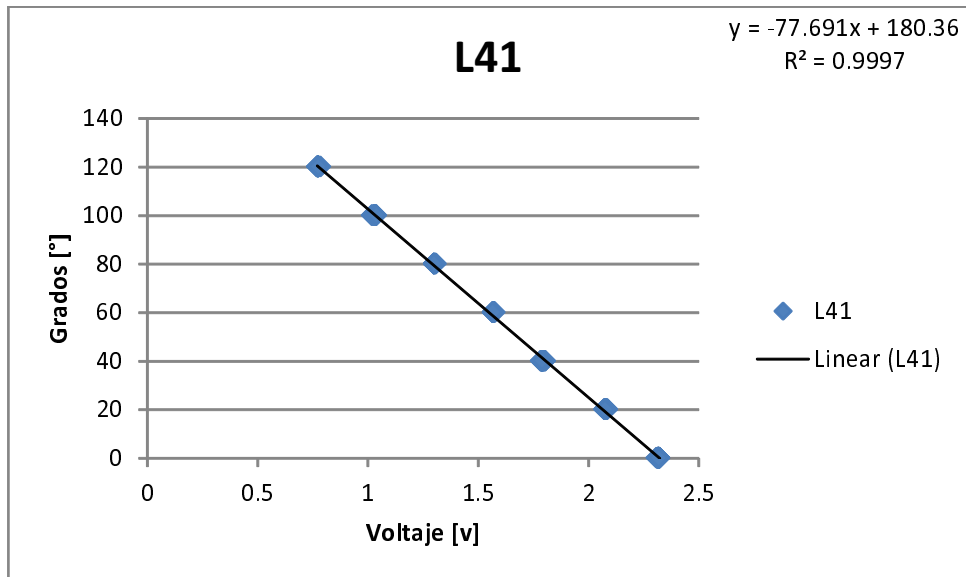
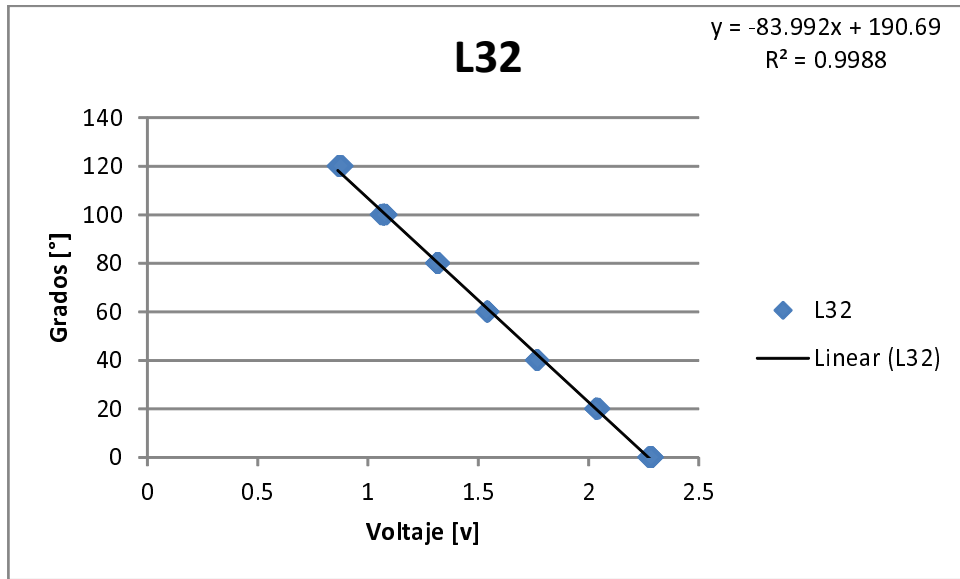


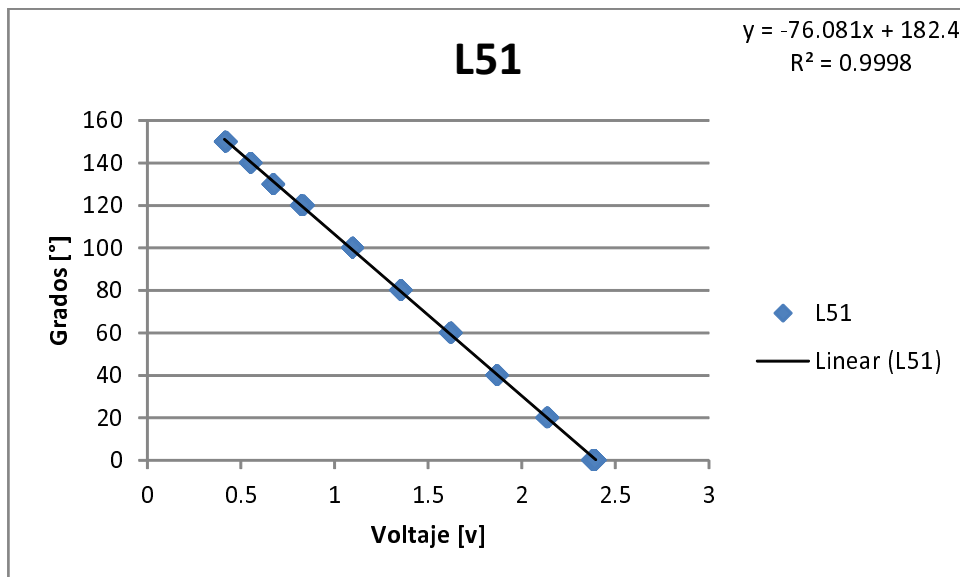
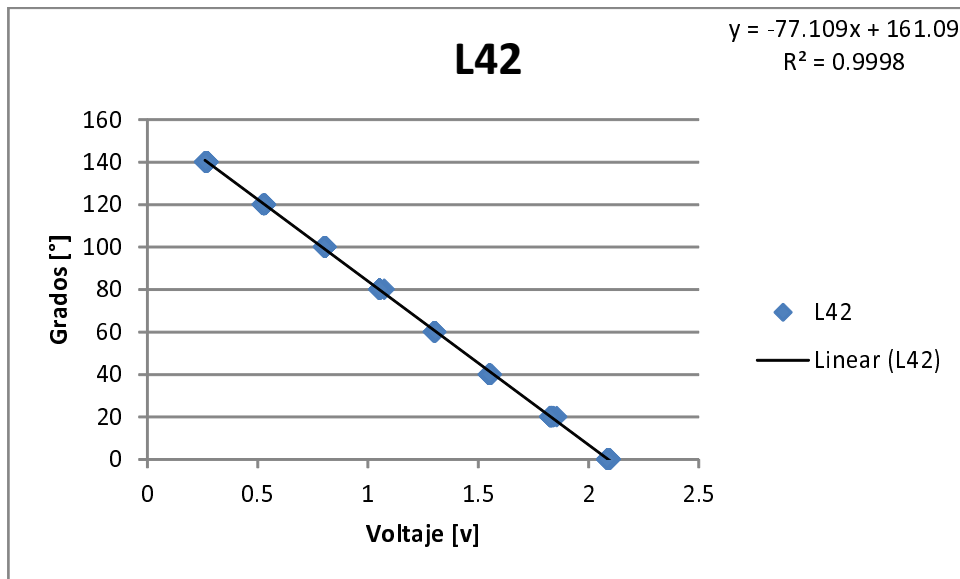
A.2. Funciones que describen el comportamiento de la curvas obtenidas a partir de la caracterización de los potenciómetros de los servomotores

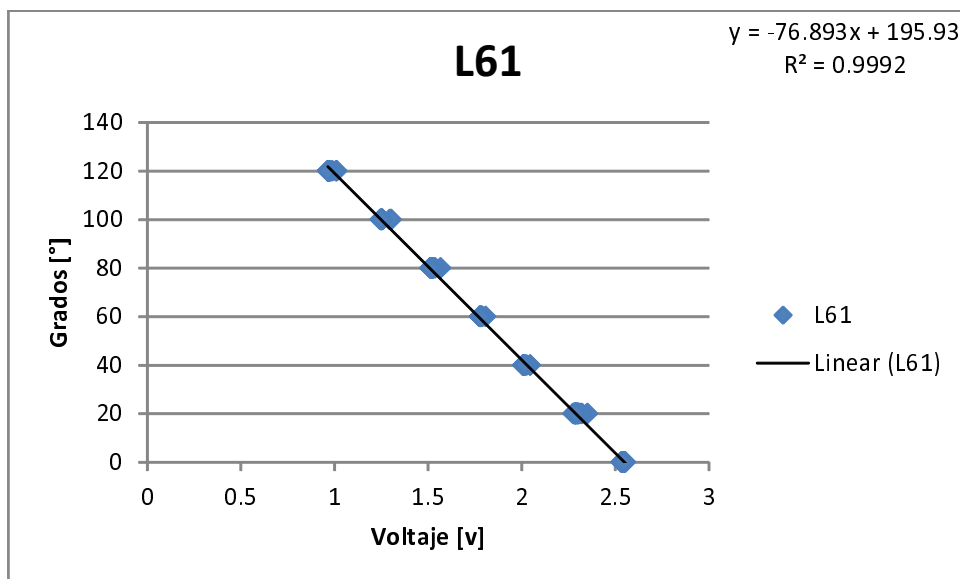
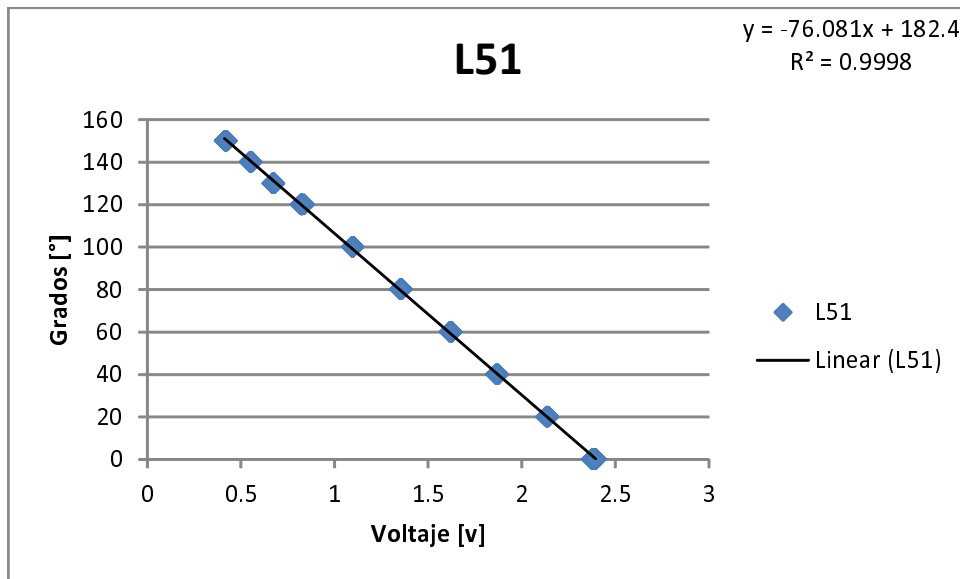


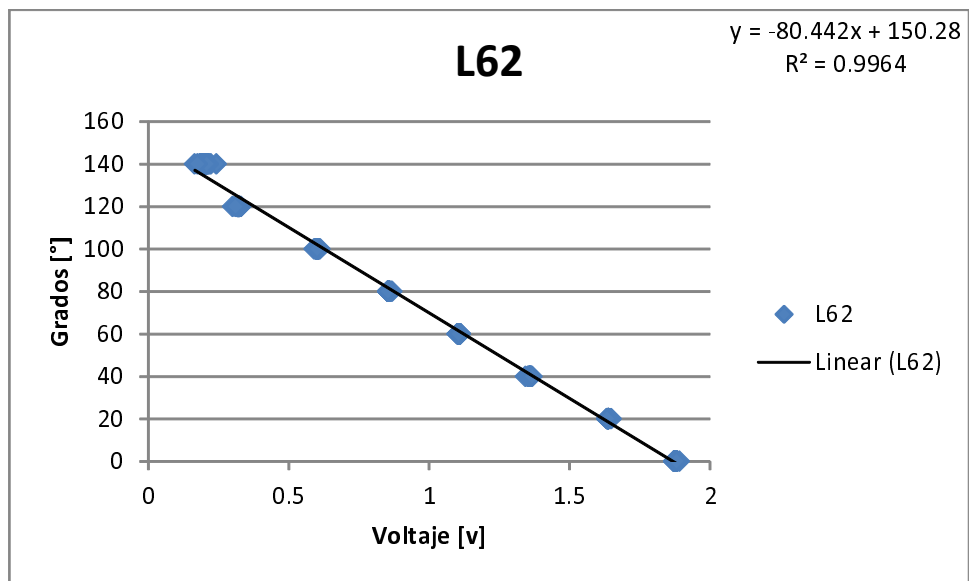












Apéndice B

Programas utilizados para la instrumentación

B.1. Programa para sensor ArduIMU V2. Controlador PID y filtro de Kalman

B.1.1. Programa principal

```
#include <SoftwareServo.h>
SoftwareServo myservo1,myservo2;

int posicion1=70,posicion2=63,grados; //Grados iniciales de los servomotores
float kp=.1,ki=.001,kd=.001,PID_error; //Constantes del PID obtenidas experimentalmente
int heading_previous_error; //Almacena la acumulación de error
float heading_I; //Almacena datos del Integrados

#define GRAVITY 101 //this equivalent to 1G in the raw data coming from the accelerometer
#define Accel_Scale(x) x*(GRAVITY/9.81)//Scaling the raw data of the accel to actual acceleration

#define Gyro_Gain 2.5 //2.5 Gyro gain
#define Gyro_Scaled(x) x*((Gyro_Gain*PI)/360)//Return the scaled ADC raw data of the gyro in radians
#define G_Dt(x) x*.02 //DT .02 = 20 miliseconds, value used in derivations and integrations

#define ToRad(x) (x*PI)/180.0
#define ToDeg(x) (x*180.0)/PI

#define PRINT_ANALOGS 0 // If 1 will print the analog raw data
#define PRINT_KALMAN 1 //Print kalman
#define PRINT_GPS 0

#define PROCESS_GPS 1

//Sensor: GYROX, GYROY, GYROZ, ACCELX, ACCELY, ACCELZ
float SENSOR_SIGN[]={-1,-1,-1,-1,1,-1}; //{1,1,-1,1,-1,1}Used to change the polarity of the sensor

int long timer=0; //general porpuse timer
int long timer24=0; //Second timer used to print values
int AN[8]; //array that store the 6 ADC filtered data
int AN_OFFSET[8]; //Array that stores the Offset of the gyrosa
```

```

int EX[8]; //General porpuse array to send information

unsigned long Now = millis();
unsigned long lastread = Now;

float dt; // = .016384;

//GPS

//GPS stuff
union long_union {
    int32_t dword;
    uint8_t byte[4];
} longUnion;

union int_union {
    int16_t word;
    uint8_t byte[2];
} intUnion;

/*Flight GPS variables*/
int gpsFix=1; //This variable store the status of the GPS
float lat=0; // store the Latitude from the gps
float lon=0; // Store guess what?
float alt_MSL=0; //This is the alt.
float ground_speed=0; // This is the velocity your "plane" is traveling in meters for second,
float ground_course=90; //This is the runaway direction of you "plane" in degrees
float climb_rate=0; //This is the velocity you plane will impact the ground (in case of being
char data_update_event=0;

//uBlox Checksum
byte ck_a=0;
byte ck_b=0;
long iTOW=0; //GPS Millisecond Time of Week
long alt=0; //Height above Ellipsoid
float speed_3d=0; //Speed (3-D) (not used)

volatile uint8_t MuxSel=0;
volatile uint8_t analog_reference = DEFAULT;
volatile int16_t analog_buffer[8];

#include <math.h>
#include "kalman.h"

KALDATA xkaldata;
KALDATA ykaldata;

void test(float value[9], int pos)
{
    Serial.print(convert_to_dec(value[pos]));
}

void setup()
{
    myservo1.attach(9); //L52
    myservo2.attach(10); //L51

```

```

    myservo1.write(70);
    myservo2.write(63);

//L52
myservo1.setMinimumPulse(900);
myservo1.setMaximumPulse(1745);
//L51
myservo2.setMinimumPulse(1500);
myservo2.setMaximumPulse(2347);

Serial.begin(38400);
pinMode(2,OUTPUT); //Serial Mux
digitalWrite(2,HIGH); //Serial Mux
pinMode(5,OUTPUT); //Red LED
pinMode(6,OUTPUT); // BLue LED
pinMode(7,OUTPUT); // Yellow LED
Analog_Reference(EXTERNAL); //Using external analog reference
Analog_Init();

for(int c=0; c<75; c++)
{
    read_adc_raw();

    digitalWrite(7,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(5,LOW);
    delay(50);
    digitalWrite(7,HIGH);
    digitalWrite(6,LOW);
    digitalWrite(5,HIGH);
    delay(50);
}
digitalWrite(5,LOW);
digitalWrite(7,LOW);
for(int y=0; y<=7; y++)
{
    AN_OFFSET[y]=AN[y];
    Serial.println((int)AN_OFFSET[y]);
}
    AN_OFFSET[5]=AN[5]-GRAVITY;

    kalmanInitState(&xkaldata);
    kalmanInitState(&ykaldata);
}
void loop()
{
    Now = millis();
    dt = (Now - lastread) * .001; //compute the time delta since last iteration, in sec.

    read_adc_raw(); //ADC Stuff

    state_update(Gyro_Scaled(read_adc(1)),&xkaldata, dt);
    state_update(Gyro_Scaled(read_adc(0)),&ykaldata, dt);

    kalman_update( (atan2(-read_adc(5),read_adc(3))-(3.14159/2)) ,&xkaldata);
    kalman_update( (atan2(-read_adc(5),read_adc(4))-(3.14159/2)) ,&ykaldata);
}

```

```

    if((millis()-timer24)>=100)
    {
        timer24=millis();
        #if PROCESS_GPS == 1
        decode_gps();
        #endif
        printdata(); //Send info via serial
        servo_control();
        SoftwareServo::refresh();
    }
    lastread = Now;
}

```

B.1.2. Filtro de Kalman

B.1.2.1. Kalman.h

```

#ifndef _KALMAN_H_
#define _KALMAN_H_
struct KALDATA
    float P[2][2];    //covariance matrix.
    float angle;
    float q_bias;    //gyro estimate
    float rate;
    float Pdot[4];
    float err;
};
void kalmanInitState(KALDATA p_kd);
void state_update(
    const float          q_m    /* Pitch gyro measurement */,
    KALDATA p_kd,
    float dt
);
void
kalman_update(
    const float          angle_m, //measured angle
    KALDATA p_kd
);
#endif

```

B.1.2.2. Kalman

```

//Adapted from tilt.c

/* -*- indent-tabs-mode:T; c-basic-offset:8; tab-width:8; -*- vi: set ts=8:
 * $Id: tilt.c,v 1.1 2003/07/09 18:23:29 john Exp $
 *
 * 1 dimensional tilt sensor using a dual axis accelerometer
 * and single axis angular rate gyro. The two sensors are fused
 * via a two state Kalman filter, with one state being the angle
 * and the other state being the gyro bias.
 *
 * Gyro bias is automatically tracked by the filter. This seems
 * like magic.
 *
 * Please note that there are lots of comments in the functions and
 * in blocks before the functions. Kalman filtering is an already complex

```



```

* subject, made even more so by extensive hand optimizations to the C code
* that implements the filter. I've tried to make an effort of explaining
* the optimizations, but feel free to send mail to the mailing list,
* autopilot-devel@lists.sf.net, with questions about this code.
*
*
* (c) 2003 Trammell Hudson <hudson@rotomotion.com>
*
*****
*
* This file is part of the autopilot onboard code package.
*
* Autopilot is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* Autopilot is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with Autopilot; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*
*/
#include "kalman.h"

/*
* Our update rate. This is how often our state is updated with
* gyro rate measurements. For now, we do it every time an
* 8 bit counter running at CLK/1024 expires. You will have to
* change this value if you update at a different rate.
*/

/* This variables now replaced by the KALDATA struct in 'kalman.h' */
#if 0
//static const float dt = ( 1024.0 * 256.0 ) / 8000000.0;

/*
* Our covariance matrix. This is updated at every time step to
* determine how well the sensors are tracking the actual state.
*/

static float P[2][2] = {
    { 1, 0 },
    { 0, 1 },
};

/*
* Our two states, the angle and the gyro bias. As a byproduct of computing

```

```

* the angle, we also have an unbiased angular rate available.  These are
* read-only to the user of the module.
*/
float          angle;
float          q_bias;
float          rate;
#endif

/*
* R represents the measurement covariance noise.  In this case,
* it is a 1x1 matrix that says that we expect 0.3 rad jitter
* from the accelerometer.
*/
static const float    R_angle = 0.4;
//static const float    R_angle = 0.3*57.2958;

/*
* Q is a 2x2 matrix that represents the process covariance noise.
* In this case, it indicates how much we trust the accelerometer
* relative to the gyros.
*/

//static const float    Q_angle = 0.001;
//static const float    Q_gyro  = 0.003;

static const float    Q_angle = 0.01 * 57.2958;
static const float    Q_gyro  = 0.03 * 57.2958;

void kalmanInitState(struct KALDATA *p_kd){
    p_kd->P[0][0] = 1.0;
    p_kd->P[0][1] = 0.0;
    p_kd->P[1][0] = 0.0;
    p_kd->P[1][1] = 1.0;
    p_kd->angle = 0;
    p_kd->q_bias = 0;
    p_kd->rate = 0;
    p_kd->Pdot[0] = 0;
    p_kd->Pdot[1] = 0;
    p_kd->Pdot[2] = 0;
    p_kd->Pdot[3] = 0;
}

/*
* state_update is called every dt with a biased gyro measurement
* by the user of the module.  It updates the current angle and
* rate estimate.
*
* The pitch gyro measurement should be scaled into real units, but
* does not need any bias removal.  The filter will track the bias.
*
* Our state vector is:
*
*     X = [ angle, gyro_bias ]
*
*/

```

```

* It runs the state estimation forward via the state functions:
*
*   Xdot = [ angle_dot, gyro_bias_dot ]
*
*   angle_dot      = gyro - gyro_bias
*   gyro_bias_dot  = 0
*
* And updates the covariance matrix via the function:
*
*   Pdot = A*P + P*A' + Q
*
* A is the Jacobian of Xdot with respect to the states:
*
*   A = [ d(angle_dot)/d(angle)      d(angle_dot)/d(gyro_bias) ]
*        [ d(gyro_bias_dot)/d(angle) d(gyro_bias_dot)/d(gyro_bias) ]
*
*   = [ 0 -1 ]
*      [ 0  0 ]
*
* Due to the small CPU available on the microcontroller, we've
* hand optimized the C code to only compute the terms that are
* explicitly non-zero, as well as expanded out the matrix math
* to be done in as few steps as possible. This does make it harder
* to read, debug and extend, but also allows us to do this with
* very little CPU time.
*/
void
state_update(
    float          q_m      /* Pitch gyro measurement */,
    struct KALDATA *p_kd,
    float dt
)
{
    float          q;//, dt;

    q = q_m - p_kd->q_bias;      /* Unbias our gyro */
    //dt = p_kd->dt;

    /*
     * Compute the derivative of the covariance matrix
     */
    Pdot = A*P + P*A' + Q
    /*
     * We've hand computed the expansion of A = [ 0 -1, 0 0 ] multiplied
     * by P and P multiplied by A' = [ 0 0, -1, 0 ]. This is then added
     * to the diagonal elements of Q, which are Q_angle and Q_gyro.
     */
    p_kd->Pdot[0] = Q_angle - p_kd->P[0][1] - p_kd->P[1][0]; /* 0,0 */
    p_kd->Pdot[1] = 0 - p_kd->P[1][1];                       /* 0,1 */
    p_kd->Pdot[2] = 0 - p_kd->P[1][1];                       /* 1,0 */
    p_kd->Pdot[3] = Q_gyro;                                  /* 1,1 */

    /* Store our unbias gyro estimate */
    p_kd->rate = q;

```

```

/*
 * Update our angle estimate
 * angle += angle_dot * dt
 *       += (gyro - gyro_bias) * dt
 *       += q * dt
 */
p_kd->angle += (q * dt);

/* Update the covariance matrix */
p_kd->P[0][0] += (p_kd->Pdot[0] * dt);
p_kd->P[0][1] += (p_kd->Pdot[1] * dt);
p_kd->P[1][0] += (p_kd->Pdot[2] * dt);
p_kd->P[1][1] += (p_kd->Pdot[3] * dt);
}

/*
 * kalman_update is called by a user of the module when a new
 * accelerometer measurement is available. axy_m and az_m do not
 * need to be scaled into actual units, but must be zeroed and have
 * the same scale.
 *
 * This does not need to be called every time step, but can be if
 * the accelerometer data are available at the same rate as the
 * rate gyro measurement.
 *
 * For a two-axis accelerometer mounted perpendicular to the rotation
 * axis, we can compute the angle for the full 360 degree rotation
 * with no linearization errors by using the arctangent of the two
 * readings.
 *
 * As commented in state_update, the math here is simplified to
 * make it possible to execute on a small microcontroller with no
 * floating point unit. It will be hard to read the actual code and
 * see what is happening, which is why there is this extensive
 * comment block.
 *
 * The C matrix is a 1x2 (measurements x states) matrix that
 * is the Jacobian matrix of the measurement value with respect
 * to the states. In this case, C is:
 *
 *      C = [ d(angle_m)/d(angle)  d(angle_m)/d(gyro_bias) ]
 *           = [ 1 0 ]
 *
 * because the angle measurement directly corresponds to the angle
 * estimate and the angle measurement has no relation to the gyro
 * bias.
 * CHANGED SUCH THAT ANGLE IS CALCULATED EXTERNALLY
 */
void
kalman_update(
    float          angle_m, //measured angle
    struct KALDATA *p_kd
)
{
    float          angle_err;

```

```

float          C_0;
float          Pct_0;
float          Pct_1;
float          E;
float          K_0, K_1;
float          t_0;
float          t_1;

/* Compute our measured angle and the error in our estimate */
angle_err = angle_m - p_kd->angle;

p_kd->err = angle_err;
/*
 * C_0 shows how the state measurement directly relates to
 * the state estimate.
 *
 * The C_1 shows that the state measurement does not relate
 * to the gyro bias estimate. We don't actually use this, so
 * we comment it out.
 */
C_0 = 1;
/* const float          C_1 = 0; */

/*
 * Pct<2,1> = P<2,2> * C'<2,1>, which we use twice. This makes
 * it worthwhile to precompute and store the two values.
 * Note that C[0,1] = C_1 is zero, so we do not compute that
 * term.
 */
Pct_0 = C_0 * p_kd->P[0][0]; /* + C_1 * P[0][1] = 0 */
Pct_1 = C_0 * p_kd->P[1][0]; /* + C_1 * P[1][1] = 0 */

/*
 * Compute the error estimate. From the Kalman filter paper:
 *
 *      E = C P C' + R
 *
 * Dimensionally,
 *
 *      E<1,1> = C<1,2> P<2,2> C'<2,1> + R<1,1>
 *
 * Again, note that C_1 is zero, so we do not compute the term.
 */
E =      R_angle + C_0 * Pct_0; /*      + C_1 * Pct_1 = 0 */

/*
 * Compute the Kalman filter gains. From the Kalman paper:
 *
 *      K = P C' inv(E)
 *
 * Dimensionally:
 *
 *      K<2,1> = P<2,2> C'<2,1> inv(E)<1,1>
 *
 * Luckily, E is <1,1>, so the inverse of E is just 1/E.

```

```

    */
    K_0 = PCt_0 / E;
    K_1 = PCt_1 / E;

    /*
    * Update covariance matrix. Again, from the Kalman filter paper:
    *
    *      P = P - K C P
    *
    * Dimensionally:
    *
    *      P<2,2> -= K<2,1> C<1,2> P<2,2>
    *
    * We first compute t<1,2> = C P. Note that:
    *
    *      t[0,0] = C[0,0] * P[0,0] + C[0,1] * P[1,0]
    *
    * But, since C_1 is zero, we have:
    *
    *      t[0,0] = C[0,0] * P[0,0] = PCt[0,0]
    *
    * This saves us a floating point multiply.
    */
    t_0 = PCt_0; /* C_0 * P[0][0] + C_1 * P[1][0] */
    t_1 = C_0 * p_kd->P[0][1]; /* + C_1 * P[1][1] = 0 */

    p_kd->P[0][0] -= (K_0 * t_0);
    p_kd->P[0][1] -= (K_0 * t_1);
    p_kd->P[1][0] -= (K_1 * t_0);
    p_kd->P[1][1] -= (K_1 * t_1);

    /*
    * Update our state estimate. Again, from the Kalman paper:
    *
    *      X += K * err
    *
    * And, dimensionally,
    *
    *      X<2> = X<2> + K<2,1> * err<1,1>
    *
    * err is a measurement of the difference in the measured state
    * and the estimate state. In our case, it is just the difference
    * between the two accelerometer measured angle and our estimated
    * angle.
    */
    p_kd->angle += (K_0 * angle_err);
    p_kd->q_bias += (K_1 * angle_err);

```

B.1.3. Control PID

```

//Se presenta un controlador PID para los servomotores L51 y L52 del robot
bipedo Scout
void servo_control(void)
{
    static unsigned int heading_PID_timer; //Timer para calcular el dt del PID
    static float heading_D; // Almacena los resultados del derivador
    static int heading_output; //Almacena los resultados del controlador PID

```

```

float dt=(float)(millis()-heading_PID_timer)/1000;// Calcula el dt
posicion1=constrain(posicion1,0,120); //Restringe los ángulos de los
servomotores
posicion2=constrain(posicion2,0,140);
PID_error=ToDeg(xkaldata.angle); //El error es igual al ángulo de inclinación
if(abs(PID_error)>4) // Se configura una tolerancia de movimiento
{
    posicion1=posicion1-heading_output;
    posicion2=posicion2+heading_output;
}
Serial.println("POSICION");
Serial.print(posicion1);
if( abs(PID_error<3)) //Se configura una tolerancia de error
{
    PID_error=PID_error*0;
    heading_output=0;
}
if(posicion1>140 || posicion1<0) //Se limitan los errores y se restauran
valores
{
    grados=0;
    heading_I=0;
    posicion1=70;
    posicion2=63;
}
myservo1.write(posicion1); //Se envían los grados a los servos
myservo2.write(posicion2);
//Parte Integral
heading_I+= (float)PID_error*(float)dt; //1000 microsegundos / 1000 = 1
millisegundo
//Parte Derivativa
heading_D=((float)PID_error-(float)heading_previous_error)/(float)dt;
heading_output=0;//Limpia la variable
//Parte proporcional
heading_output=(kp*PID_error);
//Suma de las partes integral y derivativa
heading_output+= (ki*heading_I);
heading_output+= (kd*heading_D);
heading_previous_error=PID_error;//Se guarda el error actual para usarlo en la
parte derivativa
heading_PID_timer=millis();// Se guarda el tiempo de ejecución para calcular
el dt
}

```

B.2. Programa en Matlab para cálculo de ZMP

```

% Programa que calcula y grafica los puntos generados por el criterio de
% ZMP, medidos a través de sensores de presión y procesados con LabVIEW y
% Arduino
datos=load('C:/Users/VICKY/Dropbox/Tesis_2.0/Resultados/ZMP/paso4.txt','r');
lectura=datos;
volt1=lectura(:,1); %Se separa las lecturas de cada sensor
volt2=lectura(:,2);
volt3=lectura(:,3);
volt4=lectura(:,4);

```

```

volt5=lectura (:,5);
volt6=lectura (:,6);
volt7=lectura (:,7);
volt8=lectura (:,8);

[V11,V12]=size(volt1);
[V21,V22]=size(volt2);
[V31,V32]=size(volt3);
[V41,V42]=size(volt4);

[V51,V52]=size(volt5);
[V61,V62]=size(volt6);
[V71,V72]=size(volt7);
[V81,V82]=size(volt8);

sen7=zeros(V11,V12);
sen4=zeros(V21,V22);
sen3=zeros(V31,V32);
sen8=zeros(V41,V42);

sen5=zeros(V51,V52);
sen1=zeros(V61,V62);
sen6=zeros(V71,V72);
sen2=zeros(V81,V82);

INDICATOR=0;
BOUCLE=0;

for(k=1:1:V11)
    for(l=1:1:V12)

        %se convierte de un voltaje a newtons para sensores de pierna derecha
        if(volt1(k,l)>0)
            sen7(k,l)=((88.01*volt1(k,l)^3)-(185.01*volt1(k,l)^2)+(381.01*volt1(k,l))
                +4.3455)*(9.81/1000);
        end
        if(volt2(k,l)>0)
            sen4(k,l)=((324.89*volt2(k,l)^3)-(822.17*volt2(k,l)^2)+1134.8*volt2(k,l)
                -6.5322)*(9.81/1000);
        end
        if(volt3(k,l)>0)
            sen3(k,l)=((131.54*volt3(k,l)^3)-(389.02*volt3(k,l)^2)+655.89*
                volt3(k,l)+0.8805)*(9.81/1000);
        end
        if(volt4(k,l)>0)
            sen8(k,l)=((65.518*volt4(k,l)^3)-(234.65*volt4(k,l)^2)+471.91*volt4(k,l)
                -0.2033)*(9.81/1000);
        end

        %se convierte de un voltaje a newtons para sensores de pierna izquierda
        if(volt5(k,l)>0)
            sen5(k,l)=((44.435*volt5(k,l)^3)-(110.79*volt5(k,l)^2)+227.86*
                volt5(k,l)-0.8806)*(9.81/1000);
        end
        if(volt6(k,l)>0)
            sen1(k,l)=((101.83*volt6(k,l)^3)-(240.86*volt6(k,l)^2)+(453.22*volt6

```



```

        (k,l)) - 0.1917)*(9.81/1000);
    end
    if (volt7(k,l)>0)
        sen6(k,l)=((52.543*volt7(k,l)^3) - (85.613*volt7(k,l)^2) + 154.37*
            volt7(k,l) - 0.2129)*(9.81/1000);
    end
    if (volt8(k,l)>0)
        sen2(k,l)=((58.448*volt8(k,l)^3) - (149.59*volt8(k,l)^2)+ 384.52*volt8
            (k,l) - 0.1215)*(9.81/1000);
    end
    if (sen1(k,l)==0 && sen2(k,l)==0 && sen3(k,l)==0 && sen4(k,l)==0 && sen5(k,
        l)==0 && sen6(k,l)==0 && sen7(k,l)==0 && sen8(k,l)==0)
        INDICATOR=INDICATOR+1;
        while (BOUCLE==0)
            kk=k;
            ll=l;
            BOUCLE=1;
        end
    end
end
end
end
yzmp_der=((5.4).*(sen4+sen8)-(sen3+sen7).*(1.3))./(sen7+sen4+sen3+sen8); % Calculo
del ZMP pierna derecha
xzmp_der=((1.35).*(sen4+sen3)-(sen8+sen7).*(2.85))./(sen7+sen4+sen3+sen8);

yzmp_iz=((5.4).*(sen2+sen5)-(sen1+sen6).*(1.3))./(sen2+sen5+sen1+sen6); % Calculo
del ZMP pierna izquierda
xzmp_iz=((1.35).*(sen2+sen1)-(sen6+sen5).*(2.85))./(sen2+sen1+sen5+sen6);

esc=.1;

pd1x = esc *44.33; % Dimensiones de la planta del pie
pd1y = esc *83.33;
pd2x = esc *15.7;
pd2y = esc *83.33;
pd3x = esc *15.7;
pd3y = esc *64.34;
pd4x = esc *3.03;
pd4y = esc *64.34;
pd5x = esc *3.03;
pd5y = esc *83.33;
pd6x = esc * -26.18;
pd6y = esc *83.33;
pd7x = esc * -26.18;
pd7y = esc *64.34;
pd8x = esc * -28.18;
pd8y = esc *64.81;
pd9x = esc * -28.18;
pd9y = esc * -24.20;
pd10x = esc * -26.18;
pd10y = esc * -24.70;
pd11x = esc * -26.18;
pd11y = esc * -43.70;
pd12x = esc *3.03;
pd12y = esc * -43.70;
pd13x = esc *3.03;

```

```

pd13y = esc * -24.7;
pd14x = esc * 15.7;
pd14y = esc * -24.7;
pd15x = esc * 15.7;
pd15y = esc * -43.7;
pd16x = esc * 44.33;
pd16y = esc * -43.7;
pd17x = esc * 44.33;
pd17y = esc * -24.7;
pd18x = esc * 46.33;
pd18y = esc * -24.2;
pd19x = esc * 46.33;
pd19y = esc * 64.81;
pd20x = esc * 44.33;
pd20y = esc * 64.34;

Tpx=[pd1x , pd2x , pd3x , pd4x , pd5x , pd6x , pd7x , pd8x , pd9x , pd10x , pd11x , pd12x , pd13x , pd14x ,
      pd15x , pd16x , pd17x , pd18x , pd19x , pd20x , pd1x ];
Tpy=[pd1y , pd2y , pd3y , pd4y , pd5y , pd6y , pd7y , pd8y , pd9y , pd10y , pd11y , pd12y , pd13y , pd14y ,
      pd15y , pd16y , pd17y , pd18y , pd19y , pd20y , pd1y ];

ver_x1 = 3.133;
ver_x2 = -1.318;
ver_y1 = -1.87;
ver_y2 = 6;

Traslacion1=esc * 9.3650;
Traslacion2=esc * 7.26;
Tpxd=Tpx-Traslacion1-Traslacion2;
Tpxi=Tpx-Traslacion1-Traslacion2;
Tpxd=-Tpxd;

ver_x1d = 3.133-Traslacion1-Traslacion2;
ver_x2d = -1.318-Traslacion1-Traslacion2;
ver_x1d = -ver_x1d;
ver_x2d = -ver_x2d;
ver_x1i = 3.133-Traslacion1-Traslacion2;
ver_x2i = -1.318-Traslacion1-Traslacion2;
ver_y1 = -1.87;
ver_y2 = 6;

    % Trazo del área de seguridad:
    trayxd = [ver_x1d , ver_x2d , ver_x2d , ver_x1d , ver_x1d];
    trayxi = [ver_x1i , ver_x2i , ver_x2i , ver_x1i , ver_x1i];
    trayy = [ver_y1 , ver_y1 , ver_y2 , ver_y2 , ver_y1];

% Gráfica de los pies con ZMP
figure(2); % derecha
plot(Tpxd,Tpy, trayxd , trayy);
hold on;
plot(-xmp_der , ymp_der , 'ro')
plot(0.7547,2.04 , 'bd' , 'Linewidth' , 1.5) % punto ideal
hold on
plot(1.6357,2.6284 , 'mp' , 'Linewidth' , 1.5) % media
axis equal;

```

```

figure(1); %izquierda
plot(Tpxi,Tpy, trayxi , trayy);
hold on;
plot(xzmp_iz , yzmp_iz , 'ro')
plot(-0.7553,2.04, 'bd', 'Linewidth',1.5) % punto ideal
hold on
plot(-0.5164,4.2788, 'mp', 'Linewidth',1.5) % media
axis equal;

```

B.3. Programa para obtener los ángulos rotacionales de los potenciómetros

```

% Programa que convierte las señales analógicas adquiridas
% de los potenciómetros en ángulos
datos_pot=load('C:/Users/VICKY/Dropbox/Tesis_2.0/Resutaldos/pot/2_marcha.txt','r')
;
[m n]=size(datos_pot);
lectura=datos_pot(:,1:12).*(4.96/1023);
pot1=lectura(:,1); %Se separa las lecturas de cada sensor
pot2=lectura(:,2);
pot3=lectura(:,3);
pot4=lectura(:,4);
pot5=lectura(:,5);
pot6=lectura(:,6);
pot7=lectura(:,7);
pot8=lectura(:,8);
pot9=lectura(:,9);
pot10=lectura(:,10);
pot11=lectura(:,11);
pot12=lectura(:,12);

L12=-80.857.*pot1+212.06;
L11=-83.47.*pot2+217.37;
L22=-84.597.*pot3+153.74;
L21=-81.905.*pot4+216.61;
L32=-83.992.*pot5+190.69;
L31=-79.651.*pot6+181.56;
L42=-77.109.*pot7+161.09;
L41=-77.691.*pot8+180.36;
L52=-76.938.*pot9+166.38;
L51=-76.08.*pot10+182.4;
L62=-80.441.*pot11+150.28;
L61=-76.893.*pot12+195.93;
mapeo=[L12, L11, L22, L21, L32, L31, L42, L41, L52, L51, L62, L61];
archivo=xlswrite('C:/Users/VICKY/Dropbox/Tesis_2.0/Resutaldos/pot/Angulos',mapeo,'
grados','A1')

```