



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA

FACULTAD DE INGENIERÍA

Análisis Cinemático y Dinámico de un Robot
Bípedo de 12 GDL Internos Utilizando la
Formulación Newton-Euler

TESIS

QUE PARA OPTAR POR EL GRADO DE

**MAESTRO EN INGENIERÍA
MECÁNICA-MECATRÓNICA**

PRESENTA:

ING. OCTAVIO RUIZ HERNÁNDEZ

TUTOR:

Dr. EDMUNDO GABRIEL ROCHA CÓZATL

MÉXICO, D.F. ENERO DEL 2014



Jurado Asignado

Presidente: Dr. González Villela Víctor Javier
Secretario: Dr. Cuenca Jiménez Francisco
Vocal: Dr. Rocha Cózatl Edmundo Gabriel
1er. Suplente: M. I. Díaz Hernández Octavio
2do. Suplente: M. I. Castañeda Cedeño Serafín

Ciudad Universitaria, México D.F.

Tutor de Tesis

Dr. Rocha Cózatl Edmundo Gabriel

FIRMA



Análisis Cinemático y Dinámico de un Robot Bípedo de Doce Grados de Libertad Internos Utilizando la Formulación Newton-Euler

Octavio Ruiz Hernández

Enero del 2014



¡Un extraordinario laberinto!, pero no sin solución.

Alexander Pope



Agradecimientos

A mi familia por su apoyo y su manera de siempre estar conmigo.

A mis amigos por brindarme ánimo en mis momentos de desesperanza.
En especial a Sergio y Alberto con quien viví estos dos increíbles años.

Al Dr. Edmundo cuya orientación fue indispensable en el presente trabajo.

A Alessia por la compañía y por ayudarme a desenredar mi mente.

Al Dr. Emmanuel Merchán por inspirarme a estudiar una maestría.

Al Ing. Alejandro Rodríguez por los consejos de vida.

Al CONACyT por el patrocinio de mis estudios.

Al IPN y a la UNAM que son pilares en mi desarrollo humano.



INDICE GENERAL

Resumen.....	8
Abstract	9
1. Estado del Arte.....	10
1.1 Introducción.....	11
1.2 Clasificación de Robots Bípedos.....	12
1.3 Exoesqueletos	14
1.4 Robots Bípedos Comerciales.....	15
1.5 Formulaciones Matemáticas de la Dinámica de los Robots	16
1.6 Futuro y Tendencias de la Robótica Bípeda	17
1.7 Objetivos de la Tesis.....	19
2. Marco Teórico.....	20
2.1 Locomoción	21
2.2 Planos de Referencia y Ejes de Movimiento.....	23
2.3 Cinemática Directa.....	23
2.4 Cinemática Inversa	25
2.5 Modelo Dinámico	28
2.6 Controladores en Robots Bípedos	30
2.7 Arquitectura del Bípedo Scout.....	32
3. Análisis Cinemático	35
3.1 Cinemática del Robot Bípedo Scout	36
3.1.1 Cinemática Directa del Robot Bípedo Scout	36
3.1.2 Cinemática Inversa del Bípedo Scout	46
3.2 Simulación del Robot Bípedo Scout	50
3.2.1 Trayectorias Propuestas	51
4. Análisis Dinámico.....	58
4.1 Introducción	59
4.2 Dinámica inversa (Formulación de Newton-Euler).....	59
4.2.1. Sistema de coordenadas en movimiento	59
4.2.2. Cinemática de los elementos.....	60
4.2.3. Ecuaciones de movimiento recursivas.....	63
4.2.4. Algoritmo Computacional.....	66
4.2 Análisis de la Problemática de Equilibrio	71
4.3 Cálculo del ZMP	74
4.3.1 Análisis del Pie de Apoyo	77
4.3.2 Cálculo de las Velocidades y Aceleraciones del COG.....	78



5. Conclusiones y Trabajo a Futuro	87
5.1 Conclusiones	88
5.2 Trabajo a Futuro.....	89
A. Anexo A.....	91
A.1 Especificaciones del Bípido Scout	92
A.2 Parámetros Geométricos	96
A.3 Masa de los Eslabones.....	96
B. Anexo B	97
B.1 Programa “BPDPIA2”.-.....	98
B.2 Programa “BPDPIF”.-.....	104
B.3 Programa “EcuacionesBipeDF”.-.....	109
B.5 Programa “VELACELBPD”.-.....	111
B.6 Programa “plotbipedo”.-.....	113
B.7 Programa “elipse”.-.....	114
B.8 Programa “rotx”.-.....	115
B.9 Programa “roty”.-.....	115
B.10 Programa “rotz”.-.....	115
B.11 Programa “Torques”.-.....	116
C. Anexo C	119
C.1 Algebra de matrices.....	120
C.2 Suma de matrices	120
C.3 Multiplicación de matrices.....	120
C.4 Matrices adjuntas e inversas.....	121
C.5 Traza de una matriz	122
Bibliografía.....	123



INDICE DE FIGURAS

<i>Figura 1.1. Robots bípedos: Izquierda ASIMO, Derecha QRIO.....</i>	<i>11</i>
<i>Figura 1.2. Robot bípedo HRP-3 en el laboratorio de Mecatrónica de Kawada.</i>	<i>13</i>
<i>Figura 1.3. Honda ASIMO dirigiendo una orquesta sinfónica, cargando una charola.</i>	<i>13</i>
<i>Figura 1.4. Pruebas preliminares del exoesqueleto XOS (Dollar & Herr, 2008)</i>	<i>14</i>
<i>Figura 1.5. Computadora y sistemas de control del Sarcos XOS (Dollar & Herr, 2008)</i>	<i>15</i>
<i>Figura 1.6. Robots bípedos comerciales, BRAT (6 GDL), Scout (12 GDL) y Robonova.....</i>	<i>15</i>
<i>Figura 1.8. ASIMO jugando con niños humanos.</i>	<i>18</i>
<i>Figura 1.9. FIRSTSTEP, construido por la Escuela de Ingeniería Electricidad y Electrónica, Politécnica de Singapur. Participante en Robocup 2003.</i>	<i>19</i>
<i>Figura 2.1. Varias fases de la marcha bípeda.....</i>	<i>22</i>
<i>Figura 2.2 a) Plano sagital, b) Plano coronal, c) Plano transversal (Iguar, et al., 1996).</i>	<i>23</i>
<i>Figura 2.3. Cinemática Directa (Fu, et al, 1988).....</i>	<i>24</i>
<i>Figura 2.4. Cinemática Inversa (Fu, et al, 1988)</i>	<i>26</i>
<i>Figura 2.5. Modelos de péndulo invertido 3D y modelo car-table.....</i>	<i>29</i>
<i>Figura 2.6. El punto P es el ZMP.....</i>	<i>30</i>
<i>Figura 2.7. Caminata deseada (Jonghoon Park et al (2007)).....</i>	<i>31</i>
<i>Figura 2.8. Diagrama del controlador ZMP-FLC (Narong and Manukid et al (2009))</i>	<i>31</i>
<i>Figura 2.9. Diseño del controlador de los robots HONDA (Chevallereau et al (2007))</i>	<i>31</i>
<i>Figura 2.10. Modelo CAD 3D del bípedo Scout (Narvárez, 2010)</i>	<i>32</i>
<i>Figura 2.11. Modelo CAD 3D simplificado del bípedo Scout (Narvárez, 2010).....</i>	<i>33</i>
<i>Figura 2.12. Vista sagital derecha del modelo simplificado del bípedo. Las cotas indican las distancias con relevancia cinemática (Narvárez, 2010).</i>	<i>34</i>
<i>Figura 3.1 Diagrama unifilar del Bípedo Scout.....</i>	<i>36</i>
<i>Figura 3.2. Trayectoria elíptica para la cadera y el pie flotante.</i>	<i>51</i>
<i>Figura 3.3. El robot bípedo siguiendo una trayectoria elíptica para la cadera y el pie flotante....</i>	<i>52</i>
<i>Figura 3.4. El robot bípedo siguiendo una trayectoria elíptica para la cadera y el pie derecho flotante.</i>	<i>52</i>
<i>Figura 3.5. El robot bípedo siguiendo una trayectoria elíptica para la cadera y el pie izquierdo flotante.</i>	<i>53</i>
<i>Figura 3.6. Ángulos de las articulaciones del pie izquierdo durante la trayectoria</i>	<i>55</i>
<i>Figura 3.7. Ángulos de las articulaciones del pie derecho durante la trayectoria</i>	<i>56</i>



Figura 4.1. Sistema de coordenadas en movimiento	59
Figura 4.2. Relación entre los sistemas O, O^* y O'	61
Figura 4.3. Fuerzas y momentos sobre el elemento i.....	63
Figura 4.4. Torques calculados para los 6 GDL del pie Izquierdo	69
Figura 4.5. Torques calculados para los 6 GDL del pie Derecho.....	70
Figura 4.6. Fuerzas y momentos aplicados sobre el pie de apoyo	71
Figura 4.7. Compensación de los momentos en el pie de apoyo	72
Figura 4.8. Rotación del pie de apoyo a partir de uno de sus bordes. Robot inestable.....	74
Figura 4.9. Modelo simplificado del robot bípedo (péndulo invertido 3D).....	75
Figura 4.10. Análisis del pie de apoyo.....	77
Figura 4.11. Un punto $i r^i$ que se refiere a la cadera del robot bípedo Scout	79
Figura 4.12. Velocidad Lineal del COG en sus componentes “X, Y, Z”	82
Figura 4.13. Aceleración Lineal del COG en sus componentes “X, Y, Z”	83
Figura 4.14. Calculo del ZMP en el pie Izquierdo	84
Figura 4.15. Calculo del ZMP en el pie Derecho.....	85
Figura 5.1. Esquema de control del robot bípedo Scout.....	90

INDICE DE TABLAS

Tabla 2.1.- Comparación del número de operaciones realizadas por los métodos de Lagrange-Euler y Newton-Euler.	28
Tabla 3.1 Parámetros D-H del Bípedo (pie izquierdo).	37
Tabla 3.2 Parámetros D-H del Bípedo (Pie derecho).....	42
Tabla 4.1. Ecuaciones de movimiento de Newton-Euler recursivas eficientes	67
Tabla A.1. Lista de elementos.....	92



Resumen

Este trabajo de investigación presenta el desarrollo de los modelos cinemático y dinámico para el robot bípedo Scout que cuenta con 12 GDL internos. Se encuentra la solución de la cinemática directa e inversa basada en el método de Denavit-Hartenberg de matrices de transformación homogénea con el fin de obtener la posición, velocidad y aceleración de cada uno de los GDL internos.

El modelo dinámico de este proyecto se basa en la formulación de Newton-Euler despreciando las fuerzas disipativas en las juntas rotacionales del bípedo y los términos de Coriolis y centrífugo, además asumiendo que no existe deslizamiento entre los pies y el piso. También se utilizó el modelo del Péndulo invertido 3D para calcular el ZMP basado en la trayectoria que sigue el COG y a si tener un criterio de estabilidad en la marcha para futuros sistemas de control aplicados al robot bípedo Scout.

Se generó una simulación del modelo cinemático del bípedo scout por computadora en la que el robot sigue trayectorias elípticas para el pie flotante y la cadera y que aunado al cálculo del ZMP sirve para elegir trayectorias estables en el proceso de caminata.

En tanto que la generalidad de la formulación de ambos modelos permite describir a robots bípedos de " n " GDL's constituidos por juntas rotacionales actuadas este trabajo también podrá ser utilizado para otro ramo como el diseño de prótesis, órtesis robóticas o exoesqueletos que le permitan al humano realizar tareas imposibles o incluso volver a caminar.



Abstract

This investigation presents a development of dynamic and cinematic methods for a biped robot Scout which one counts with 12 internal GDL. The direct and inverse kinematic solution was founded by using the Denavit-Hartenberg method of homogeneity transformation matrices to obtain the position, velocity and acceleration of each one of the internal GDL.

The dynamic model in this project is based on the Newton-Euler formulation ignoring the wasted forces in the rotational joints of the biped and the Coriolis and centrifuge terms. The inverted 3D pendulum was also used to calculate the ZMP based on the trajectory followed by the COG so this way, we could have a stability standard for future control systems applied to the biped robot Scout.

A cinematic model simulation of the biped was generated by computer which one on where the robot follows elliptic trajectories for the floating feet and the hip, this and the ZMP are used for choose stables trajectories at the walking process.

Talking about the generality of both model's formulations, it allows us to discover an "n" GDLs biped robot made by rotational actuated joints. This work can also be used for another branch of science like the robotic prosthesis, orthosis or exoskeletons that could allow humans to realize impossible works and even to walk again.



1

Estado del Arte

En este capítulo se muestran los avances tecnológicos más importantes que se han logrado en los últimos años dentro del área de robots bípedos.



1.1 Introducción

Transportarse en la actualidad es algo que hacemos todos los días y se puede hacer de diferentes formas; ya sea en un automóvil, avión, barco, bicicleta y desde luego, podemos transportarnos con nuestras propias piernas. Los seres humanos decidimos en que trasportarnos por el mundo dependiendo de las condiciones del camino a seguir, un robot móvil tiene que sortear las mismas condiciones de camino y es por ello que existen varios tipos de robots orientados a diferentes tipos de medio ambiente. La robótica bípeda se enfoca en robots capaces de moverse por el mundo usando dos extremidades inferiores emulando la caminata del humano con el fin de poder interactuar con él en diferentes tareas en las cuales otros tipos de robots serian inútiles.

En el campo de la robótica existen grandes avances en el desarrollo de robots bípedos en todo el mundo, tal es el caso de Honda con su robot ASIMO o Sony con su QRIO, etc. (Figura 1.1). Pretender rivalizar con estas grandes compañías sería iluso de mi parte, debido a desventajas como infraestructura requerida, presupuesto y tiempo que estas empresas han podido dedicar a la investigación de los robots bípedos, es por esto que mucha de la bibliografía consultada con respecto a los modelos matemáticos de la cinemática y de la dinámica de los robots bípedos es alrededor de los años 1970-1990 ya que estas grandes compañías no sacan a la luz sus avances en esta área.

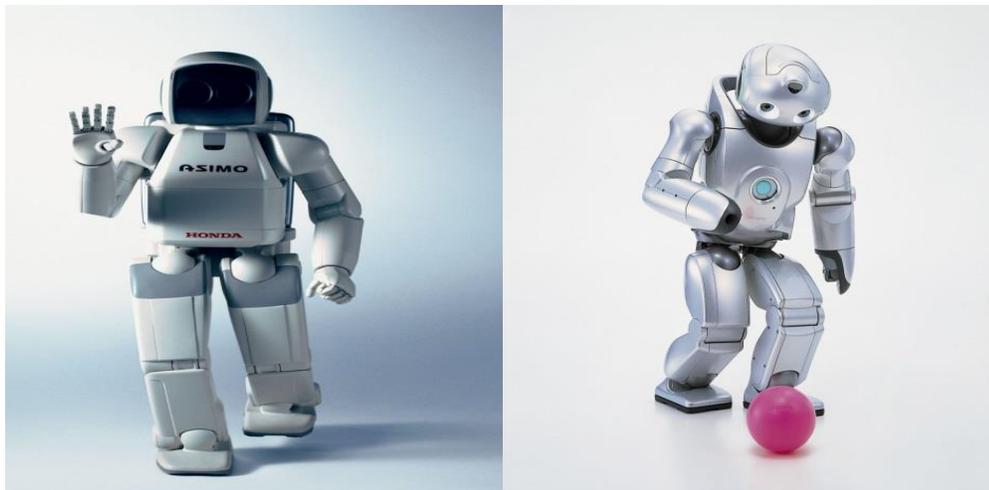


Figura 1.1. Robots bípedos: Izquierda ASIMO, Derecha QRIO

La principal ventaja de los robots bípedos sobre otro tipo de robot móvil es que pueden desenvolverse en superficies irregulares, escaleras, etc., sin embargo, la complejidad en el diseño y control aumenta considerablemente por lo que en la actualidad son un desafío de ingeniería.



1.2 Clasificación de Robots Bípedos

De acuerdo a la complejidad de sus movimientos, los robots bípedos pueden describirse realizando combinaciones de los rasgos definidos en cada una de las siguientes categorías (Westervelt and de Wit, 2007):

Bidimensionales: El movimiento de sus eslabones solo tiene lugar en el plano sagital.

Tridimensionales: El movimiento de sus eslabones tiene lugar dentro y fuera del plano sagital.

Minimalistas: Poseen menos de seis grados de libertad (GDL) en cada una de sus piernas.

Antropomórficos: Tienen seis o más GDL en cada una de sus piernas.

Pasivos: Desarrollan su ciclo de marcha sin un control directo de la cinemática de sus eslabones.

Activos: Caminan controlando la posición de los actuadores en cada uno de sus eslabones.

Adicionalmente, de acuerdo al problema de diseño mecánico que motiva su construcción, los bípedos se pueden clasificar como:

Eficientes: Buscan que la intervención humana se reduzca al mínimo para asegurar la recarga de su fuente de energía.

Versátiles: Persiguen ser suficientemente robustos para llevar acabo la caminata de forma estable a pesar de su exposición a perturbaciones externas e incertidumbres, tales como la presencia de obstáculos en la superficie de desplazamiento.

Existen dos que son plataformas avanzadas para el desarrollo de conocimiento en el tema de la caminata, los cuales son el HRP-3 y el famoso ASIMO.

En 2007 Kawada hizo público en HRP-3 Promet MK-II (figura 1.2), este tiene una altura de 1.6 [m], un peso de 68[kg] y 42 GDL internos. Se le ha presentado caminando establemente sobre superficies con agua y arena, reconociendo objetos, bailando y usando taladros para ejecución de tareas sencillas. Sus sistemas de enfriamiento y filtrado de polvo y humedad le permiten funcionar durante dos horas en condiciones de lluvia (Kawada-Industries-Inc., 2009).

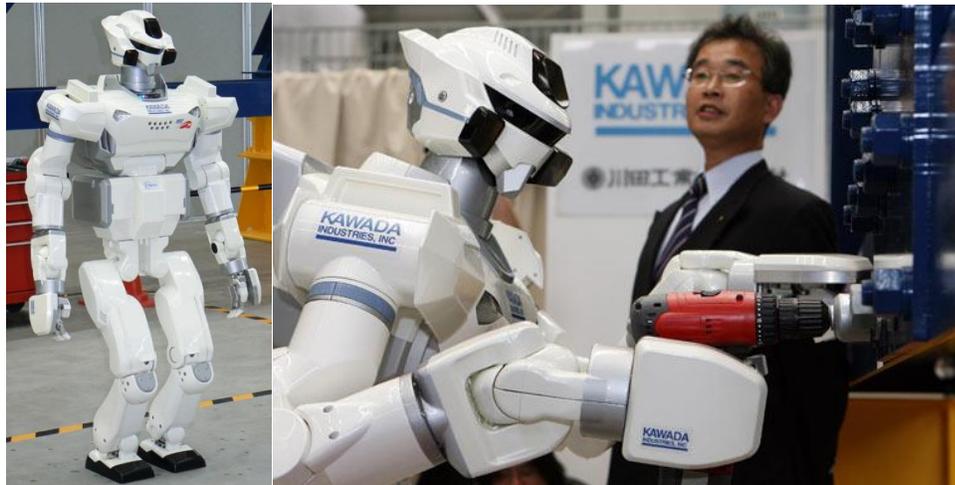


Figura 1.2. Robot bípido HRP-3 en el laboratorio de Mecatrónica de Kawada.

De entre todos los robots bípedos de la actualidad ASIMO (Advanced Step in Innovative Mobility) es el más avanzado de todos y es capaz de realizar muchas funciones (Honda-Worldwide, 2010). Con una altura de 1.3 [m], un peso total de 54[kg] y 34 GDL; puede caminar a 0.75 [m/s], correr a una velocidad de 1.67 [m/s], dar vueltas, subir y descender escaleras, manipular interruptores, abrir puertas, regular la fuerza de presión de sus manos, esquivar obstáculos en su trayectoria de movimiento, dirigir una orquesta sinfónica, cargar charolas de comida, empujar un carro de compras etc. (Figura 1.3). Con las cámaras en su cabeza y su tecnología de inteligencia artificial puede reconocer su ambiente, objetos en movimiento, la postura y gestos de personas a su alrededor, identificar voces y los focos de producción de sonidos para ubicar a su interlocutor y seguir sus órdenes. Uno de los nuevos avances de ASIMO es que puede comunicarse con las demás personas utilizando comandos de voz que emanan de él.



Figura 1.3. Honda ASIMO dirigiendo una orquesta sinfónica, cargando una charola.



1.3 Exoesqueletos

Los beneficios de los sistemas robóticos no se limitan solamente a la imitación del hombre en sus tareas diarias. La aplicación de estas tecnologías para usos militares permite que los soldados puedan cargar más peso y caminar grandes distancias. Sin embargo, el estilo de las últimas guerras crea necesidades adicionales de asistencia robótica: mientras que la cifra de muertos ha sido dramáticamente reducida (10% de los heridos murió en Irak frente a un 30% en la Segunda Guerra Mundial), un 6% de los sobrevivientes de lesiones requiere amputación (frente al 3% en guerras anteriores) y 20% de los sobrevivientes de lesiones necesitará asistencia permanente para el resto de sus vidas (Dellon & Matsuoka, 2007).

El exoesqueleto Sarcos XOS es el traje robótico más avanzado en la actualidad, desarrollado por Steve Jacobsen y los ingenieros de la compañía SARCOS (Figura 1.4).

De tecnología estadounidense, este increíble sistema le permite al usuario incrementar sus capacidades físicas para realizar tareas que son muy difíciles para un ser humano normal.



Figura 1.4. Pruebas preliminares del exoesqueleto XOS (Dollar & Herr, 2008)

Este tipo de sistemas están muy cerca de proveer un aumento real de poder para soldados, bomberos, personal de emergencias y trabajadores de la industria para poder transportar cargas pesadas y extender sus habilidades físicas (Kaserooni, 2005).

Actualmente, este exoesqueleto se encuentra en la etapa de desarrollo, pues el plan es utilizarse como un auxiliar para los soldados en combate. Existen algunos inconvenientes en el diseño, por ejemplo, el tamaño de la computadora y los sistemas de control y quizás la más significativa, que el sistema debe estar conectado a una fuente de poder por medio de cables (figura 1.5). Pero los creadores están seguros que, conforme avance la ciencia y la tecnología, pronto estará funcionando al 100% para la milicia estadounidense (Dollar & Herr, 2008).



Figura 1.5. Computadora y sistemas de control del Sarcos XOS (Dollar & Herr, 2008)

1.4 Robots Bípedos Comerciales

Algunos de los bípedos que pueden encontrarse en el mercado son BRAT y Scout de la compañía estadounidense Lynxmotion y el humanoide Robonova de Hitec Robotics con 16 GDL, 0.4 [m] de altura y un peso de 1.6 [kg] (Figura 1.6). La estructura de estos robots se encuentra restringida por la geometría de los servomotores comercializados por Hitec, empresa coreana de buena reputación en aplicaciones de radiocontrol. El Scout, seleccionado como bípedo de estudio en este trabajo, será descrito de forma detallada en el capítulo 3.



Figura 1.6. Robots bípedos comerciales, BRAT (6 GDL), Scout (12 GDL) y Robonova.



1.5 Formulaciones Matemáticas de la Dinámica de los Robots

La dinámica del robot trata con las formulaciones matemáticas de las ecuaciones de movimiento del robot. Las ecuaciones de movimiento de un manipulador son un conjunto de ecuaciones matemáticas que describen su conducta dinámica. Tales ecuaciones son útiles para la simulación en computadora del movimiento del robot, en diseño de ecuaciones de control apropiadas para el robot y la evaluación del diseño y estructura del robot. En el capítulo 4 nos concentraremos en la formulación, características y propiedades de las ecuaciones dinámicas de movimiento.

El modelo dinámico de un robot se puede obtener a partir de leyes físicas conocidas tales como las leyes de la mecánica Newtoniana y la Lagrangiana. Esto conduce al desarrollo de las ecuaciones de movimiento dinámico para las diversas articulaciones del robot bípedo en término de los parámetros geométricos e iniciales de los elementos. Métodos convencionales como las formulaciones de Lagrange-Euler (L-E) y Newton-Euler (N-E) se pueden aplicar entonces sistemáticamente para desarrollar las ecuaciones de movimiento del robot bípedo, de estas 2 formulaciones se obtienen diferentes formas de describir la dinámica del robot bípedo tales como las ecuaciones de Lagrange-Euler de Uicker (Uicker 1965, Bejczy 1974), las ecuaciones recursivas de Lagrange de Hollerbach (Hollerbach 1980), las ecuaciones de Newton-Euler de Luh (Luh 1980). Estas ecuaciones de movimiento son equivalentes unas a otras en el sentido de que describen la conducta dinámica del mismo robot físico. Sin embargo sus estructuras pueden diferir por que se obtienen por diversas razones y objetivos. Algunas se obtienen para lograr tiempos de cálculo rápido en la evaluación de los torques de las articulaciones nominales para controlar el robot bípedo, otras se obtienen para facilitar el análisis y la síntesis de control, y todavía otras se obtienen para mejorar la simulación en computadora del movimiento del robot.

La obtención del modelo dinámico de un robot bípedo basado en la formulación de L-E es simple y sistemática. Suponiendo el movimiento del cuerpo rígido, las ecuaciones de movimiento resultante, excluyendo la dinámica de los dispositivos de control electrónico y el rozamiento de los engranes, son un conjunto de ecuaciones diferenciales no lineales acopladas de segundo orden (Bejczy 1974), utilizando la representación de la matriz de transformación homogénea 4x4 de la cadena cinemática y la formulación lagrangiana, ha demostrado que las ecuaciones de movimiento dinámico para un robot bípedo de diez articulaciones son fuertemente no lineales y constan de carga inercial, fuerzas de reacción de acoplo entre las articulaciones (Coriolis y centrífuga) y efectos de carga de la gravedad. Más aún, estos torques/fuerza, dependen de los parámetros físicos del robot bípedo, de la configuración instantánea de las configuraciones, de la velocidad y aceleración de las articulaciones y de la carga que está soportando el robot. *Las ecuaciones de movimiento L-E proporcionan ecuaciones de estado implícitas para la dinámica del robot. En una menor medida se están utilizando para resolver el problema dinámico directo, esto es, dadas las fuerzas/torque deseadas, se utilizan las ecuaciones dinámicas para resolver las aceleraciones de las articulaciones que se integran a continuación para obtener las coordenadas y velocidades generalizadas; o para el problema dinámico inverso, esto es dadas las coordenadas generalizadas deseadas y sus primeras dos derivadas con respecto al tiempo se calculan las fuerzas/torques generalizados.* Desgraciadamente, el cálculo de estos



coeficientes requiere una relativa cantidad de operaciones aritméticas. Así las ecuaciones de L-E son muy difíciles de utilizar con fines de control en tiempo real a menos que se simplifiquen.

Como alternativa para derivar ecuaciones de movimiento más eficientes se dirigió la atención a desarrollar algoritmos para calcular las fuerzas/torque generalizados basados en las ecuaciones de movimiento de N-E (Armstrong 1979, Orin 1979, Luh 1980). La obtención es simple, pero engorrosa e implica términos de producto vectorial. Las ecuaciones dinámicas resultantes, excluyendo la dinámica de dispositivo de control y rozamiento de los engranajes, son un conjunto de ecuaciones recursivas hacia delante y hacia atrás. Este conjunto de ecuaciones se puede aplicar secuencialmente a los elementos del robot. La recursión hacia adelante propaga la información cinemática tal como velocidades lineales, velocidades angulares, aceleraciones angulares y aceleraciones lineales del centro de masa de cada elemento, desde el sistema de coordenada inercial hasta el sistema de coordenadas del pie de robot bípedo. La recursión hacia atrás propaga las fuerzas y momentos ejercidos sobre cada elemento del pie del robot bípedo hasta el sistema de referencia de la base. El resultado más significativo de esta formulación es que el tiempo de cálculo de las fuerzas/torque generalizados se encuentra que es linealmente proporcional al número de articulaciones del robot bípedo e independiente de la configuración del mismo. Con este algoritmo se pueden realizar el control del tiempo real del robot bípedo.

La ineficacia de las ecuaciones de movimiento de L-E surgen parcialmente de las matrices homogéneas 4x4 que describen la cadena cinemática, mientras que la eficacia de la formulación de N-E se basa en la formulación vectorial y en su naturaleza recursiva. En el capítulo 4 se abordará y explicará a detalle la formulación N-E con el fin de obtener una formulación matemática de la dinámica del robot bípedo y así mismo, una simulación computacional.

1.6 Futuro y Tendencias de la Robótica Bípeda

La intención de las compañías que financian investigaciones en robots humanoides es hacer que estos mismos puedan interactuar con humanos, pensando a la larga para ayudar a discapacitados y gente mayor. Se trata de continuar investigando en el desarrollo de un robot inteligente, que pueda crear estrategias de resolución de problemas mediante el reconocimiento y el análisis, la asociación y combinación de datos, la planificación y la toma de decisiones. Aunque aún están lejos de este objetivo, se están realizando pasos en esta dirección.

De momento, ASIMO es solo un proyecto de los robots que probablemente nos esperan en un futuro no muy lejano, hay demasiadas personas investigado acerca de cada una de las habilidades que la robótica puede conferir a este tipo de dispositivos, muy pronto veremos en las calles a robots caminantes capaces de realizar tareas en ocasiones, imposibles para los humanos.



Figura 1.8. ASIMO jugando con niños humanos.

Los avances de la inteligencia artificial, pueden llegar al extremo de hacer posible la existencia de un equipo de robots capaz de ganar una copa mundial de futbol contra un equipo humano jugando de acuerdo con las normas de la FIFA. Existe un proyecto marcando el año 2050 como límite de tiempo para que se pudiera alcanzar este objetivo. El proyecto se ha iniciado mediante la realización periódica de la copa mundial de futbol jugada entre sí por equipos de robots, la ROBOCUP cuya primera cede fue en Nagoya Japón en 1997 y contó con 35 equipos de 12 países. En consecuencia los equipos que participan en este evento tienen que generar tecnología en el ámbito de Ciencia de los Materiales, Electrónica, Robótica y Mecatrónica.

Para responder a los rápidos movimientos de los adversarios y del balón se habrá de diseñar complejos sistemas de visión interactivos con el entorno. Para resolver problemas de choques y tropezones habrá que incorporar sensores de diferentes clases que faciliten el procesamiento de la integración multisensorial. Los robots han de ser capaces de saltar, correr, pegarle al balón, para ello se requiere del desarrollo de refinados pero prácticos mecanismos omnidireccionales de movimiento que superen a los sistemas actuales de motores y engranajes que son muy frágiles y limitados. Superando los anteriores desafíos, el problema principal se fundamentará en el aprendizaje, la colaboración y la estrategia de equipo.

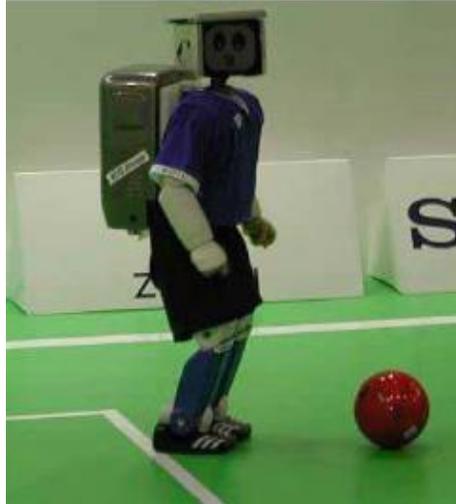


Figura 1.9. FIRSTSTEP, construido por la Escuela de Ingeniería Electricidad y Electrónica, Politécnica de Singapur. Participante en Robocup 2003.

La cantidad de desarrollo que se requiere es inmensa, pero logrando hacer que un robot pueda jugar fútbol igual o mejor que un ser humano podríamos decir que se ha cumplido con la parte de crear algo mecánico bastante parecido a un ser humano en apariencia y en ámbitos fisiológicos. En ámbitos psicológicos aún no se puede decir que un robot pueda igualar a un ser humano pero ese es otro tema.

1.7 Objetivos de la Tesis

Tomando en cuenta los avances alcanzados en el área de robótica bípeda, el objetivo principal consiste en obtener la formulación de la dinámica del robot bípedo basada en la formulación de Newton-Euler a través de la solución de la cinemática inversa y las ecuaciones de la dinámica directa para ofrecer mayores recursos que permitan el entendimiento de esta área, además de contemplar al cálculo del ZMP como criterio de estabilidad.

Esta investigación se realiza con el fin de comparar los resultados obtenidos con la formulación de Newton-Euler contra los obtenidos con la formulación de Lagrange-Euler, los resultados obtenidos de esta última formulación se encuentran expuestos en la tesis de Narváez 2010, y que fueron piedra angular del presente trabajo.

Para esto, se diseñará una simulación en el software MatLab® de la caminata a seguir por el robot bípedo Scout, para poder probar diferentes trayectorias y de ellas elegir la que más se adecue a las condiciones reales de la marcha humana. En el capítulo 2, Marco Teórico, se ofrece una introducción formal a los conocimientos necesarios para poder obtener las ecuaciones dinámicas y cinemáticas del bípedo en cuestión.



2

Marco Teórico

En este capítulo se muestran los antecedentes históricos de la locomoción bípeda, así como los conceptos necesarios para tener un panorama más amplio sobre esta disciplina.



2.1 Locomoción

Locomoción, la capacidad de un cuerpo para moverse de un lugar a otro, es una característica definitoria de la vida animal. Esto se logra mediante la manipulación del cuerpo con respecto al medio ambiente. En condiciones normales, la locomoción tiene muchas formas, ya sea el nadar de los peces, el volar de los pájaros o el caminar de los humanos. La diversidad de la locomoción animal es realmente sorprendente y extremadamente compleja. Lo mismo es cierto en los objetos elaborados por el hombre: los aviones tienen alas que los hacen ascender para el vuelo, los tanques tienen pistas para recorrer terreno irregular, los coches tienen ruedas para rodar eficientemente y los robots suelen ahora caminar sobre sus propias piernas en entornos con tierra discontinua, tales como un pendiente rocosa o los peldaños de una escalera, se puede argumentar que el medio más adecuado y versátil para la locomoción son las piernas.

Las piernas permiten la facilidad de apoyo en el medio ambiente al pasar por encima de las discontinuidades. Además, las piernas son una opción obvia para la locomoción en entornos creados para la marcha humana: correr y trepar. “En la medida en que una máquina equipada con dos piernas puede imitar la marcha de un ser humano, mejor puede interactuar con él en su medio ambiente” Westervelt, Grizzle, Chevallereau & Ho Choi, 2007.

Los robots bípedos forman una subclase de robots con patas. En el aspecto práctico, el estudio de la locomoción con patas mecánicas ha sido motivado por su uso potencial como medio de locomoción en terreno accidentado, o ambientes con discontinuidades. También hay que reconocer que gran parte del interés actual en robots con patas se deriva de la existencia de máquinas que operan en formas antropomórficas o de forma animal (tenemos en cuenta varios bípedos conocidos y juguetes cuadrúpedos). La motivación para el estudio de robots bípedos, en particular, surge de diversos intereses sociológicos y comerciales, que van desde el deseo de reemplazar a los humanos en ocupaciones peligrosas (remoción de minas, la inspección de las centrales nucleares, las intervenciones militares, etc.), a la restauración del movimiento en las personas con discapacidad (controlado dinámicamente los miembros inferiores, prótesis robótica de rehabilitación y estimulación neural funcional).

El proceso de caminar es mucho más complejo de lo que parece ya que para dar pasos están involucrados pies, cadera, torso, brazos, hombros, cabeza, etc. Tomando en cuenta el objetivo principal de este proyecto, conocer los principios básicos que forman parte de la caminata humana es fundamental además de constituir una base para dar pie al modelo dinámico del robot bípedo.

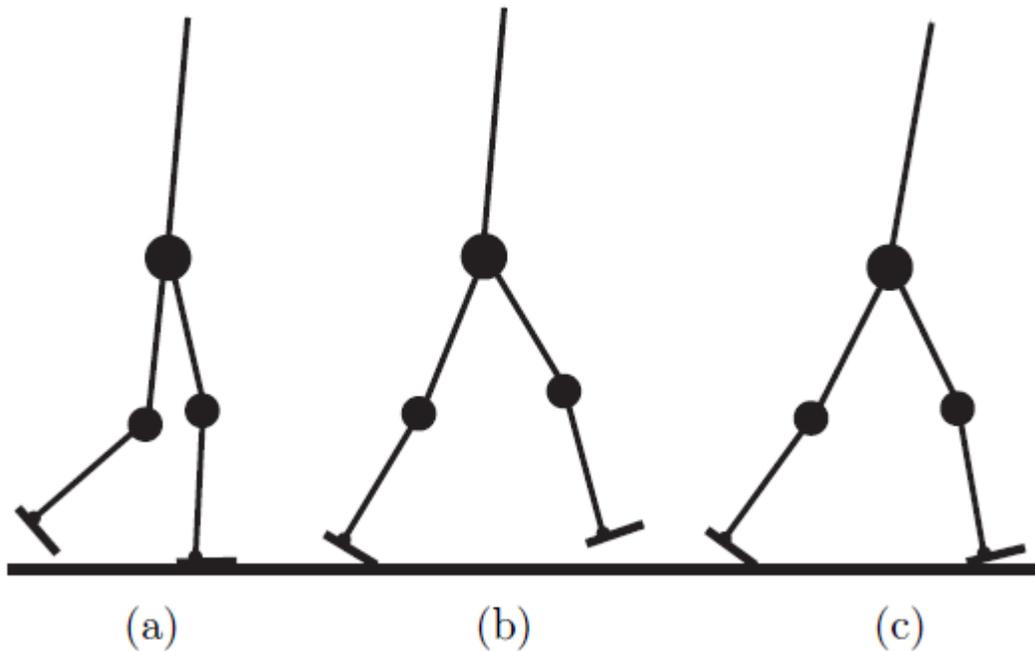


Figura 2.1. Varias fases de la marcha bípeda.

Como se muestra en la figura 2.1 la fase de simple apoyo (también llamada la fase de oscilación) se muestra en (a) y (b), mientras que una fase de doble apoyo se representa en (c). La caminata comienza con los dos pies extendidos y sobre el suelo, en donde el equilibrio no es muy significativo, el gran problema comienza al levantar uno de los dos pies para realizar el movimiento de simple apoyo ya que la tendencia del bípedo a caer es alta, para evitar que el robot bípedo caiga se deben realizar correctivos a los movimientos del robot permitiendo así la estabilidad dinámica de la caminata.

El estudio de la caminata bípeda es un tema muy amplio; una primera subdivisión es realizada en función de la energía que requiere el sistema (robot bípedo, ser humano, etc.) para caminar:

Caminata Pasiva: Refiriéndonos a robots bípedos, la caminata pasiva hace referencia a sistemas mecánicos que pueden caminar sin necesidad de control ni de ningún tipo de actuadores por tanto no requiere de energía suministrada al sistema más que la energía potencial (McGeer, 1989).

Caminata Activa: En oposición a la caminata pasiva, La caminata activa está directamente relacionada con sistemas complejos de control, así como la utilización de actuadores (motores, pistones, etc.) requiriendo de energía extra para activar estos actuadores.



2.2 Planos de Referencia y Ejes de Movimiento.

Un eje es la línea alrededor de la cual se realiza el movimiento y plano es la superficie que se haya en ángulo recto con aquel y en la que se produce el movimiento. Estos términos se usan para facilitar la descripción del movimiento o dirección y por lo que se refiere a ejes y planos del movimiento articular, se describen con relación al cuerpo en posición anatómica (Gardiner, 1964)

La flexión y extensión, movimientos que se estudian en este proyecto, utilizan como plano de deslizamiento el sagital y el eje coronal (Figura 2.2 a y b). El plano sagital está trazado de forma vertical de atrás hacia delante de manera que divide al cuerpo en lado derecho y en lado izquierdo (Figura 2.2 a). El eje coronal va perpendicular respecto al plano de movimiento sagital, es de forma horizontal y está situado en el plano coronal (Figura 2.2 a y b) (Iguar, 1996).

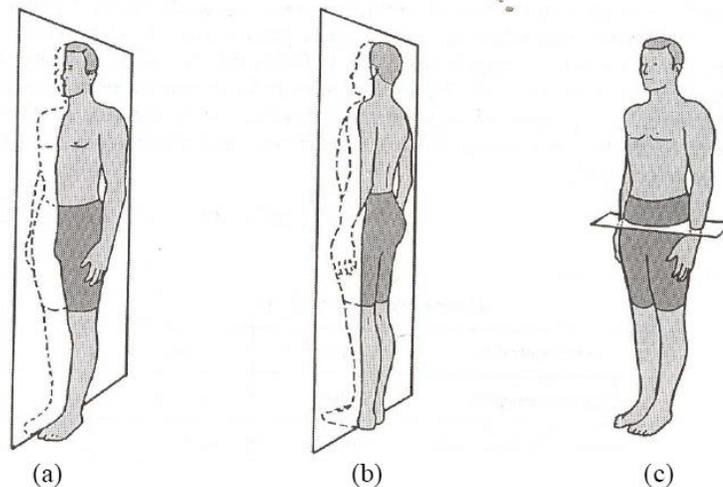


Figura 2.2 a) Plano sagital, b) Plano coronal, c) Plano transversal (Iguar, et al., 1996).

2.3 Cinemática Directa

Primero se desarrollará un análisis cinemático del bípedo, el cual no toma en cuenta las fuerzas que producen el movimiento, esto a través de la cinemática directa. Cuando se habla de cinemática directa, nos referimos a ciertos ángulos dados y los parámetros de los elementos del sistema para encontrar las posiciones y orientaciones deseadas de los elementos (Figura 2.3).

Para resolver el problema cinemático inverso se tiene que encontrar la matriz de transformación que relacione al sistema ligado al cuerpo con respecto al sistema de referencia. Se utiliza una matriz de rotación 3×3 para describir las operaciones rotacionales del sistema ligado al cuerpo con respecto al sistema de referencia. Se utilizarán entonces las coordenadas homogéneas para representar vectores de posición en un espacio tridimensional y las matrices de rotación se ampliarán a



matrices de transformación homogénea 4x4 para incluir las operaciones traslacionales del sistema ligado al cuerpo (Fu, et al, 1988).

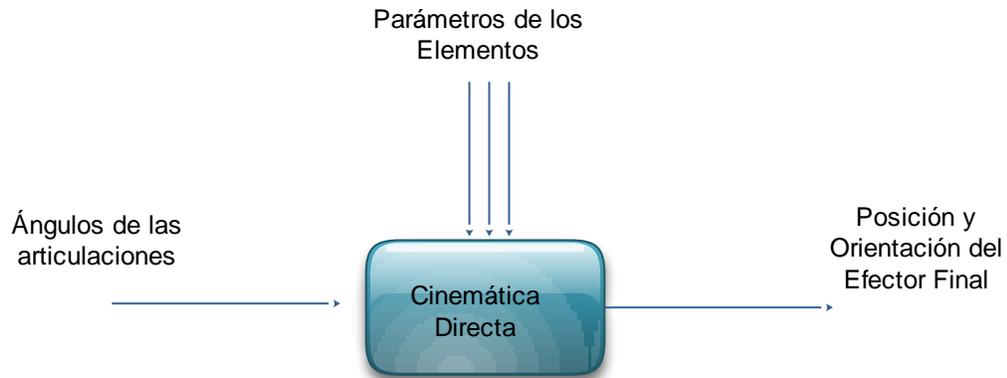


Figura 2.3. Cinemática Directa (Fu, et al, 1988).

Las coordenadas homogéneas en un espacio euclidiano tridimensional son útiles para desarrollar transformaciones matriciales que incluyan rotación, translación, escalado y transformación de perspectiva, por lo que se puede considerar a una matriz de transformación homogénea que consiste en cuatro submatrices.

$$T = \begin{bmatrix} R_{3x3} & p_{3x1} \\ f_{1x3} & 1 \end{bmatrix} = \begin{bmatrix} \text{matriz de rotación} & \vdots & \text{vector de posición} \\ \text{transformación de perspectiva} & & \text{escalado} \end{bmatrix} \quad (2.1)$$

Donde la submatriz 3x3 superior izquierda representa la matriz de rotación; la submatriz superior derecha 3x1 representa el vector de posición con respecto al sistema de referencia, la submatriz inferior izquierda 1x3 representa la transformación de perspectiva y el cuarto elemento diagonal es el factor de escala global.

En resumen una matriz de transformación homogénea 4x4 transforma un vector expresado en coordenadas homogéneas con respecto al sistema de coordenadas OUVW en el sistema de referencia OXYZ con escala igual a 1.

$$\hat{p}_{xyz} = T \hat{p}_{uvw} \quad (2.2)$$

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Para describir la relación traslacional y rotacional entre elementos adyacentes, Denavit y Hartenberg, propusieron un método matricial para establecer de forma sistemática un sistema de coordenadas (sistema ligado al cuerpo) para cada elemento de una cadena articulada. La representación de Denavit-Hartenberg (D-H) resulta una matriz de transformación



homogénea 4x4 que representa cada uno de los sistemas de coordenadas de los elementos en la articulación con respecto al sistema de coordenadas del elemento previo. Se puede establecer para cada elemento en sus ejes de articulación un sistema de coordenadas cartesiano ortogonal (x_i, y_i, z_i), donde: $i = 1, 2, \dots, n$ (n =número de grados de libertad), más el sistema de coordenadas de la base.

Cada sistema de coordenadas se determina y establece sobre la base de tres reglas que son indispensables:

1. El eje z_{i-1} yace a lo largo del eje de la articulación.
2. El eje x_i es normal al eje z_{i-1} y apunta hacia fuera de él.
3. El eje y_i completa el sistema de coordenadas dextrógiro según se requiera.

La representación D-H de un elemento rígido depende de cuatro parámetros geométricos asociados con cada elemento. Estos cuatro parámetros describen completamente cualquier articulación prismática o de revolución (Merchán, 2000).

θ_i . Es el ángulo de la articulación del eje x_{i-1} al eje x_i respecto del eje z_{i-1} .

d_i . Es la distancia entre el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje z_{i-1} con el eje x_i a lo largo del eje z_{i-1} .

a_i . Es la distancia de separación desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo a lo largo del eje x_i .

α_i . Es el ángulo de separación del eje z_{i-1} al eje z_i respecto al eje x_i .

Conociendo estos puntos mencionados se hará un diagrama unifilar que representa al robot bípedo asignando debidamente los sistemas de coordenadas (Capítulo 3).

2.4 Cinemática Inversa

Cuando hablamos de cinemática inversa, dada la posición y orientación del efector final de un robot así como sus parámetros de articulación y elementos, se busca encontrar los ángulos de articulación correspondientes del robot de manera que se pueda posicionar como se desee el efector final (Figura 2.4).

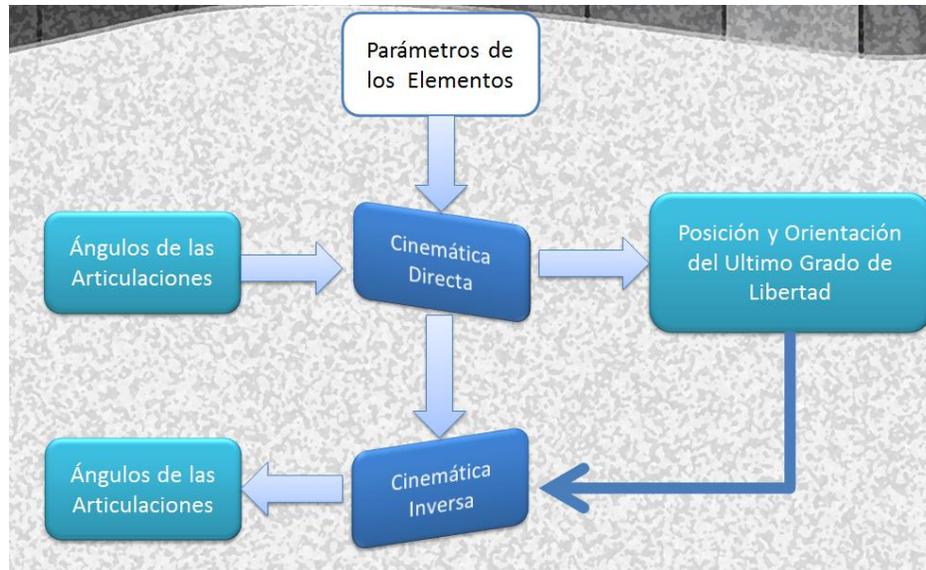


Figura 2.4. Cinemática Inversa (Fu, et al, 1988)

La cinemática inversa consiste en determinar los valores de las articulaciones que satisfagan condiciones deseadas de posición, velocidad y aceleración en el espacio cartesiano. De lo anterior podemos identificar tres problemas, la cinemática inversa para posiciones, cinemática inversa para velocidades y cinemática inversa para aceleraciones.

En cuanto a cinemática inversa para posiciones, la cual representa un sistema de ecuaciones no lineales debido a que cada una de las ecuaciones depende de las variables de cada GDL además de ser un sistema sobredeterminado porque tenemos más ecuaciones que incógnitas.

Como el sistema de ecuaciones no es lineal, primero se linealiza con ayuda de algún método iterativo. El método matemático de Newton-Rhapson es el siguiente:

$$f_i(x_1, x_2, \dots, x_n) = 0, \quad i = 1, 2, \dots, n \quad (2.4)$$

Donde f_i es una función no lineal de la estimación inicial.

Una estimación inicial de la solución de dicho sistema de ecuaciones se escribe de la siguiente manera:

$$x_j = \hat{x}_j + \Delta x_j \quad (2.5)$$

Donde x_j es la estimación inicial y Δx_j es una corrección no conocida. Si expandimos la ecuación 2.5 para obtener un polinomio de Taylor truncado de primer orden alrededor \hat{x}_j obtenemos:



$$\sum_j \frac{\partial f_i}{\partial x_j} \Delta x_j = -f_i(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \quad (2.6)$$

Donde las derivadas parciales se evalúan con las condiciones iniciales ecuación 2.6 puede escribirse en forma de matriz como:

$$J\Delta = -f \quad (2.7)$$

Donde J es la matriz Jacobiana:

$$J = \left[\frac{\partial f_i}{\partial x_j} \right] \quad (2.8)$$

$$\Delta x = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \quad (2.9)$$

$$f = \begin{bmatrix} f_1(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \\ f_2(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \\ \vdots \\ f_3(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \end{bmatrix} \quad (2.10)$$

Las derivadas parciales pueden evaluarse con una aproximación de diferencia de la forma siguiente:

$$\frac{\partial y}{\partial x} \approx \frac{f_i(\hat{x}_1, \dots, \hat{x}_j + \sigma x_j, \dots, \hat{x}_n) - f_i(\hat{x}_1, \dots, \hat{x}_j, \dots, \hat{x}_n)}{\sigma x_j} \quad (2.11)$$

De estas ecuaciones se obtiene el Jacobiano para poder resolver el siguiente sistema lineal y de esta manera encontrar los ángulos necesarios que satisfagan al sistema para cierto punto deseado. El sistema resultante es:

$$J\theta = f \quad (2.12)$$

Una vez que el sistema de ecuaciones es lineal, se procede a resolverlo con métodos iterativos, en este caso la resolución está basada en el método Moore-Penrose que hace uso de la matriz pseudoinversa:

$$\theta = (J^T J)^{-1} J^T (f) = J^\dagger (f) \quad (2.13)$$



2.5 Modelo Dinámico

Elaborar el análisis dinámico de un sistema mecánico implica encontrar las relaciones matemáticas entre sus coordenadas “q” y fuerzas “Q” generalizadas. De acuerdo a Spong et al. (2006), en la deducción de dichas relaciones se pueden distinguir dos objetivos principales:

- Obtener ecuaciones de forma cerrada que describan la evolución en el tiempo de las coordenadas generalizadas $\mathbf{q}(t)$.
- Conocer cuáles son las fuerzas generalizadas $\mathbf{Q}(t)$ que deben ser aplicadas para producir una evolución particular en el tiempo de las coordenadas generalizadas $\mathbf{q}(t)$, sin preocuparse de la relación funcional entre ambas.

Aunque se sabe que las formulaciones de Lagrange-Euler y Newton-Euler para deducir las ecuaciones dinámicas de un sistema conducen a los mismos resultados; se considera que el método basado en energía de Lagrange-Euler es superior para alcanzar el primer objetivo, mientras que el método recursivo de Newton-Euler resulta más apropiado para el segundo objetivo.

Otra de las consideraciones a tomar es el tiempo de cálculo de la dinámica, en este caso de la dinámica inversa para obtener las torques necesarias en las articulaciones y así el bípedo pueda caminar de manera más suave. (Fu et al 1994)

El modelo más difícilmente computable es el Lagrange-Euler, el modelo más rápido es el de Newton-Euler esto se puede apreciar en la tabla 1 que muestra el número de operaciones, aunque siendo realistas los dos modelos son bastante tardados a la hora de computarlos.

Tabla 2.1.- Comparación del número de operaciones realizadas por los métodos de Lagrange-Euler y Newton-Euler.

Método	Lagrange-Euler	Newton-Euler
Multiplicaciones	$\frac{128}{3}n^4 + \frac{512}{3}n^3 + \frac{739}{3}n^2 + \frac{160}{3}n$	$132n$
Sumas	$\frac{98}{3}n^4 + \frac{781}{6}n^3 + \frac{559}{3}n^2 + \frac{245}{6}n$	$111n - 4$
Representación cinemática	Matrices homogéneas 4×4	Matrices de rotación y vectores de posición
Ecuaciones de movimiento	Ecuaciones diferenciales en forma cerrada	Ecuaciones recursivas



La gran desventaja del modelo de Newton-Euler es que solo funciona para velocidades bajas donde los elementos de Coriolis y centrífugo, que no toman en cuenta, son despreciables. Y la gran desventaja del método de Lagrange-Euler es la cantidad de cálculos que se deben de hacer por lo que varios científicos han optado por simplificar el modelo dinámico del robot bípedo, quedando las siguientes dos simplificaciones:

La primera es que un bípedo puede ser representado por el modelo de péndulo invertido en 3D considerando la masa del péndulo como el centro geométrico de la suma de las masas de las piernas del bípedo, este punto es llamado el *Centre Of Gravity* (COG).

La segunda, algo parecida a la anterior, es que este puede ser representado por el modelo de car-table propuesto por Kajita et al (2005) el cual también usa el COG para modelar matemáticamente, esto se observa en la figura 2.5.

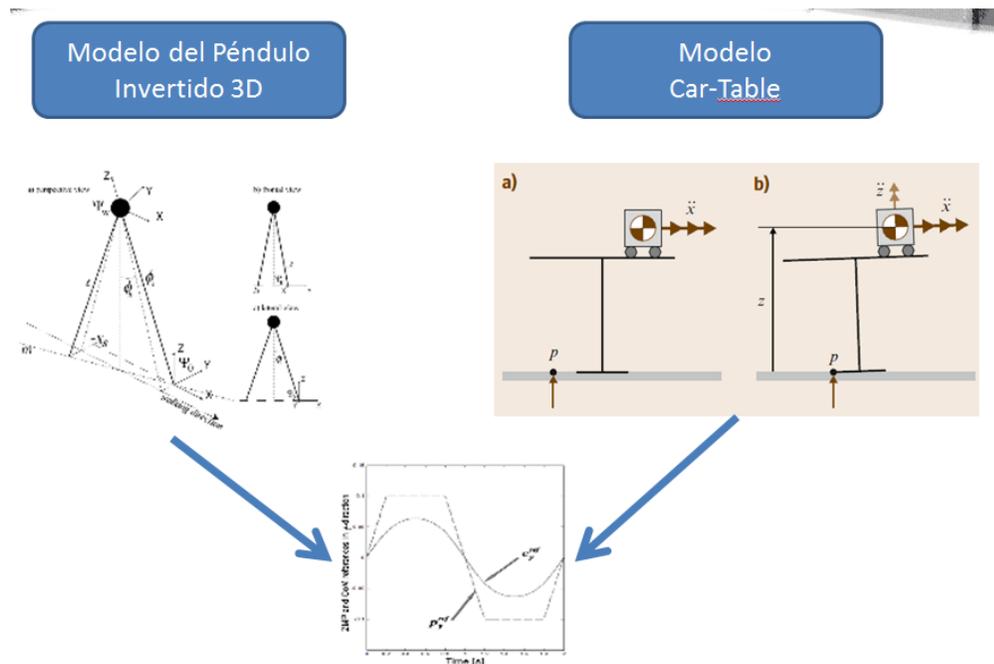


Figura 2.5. Modelos de péndulo invertido 3D y modelo car-table

Estos modelos necesitan de un punto en el suelo para poder ser controlados, obviamente se deduce que para que el robot no caiga este punto en el suelo no puede salir de un cierto parámetro, este parámetro se llama “Polígono de Soporte”. El punto en el suelo es llamado ZMP (Zero Moment Point) y es un criterio de estabilidad que se define como “El punto en el suelo donde los momentos resultantes son igual a cero” (Vukobratovic et al., 1975) esto se observa en la figura 2.6.

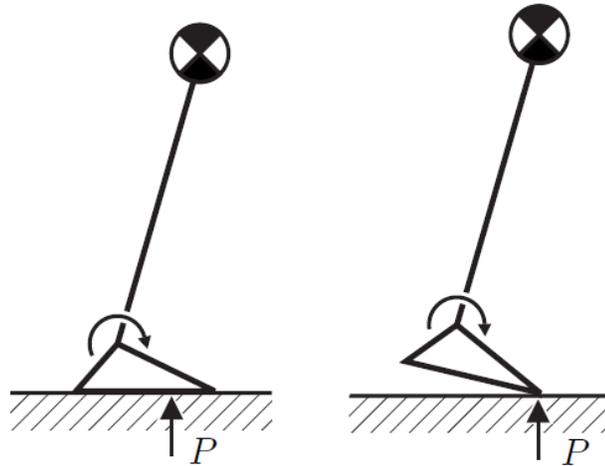


Figura 2.6. El punto P es el ZMP

2.6 Controladores en Robots Bipedos

La idea de un controlador en un robot bípedo es hacer que el bípedo pueda caminar sin caerse, esto se logra haciendo que ciertos puntos del bípedo (como pueden ser la cadera o el pie) sigan trayectorias definidas tales que emulen la marcha humana, esto aunado a que el ZMP siempre se mantenga dentro del polígono de soporte, es por eso que los puntos importantes del controlador son los siguientes:

- Hacer que el robot bípedo llegue a los ángulos calculados por la cinemática inversa de cada articulación para así seguir una trayectoria deseada de la cadera y el pie.
- Tomando en cuenta las trayectorias de la cadera y el pie, hacer que el ZMP no salga del polígono de soporte.

Estos puntos son la base para poder generar un diagrama de control que tenga por lo menos el control de las articulaciones y el control del ZMP como lo podemos ver en la figura 2.7 que tiene los elementos básicos para poder hacer que el robot bípedo camine sin caerse, claro está que dentro de los diagramas de control propuestos se puede hacer uso de métodos heurísticos con el fin de proporcionar robustez y que el bípedo pueda caminar por diferentes terrenos como se muestra en el diagrama en la figura 2.8 que hace uso de una lógica difusa.

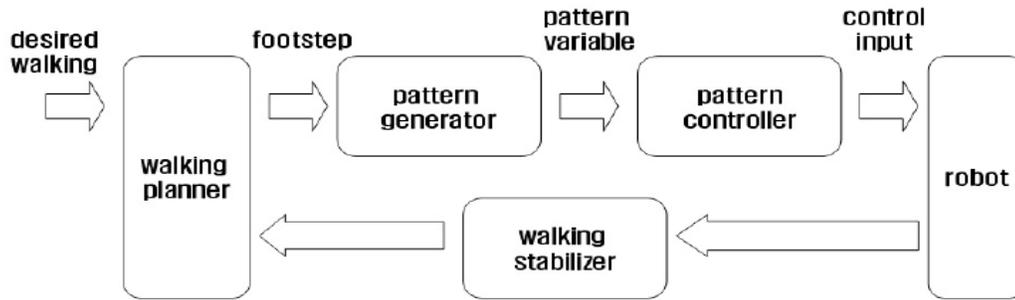


Figura 2.7. Caminata deseada (Jonghoon Park et al (2007))

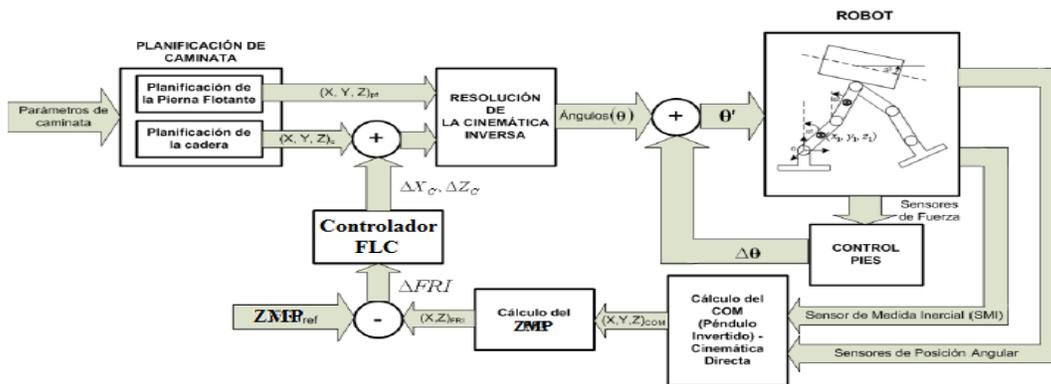


Figura 2.8. Diagrama del controlador ZMP-FLC (Narong and Manukid et al (2009))

Como ya se ha mencionado antes el robot humanoide ASIMO es el más avanzado en cuanto a investigación para robots bípedos se refiere, es por ello que al ver el diagrama de control que utilizan en la fase de marcha del mismo (Figura 2.9) es parecido al expuesto en la figura 2.7 pero con ciertas diferencias como el control de inclinación de cuerpo, el control de la reacción de suelo etc. Estos diagramas de control expuestos serán de ayuda para proponer un diagrama de control para el bípedo Scout, diagrama que será presentado en el capítulo 5.

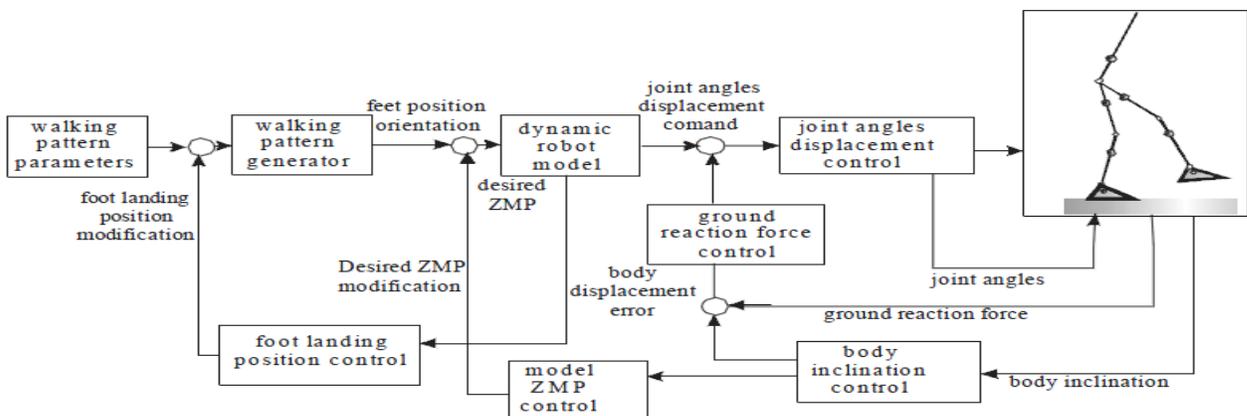


Figura 2.9. Diseño del controlador de los robots HONDA (Chevallereau et al (2007))



2.7 Arquitectura del Bípido Scout

El bípido Scout está constituido por piernas de seis eslabones unidas entre sí mediante un eslabón central al que, en este proyecto se le llamará cadera, sus trece eslabones se conectan entre sí a través de juntas rotacionales actuadas por servomotores. En la figura 2.10 se muestra el modelo CAD 3D del bípido en la configuración espacial que adoptan sus servomotores en posición neutral.

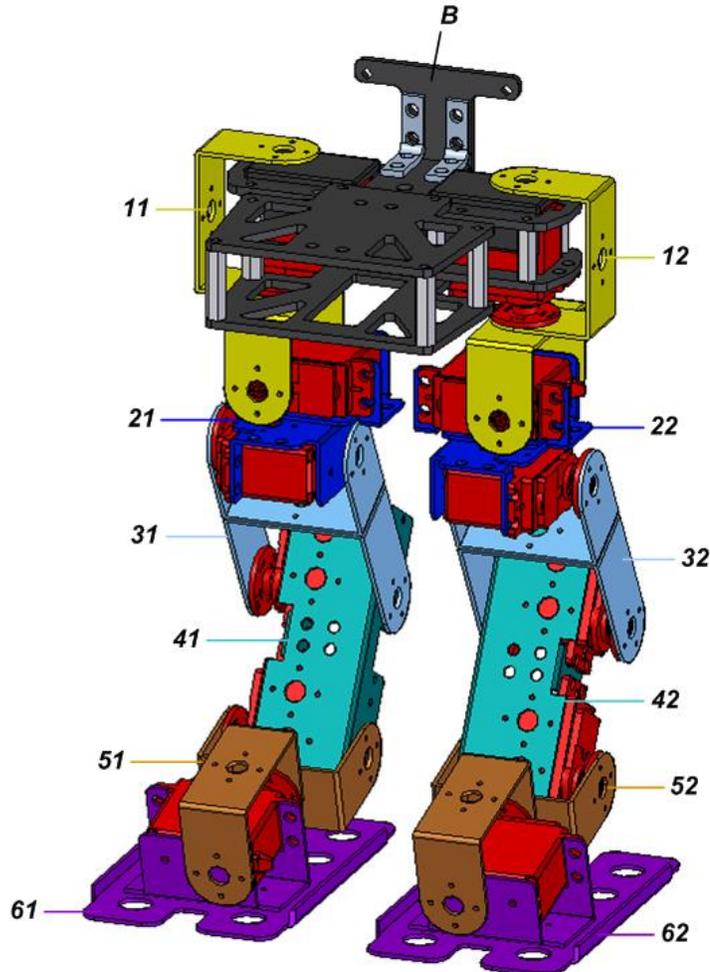


Figura 2.10. Modelo CAD 3D del bípido Scout (Narváez, 2010)

En la figura 2.10 se caracterizan las piezas de los eslabones por medio de diferentes colores y los servomotores en rojo. La cadera se identifica con la letra B y los eslabones de las piernas con las etiquetas " ni "; donde " n " es el número de eslabón e " i " es el número de pierna, se asignará $i=1$ para la pierna izquierda e $i=2$ para la derecha.

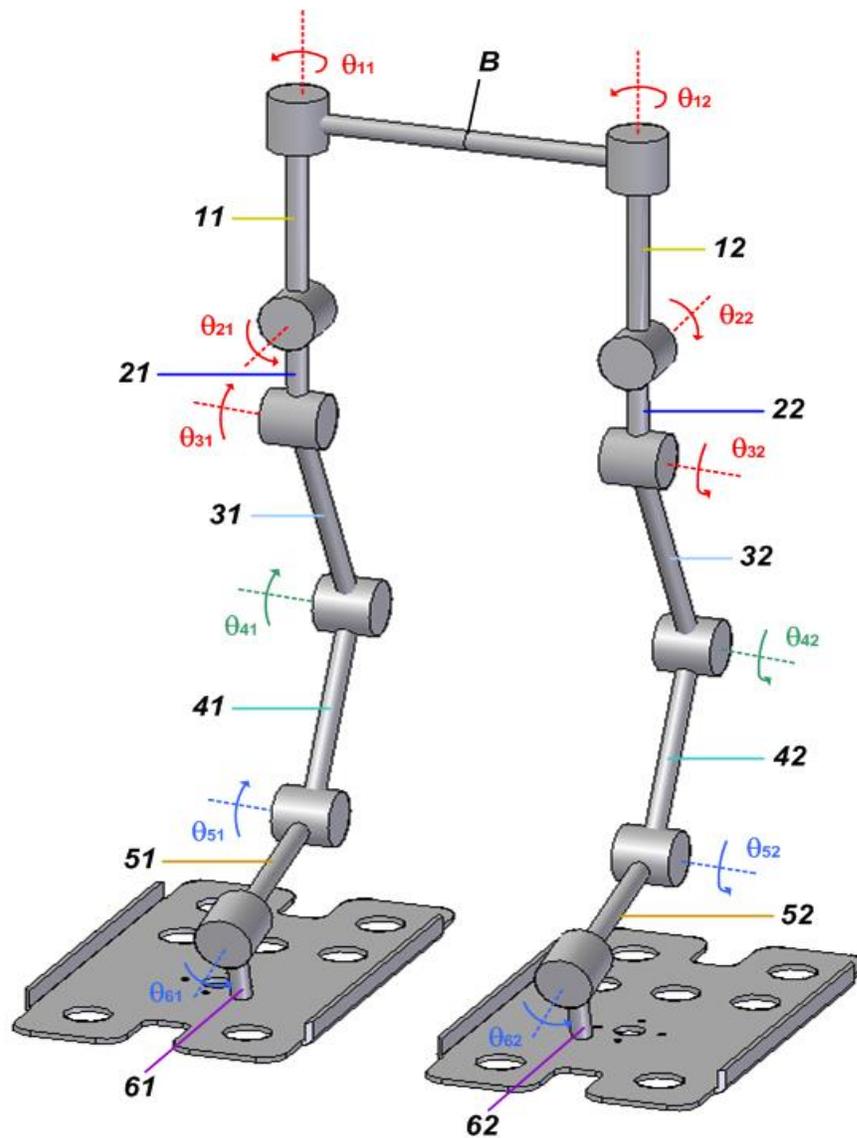


Figura 2.11. Modelo CAD 3D simplificado del bípido Scout (Narváez, 2010)

Un modelo simplificado como el mostrado en la figura 2.11 es necesario para el análisis cinemático, en este se aprecia que las juntas son representadas por medio de cilindros cuyos ejes coinciden con el de los servomotores y se unen a través de barras. La rotación de los ejes de los servomotores se describirá por medio de los ángulos " θ_{ni} " que obedece a la nomenclatura antes descrita.

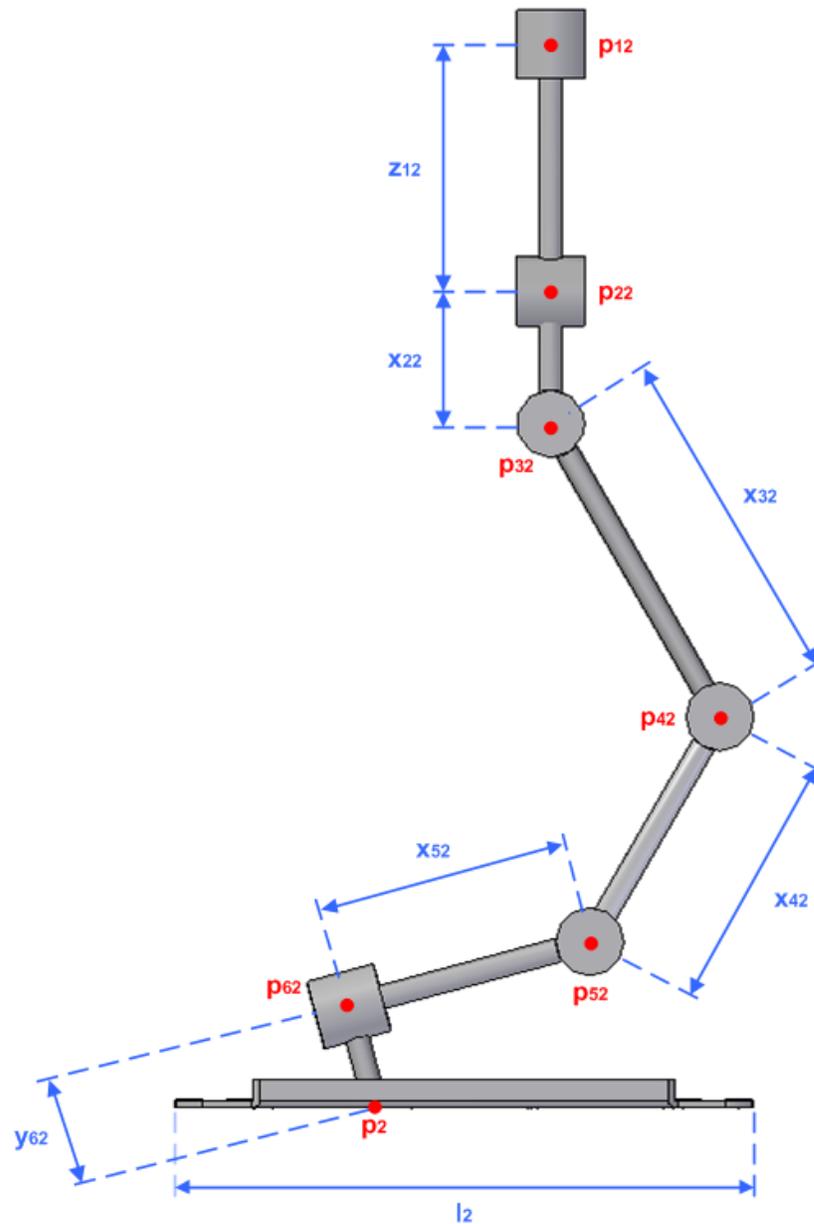


Figura 2.12. Vista sagital derecha del modelo simplificado del bípido. Las cotas indican las distancias con relevancia cinemática (Narváez, 2010).

Como se muestra en la figura 2.12 las cotas marcadas en azul son las longitudes de cada eslabón que conforman la pierna izquierda del bípido Scout de manera similar las distancias marcadas también son las mismas en la pierna derecha.

En el Anexo A se pueden consultar las listas de los elementos constituyentes del bípido, planos de su ensamble y su modelo simplificado, los valores de los parámetros geométricos determinados mediante su modelado asistido por computadora, las masas de las articulaciones, sus matrices de inercia entre otros datos que servirán para su análisis dinámico en el capítulo 4.



3

Análisis Cinemático

En este capítulo se generará una simulación por computadora del robot bípedo Scout siguiendo una trayectoria deseada, tratando de emular la marcha humana.



3.1 Cinemática del Robot Bípido Scout

Como ya se explicó en el capítulo pasado la cinemática es el estudio del movimiento de los cuerpos sin tomar en cuenta las fuerzas que lo originan y esta se divide en dos, la cinemática directa y la cinemática inversa. En este caso, la que nos importa es la cinemática inversa ya que lo que se hará es proponer trayectorias de ciertos puntos del robot bípido y este tiene que seguirlas, por lo tanto el dato de entrada es la posición y orientación de cierto punto y el de salida se refiere a los ángulos necesarios de las articulaciones para llegar a él. Es obvio que la cinemática directa es importante también ya que sin ella no se podría resolver la cinemática inversa.

3.1.1 Cinemática Directa del Robot Bípido Scout

Se identifican los valores de la cinemática directa para dos puntos importantes del robot bípido Scout, estos puntos son la cadera y el pie flotante tomando como referencia el pie de apoyo, se observan en la figura 3.1 siendo “A” el pie de apoyo, “B” la cadera y “C” el pie flotante, es obvio que entre “A” y “C” cambiarán el rol de pie flotante a pie de apoyo dependiendo los pasos que se den en la marcha.

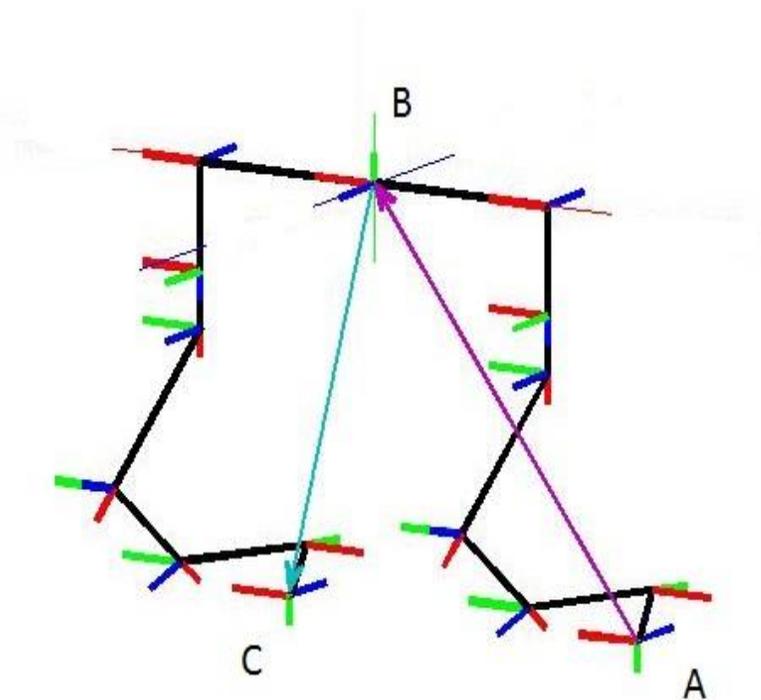


Figura 3.1 Diagrama unifilar del Bípido Scout.

Como se observa en la figura 3.1 primero se obtiene la cinemática directa de la cadera con respecto al pie de apoyo y después la cinemática directa del pie flotante con respecto a la cadera, de esta forma se obtienen los puntos de interés referenciados al pie de apoyo para posteriormente calcular la cinemática inversa.



Para obtener la cinemática directa se utiliza el método de Denavit-Hartenberg ya que tiene la virtud de posicionar siempre el eje de rotación de cada articulación al eje Z de cada uno de los marcos de referencia. Como ya se explicó el procedimiento en el capítulo anterior solo resta mencionar los parámetros de Denavit-Hartenberg para cada GDL, cabe señalar que para esto se calcula la cinemática directa del pie de apoyo izquierdo a la cadera, posteriormente se encuentra la cinemática directa de la cadera al pie flotante. Los parámetros de Denavit-Hartenberg se muestran en la tabla 3.1.

Tabla 3.1 Parámetros D-H del Bípodo (pie izquierdo).

Articulación i	θ_i	α_i	a_i	d_i
1	0	0	0	-0.05458
2	$\pi/4$	0	0.05729	0
3	$\pi/3$	0	0.07464	0
4	$-\pi/6$	$\pi/2$	0.02908	0
5	0	0	0.05458	0
6	0	π	0.04454	0

Tomando en cuenta las distancias máximas de los eslabones, las matrices de transformación homogénea y su relación directa entre eslabones se determina la matriz final desde el último sistema de coordenadas con respecto a la base o sistema de referencia con base en los parámetros de D-H ya establecidos. Se observa que como es con respecto al pie de apoyo se usan 6 GDL para así tomar en cuenta el movimiento de una sola pierna del bípodo. Es importante aclarar que la articulación 1 empieza en el punto A de la figura 3.1 y la articulación 6 termina en el punto B de la misma.

Una vez que se establece el sistema de coordenadas D-H para cada elemento se procede a desarrollar una matriz de transformación homogénea que relacione el sistema de coordenadas i-ésimo con el sistema de coordenadas (i - 1)-ésimo dando como resultado una matriz compuesta ${}^{i-1}T_i$, conocida como la matriz de transformación D-H para sistemas de coordenadas adyacentes i e i-1.

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$



Seguendo la nomenclatura mostrada en la ecuación 3.1 y haciendo la reducción de los términos que se multiplican por cero. Las matrices de transformación homogénea son las siguientes:

$${}^0T_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$${}^1T_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$${}^2T_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$${}^3T_4 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & a_4 \cos(\theta_4) \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & a_4 \sin(\theta_4) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$${}^4T_5 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & a_5 \cos(\theta_5) \\ \sin(\theta_5) & \cos(\theta_5) & 0 & a_5 \sin(\theta_5) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$${}^5T_6 = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & a_6 \cos(\theta_6) \\ \sin(\theta_6) & \cos(\theta_6) & 0 & a_6 \sin(\theta_6) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

Como el sistema de coordenadas de la base (pie de apoyo) no coincidía con el sistema de coordenadas global se ha tenido que multiplicar por la matriz “A” además de que también se tuvo que multiplicar por una matriz de rotación (“B”) para que el sistema de referencia del primer GDL estuviera correctamente orientado, las matrices se muestran a continuación:

$$A = \begin{bmatrix} 1 & 0 & 0 & df1 \\ 0 & -1 & 0 & df2 \\ 0 & 0 & -1 & df3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$



$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(h_1) & -\sin(h_1) & 0 \\ 0 & \sin(h_1) & \cos(h_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

Corregido esto, se procede a multiplicar las matrices de transformación homogénea (ecuación 3.10) para así referenciar el sistema de coordenadas de la cadera al sistema de coordenadas del pie de apoyo.

$${}^0T_6 = A * B * {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6 \quad (3.10)$$

$${}^0T_6 = \begin{bmatrix} f1 & f4 & f7 & f10 \\ f2 & f5 & f8 & f11 \\ f3 & f6 & f9 & f12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

Como la matriz resultante es muy extensa se opta por fragmentarla siguiendo la ecuación 3.11, también se decide seguir la siguiente nomenclatura, $C_i = \cos(\theta_i)$ y $S_i = \sin(\theta_i)$.

$$f1 = C_6 \left(C_1 C_5 + S_5 (C_4 (C_2 S_1 S_3 + C_3 S_1 S_2) - S_4 (S_1 S_2 S_3 - C_2 C_3 S_1)) \right) + S_6 (C_4 (S_1 S_2 S_3 - C_2 C_3 S_1) + S_4 (C_2 S_1 S_3 + C_3 S_1 S_2)) \quad (3.12)$$

$$f2 = S_6 \left(C_4 (C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2)) \right) + S_4 (C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2)) + C_6 (S_5 \left(C_4 (C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2)) - S_4 (C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2)) \right) + S_{h_1} C_5 S_1) \quad (3.13)$$

$$f3 = S_6 \left(C_4 (C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2)) \right) + S_4 (C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2)) + C_6 (S_5 \left(C_4 (C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2)) - S_4 (C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2)) \right) + S_{h_1} C_5 S_1) \quad (3.14)$$



$$f4 = S_6 \left(C_1 C_5 + S_5 (C_4 (C_2 S_1 S_3 + C_3 S_1 S_2) - S_4 (S_1 S_2 S_3 - C_2 C_3 S_1)) \right) - C_6 (C_4 (S_1 S_2 S_3 - C_2 C_3 S_1) + S_4 (C_2 S_1 S_3 + C_3 S_1 S_2)) \quad (3.15)$$

$$f5 = S_6 \left(S_5 \left(C_4 \left(C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) \right) - S_4 \left(C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) \right) \right) + S_{h_1} C_5 S_1 \right) - C_6 \left(C_4 \left(C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) \right) + S_4 \left(C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) \right) \right) \quad (3.16)$$

$$f6 = S_6 \left(S_5 \left(C_4 \left(C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) \right) - S_4 \left(C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) \right) \right) - C_{h_1} C_5 S_1 \right) - C_6 \left(C_4 \left(C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) \right) + S_4 \left(C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) \right) \right) \quad (3.17)$$

$$f7 = C_1 S_5 - C_5 (C_4 (C_2 S_1 S_3 + C_3 S_1 S_2) - S_4 (S_1 S_2 S_3 - C_2 C_3 S_1)) \quad (3.18)$$

$$f8 = S_{h_1} S_1 S_5 - C_5 \left(C_4 \left(C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) \right) - S_4 \left(C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) \right) \right) \quad (3.19)$$

$$f9 = -C_5 \left(C_4 \left(C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) \right) - S_4 \left(C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) \right) \right) - C_{h_1} S_1 S_5 \quad (3.20)$$



$$\begin{aligned}
f10 = & df1 + a_6 C_6 \left(C_1 C_5 + S_5 (C_4 (C_2 S_1 S_3 + C_3 S_1 S_2) - S_4 (S_1 S_2 S_3 - C_2 C_3 S_1)) \right) - a_4 C_4 (C_2 S_1 S_3 + C_3 S_1 S_2) + \\
& a_4 S_4 (S_1 S_2 S_3 - C_2 C_3 S_1) - a_5 C_5 \left(C_4 (C_2 S_1 S_3 + C_3 S_1 S_2) - S_4 (S_1 S_2 S_3 - C_2 C_3 S_1) \right) + a_6 C_6 \left(C_4 (S_1 S_2 S_3 - C_2 C_3 S_1) + \right. \\
& \left. S_4 (C_2 S_1 S_3 + C_3 S_1 S_2) \right) + a_5 C_1 S_5 - a_2 S_1 S_2 - a_3 C_2 S_1 S_3 - a_3 C_3 S_1 S_2
\end{aligned} \tag{3.21}$$

$$\begin{aligned}
f11 = & df2 + d_1 C_{h_1} + df S_{h_1} - a_2 C_{h_1} C_2 + a_6 C_6 \left(S_5 \left(C_4 \left(C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) \right) - \right. \right. \\
& \left. \left. S_4 \left(C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) \right) \right) + S_{h_1} C_5 S_1 \right) - a_5 C_5 \left(C_4 \left(C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + \right. \right. \\
& \left. \left. S_{h_1} C_1 C_2) \right) - S_4 \left(C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) \right) \right) + a_6 S_6 \left(C_4 \left(C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - \right. \right. \\
& \left. \left. S_{h_1} C_1 S_2) \right) + S_4 \left(C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) \right) \right) - a_4 C_4 \left(C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) - S_3 (C_{h_1} S_2 + \right. \\
& \left. S_{h_1} C_1 C_2) \right) - a_3 C_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) + a_4 S_4 \left(C_3 (C_{h_1} S_2 + S_{h_1} C_1 C_2) + S_3 (C_{h_1} C_2 - S_{h_1} C_1 S_2) \right) + a_3 S_3 (C_{h_1} S_2 + \\
& S_{h_1} C_1 C_2) + a_2 S_{h_1} C_1 S_2 + a_5 S_{h_1} S_1 S_5
\end{aligned} \tag{3.22}$$

$$\begin{aligned}
f12 = & df3 + d_1 S_{h_1} - df C_{h_1} - a_2 S_{h_1} C_2 - a_5 C_5 \left(C_4 \left(C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) \right) - \right. \\
& \left. S_4 \left(C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) \right) \right) + a_6 C_6 \left(S_5 \left(C_4 \left(C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) \right) - \right. \right. \\
& \left. \left. S_4 \left(C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) \right) \right) - C_{h_1} C_5 S_1 \right) + a_6 S_6 \left(C_4 \left(C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + \right. \right. \\
& \left. \left. C_{h_1} C_1 S_2) \right) + S_4 \left(C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) \right) \right) - a_4 C_4 \left(C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) - S_3 (S_{h_1} S_2 - \right. \\
& \left. C_{h_1} C_1 C_2) \right) - a_3 C_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) + a_4 S_4 \left(C_3 (S_{h_1} S_2 - C_{h_1} C_1 C_2) + S_3 (S_{h_1} C_2 + C_{h_1} C_1 S_2) \right) + a_3 S_3 (S_{h_1} S_2 - \\
& C_{h_1} C_1 C_2) - a_2 C_{h_1} C_1 S_2 - a_5 C_{h_1} S_1 S_5
\end{aligned} \tag{3.23}$$



A continuación se obtienen los parámetros de Denavit–Hartenberg del pie derecho (pie flotante) partiendo de la cadera, estos parámetros están representados en la tabla 3.2.

Tabla 3.2 Parámetros D-H del Bípodo (Pie derecho)

Articulació i	θ_i	α_i	a_i	d_i
7	0	0	0	-0.0545
8	0	$\pi/2$	0.0290	0
9	$\pi/6$	0	0.0746	0
10	$-\pi/3$	0	0.0572	0
11	$-\pi/4$	0	0.0545	0
12	$\pi/2$	0	0.0234	0

Se observa que como es con respecto a la cadera se usan 6 GDL para así tomar en cuenta el movimiento de una sola pierna del bípodo. Es importante aclarar que la articulación 7 empieza en el punto B de la figura 3.1 y la articulación 12 termina en el punto C de la misma.

Siguiendo la nomenclatura mostrada en la ecuación 3.1 y haciendo la reducción de los términos que se multiplican por cero, las matrices de transformación homogénea son las siguientes:

$${}^6T_7 = \begin{bmatrix} \cos(\theta_7) & -\sin(\theta_7) & 0 & 0 \\ \sin(\theta_7) & \cos(\theta_7) & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.24)$$

$${}^7T_8 = \begin{bmatrix} \cos(\theta_8) & 0 & \sin(\theta_8) & a_8 \cos(\theta_8) \\ \sin(\theta_8) & 0 & -\cos(\theta_8) & a_8 \sin(\theta_8) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.25)$$

$${}^8T_9 = \begin{bmatrix} \cos(\theta_9) & -\sin(\theta_9) & 0 & a_9 \cos(\theta_9) \\ \sin(\theta_9) & \cos(\theta_9) & 0 & a_9 \sin(\theta_9) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.26)$$



$${}^9T_{10} = \begin{bmatrix} \cos(\theta_{10}) & -\sin(\theta_{10}) & 0 & a_{10}\cos(\theta_{10}) \\ \sin(\theta_{10}) & \cos(\theta_{10}) & 0 & a_{10}\sin(\theta_{10}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

$${}^{10}T_{11} = \begin{bmatrix} \cos(\theta_{11}) & -\sin(\theta_{11}) & 0 & a_{11}\cos(\theta_{11}) \\ \sin(\theta_{11}) & \cos(\theta_{11}) & 0 & a_{11}\sin(\theta_{11}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.28)$$

$${}^{11}T_{12} = \begin{bmatrix} \cos(\theta_{12}) & -\sin(\theta_{12}) & 0 & a_{12}\cos(\theta_{12}) \\ \sin(\theta_{12}) & \cos(\theta_{12}) & 0 & a_{12}\sin(\theta_{12}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.29)$$

Por último se multiplica por una matriz de rotación (“C”) para hacer que el pie flotante siempre este orientado de manera perpendicular al suelo.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(h_2) & -\sin(h_2) & 0 \\ 0 & \sin(h_2) & \cos(h_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.30)$$

Se procede a multiplicar las matrices de trasformación homogénea (ecuación 3.18) para así referenciar el sistema de coordenadas del pie flotante al sistema de coordenadas de la cadera.

$${}^6T_{12} = {}^6T_7 * {}^7T_8 * {}^8T_9 * {}^9T_{10} * {}^{10}T_{11} * {}^{11}T_{12} * C \quad (3.31)$$

$${}^6T_{12} = \begin{bmatrix} f13 & f16 & f19 & f22 \\ f14 & f17 & f20 & f23 \\ f15 & f18 & f21 & f24 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

Como la matriz resultante es de igual manera, muy extensa (ecuación 3.11) se opta por fragmentarla siguiendo la ecuación 3.31, también se decide seguir la siguiente nomenclatura, $C_i = \cos(\theta_i)$ y $S_i = \sin(\theta_i)$.



$$f13 = C_6 \left(C_5 (C_4 (C_3 S_1 + C_1 S_2 S_3) - S_4 (S_1 S_3 - C_1 C_3 S_2)) \right. \\ \left. - S_5 (C_4 (S_1 S_3 - C_1 C_3 S_2) + S_4 (C_3 S_1 + C_1 S_2 S_3)) \right) + C_1 C_2 S_6 \quad (3.33)$$

$$f14 = C_6 \left(C_5 (C_4 (C_1 C_3 - S_1 S_2 S_3) - S_4 (C_1 S_3 + C_3 S_1 S_2)) \right. \\ \left. - S_5 (C_4 (C_1 S_3 + C_3 S_1 S_2) + S_4 (C_1 C_3 - S_1 S_2 S_3)) \right) - C_2 S_1 S_6 \quad (3.34)$$

$$f15 = C_6 (C_5 (C_2 C_3 S_4 + C_2 C_4 S_3) - S_5 (C_2 S_3 S_4 - C_2 C_3 S_4)) - S_2 S_6 \quad (3.35)$$

$$f16 = S_{h_2} \left(S_6 \left(C_5 (C_4 (C_3 S_1 + C_1 S_2 S_3) - S_4 (S_1 S_3 - C_1 C_3 S_2)) \right. \right. \\ \left. \left. - S_5 (C_4 (S_1 S_3 - C_1 C_3 S_2) + S_4 (C_3 S_1 + C_1 S_2 S_3)) \right) - C_1 C_2 C_6 \right) \\ + C_{h_2} \left(C_5 (C_4 (S_1 S_3 - C_1 C_3 S_2) + S_4 (C_3 S_1 + C_1 S_2 S_3)) \right) \\ + S_5 (C_4 (C_3 S_1 + C_1 S_2 S_3) - S_4 (S_1 S_3 - C_1 C_3 S_2)) \quad (3.36)$$

$$f17 = S_{h_2} \left(S_6 \left(C_5 (C_4 (C_1 C_3 - S_1 S_2 S_3) - S_4 (C_1 S_3 + C_3 S_1 S_2)) \right. \right. \\ \left. \left. - S_5 (C_4 (C_1 S_3 + C_3 S_1 S_2) + S_4 (C_1 C_3 - S_1 S_2 S_3)) \right) + C_2 C_6 S_1 \right) \\ + C_{h_2} \left(C_5 (C_4 (C_1 S_3 + C_3 S_1 S_2) + S_4 (C_1 C_3 - S_1 S_2 S_3)) \right) \\ + S_5 (C_4 (C_1 C_3 - S_1 S_2 S_3) - S_4 (C_1 S_3 + C_3 S_1 S_2)) \quad (3.37)$$

$$f18 = S_{h_2} \left(C_6 S_2 + S_6 (C_5 (C_2 C_3 S_4 + C_2 C_4 S_3) - S_5 (C_2 S_3 S_4 - C_2 C_3 S_4)) \right) + C_{h_2} (C_5 (C_2 S_3 S_4 - C_2 C_3 S_4) + \\ S_5 (C_2 C_3 S_4 + C_2 C_4 S_3)) \quad (3.38)$$



$$\begin{aligned} f19 = & C_{h_2} \left(S_6 \left(C_5 \left(C_4 (C_3 S_1 + C_1 S_2 S_3) - S_4 (S_1 S_3 - C_1 C_3 S_2) \right) \right. \right. \\ & \left. \left. - S_5 \left(C_4 (S_1 S_3 - C_1 C_3 S_2) + S_4 (C_3 S_1 + C_1 S_2 S_3) \right) \right) - C_1 C_2 C_6 \right) \\ & - S_{h_2} \left(C_5 \left(C_4 (S_1 S_3 - C_1 C_3 S_2) + S_4 (C_3 S_1 + C_1 S_2 S_3) \right) \right) \\ & + S_5 \left(C_4 (C_3 S_1 + C_1 S_2 S_3) - S_4 (S_1 S_3 - C_1 C_3 S_2) \right) \end{aligned} \quad (3.39)$$

$$\begin{aligned} f20 = & C_{h_2} \left(S_6 \left(C_5 \left(C_4 (C_1 C_3 - S_1 S_2 S_3) - S_4 (C_1 S_3 + C_3 S_1 S_2) \right) \right. \right. \\ & \left. \left. - S_5 \left(C_4 (C_1 S_3 + C_3 S_1 S_2) + S_4 (C_1 C_3 - S_1 S_2 S_3) \right) \right) + C_2 C_6 S_1 \right) \\ & - S_{h_2} \left(C_5 \left(C_4 (C_1 S_3 + C_3 S_1 S_2) + S_4 (C_1 C_3 - S_1 S_2 S_3) \right) \right) \\ & + S_5 \left(C_4 (C_1 C_3 - S_1 S_2 S_3) - S_4 (C_1 S_3 + C_3 S_1 S_2) \right) \end{aligned} \quad (3.40)$$

$$\begin{aligned} f21 = & C_{h_2} \left(C_6 S_2 + S_6 \left(C_5 (C_2 C_3 S_4 + C_2 C_4 S_3) - S_5 (C_2 S_3 S_4 - C_2 C_3 C_4) \right) \right) - S_{h_2} (C_5 (C_2 S_3 S_4 - C_2 C_3 C_4) + \\ & S_5 (C_2 C_3 S_4 + C_2 C_4 S_3)) \end{aligned} \quad (3.41)$$

$$\begin{aligned} f22 = & df + a_4 C_4 (S_1 S_3 - C_1 C_3 S_2) + a_4 S_4 (C_3 S_1 + C_1 S_2 S_3) + a_6 S_6 \left(C_5 \left(C_4 (C_3 S_1 + C_1 S_2 S_3) - S_4 (S_1 S_3 - \right. \right. \\ & \left. \left. C_1 C_3 S_2) \right) - S_5 \left(C_4 (S_1 S_3 - C_1 C_3 S_2) + S_4 (C_3 S_1 + C_1 S_2 S_3) \right) \right) + a_5 C_5 \left(C_4 (S_1 S_3 - C_1 C_3 S_2) + S_4 (C_3 S_1 + C_1 S_2 S_3) \right) + \\ & a_5 S_5 \left(C_4 (C_3 S_1 + C_1 S_2 S_3) - S_4 (S_1 S_3 - C_1 C_3 S_2) \right) + a_3 (S_1 S_3 - C_1 C_3 S_2) - a_6 C_1 C_2 C_6 - a_2 C_1 S_2 \end{aligned} \quad (3.42)$$

$$\begin{aligned} f23 = & a_4 C_4 (C_1 S_3 + C_3 S_1 S_2) + a_4 S_4 (C_1 C_3 - S_1 S_2 S_3) + a_6 S_6 \left(C_5 \left(C_4 (C_1 C_3 - S_1 S_2 S_3) - S_4 (C_1 S_3 + C_3 S_1 S_2) \right) - \right. \\ & \left. S_5 \left(C_4 (C_1 S_3 + C_3 S_1 S_2) + S_4 (C_1 C_3 - S_1 S_2 S_3) \right) \right) + a_5 C_5 \left(C_4 (C_1 S_3 + C_3 S_1 S_2) + S_4 (C_1 C_3 - S_1 S_2 S_3) \right) + a_5 S_5 \left(C_4 (C_1 C_3 - \right. \\ & \left. S_1 S_2 S_3) - S_4 (C_1 S_3 + C_3 S_1 S_2) \right) + a_3 (C_1 S_3 + C_3 S_1 S_2) + a_6 C_2 C_6 S_1 + a_2 S_1 S_2 \end{aligned} \quad (3.43)$$



$$f_{24} = -d_1 - a_2 C_2 - a_3 C_2 C_3 + a_6 C_6 S_2 + a_6 S_6 (C_5 (C_2 C_3 S_4 + C_2 C_4 S_3) - S_5 (C_2 S_3 S_4 - C_2 C_3 C_4)) + a_5 C_5 (C_2 S_3 S_4 - C_2 C_3 C_4) + a_5 S_5 (C_2 C_3 S_4 + C_2 C_4 S_3) - a_4 (C_2 C_3 C_4 - C_2 S_3 S_4) \quad (3.44)$$

Teniendo en cuenta la matriz resultante que está referenciando el sistema de coordenadas del pie flotante al sistema de coordenadas de la cadera es posible referenciar el pie flotante con el sistema de coordenadas del pie de apoyo esto con solo multiplicar las matrices de la ecuación 3.11 y 3.31 como se muestra en la ecuación 3.45.

$${}^0T_{12} = {}^0T_6 * {}^6T_{12} \quad (3.45)$$

La ecuación 3.45 es de suma relevancia al resolver la cinemática inversa para una trayectoria dada en el pie flotante ya que este punto debe de estar referenciado al pie de apoyo. Terminando esta sección los resultados que se obtienen son los siguientes:

1. La cinemática directa del robot bípedo.
2. Se referencia el sistema de coordenadas de la cadera al sistema de coordenadas del pie de apoyo.
3. Se referencia el sistema de coordenadas del pie flotante al sistema de coordenadas de la cadera.
4. Se referencia el sistema de coordenadas del pie flotante al sistema de coordenadas del pie de apoyo.

Los puntos 2 y 4 serán utilizados más adelante para la resolución de la cinemática inversa y así proponer trayectorias para la cadera y el pie flotante con el fin de poder generar una simulación por computadora.

3.1.2 Cinemática Inversa del Bípedo Scout

Uno de los problemas fundamentales y complejos en robótica es la resolución del problema cinemático inverso debido a que no se puede encontrar una metodología de solución aplicable a todos los robots, siendo factores determinantes: el número de grados de libertad, el número de efectores finales, el tipo de cadenas cinemáticas (abierta o cerrada), la redundancia, etc.

El problema cinemático inverso consiste en encontrar el estado de cada una de las articulaciones que conforman un robot, para lograr una posición y orientación deseadas del efector final (cadera o pie flotante en el caso del bípedo). En el robot bípedo se tienen que resolver dos cinemáticas inversas, la de la cadera y la del pie flotante. Comenzaremos con la de la cadera.



3.1.2.1 Resolución de la Cinemática Inversa de la Cadera del Robot Bípedo Scout

Como se puede observar en la ecuación 3.11 cada una de las “ f ” representan términos de la posición u orientación, estos términos dependen única y exclusivamente de las variables que deseamos resolver en la cinemática inversa, los términos de $df1, df2$ y $df3$ junto con h_1 y h_2 son términos que si tienen un valor numérico pero que en este caso se omiten para que las ecuaciones de las “ f ” no tengan términos decimales. En la ecuación 3.46 se observa el problema, en el lado derecho de la igualdad se pone la orientación y posición del punto deseado de la cadera.

$$\begin{bmatrix} f1 & f4 & f7 & f10 \\ f2 & f5 & f8 & f11 \\ f3 & f6 & f9 & f12 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.46)$$

Como el problema de la cinemática inversa es demasiado complejo debido a que no es lineal por que las “ f ” dependen de uno o más términos que deseamos resolver, se opta por linealizar el sistema de ecuaciones; una simplificación necesaria es la siguiente:

$$\begin{bmatrix} f1 \\ f2 \\ f3 \\ f4 \\ f5 \\ f6 \\ f7 \\ f8 \\ f9 \\ f10 \\ f11 \\ f12 \end{bmatrix} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ s_x \\ s_y \\ s_z \\ a_x \\ a_y \\ a_z \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad (3.47)$$

Dónde:

$$f(\bar{x}) = \begin{bmatrix} f1 \\ f2 \\ f3 \\ f4 \\ f5 \\ f6 \\ f7 \\ f8 \\ f9 \\ f10 \\ f11 \\ f12 \end{bmatrix} \quad f(x) = \begin{bmatrix} n_x \\ n_y \\ n_z \\ s_x \\ s_y \\ s_z \\ a_x \\ a_y \\ a_z \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad (3.48 \text{ y } 3.49)$$



Ahora utilizamos la serie de Taylor en cada función “ f ” para linealizarla de manera en que cada función dependa de solo una variable a solucionar, la serie de Taylor se trunca en el segundo sumando ya que es una buena aproximación hasta ese punto y requiere menos cálculos. Las ecuaciones 3.50, 3.51 y 3.52 son el procedimiento elegido para obtener la linealización de todas las funciones.

$$f(x_j) = f(\bar{x}_j) + \sum_{i=1}^j \left. \frac{\partial f(\bar{x}_j)}{\partial \theta_i} \right|_{\theta=\bar{\theta}} (\theta_i - \bar{\theta}_i) \quad (3.50)$$

Dónde:

- $i = \text{Número de grados de libertad}$
- $j = \text{Número de funciones a linealizar}$

Realizando las operaciones y sabiendo que el número de grados de libertad es 6 y el número de funciones a linealizar es 12, podemos hacer que las ecuaciones resultantes estén expresadas en forma de matrices con el fin de simplificar las operaciones a realizar como se muestra en la ecuación 3.51 y 3.52.

$$\begin{bmatrix} f(x_1) - f(\bar{x}_1) \\ \vdots \\ f(x_{12}) - f(\bar{x}_{12}) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(\bar{x}_1)}{\partial \theta_1} & \dots & \frac{\partial f(\bar{x}_1)}{\partial \theta_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(\bar{x}_{12})}{\partial \theta_1} & \dots & \frac{\partial f(\bar{x}_{12})}{\partial \theta_6} \end{bmatrix} \begin{bmatrix} \theta_1 - \bar{\theta}_1 \\ \vdots \\ \theta_6 - \bar{\theta}_6 \end{bmatrix} \quad (3.51)$$

$$\nabla f = J * \nabla \theta \quad (3.52)$$

En la ecuación 3.52 el término “ J ” es la matriz Jacobiana del sistema de ecuaciones la cual está formada por las derivadas parciales de primer orden de un vector de funciones, esta matriz Jacobiana nos otorga la posibilidad de aproximar linealmente a las funciones en un punto.

Para encontrar la solución de diferencia de los ángulos de las articulaciones solo basta con despejar la ecuación 3.52, pero para encontrar la inversa de la matriz Jacobiana debemos recurrir al teorema de Moore-Penrose que se explicó en el capítulo anterior.

$$\nabla \theta = J^{-1} * \nabla f \quad (3.53)$$

$$\nabla \theta = (J^T J)^{-1} J^T * \nabla f \quad (3.54)$$



Para obtener los ángulos calculados simplemente se le suman los ángulos propuestos en un inicio a la solución de la diferencia de los ángulos de las articulaciones.

$$\theta = \bar{\theta} + \nabla\theta \quad (3.55)$$

3.1.2.2 Solución de la Cinemática Inversa del Pie Flotante del Robot Bípodo Scout

Para encontrar la cinemática inversa del pie flotante se realiza un procedimiento parecido con la única diferencia de que al encontrar la cinemática inversa de la cadera se resuelve la cinemática directa de la misma y este resultado se multiplica por las funciones que van de la cadera al pie flotante.

Primero se evalúa la ecuación 3.10 con los ángulos calculados con la cinemática inversa de la cadera, este valor (numérico) se multiplica por la ecuación 3.30 para así proceder a calcular la cinemática inversa del pie flotante con respecto a la cadera.

Después de esto se vuelve a realizar el procedimiento de las ecuaciones 3.50, 3.51 y 3.52 con el fin de linealizar las funciones y obtener la matriz Jacobiana para así calcular los ángulos que resolverán la cinemática inversa para cierto punto en el espacio. Las ecuaciones siguientes son las únicas correcciones que hay que hacer para resolver la cinemática inversa del pie flotante.

$$\begin{bmatrix} f13 & f16 & f19 & f22 \\ f14 & f17 & f20 & f23 \\ f15 & f18 & f21 & f24 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.56)$$

$$f(\bar{x}) = \begin{bmatrix} f13 \\ f14 \\ f15 \\ f16 \\ f17 \\ f18 \\ f19 \\ f20 \\ f21 \\ f22 \\ f23 \\ f24 \end{bmatrix} \quad f(x) = \begin{bmatrix} n_x \\ n_y \\ n_z \\ s_x \\ s_y \\ s_z \\ a_x \\ a_y \\ a_z \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad (3.57 \text{ y } 3.58)$$



3.2 Simulación del Robot Bípedo Scout

Cuando se cuenta ya con los ángulos que resuelven la cinemática inversa de cierto punto en el espacio de la cadera con respecto al pie de apoyo y del pie de apoyo con respecto a la cadera se procede a realizar una simulación por computadora utilizando el software de Matlab. Todo el procedimiento anterior se programa en Matlab, este programa se encuentra en el anexo B.

Una de las complicaciones de programar el procedimiento es la matriz Jacobiana que puede ser calculada analíticamente diferenciando las funciones, sin embargo, este proceso solo es eficiente cuando se tienen pocos grados de libertad. En este caso, una mejor opción es encontrar la Jacobiana numéricamente, mediante incrementos finitos. Recordando la definición de la derivada de una variable:

$$\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (3.59)$$

De aquí que:

$$\frac{\Delta f(x)}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (3.60)$$

Si en la ecuación 3.60 Δx es lo suficientemente pequeño, esta ecuación se aproxima mucho a la ecuación 3.59, es decir, la ecuación 3.60 es el cálculo numérico de la derivada de la función $f(x)$. Este mismo principio puede ser utilizado en nuestro caso tal como muestra la ecuación 3.61.

$$J = \begin{bmatrix} \frac{f_1(\Delta\theta_1, \dots, \theta_6) - f_1(\theta_1, \dots, \theta_6)}{\Delta\theta_1} & \dots & \frac{f_1(\theta_1, \dots, \Delta\theta_6) - f_1(\theta_1, \dots, \theta_6)}{\Delta\theta_6} \\ \vdots & \ddots & \vdots \\ \frac{f_{12}(\Delta\theta_1, \dots, \theta_6) - f_{12}(\theta_1, \dots, \theta_6)}{\Delta\theta_1} & \dots & \frac{f_{12}(\theta_1, \dots, \Delta\theta_6) - f_{12}(\theta_1, \dots, \theta_6)}{\Delta\theta_6} \end{bmatrix} \quad (3.61)$$

Se elige un incremento de 0.0001 para cada articulación ($\Delta\theta_1, \Delta\theta_2, \Delta\theta_3, \Delta\theta_4, \Delta\theta_5, \Delta\theta_6$) gracias a lo cual se obtienen resultados bastante satisfactorios.



3.2.1 Trayectorias Propuestas

Con el fin de probar la simulación por computadora se proponen trayectorias elípticas para el pie flotante y para la cadera, estas trayectorias no son las definitivas sin embargo servirán para comprobar si el procedimiento funciona. Las elipses elegidas como trayectorias para la cadera y el pie flotante del robot bípedo Scout son las mismas y se ilustran mejor en la figura 3.2.

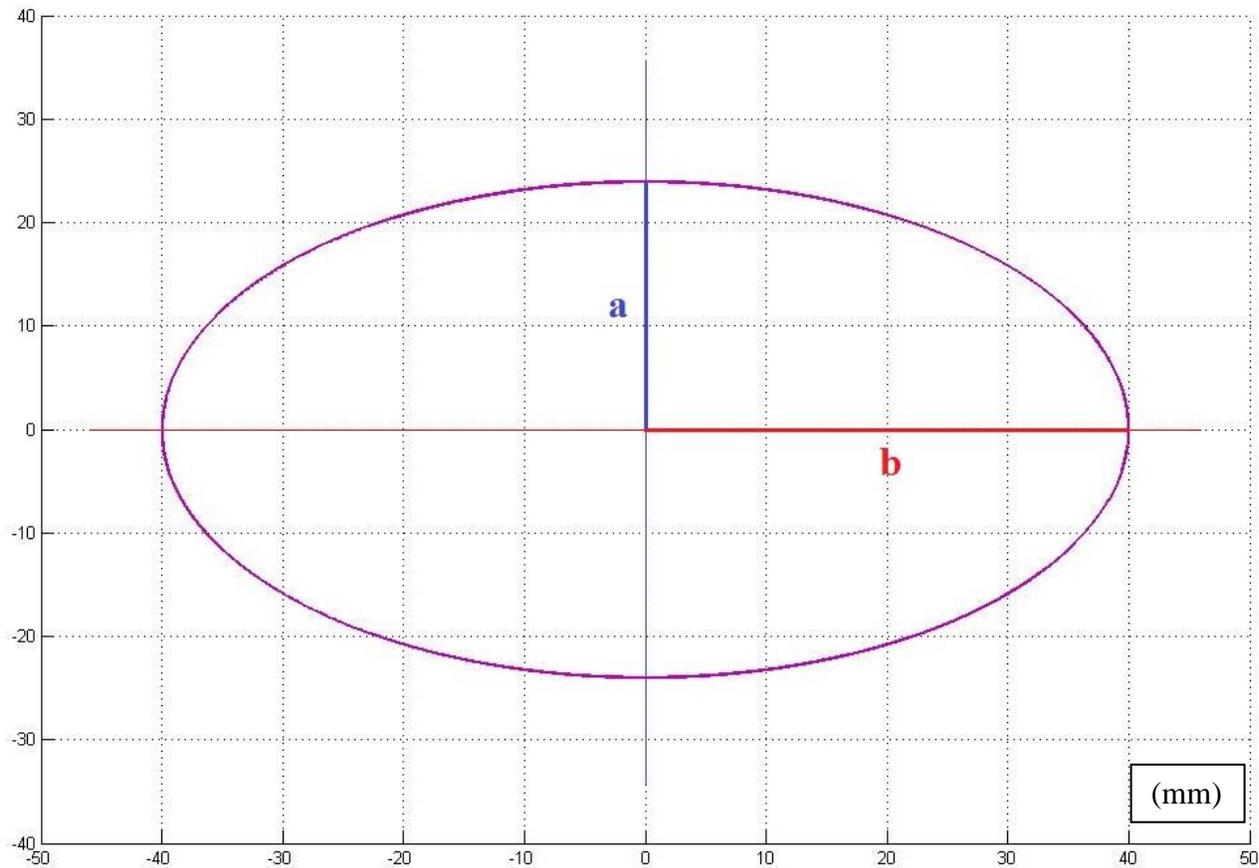


Figura 3.2. Trayectoria elíptica para la cadera y el pie flotante.

Una de las restricciones que ayudan a que el cálculo sea más rápido es el hecho de que el efector final del bípedo, ya sea la cadera o el pie flotante, no altere su orientación inicial sino que durante toda la trayectoria solo cambie la posición de dicho efector final, estas trayectorias elípticas propuestas se dividieron en 72 puntos por la misma razón y se muestran dibujadas en Matlab en la figura 3.3.

Primero se resuelve la cinemática inversa de la cadera y después la cinemática inversa del pie flotante para cada punto de las trayectorias elípticas (figura 3.3).

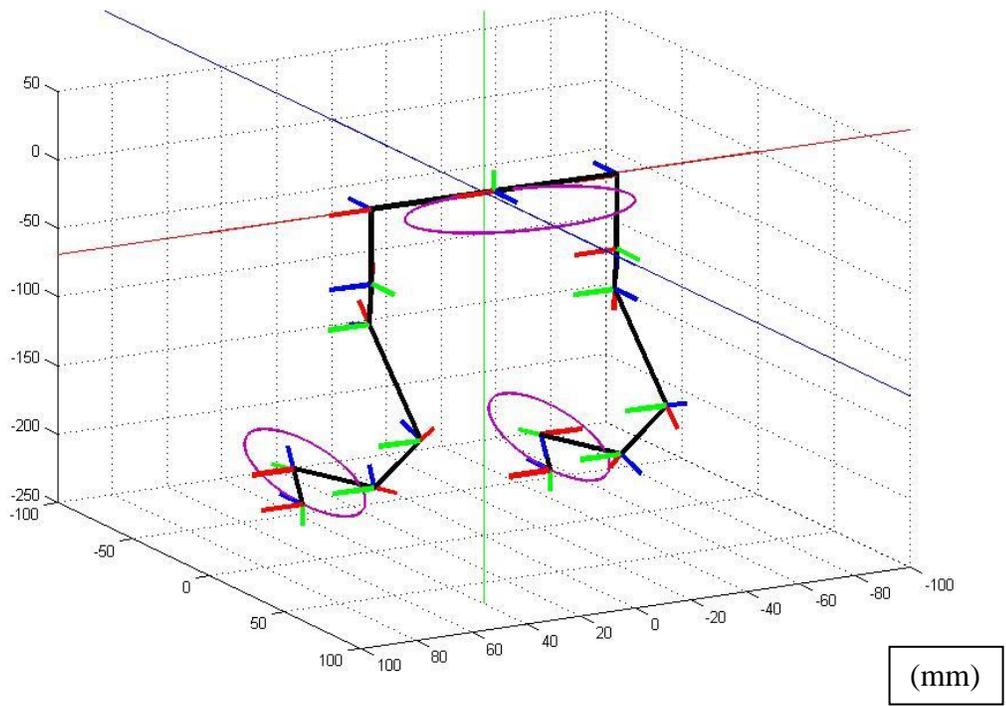


Figura 3.3. El robot bípedo siguiendo una trayectoria elíptica para la cadera y el pie flotante.

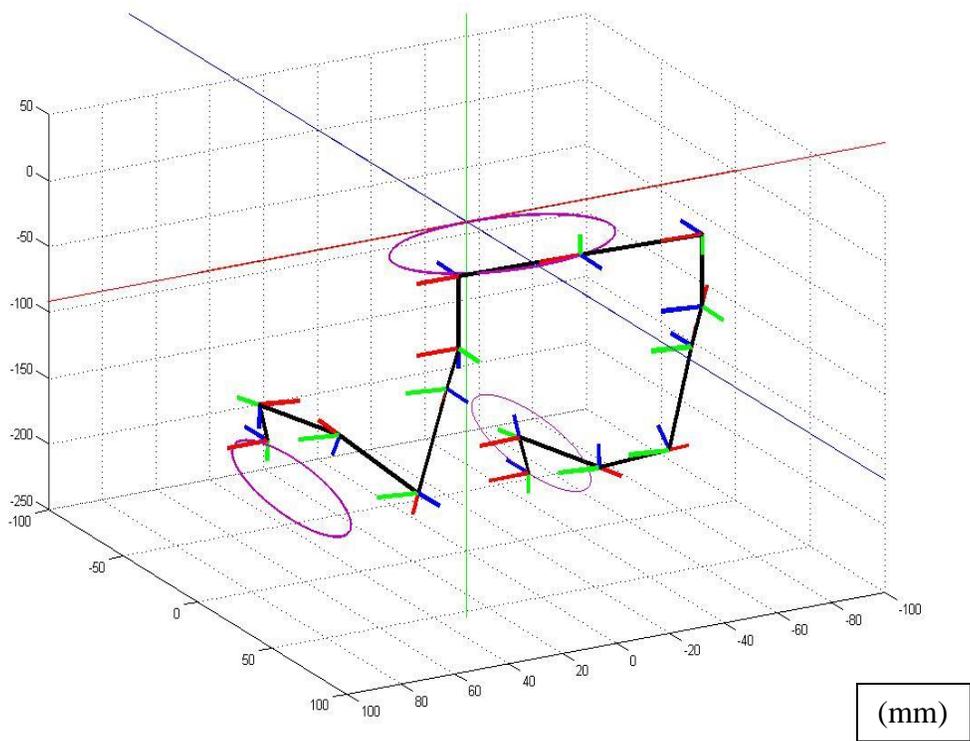


Figura 3.4. El robot bípedo siguiendo una trayectoria elíptica para la cadera y el pie derecho flotante.

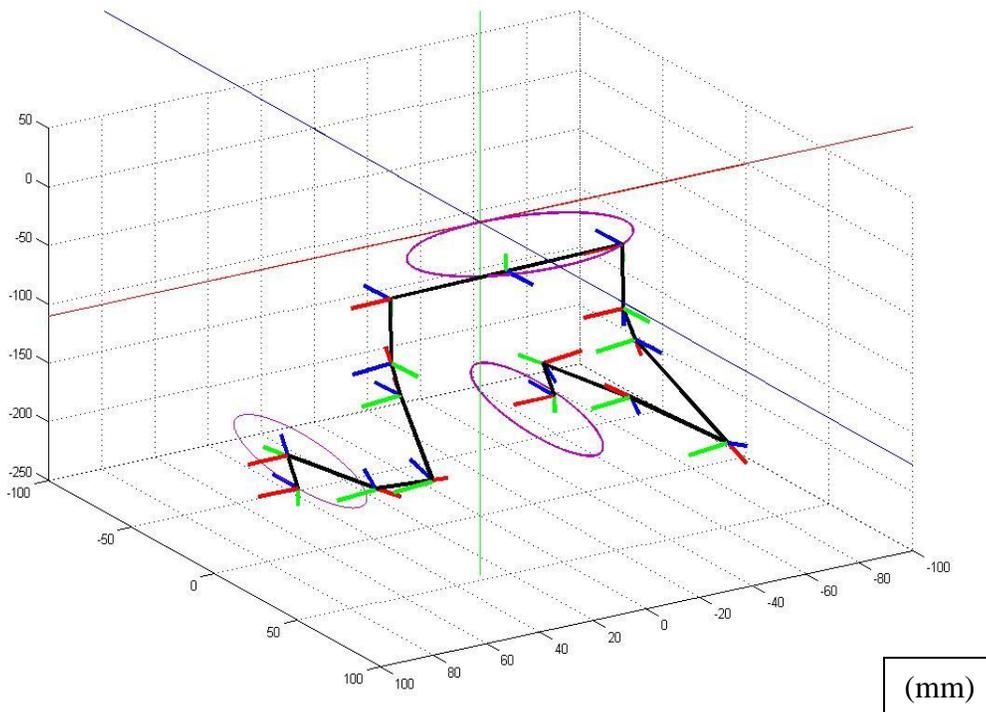


Figura 3.5. El robot bípedo siguiendo una trayectoria elíptica para la cadera y el pie izquierdo flotante.

Un punto importante es lograr que el pie de apoyo también sea el pie flotante y el pie flotante sea el pie de apoyo, esto se resuelve cambiando un dato del planteamiento de la cinemática inversa, este dato es necesario para ubicar el pie de apoyo en otro punto que no sea el sistema global de Matlab. Si se observa la matriz “A” (ecuación 3.8) es evidente que al cambiarse el parámetro de posición en el eje “x” por el negativo del mismo (ecuación 3.62) toda la cinemática se voltea haciendo que el pie de apoyo sea el pie flotante y el pie flotante sea el pie de apoyo.

$$A = \begin{bmatrix} 1 & 0 & 0 & -df1 \\ 0 & -1 & 0 & df2 \\ 0 & 0 & -1 & df3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.62)$$

La simulación consiste en lo siguiente:

- Cuando el pie izquierdo es el pie de apoyo, la cadera y el pie derecho siguen una trayectoria elíptica (Figura 3.4)
- Cuando el pie derecho es el pie de apoyo, la cadera y el pie izquierdo siguen una trayectoria elíptica (Figura 3.5)



Cada una de las trayectorias se divide solo 72 veces con el fin de hacer la simulación más rápida ya que de ser más divisiones esta se tardaría demasiado tiempo. Se eligen 72 divisiones tomando como parámetro un quinto de una circunferencia (Ecuación 3.63).

$$\text{Divisiones} = \frac{360}{5} = 72 \quad (3.63)$$

Las dimensiones de las elipses son mostradas en las ecuaciones 3.64 y 3.65 que van de acuerdo con la figura 3.2, estas trayectorias hacen que la solución de la cinemática inversa converja de manera rápida sin que tienda a la divergencia.

$$a = 40 \text{ mm} \quad (3.64)$$

$$b = 24 \text{ mm} \quad (3.65)$$

Para la simulación descrita la solución de la cinemática inversa arroja los siguientes resultados para cada una de las articulaciones:

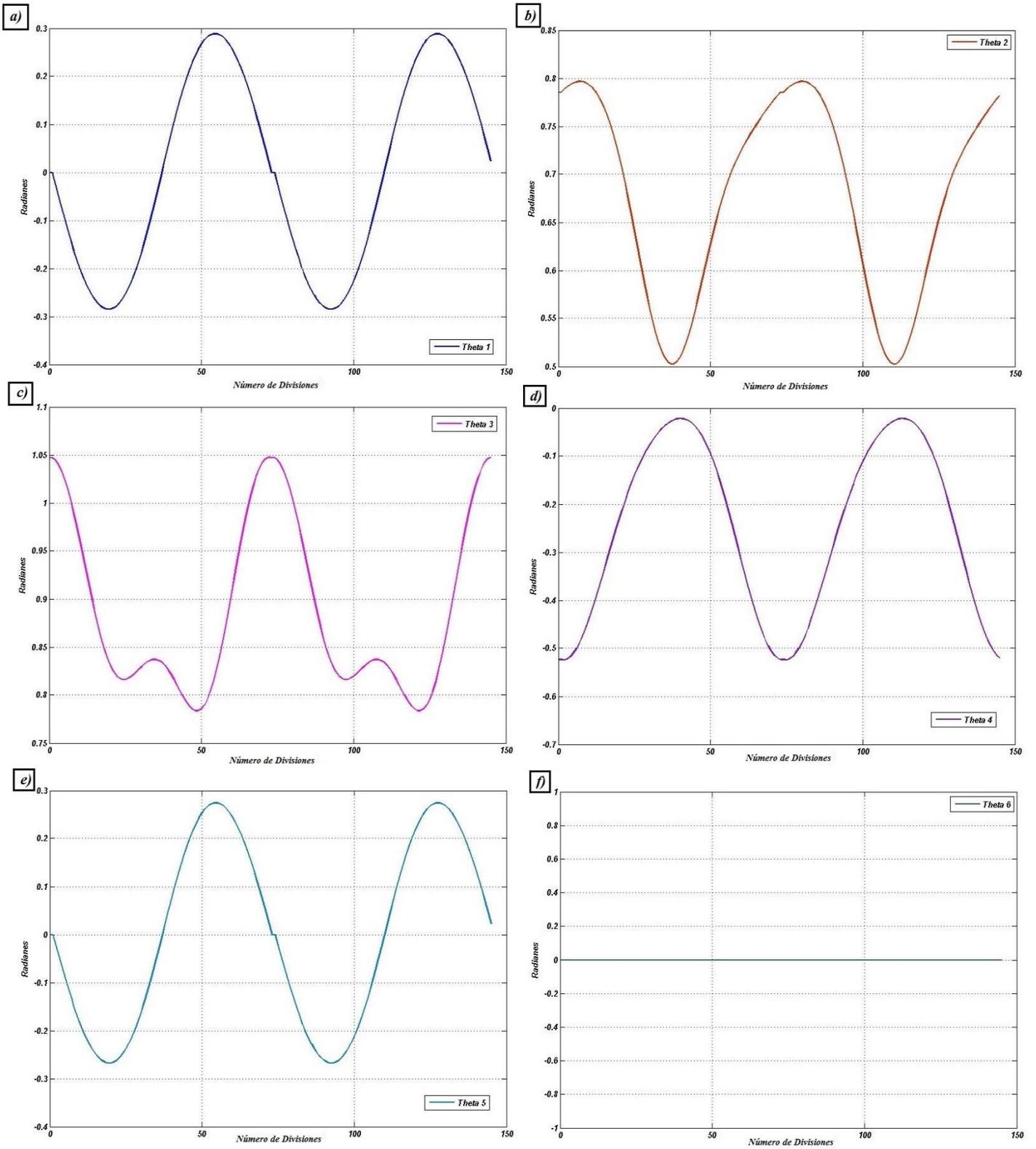


Figura 3.6. Ángulos de las articulaciones del pie izquierdo durante la trayectoria

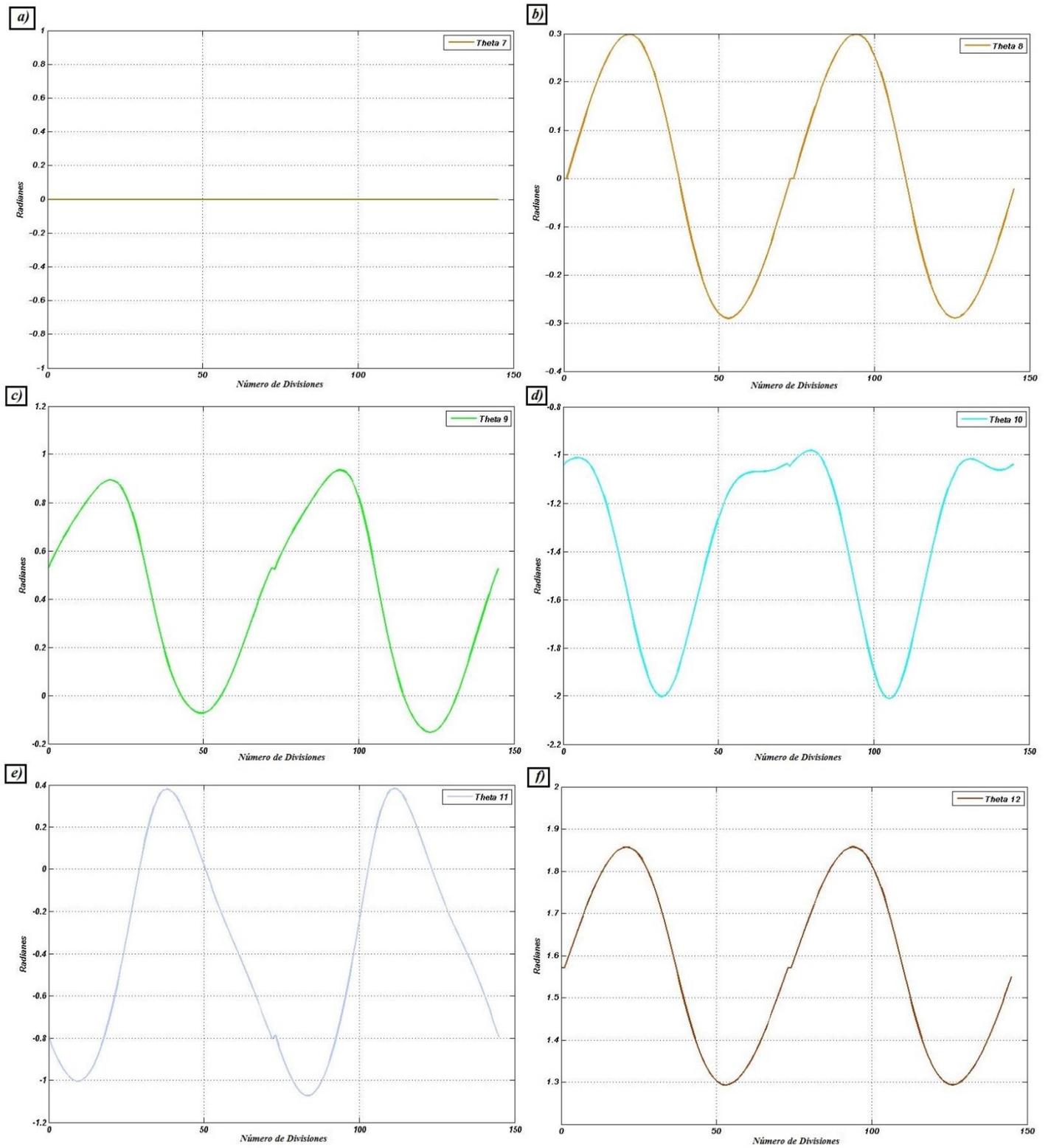


Figura 3.7. Ángulos de las articulaciones del pie derecho durante la trayectoria



Como se puede observar en las figuras 3.6 f) y 3.7 a) las articulaciones no se mueven debido a que no son necesarias pues estas articulaciones se usan para hacer que el bípedo cambie de dirección mientras camina, una de las limitaciones que se presentan es que el bípedo camine en primera instancia en línea recta. También se observa que los resultados obtenidos están en radianes, esto con el fin de en el siguiente capítulo se puedan obtener las velocidades y aceleraciones angulares que son necesarias para la formulación de la dinámica, además en las gráficas se aprecia que las coordenadas en "x" no son de tiempo si no que son en divisiones de la elipse por consiguiente se debe de asignar un tiempo promedio en el que el bípedo pase de una posición a otra para así obtener las velocidades y aceleraciones angulares y lineales.

Es importante destacar que la mayoría de las gráficas resultantes no inician en cero y esto es gracias a que los parámetros de Denavit-Hartenberg de las articulaciones están orientados con respecto a la base y nuestro sistema tiene ángulos diferentes de cero en su pose original. Por ultimo también se aprecia que los movimientos de las articulaciones son de máximo 1 radian lo cual concuerda con la simulación por que en ningún momento se da un ángulo de más de 90 grados.

La simulación por computadora del bípedo siguiendo las trayectorias elípticas es una gran ayuda visual para observar que la formulación de la cinemática directa y la solución de la cinemática inversa son correctas y muestran que el método no diverge en ningún punto establecido.



4

Análisis Dinámico

En este capítulo se planteará el análisis dinámico (Newton-Euler) así como el del péndulo invertido y las ecuaciones dinámicas para el cálculo del ZMP.



4.1 Introducción

El objetivo principal del presente capítulo es determinar la formulación necesaria que nos pueda indicar el comportamiento dinámico del robot a partir de unos cuantos parámetros. Para ello realizamos el estudio de la formulación de Newton-Euler con el fin de poder conocer los torques necesarios de los motores para poder mover el Bípido Scout, de esta forma también se realizará el estudio del ZMP, que es un indicador de comportamiento de la estabilidad del robot bípido, se planteará su definición, se realizará la exposición explícita de sus propiedades y se mencionarán las ventajas que se tienen utilizándolo como criterio de estabilidad.

Posteriormente se desarrollará el análisis dinámico del robot bípido, utilizando para este objetivo el modelo simplificado del péndulo invertido, donde es considerada la masa total del robot, concentrada en un punto en su centro de gravedad (COG).

4.2 Dinámica inversa (Formulación de Newton-Euler)

El método de Newton-Euler permite obtener un conjunto de ecuaciones recursivas hacia delante de velocidad y aceleración lineal y angular, las cuales están referidas a cada sistema de referencia articular. Las velocidades y aceleraciones de cada elemento se propagan hacia adelante desde el sistema de referencia de la base hasta el efector final. Las ecuaciones recursivas hacia atrás calculan los pares y fuerzas necesarias para cada articulación desde el efector final hasta el sistema de referencia de la base.

4.2.1. Sistema de coordenadas en movimiento

La formulación de N-E se basa en los sistemas de coordenadas en movimiento.

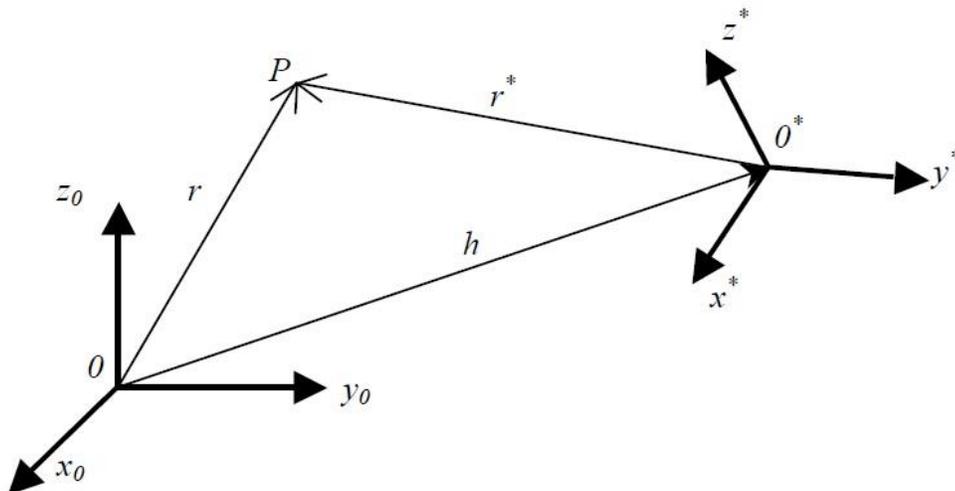


Figura 4.1. Sistema de coordenadas en movimiento



Con respecto a la figura 4.1 el sistema de coordenadas O^* se desplaza y gira en el espacio respecto del sistema de referencia de la base O , el vector que describe el origen del sistema de movimiento es h y el punto P se describe respecto del sistema O^* a través del vector r^* , de acuerdo con esto, la descripción del punto P respecto del sistema de la base es:

$$\mathbf{r} = \mathbf{r}^* + \mathbf{h} \quad (4.1)$$

$$\frac{d\mathbf{r}}{dt} = \frac{d\mathbf{r}^*}{dt} + \frac{d\mathbf{h}}{dt} = \mathbf{v}^* + \mathbf{v}_h \quad (4.2)$$

Donde \mathbf{v}^* es la velocidad del punto P respecto del origen del sistema O^* en movimiento y \mathbf{v}_h es la velocidad del origen del sistema O^* respecto a la base.

Si el P se desplaza y gira respecto al sistema O^* la ecuación 4.2 debe escribirse como:

$$\mathbf{v} = \frac{d\mathbf{r}^*}{dt} + \frac{d\mathbf{h}}{dt} = \frac{d^*\mathbf{r}^*}{dt} + \boldsymbol{\omega} \times \mathbf{r}^* + \frac{d\mathbf{h}}{dt} \quad (4.3)$$

Donde $\frac{d^*\mathbf{r}^*}{dt}$ es la velocidad lineal del punto P respecto del origen O^* y $\boldsymbol{\omega} \times \mathbf{r}^*$ es la velocidad angular del punto P respecto del origen O^* .

De manera similar la aceleración general del sistema se puede describir como:

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d^2\mathbf{r}^*}{dt^2} + \frac{d^2\mathbf{h}}{dt^2} = \mathbf{a}^* + \mathbf{a}_h \quad (4.4)$$

$$\mathbf{a} = \frac{d^{*2}\mathbf{r}^*}{dt^2} + 2\boldsymbol{\omega} \times \frac{d^*\mathbf{r}^*}{dt} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}^*) + \frac{d\boldsymbol{\omega}}{dt} \times \mathbf{r}^* + \frac{d^2\mathbf{h}}{dt^2} \quad (4.5)$$

Con esta introducción a los sistemas de coordenadas móviles, podremos aplicar este concepto a los sistemas de coordenadas de los elementos que establecimos para un robot con el fin de obtener la información cinemática de los elementos.

4.2.2. Cinemática de los elementos

El objetivo de esta sección es deducir un conjunto de ecuaciones matemáticas que, basadas en los sistemas de coordenadas en movimiento descrito en la sección anterior, describen las relaciones cinemáticas de los elementos en movimiento de un robot con respecto al sistema de coordenadas de la base.



A partir de las ecuaciones (4.1) a (4.5) de la sección anterior, se desarrolla a continuación el planteamiento general para la cinemática de los eslabones del robot.

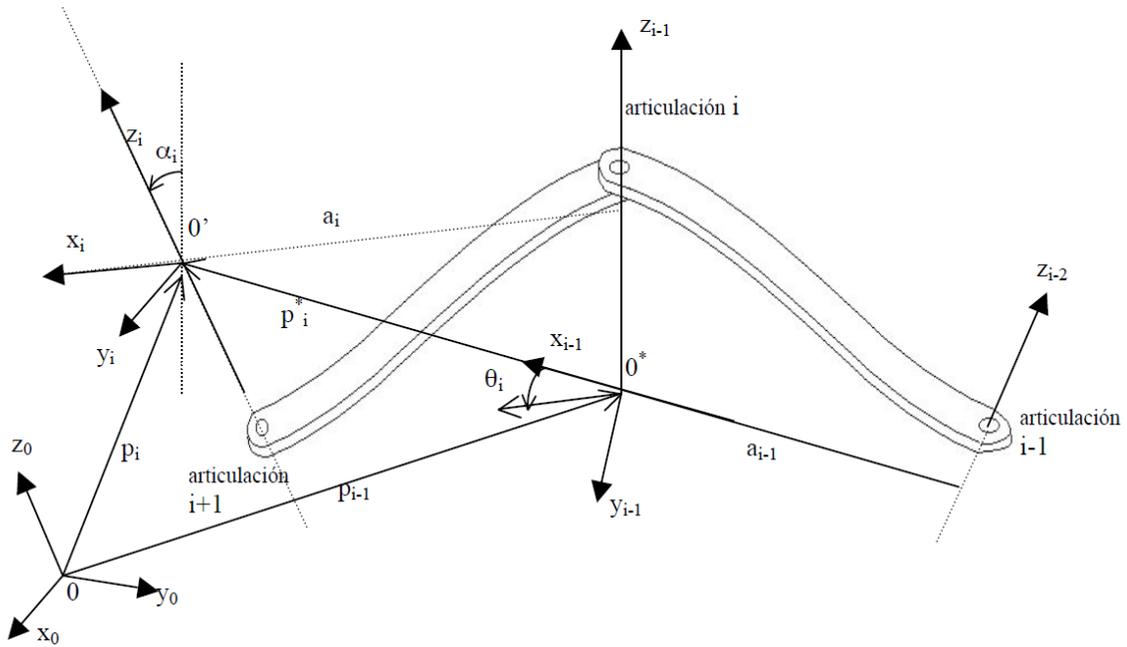


Figura 4.2. Relación entre los sistemas O, O^* y O'

De acuerdo con la figura 4.2 las ecuaciones cinemáticas para los eslabones de un robot, se pueden escribir como:

$$\begin{aligned} \mathbf{v}_i &= \frac{d^* \mathbf{P}_i^*}{dt} + \boldsymbol{\omega}_{i-1} \times \mathbf{P}_i^* + \mathbf{v}_{i-1} \\ \boldsymbol{\omega}_i &= \boldsymbol{\omega}_{i-1} + \boldsymbol{\omega}_i^* \end{aligned} \quad (4.6)$$

Debe notarse que la velocidad angular del sistema de referencia $\boldsymbol{\omega}_i$ es igual a la suma de la velocidad angular absoluta del sistema $i - 1$ más la velocidad angular relativa $\boldsymbol{\omega}_i^*$ del eslabón referida a su propio sistema de coordenadas.

La aceleración lineal del sistema de coordenadas de la articulación i es:

$$\mathbf{a}_i = \frac{d^{*2} \mathbf{P}_i^*}{dt^2} + \dot{\boldsymbol{\omega}}_{i-1} \times \mathbf{P}_i^* + 2\boldsymbol{\omega}_{i-1} \times \frac{d^* \mathbf{P}_i^*}{dt} + \boldsymbol{\omega}_{i-1} \times (\boldsymbol{\omega}_{i-1} \times \mathbf{P}_i^*) + \dot{\mathbf{v}}_{i-1} \quad (4.7)$$



$$\dot{\omega}_i = \dot{\omega}_{i-1} + \dot{\omega}_i^* \quad (D.8)$$

La aceleración angular del sistema de referencia i (x_i, y_i, z_i) respecto al sistema ($x_{i-1}, y_{i-1}, z_{i-1}$) se consigue de manera similar a la ecuación (4.3).

$$\dot{\omega}_i^* = \frac{d\omega_i^*}{dt} + \omega_{i-1} \times \omega_i^* \quad (4.9)$$

Por lo que la ecuación (4.8) queda como:

$$\dot{\omega}_i = \dot{\omega}_{i-1} + \frac{d\omega_i^*}{dt} + \omega_{i-1} \times \omega_i^* \quad (4.10)$$

En general para un robot los sistemas de coordenadas ($x_{i-1}, y_{i-1}, z_{i-1}$) y (x_i, y_i, z_i) están unidos a los eslabones $i - 1$ e i . La velocidad del eslabón i respecto del sistema de coordenadas $i - 1$ es \dot{q}_i . Si el eslabón es prismático, la velocidad será una velocidad de traslación relativa respecto del sistema ($x_{i-1}, y_{i-1}, z_{i-1}$) y si es rotacional le corresponderá una velocidad rotacional relativa de eslabón i respecto del sistema ($x_{i-1}, y_{i-1}, z_{i-1}$), en el caso del robot bípedo Scout todos sus eslabones son rotacionales por tanto las ecuaciones quedan de la siguiente forma:

$$\omega_i^* = Z_{i-1} \dot{q}_i \quad (4.11)$$

Donde \dot{q}_i es la magnitud de la velocidad angular del eslabón i con respecto al sistema de coordenadas ($x_{i-1}, y_{i-1}, z_{i-1}$). De manera similar:

$$\frac{d^* \omega_i^*}{dt} = Z_{i-1} \ddot{q}_i \quad (4.12)$$

Debe notarse que el vector de Z_{i-1} es igual a $(0, 0, 1)^T$.

Las velocidades y aceleraciones de los sistemas de coordenadas ligados a cada eslabón son absolutas y se calculan como:

$$\omega_i = \omega_{i-1} + Z_{i-1} \dot{q}_i \quad (4.13)$$

$$\dot{\omega}_i = \dot{\omega}_{i-1} + Z_{i-1} \ddot{q}_i + \omega_{i-1} \times (Z_{i-1} \dot{q}_i) \quad (4.14)$$



Las velocidades lineales de los sistemas de referencia de cada eslabón se calculan como:

$$\frac{d^*P_i}{dt} = \omega_i \times P_i^* \quad (4.15)$$

$$\frac{d^{*2}P_i^*}{dt^2} = \frac{d^*\omega_i^*}{dt} \times P_i^* + \omega_i^* \times (\omega_i^* \times P_i^*) \quad (4.16)$$

Por lo cual la velocidad lineal absoluta del sistema de coordenadas ligado a cada eslabón se calcula como:

$$v_i = \omega_i \times P_i^* + v_{i-1} \quad (4.17)$$

$$\dot{v}_i = \dot{\omega}_i \times P_i^* + \omega_i \times (\omega_i \times P_i^*) + \dot{v}_{i-1} \quad (4.18)$$

4.2.3. Ecuaciones de movimiento recursivas

A partir de las ecuaciones cinemáticas del apartado anterior y aplicando el principio de D'Alembert del equilibrio estático para todos los instantes de tiempo, se obtienen las ecuaciones recursivas de Newton-Euler.

Se utiliza la nomenclatura de la figura 4.2 sobre un eslabón cualquiera del robot, tal y como se muestra en la figura 4.3.

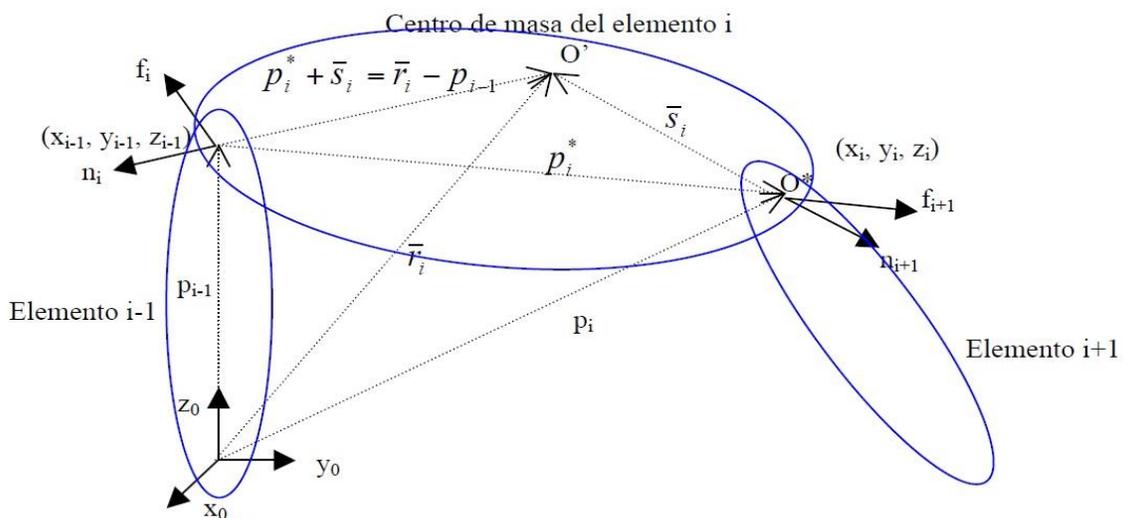


Figura 4.3. Fuerzas y momentos sobre el elemento i



Dónde:

m_i	Masa total del eslabón i .
$\bar{\mathbf{r}}_i$	Posición del centro de masa del elemento i desde el origen del sistema de referencia de la base.
$\bar{\mathbf{s}}_i$	Posición del centro de masa del elemento i desde el origen del Sistema de coordenadas $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$.
\mathbf{P}_i^*	Posición del origen de coordenadas $i - \text{ésimo}$ con respecto al sistema de coordenadas $(i - 1) - \text{ésimo}$.
$\bar{\mathbf{v}}_i = \frac{d\bar{\mathbf{r}}_i}{dt}$	Velocidad lineal del centro de masa del elemento i .
$\bar{\mathbf{a}}_i = \frac{d\bar{\mathbf{v}}_i}{dt}$	Aceleración lineal del centro de masa del elemento i .
\mathbf{F}_i	Fuerza total externa ejercida sobre el elemento i en el centro de masa.
\mathbf{N}_i	Momento total externo ejercido sobre el elemento i en el centro de masa.
\mathbf{I}_i	Matriz de inercia del elemento i respecto de su centro de masa con respecto al sistema de coordenadas $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$.
\mathbf{f}_i	Fuerza ejercida sobre el elemento i por el elemento $i - 1$ en el sistema de coordenadas $(\mathbf{x}_{i-1}, \mathbf{y}_{i-1}, \mathbf{z}_{i-1})$ para soportar al elemento i y a los elementos por encima de él.
\mathbf{n}_i	Momento ejercido sobre el elemento i por el elemento $i - 1$ en el sistema de coordenadas $(\mathbf{x}_{i-1}, \mathbf{y}_{i-1}, \mathbf{z}_{i-1})$.

Si se omiten los efectos del rozamiento viscoso en las articulaciones, y se aplica el principio de D'Alembert, se obtiene para cada eslabón:

$$\mathbf{F}_i = \frac{d(m_i \bar{\mathbf{v}}_i)}{dt} = m_i \bar{\mathbf{a}}_i \quad (4.19)$$

$$\mathbf{N}_i = \frac{d(\mathbf{I}_i \boldsymbol{\omega}_i)}{dt} = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i) \quad (4.20)$$



Realizando el balance de pares y fuerzas en la figura D.3:

$$\mathbf{F}_i = \mathbf{f}_i - \mathbf{f}_{i+1} \quad (4.21)$$

$$\mathbf{N}_i = \mathbf{n}_i - \mathbf{n}_{i+1} + (\mathbf{P}_{i-1} - \bar{\mathbf{r}}_i) \times \mathbf{f}_i - (\mathbf{P}_i - \bar{\mathbf{r}}_i) \times \mathbf{f}_{i+1} \quad (4.21)$$

$$= \mathbf{n}_i - \mathbf{n}_{i+1} + (\mathbf{P}_{i-1} - \bar{\mathbf{r}}_i) \times \mathbf{F}_i - \mathbf{P}_i^* \times \mathbf{f}_{i+1} \quad (4.22)$$

Que utilizando la relación geométrica:

$$\bar{\mathbf{r}}_i - \mathbf{P}_{i-1} = \mathbf{P}_i^* + \bar{\mathbf{s}}_i \quad (4.23)$$

Se obtienen las ecuaciones recursivas:

$$\mathbf{f}_i = \mathbf{F}_i + \mathbf{f}_{i+1} = m_i \bar{\mathbf{a}}_i + \mathbf{f}_{i+1} \quad (4.24)$$

$$\mathbf{n}_i = \mathbf{n}_{i+1} + \mathbf{P}_i^* \times \mathbf{f}_{i+1} + (\mathbf{P}_i^* - \bar{\mathbf{s}}_i) \times \mathbf{F}_i + \mathbf{N}_i \quad (4.25)$$

Se observa que estas ecuaciones son recursivas y permiten obtener las fuerzas y momentos en los elementos $i = 1, 2, \dots, n$ para un robot de n elementos. \mathbf{f}_{i+1} y \mathbf{n}_{i+1} representan la fuerza y momento ejercidos por el efector final del robot sobre un objeto externo. Por lo tanto el par de fuerza para cada articulación se expresa como:

$$\boldsymbol{\tau}_i = \mathbf{n}_i^T \mathbf{Z}_{i-1} + b_i \dot{\mathbf{q}}_i \quad (4.26)$$

Donde b_i es el coeficiente de amortiguamiento viscoso de la articulación.



4.2.4. Algoritmo Computacional

En resumen de lo anteriormente expuesto las ecuaciones de N-E consisten en un conjunto de ecuaciones recursivas hacia adelante y hacia atrás. Para obtener un algoritmo computacional se debe tener en cuenta que en las anteriores ecuaciones, las matrices de inercia I_i y los parámetros del robot $\bar{\mathbf{r}}_i, \bar{\mathbf{s}}_i, \mathbf{P}_i^*$, están referenciados respecto al sistema de coordenadas de la base.

Sea ${}^{i-1}\mathbf{R}_i$ una matriz de rotación 3×3 que transforma cualquier vector con referencia al sistema de coordenadas $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ al sistema de coordenadas $(\mathbf{x}_{i-1}, \mathbf{y}_{i-1}, \mathbf{z}_{i-1})$. Esto es la submatriz superior izquierda 3×3 de ${}^{i-1}\mathbf{A}_i$.

Ha sido demostrado que:

$$({}^{i-1}\mathbf{R}_i)^{-1} = {}^i\mathbf{R}_{i-1} = ({}^{i-1}\mathbf{R}_i)^T \quad (4.27)$$

Dónde:

$${}^{i-1}\mathbf{R}_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix}$$

$$({}^{i-1}\mathbf{R}_i)^{-1} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & 0 \\ -\cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & \sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) \end{bmatrix} \quad (4.29)$$

En lugar de calcular $\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i, \mathbf{v}_i, \bar{\mathbf{a}}_i, \mathbf{P}_i^*, \bar{\mathbf{s}}_i, \mathbf{F}_i, \mathbf{N}_i, \mathbf{f}_i, \mathbf{n}_i$ y $\boldsymbol{\tau}_i$ que se refieren al sistema de coordenadas de la base, calculamos ${}^i\mathbf{R}_0\boldsymbol{\omega}_i, {}^i\mathbf{R}_0\dot{\boldsymbol{\omega}}_i, {}^i\mathbf{R}_0\mathbf{v}_i, {}^i\mathbf{R}_0\bar{\mathbf{a}}_i, {}^i\mathbf{R}_0\mathbf{P}_i^*, {}^i\mathbf{R}_0\bar{\mathbf{s}}_i, {}^i\mathbf{R}_0\mathbf{F}_i, {}^i\mathbf{R}_0\mathbf{N}_i, {}^i\mathbf{R}_0\mathbf{f}_i, {}^i\mathbf{R}_0\mathbf{n}_i$ y ${}^i\mathbf{R}_0\boldsymbol{\tau}_i$ que se refieren a su propio sistema de coordenadas del elemento $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$. Por tanto las ecuaciones recursivas de N-E quedan expresadas en la tabla 4.1.



Tabla 4.1. Ecuaciones de movimiento de Newton-Euler recursivas eficientes

Ecuaciones hacia adelante: $i = 1, 2, \dots, n$

$${}^i\mathbf{R}_0\boldsymbol{\omega}_i = \begin{cases} {}^i\mathbf{R}_{i-1}({}^{i-1}\mathbf{R}_0\boldsymbol{\omega}_{i-1} + \mathbf{z}_0\dot{q}_i) & \text{si el elemento } i \text{ es rotacional} \\ {}^i\mathbf{R}_{i-1}({}^{i-1}\mathbf{R}_0\boldsymbol{\omega}_{i-1}) & \text{si el elemento } i \text{ es traslacional} \end{cases}$$

$${}^i\mathbf{R}_0\dot{\boldsymbol{\omega}}_i = \begin{cases} {}^i\mathbf{R}_{i-1}[{}^{i-1}\mathbf{R}_0\dot{\boldsymbol{\omega}}_{i-1} + \mathbf{z}_0\ddot{q}_i + ({}^{i-1}\mathbf{R}_0\boldsymbol{\omega}_{i-1}) \times \mathbf{z}_0\dot{q}_i] & \text{si el elemento } i \\ \text{es rotacional} \\ {}^i\mathbf{R}_{i-1}({}^{i-1}\mathbf{R}_0\dot{\boldsymbol{\omega}}_{i-1}) & \text{si el elemento } i \\ \text{es traslacional} \end{cases}$$

$${}^i\mathbf{R}_0\dot{\mathbf{v}}_i = \begin{cases} ({}^i\mathbf{R}_0\dot{\boldsymbol{\omega}}_i) \times ({}^i\mathbf{R}_0\mathbf{p}_i^*) + ({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times [({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times ({}^i\mathbf{R}_0\mathbf{p}_i^*)] + \\ + {}^i\mathbf{R}_{i-1}({}^{i-1}\mathbf{R}_0\dot{\mathbf{v}}_{i-1}) & \text{si el elemento } i \text{ es rotacional} \\ {}^i\mathbf{R}_{i-1}(\mathbf{z}_0\ddot{q}_i + {}^{i-1}\mathbf{R}_0\dot{\mathbf{v}}_{i-1}) + ({}^i\mathbf{R}_0\dot{\boldsymbol{\omega}}_i) \times ({}^i\mathbf{R}_0\mathbf{p}_i^*) + \\ + 2({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times ({}^i\mathbf{R}_{i-1}\mathbf{z}_0\dot{q}_i) + \\ + ({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times [({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times ({}^i\mathbf{R}_0\mathbf{p}_i^*)] & \text{si el elemento } i \text{ es traslacional} \end{cases}$$

$${}^i\mathbf{R}_0\bar{\mathbf{a}}_i = ({}^i\mathbf{R}_0\dot{\boldsymbol{\omega}}_i) \times ({}^i\mathbf{R}_0\bar{\mathbf{s}}_i) + ({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times [({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times ({}^i\mathbf{R}_0\bar{\mathbf{s}}_i)] + {}^i\mathbf{R}_0\dot{\mathbf{v}}_i$$

Ecuaciones hacia atrás: $i = n, n - 1, \dots, 1$

$${}^i\mathbf{R}_0\mathbf{f}_i = {}^i\mathbf{R}_{i+1}({}^{i+1}\mathbf{R}_0\mathbf{f}_{i+1}) + m_i{}^i\mathbf{R}_0\bar{\mathbf{a}}_i$$

$${}^i\mathbf{R}_0\mathbf{n}_i = {}^i\mathbf{R}_{i+1}[({}^{i+1}\mathbf{R}_0\mathbf{n}_{i+1} + ({}^{i+1}\mathbf{R}_0\mathbf{p}_{i+1}^*) \times ({}^{i+1}\mathbf{R}_0\mathbf{f}_{i+1})] + ({}^i\mathbf{R}_0\mathbf{p}_i^* + {}^i\mathbf{R}_0\bar{\mathbf{s}}_i) \times ({}^i\mathbf{R}_0\mathbf{F}_i) + \\ + ({}^i\mathbf{R}_0\mathbf{I}_i{}^0\mathbf{R}_i)({}^i\mathbf{R}_0\dot{\boldsymbol{\omega}}_i) + ({}^i\mathbf{R}_0\boldsymbol{\omega}_i) \times [({}^i\mathbf{R}_0\mathbf{I}_i{}^0\mathbf{R}_i)({}^i\mathbf{R}_0\boldsymbol{\omega}_i)]$$

$$\tau_i = \begin{cases} ({}^i\mathbf{R}_0\mathbf{n}_i)^T({}^i\mathbf{R}_{i-1}\mathbf{z}_0) + b_i\dot{q}_i & \text{si el elemento } i \text{ es rotacional} \\ ({}^i\mathbf{R}_0\mathbf{f}_i)^T({}^i\mathbf{R}_{i-1}\mathbf{z}_0) + b_i\dot{q}_i & \text{si el elemento } i \text{ es traslacional} \end{cases}$$

donde $\mathbf{z}_0 = (0, 0, 1)^T$ y b_i es el coeficiente de amortiguamiento viscoso para la articulación i . Las condiciones iniciales usuales son $\boldsymbol{\omega}_0 = \dot{\boldsymbol{\omega}}_0 = \mathbf{v}_0 = \mathbf{0}$ y $\dot{\mathbf{v}}_0 = (g_x, g_y, g_z)^T$ (para incluir la gravedad), donde $|g| = 9,8062 \text{ m/s}^2$.

Estas ecuaciones fueron programadas en Matlab para realizar la función del cálculo de los torques de los 12 GDL (Véase Anexo B) para el ciclo de marcha propuesto en el capítulo 3, es obvio que los ángulos calculados por la cinemática inversa sirven de entrada a la función de torques al igual que las velocidades angulares y aceleraciones angulares, para calcular las velocidades y aceleraciones angulares basta con saber que la velocidad es el cambio de posición de un punto a otro en un tiempo determinado y la aceleración es el cambio de velocidad de un punto a otro en un tiempo determinado, siguiendo esta filosofía se pueden obtener las ecuaciones 4.30 y 4.31 de velocidad angular y aceleración angular respectivamente tomando en cuenta los valores de los ángulos obtenidos para cada una de las articulaciones del robot bípedo scout.



$$\omega_{i_n} = (\theta_{i_n} - \theta_{i_{n-1}})/T \quad (4.30)$$

$$\alpha_{i_n} = (\omega_{i_n} - \omega_{i_{n-1}})/T \quad (4.31)$$

Dónde:

n	División de la trayectoria
θ_{i_n}	Posición angular de la articulación en la división de la trayectoria.
ω_{i_n}	Velocidad angular de la articulación en la división de la trayectoria
α_{i_n}	Aceleración angular de la articulación en la división De la trayectoria
T	Tiempo

Es importante tomar en cuenta que para realizar el primer cálculo de la aceleración se requiere de una velocidad angular que no existe “ $\omega_{i_{n-1}}$ ” por lo cual se procede a asignar un valor inicial que puede tomar cualquier velocidad lógica para el robot bípedo Scout, pero para este caso de uso se propone la velocidad inicial de cero para cada articulación, lo mismo para la aceleración inicial.

$$\omega = \begin{bmatrix} 0 \text{ rad/s} \\ 0 \text{ rad/s} \end{bmatrix} \quad (4.32)$$



$$\alpha = \begin{bmatrix} 0 \text{ rad/s}^2 \\ 0 \text{ rad/s}^2 \end{bmatrix} \quad (4.33)$$

El tiempo que se toma para realizar los cálculos debe de ser el deseado para cada división de la trayectoria con esta aseveración si se toma un tiempo de 0.14 de segundo el bípido terminará la trayectoria en 10 segundos ya que son 72 divisiones de la trayectoria. Teniendo todos estos datos además de las masas de cada eslabón (Anexo A) y tomando en cuenta que la gravedad es de $9.81 \frac{m}{s^2}$ es posible calcular el torque necesario de los motores del robot bípido Scout que dan como resultado el expresado en la figuras 4.4 y 4.5.

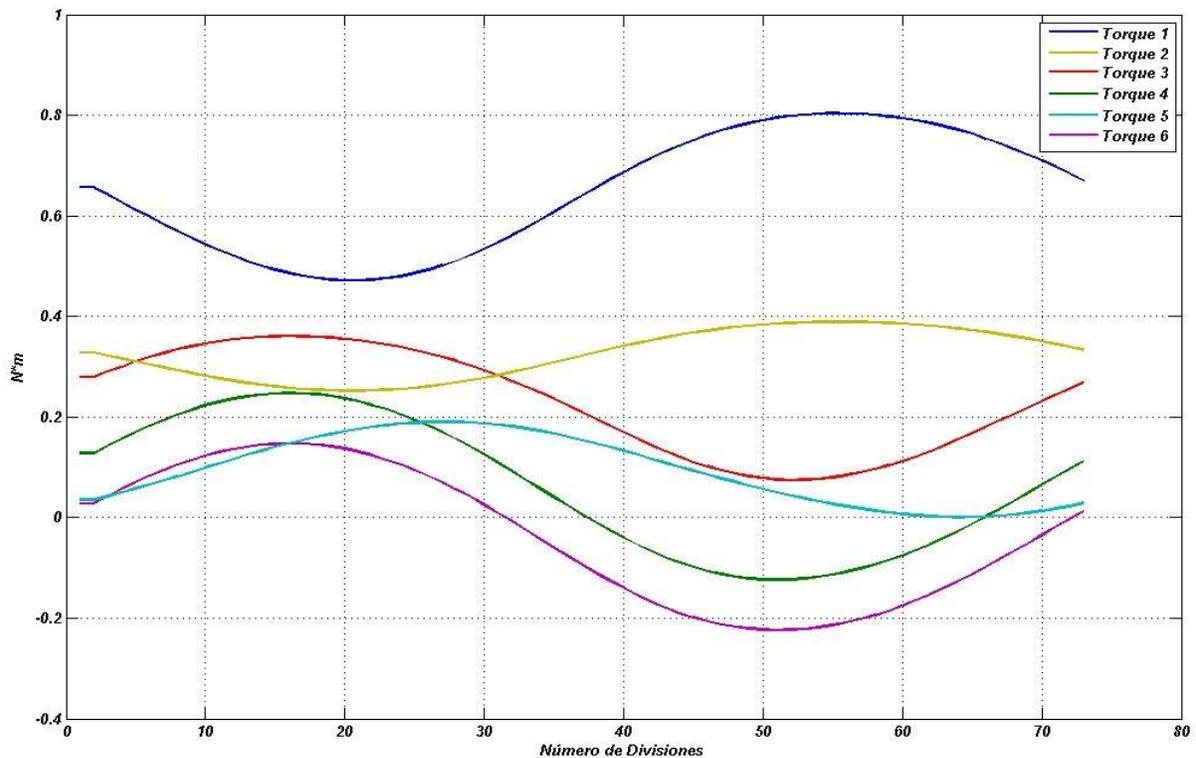


Figura 4.4. Torques calculados para los 6 GDL del pie Izquierdo

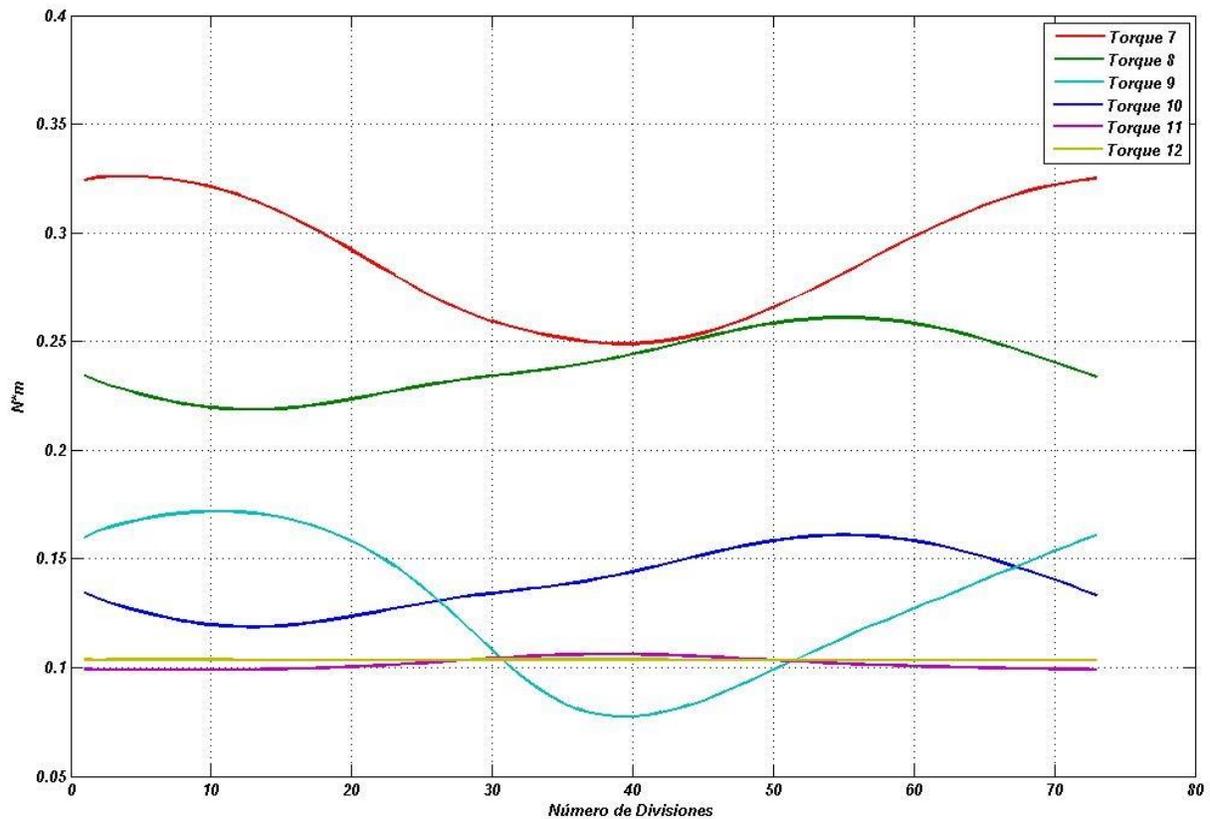


Figura 4.5. Torques calculados para los 6 GDL del pie Derecho

Tenemos un torque máximo de 0.8 Nm lo que es lógico porque el motor que más carga tiene que mover es el motor del primer grado de libertad ya que es el que en algún momento tiene que soportar toda la masa del robot scout.

También se logra apreciar la existencia de un torque negativo que matemáticamente es incoherente y a primera vista se entendería como el cambio de giro de los motores, pero esto no es así, más bien se debe a que el modelo dinámico emula la inercia del movimiento, la representación física resulta ser que en esos momentos el movimiento del robot bípedo genera fuerzas inerciales que mueven el motor sin necesidad de que este genere un torque.

Uno de los inconvenientes más grandes del cálculo de torques es que no se tiene manera de comprobar los resultados y esto es debido a que no se tiene una simulación gráfica como en la cinemática es por ello que los resultados deben ser validados por lógica y mediante pruebas físicas con el robot en cuestión.

4.2 Análisis de la Problemática de Equilibrio

La idea fundamental del control de equilibrio es evitar que el robot caiga, para esto, se dispone de un indicador de equilibrio, en este caso el ZMP.

Partamos del análisis de todas las fuerzas y momentos que actúan sobre el pie de apoyo:

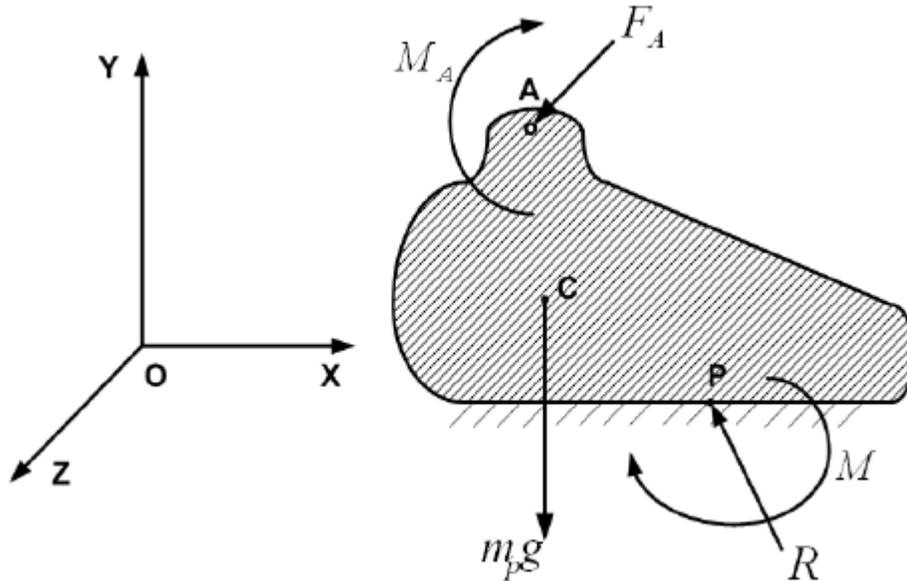


Figura 4.6. Fuerzas y momentos aplicados sobre el pie de apoyo

De acuerdo con la figura 4.6, M_A y F_A son el momento y la fuerza resultante generado por el cuerpo en movimiento, M y R son el momento y la fuerza de reacción del piso, $m_p g$ es la masa del pie. Para que el pie se encuentre en equilibrio estático:

$$\sum \vec{F} = 0 \quad (4.34)$$

$$\vec{R} + \vec{F}_A = 0 \quad (4.35)$$

$$\sum \vec{M}_o = 0 \quad (4.36)$$

$$(\vec{OP} * \vec{R}) + (\vec{OA} * \vec{F}_A) + \vec{M} + \vec{M}_A + (\vec{OC} * (m_p \vec{g})) = 0 \quad (4.37)$$



A partir de ahora se asumirá que la fricción de la superficie es lo suficientemente grande, para no permitir que exista desplazamiento entre el pie y el piso, entonces, R_x y R_z compensan las fuerzas del cuerpo en esas direcciones por lo que:

$$\sum \vec{F}_y = 0 \quad (4.38)$$

$$\vec{R}_y + \vec{F}_{Ay} = 0 \quad (4.39)$$

Se ha podido reemplazar a la fuerza de reacción neta del piso por una que actúa solo en el eje “Y”, también los momentos se han reducido al plano horizontal (ejes “X” y “Z”). La condición fundamental para que el sistema esté en equilibrio es que el punto “P”, cumpla con la ecuación 4.40.

$$M_x = M_z = 0 \quad (4.40)$$

La ecuación 4.40 nos indica que los momentos generados por la fuerza de reacción son cero, siendo posible reemplazar a la reacción del piso por una única fuerza en el eje “Y”. Este fenómeno puede ser explicado de mejor forma a través de la figura 4.7.

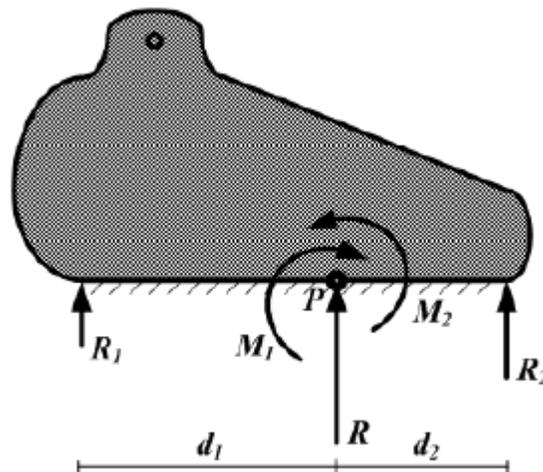


Figura 4.7. Compensación de los momentos en el pie de apoyo



En la figura 4.2 se puede observar que si $M_1 = M_2$, entonces $M_z = 0$. Cabe señalar que las longitudes de las flechas expresan el valor de la fuerza de reacción en algunos puntos del pie de apoyo. Los momentos generados por las fuerzas R_1 y R_2 son iguales ya que $R_1 d_1 = R_2 d_2$.

Cambiando el punto de referencia de “O” hacia “A” (figura 4.6) y despreciando la masa del pie se obtienen las siguientes ecuaciones:

$$\sum \vec{M}_A = 0 \quad (4.41)$$

$$(AP * R) + M_A = 0 \quad (4.42)$$

$$AP * R = -M_A \quad (4.43)$$

Claramente se puede observar que la igualdad de la ecuación 4.43 se cumple eligiendo el punto “P” adecuado, de tal forma que el momento generado por la fuerza de reacción del piso compense al momento total generado por el cuerpo en su caminata, este punto es conocido como el ZMP. El ZMP existe únicamente dentro del polígono de soporte. Cuando existe el ZMP el robot se encuentra dinámicamente estable, ya que es coincidente con el COG.

Cuando el momento generado por el cuerpo (M_A) es demasiado grande, el punto “P” puede centrarse fuera del polígono de soporte (figura 4.8), en este caso, la fuerza de reacción del piso se encuentra en el borde del pie (no puede seguir al punto “P” y no puede compensar el momento generado por el cuerpo en movimiento), si el ZMP no existe, evidentemente el robot tiende a perder su estabilidad. Un indicador físico de la pérdida de estabilidad del robot es la rotación del mismo a partir de los bordes del polígono de soporte.

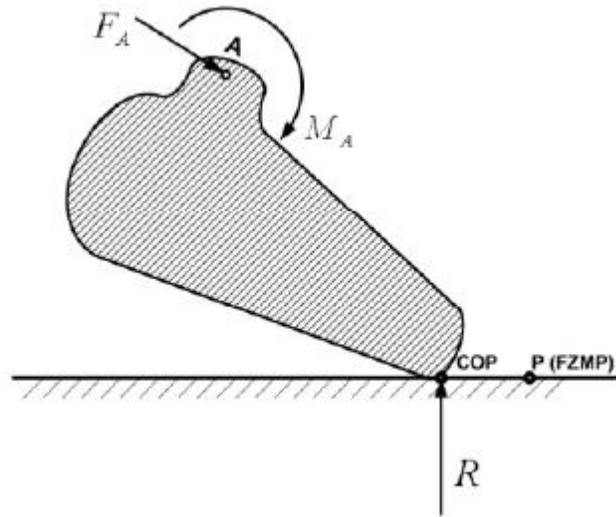


Figura 4.8. Rotación del pie de apoyo a partir de uno de sus bordes. Robot inestable

Dado que el ZMP existe únicamente dentro del polígono de soporte, esto nos da una idea de cuan estable se encuentra el robot en un determinado instante.

4.3 Cálculo del ZMP

A partir de diversos análisis de este punto (ZMP), han surgido principalmente dos tendencias para el cálculo:

Una de las formas de calcular el ZMP es a partir de la dinámica de cada uno de los eslabones que conforman el robot bípedo, de modo que, la ecuación para el equilibrio dinámico rotatorio se obtiene mediante la suma de los momentos sobre el robot calculado sobre cualquier punto de referencia estacionario, es decir; la suma de las razones de cambio del momento angular de los segmentos individuales entorno al mismo punto:

$$X_{ZMP} = \frac{\sum_i m_i z_i (a_{y_i} + g) - \sum_i m_i y_i a_{z_i} - \sum_i H_{G_{x_i}}}{\sum_i m_i (a_{y_i} + g)} \quad (4.44)$$

$$Z_{ZMP} = \frac{\sum_i m_i x_i (a_{y_i} + g) - \sum_i m_i y_i a_{x_i} - \sum_i H_{G_{z_i}}}{\sum_i m_i (a_{y_i} + g)} \quad (4.45)$$



Las ecuaciones 4.44 y 4.45 calculan con precisión las coordenadas del ZMP. Sin embargo se requieren estas, las aceleraciones, los momentos angulares, etc., del centro de gravedad de cada eslabón del robot bípodo.

Una manera más simple de calcular el ZMP es a partir de un modelo simplificado del robot bípodo, en este caso se opta por abordar el modelo del péndulo invertido 3D ya que el cálculo se realiza a partir de las coordenadas, aceleraciones del COG y el momento angular del robot bípodo.

El análisis que se presentará se enfoca principalmente en la etapa de soporte único del ciclo de caminata, durante el cual solo un pie (pie de apoyo) está en contacto con el suelo, En la caminata humana, la etapa de soporte único ocupa aproximadamente el 80% de todo el ciclo de caminata.

Considerando el modelo simplificado del robot bípodo (péndulo invertido 3D), se desarrolla la formulación del ZMP:

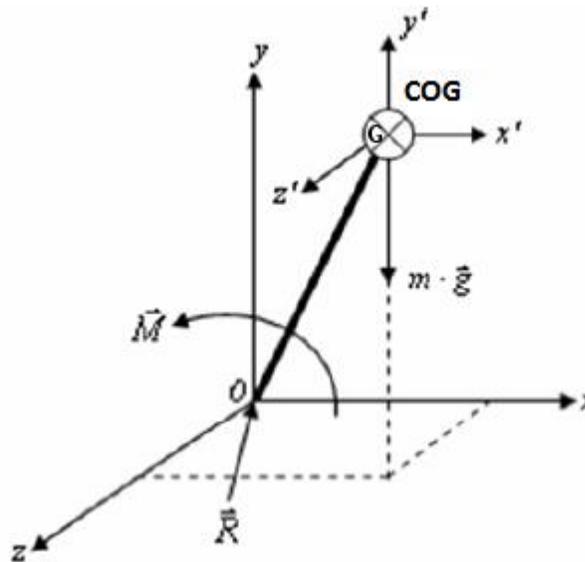


Figura 4.9. Modelo simplificado del robot bípodo (péndulo invertido 3D)

Según la figura 4.9, m es la masa total del robot bípodo concentrado en el COG, cuyas coordenadas quedan denotadas por x, y, z .

Partiendo del análisis de equilibrio de fuerzas $\sum \vec{F} = m\vec{a}$, y según la figura 4.4 tenemos que:

$$\begin{aligned} \sum F_x &= ma_x \\ R_x &= ma_x \end{aligned} \tag{4.46}$$



$$\begin{aligned}\sum F_y &= ma_y \\ R_y - mg &= ma_y \\ R_y &= m(a_y + g)\end{aligned}\tag{4.47}$$

$$\begin{aligned}\sum F_z &= ma_z \\ R_z &= ma_z\end{aligned}\tag{4.48}$$

Planteando la ecuación dinámica de momentos (sin considerar el movimiento giroscópico) $\sum M_G = \dot{H}_G = I_G \alpha_G$ obtenemos:

$$\begin{aligned}\sum M_{GX} &= I_{GX} \alpha_{GX} \\ M_X - (R_Z * y) + (R_Y * z) &= I_{GX} \alpha_{GX}\end{aligned}\tag{4.49}$$

Reemplazando en la ecuación 4.49 las ecuaciones 4.46 y 4.47, se obtiene la ecuación 4.50.

$$M_X = ma_z y - m(a_y + g)z + I_{GX} \alpha_{GX}\tag{4.50}$$

La ecuación en el eje "Z" queda de la siguiente forma:

$$\begin{aligned}\sum M_{GZ} &= I_{GZ} \alpha_{GZ} \\ M_Z - (R_Y * z) + (R_X * y) &= I_{GZ} \alpha_{GZ}\end{aligned}\tag{4.51}$$

Reemplazando en la ecuación 4.51 las ecuaciones 4.46 y 4.47, se obtiene la ecuación 4.52.

$$M_Z = m(a_y + g)x - ma_x y + I_{GZ} \alpha_{GZ}\tag{4.52}$$



4.3.1 Análisis del Pie de Apoyo

Para este análisis debemos considerar al pie en equilibrio estático, es decir, que al entrar este en contacto con el suelo no debe experimentar ningún tipo de deslizamiento en dirección al eje “X” ni al eje “Z”, que son los ejes que forman el plano horizontal sobre el cual se desplaza el robot.

El éxito del equilibrio de cualquier robot bípedo se encuentra en considerar el pie de apoyo correctamente asentado.

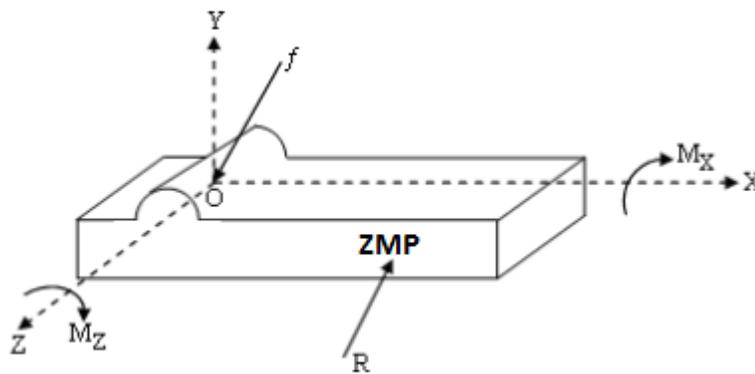


Figura 4.10. Análisis del pie de apoyo

$$\begin{aligned} \sum M_x &= 0 \\ -M_x - (R_y * Z_{ZMP}) &= 0 \\ Z_{ZMP} &= -\frac{M_x}{R_y} \end{aligned} \tag{4.53}$$

Finalmente en la ecuación 4.53, se sustituyen las ecuaciones 4.47 y 4.50 para obtener lo que se muestra en la ecuación 4.54.

$$Z_{ZMP} = \frac{m(a_y + g)z - ma_{zy} - I_{GX}\alpha_{GX}}{m(a_y + g)} \tag{4.54}$$



Siguiendo un proceso similar:

$$\begin{aligned}\sum M_Z &= 0 \\ -M_Z - (R_Y * X_{ZMP}) &= 0 \\ X_{ZMP} &= \frac{M_Z}{R_Y}\end{aligned}\tag{4.55}$$

$$X_{ZMP} = \frac{m(a_Y + g)x - ma_{Xy} + I_{GZ}\alpha_{GZ}}{m(a_Y + g)}\tag{4.56}$$

De esta manera obtenemos las ecuaciones del ZMP, que se encuentra en función netamente del comportamiento dinámico del robot bípedo. Replanteando las ecuaciones 4.55 y 4.56. Al considerar que $\dot{H}_G = I_G\alpha_G$ tenemos finalmente que:

$$X_{ZMP} = \frac{m(a_Y + g)x - ma_{Xy} + \dot{H}_Z}{m(a_Y + g)}\tag{4.57}$$

$$Z_{ZMP} = \frac{m(a_Y + g)z - ma_{Zy} - \dot{H}_X}{m(a_Y + g)}\tag{4.58}$$

4.3.2 Cálculo de las Velocidades y Aceleraciones del COG

Para calcular las velocidades y aceleraciones del COG del robot bípedo Scout es importante poder ubicarlo en el espacio y esto se logra identificando el COG de cada uno de los eslabones que conforman al robot bípedo y haciendo la sumatoria de cada COG como se muestra en las ecuaciones 4.59, 4.60 y 4.61.

$$X_{COG} = \frac{\sum_{i=1}^{12} m_i x_i}{\sum_{i=1}^{12} m_i}\tag{4.59}$$

$$Y_{COG} = \frac{\sum_{i=1}^{12} m_i y_i}{\sum_{i=1}^{12} m_i}\tag{4.60}$$

$$Z_{COG} = \frac{\sum_{i=1}^{12} m_i z_i}{\sum_{i=1}^{12} m_i}\tag{4.61}$$



Una vez calculada la posición del COG nos queda obtener las velocidades y aceleraciones lineales del mismo y así poder resolver el ZMP, se obtienen las velocidades lineales de cada eslabón para calcular la velocidad lineal del COG. Con la aceleración se hace lo mismo, aunado a que existe un impedimento para poder realizarlo ya que en la vida real se le tendrían que acoplar acelerómetros a cada uno de los 12 eslabones para así obtener estos valores, en la simulación no es necesario pues todo se programa de manera simultánea.

Teniendo en cuenta este impedimento se decidió aproximar el cálculo del COG a la posición de la cadera ya que en cálculos se encuentran relativamente cerca, esto da la ventaja de que en la vida real solo se acople un acelerómetro en donde se encuentra la cadera del bípedo Scout. De esta forma la posición y orientación del COG es igual a la posición y orientación de la cadera (Larriva, Vele, 2006).

Las consideraciones pasadas y del uso de la cinemática directa, sirven de apoyo para poder calcular las velocidades y aceleraciones lineales de una forma más fácil y sistemática.

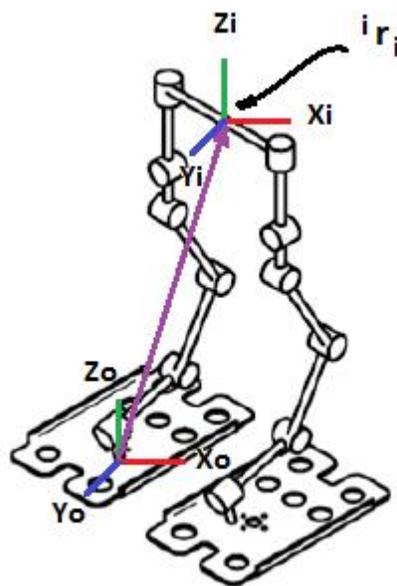


Figura 4.11. Un punto ${}^i r_i$ que se refiere a la cadera del robot bípedo Scout

Con referencia en la figura 4.11, sea ${}^i r_i$ un punto fijo y en reposo de la cadera expresado en coordenadas homogéneas con respecto al sistema de coordenadas del elemento i -ésimo.



$${}^i\mathbf{r}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = (x_i, y_i, z_i, 1)^T \quad (4.62)$$

Sea ${}^0\mathbf{r}_i$ el mismo punto ${}^i\mathbf{r}_i$ con respecto al sistema de coordenadas de la base, ${}^{i-1}\mathbf{T}_i$ la matriz de transformación de coordenadas homogéneas que relaciona el desplazamiento espacial del sistema de coordenadas del elemento i -ésimo con respecto al sistema de coordenadas del elemento $(i-1)$ -ésimo y ${}^0\mathbf{T}_i$ la matriz de transformación de coordenadas que relaciona el sistema de coordenadas i -ésimo con el sistema de coordenadas de la base; entonces ${}^0\mathbf{r}_i$ está relacionado con el punto ${}^i\mathbf{r}_i$ por:

$${}^0\mathbf{r}_i = {}^0\mathbf{T}_i {}^i\mathbf{r}_i \quad (4.63)$$

Dónde:

$${}^0\mathbf{T}_i = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \dots {}^{i-1}\mathbf{T}_i \quad (4.64)$$

Como el punto ${}^i\mathbf{r}_i$ está en reposo en la cadera y suponiendo el movimiento del cuerpo rígido, otros puntos así como el punto ${}^i\mathbf{r}_i$ fijado en la cadera y expresado con respecto al sistema de coordenadas i -ésimo tendrán velocidad nula con respecto a dicho sistema de coordenadas (que no es un sistema inercial). La velocidad de ${}^i\mathbf{r}_i$ expresada en el sistema de coordenadas de la base (que es un sistema inercial) se puede expresar como:

$$\begin{aligned} {}^0\mathbf{v}_i \equiv \mathbf{v}_i &= \frac{d({}^0\mathbf{r}_i)}{dt} = \frac{d({}^0\mathbf{T}_i {}^i\mathbf{r}_i)}{dt} = \\ &= {}^0\dot{\mathbf{T}}_1 {}^1\mathbf{T}_2 \dots {}^{i-1}\mathbf{T}_i {}^i\mathbf{r}_i + {}^0\mathbf{T}_1 {}^1\dot{\mathbf{T}}_2 \dots {}^{i-1}\mathbf{T}_i {}^i\mathbf{r}_i \\ &+ {}^0\mathbf{T}_1 \dots {}^{i-1}\dot{\mathbf{T}}_i {}^i\mathbf{r}_i + \dots + {}^0\mathbf{T}_i {}^i\dot{\mathbf{r}}_i = \left(\sum_{j=1}^i \frac{\partial {}^0\mathbf{T}_i}{\partial \theta_j} \dot{\theta}_j \right) \end{aligned} \quad (4.65)$$

La forma compacta anterior se obtiene por que ${}^i\dot{\mathbf{r}}_i = 0$. La derivada parcial de ${}^0\mathbf{T}_i$ con respecto a θ_j se puede calcular fácilmente con la ayuda de una matriz \mathbf{Q}_i que para una articulación de revolución, se define como:

$$\mathbf{Q}_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.66)$$



Entonces se sigue que:

$$\frac{\partial^{i-1}\mathbf{T}_i}{\partial\theta_j} = \mathbf{Q}_i^{i-1}\mathbf{T}_i \quad (4.67)$$

La matriz mostrada en la ecuación 4.66 funciona exclusivamente cuando las rotaciones de los sistemas de coordenadas de cada articulación son efectuadas en el eje “Z”, esto se cumple porque una de las virtudes del método de Denavit-Hartenberg para la cinemática directa es que en los centros de rotación siempre este orientado el eje “Z”.

Para aplicar la ecuación 4.65 se toma en cuenta que los grados de libertad de la base del pie de apoyo hasta la cadera son seis y es por ello que la ecuación de velocidades para el robot bípedo Scout queda de la siguiente manera tomando en cuenta la nomenclatura de la cinemática directa del capítulo 3:

$$\begin{aligned} {}^0\mathbf{v}_6 &= \left(\sum_{j=1}^6 \frac{\partial {}^0\mathbf{T}_6}{\partial \theta_j} \dot{\theta}_j \right) {}^i\mathbf{r}_i = \\ &= (AB\mathbf{Q}_1{}^0T_1{}^1T_2{}^2T_3{}^3T_4{}^4T_5{}^5T_6\dot{\theta}_1 + AB{}^0T_1\mathbf{Q}_2{}^1T_2{}^2T_3{}^3T_4{}^4T_5{}^5T_6\dot{\theta}_2 \\ &+ AB{}^0T_1{}^1T_2\mathbf{Q}_3{}^2T_3{}^3T_4{}^4T_5{}^5T_6\dot{\theta}_3 + AB{}^0T_1{}^1T_2{}^2T_3\mathbf{Q}_4{}^3T_4{}^4T_5{}^5T_6\dot{\theta}_4 \\ &+ AB{}^0T_1{}^1T_2{}^2T_3{}^3T_4\mathbf{Q}_5{}^4T_5{}^5T_6\dot{\theta}_5 + AB{}^0T_1{}^1T_2{}^2T_3{}^3T_4{}^4T_5\mathbf{Q}_6{}^5T_6\dot{\theta}_6) {}^6\mathbf{r}_6 \end{aligned} \quad (4.68)$$

Ahora bien para obtener la aceleración lineal del COG se procede a derivar con respecto al tiempo la ecuación 4.65, obteniendo:

$${}^0\mathbf{a}_i \equiv \frac{d(\mathbf{v}_i)}{dt} = \left(\sum_{j=1}^i \frac{\partial^2 {}^0\mathbf{T}_i}{\partial \theta_j^2} \dot{\theta}_j \ddot{\theta}_j \right) {}^i\mathbf{r}_i \quad (4.69)$$

Para aplicar la ecuación 4.69 se toma en cuenta que los grados de libertad de la base del pie de apoyo hasta la cadera son seis y es por ello que la ecuación de aceleraciones para el robot bípedo Scout queda de la siguiente manera:



$$\begin{aligned} {}^0 \mathbf{a}_6 &= \left(\sum_{j=1}^6 \frac{\partial^2 {}^0 \mathbf{T}_6}{\partial \theta_j^2} \dot{\theta}_j \ddot{\theta}_j \right) \mathbf{r}_i = \\ &= (AB \mathbf{Q}_1 \mathbf{Q}_1 {}^0 T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5 {}^5 T_6 \dot{\theta}_1 \ddot{\theta}_1 \\ &+ AB {}^0 T_1 \mathbf{Q}_2 \mathbf{Q}_2 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5 {}^5 T_6 \dot{\theta}_2 \ddot{\theta}_2 \\ &+ AB {}^0 T_1 {}^1 T_2 \mathbf{Q}_3 \mathbf{Q}_3 {}^2 T_3 {}^3 T_4 {}^4 T_5 {}^5 T_6 \dot{\theta}_3 \ddot{\theta}_3 \\ &+ AB {}^0 T_1 {}^1 T_2 {}^2 T_3 \mathbf{Q}_4 \mathbf{Q}_4 {}^3 T_4 {}^4 T_5 {}^5 T_6 \dot{\theta}_4 \ddot{\theta}_4 \\ &+ AB {}^0 T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 \mathbf{Q}_5 \mathbf{Q}_5 {}^4 T_5 {}^5 T_6 \dot{\theta}_5 \ddot{\theta}_5 \\ &+ AB {}^0 T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5 \mathbf{Q}_6 \mathbf{Q}_6 {}^5 T_6 \dot{\theta}_6 \ddot{\theta}_6) {}^6 \mathbf{r}_6 \end{aligned} \quad (4.70)$$

Al aplicar las ecuaciones 4.68 y 4.70 a la simulación propuesta en el capítulo 3 (donde el robot bípedo sigue las trayectorias elípticas) se presentan los siguientes resultados expresados en las figuras 4.12 a la 4.13 que se refieren a la velocidad y aceleración del COG del robot bípedo Scout.

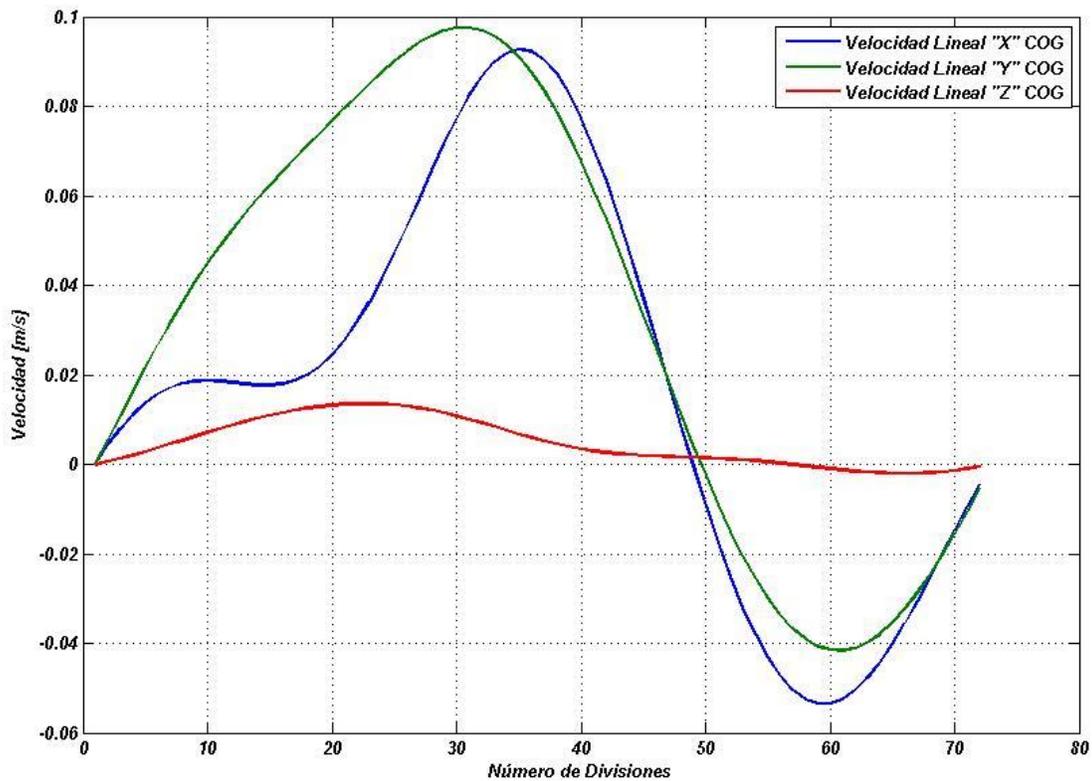


Figura 4.12. Velocidad Lineal del COG en sus componentes "X, Y, Z"

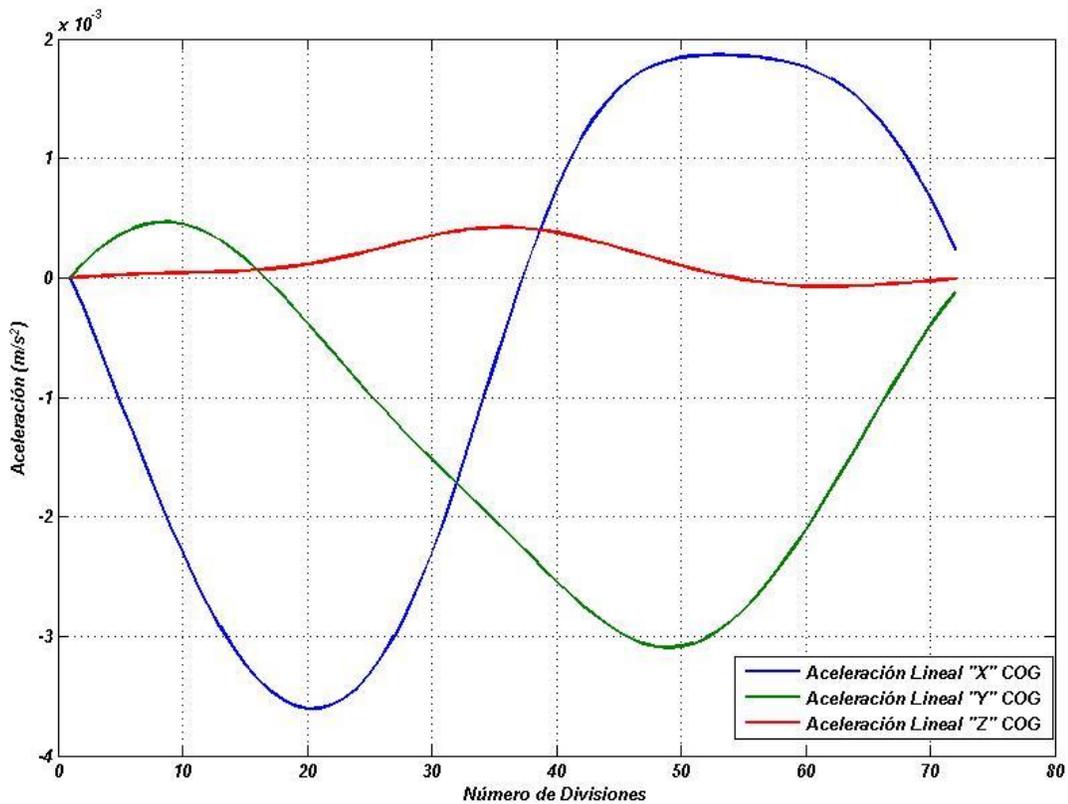


Figura 4.13. Aceleración Lineal del COG en sus componentes “X, Y, Z”

La aceleración lineal del COG es la que se utiliza para el cálculo del ZMP, es importante remarcar que estas aceleraciones están en $[m/s]$ y para poder realizar esta simulación se tienen que proponer las velocidades y aceleraciones angulares de cada una de las articulaciones del robot, estas están expresadas en la ecuación 4.32 y 4.33.

Como se observa en las imágenes de la velocidad y aceleración del COG la velocidad máxima que alcanza es del orden de los 15-18 $[cm/s]$ y de la aceleración es del orden de los 3-6 $[mm/s]$, estos datos parecen ser bastante similares a los mostrados por el bípedo Scouts en pruebas anteriores.

Una vez que se obtienen las velocidades y aceleraciones lineales y se proponen las velocidades y aceleraciones angulares se sustituyen en las ecuaciones 4.57 y 4.58 para cada división de la trayectoria propuesta, además se toma en cuenta la masa total del bípedo (Anexo A) y la gravedad que debería ser de $9.81 \frac{m}{s^2}$, esto arroja los resultados de la posición del ZMP (figuras 4.14, 4.15).

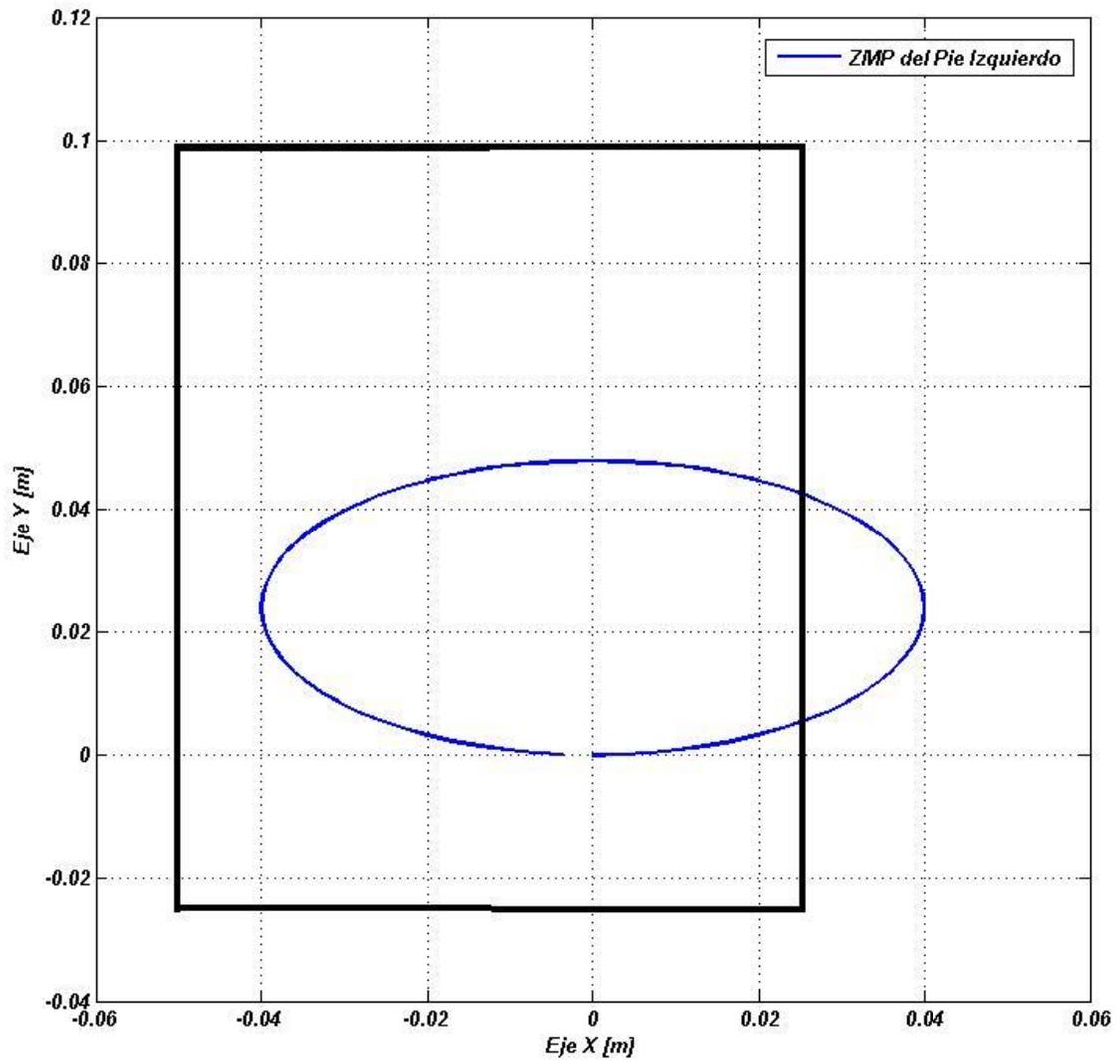


Figura 4.14. Calculo del ZMP en el pie Izquierdo

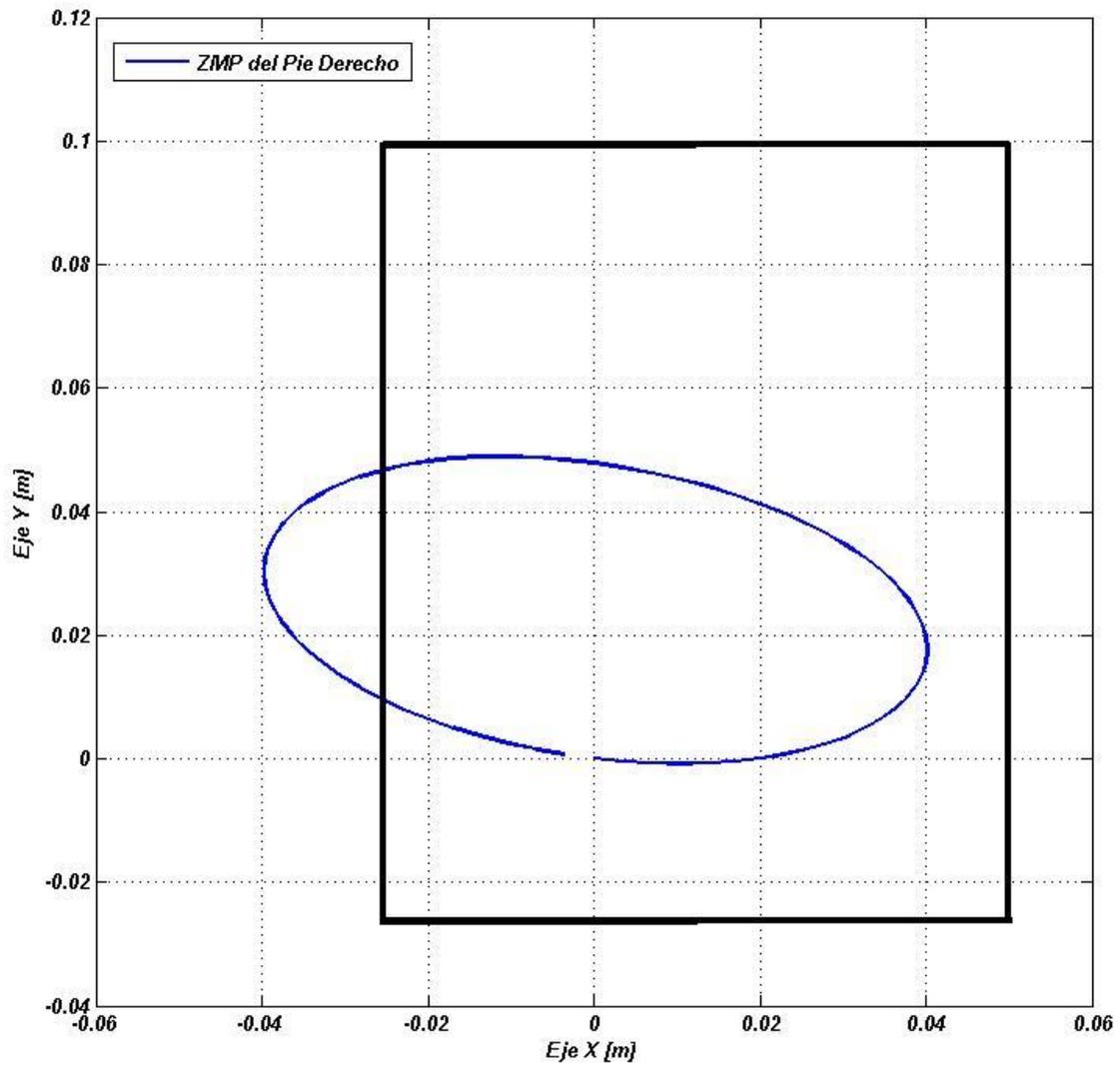


Figura 4.15. Calculo del ZMP en el pie Derecho



Como se puede apreciar en la figuras 4.14 y 4.15 se dibuja un polígono de soporte (Cuadrado) que simula la suela de los pies del bípedo Scout, la gráfica en azul muestra la ubicación del ZMP para cada pie cuando son el pie de soporte, lo que esto nos indica es que para cuando el ZMP está fuera del polígono de soporte el bípedo tiende a caer y que mientras el ZMP se encuentra dentro del polígono de soporte el bípedo se mantiene estable en la caminata. Como también se aprecia en las gráficas la trayectoria del ZMP se sigue de derecha a izquierda y esto es gracias a que la trayectoria de la cadera lo hace del mismo modo.

Los polígonos de soporte no son exactamente iguales a los reales y esto se debe a que se aproximó el área a una más simple con el fin de poder apreciar los resultados de manera más concisa, aun así esta aproximación es la que se debe de utilizar en el diseño del controlador por que al ver el polígono de soporte real (Anexo A) se logra apreciar que en diferentes áreas que están fuera del polígono aún puede existir estabilidad en la caminata.

En términos de control se debe definir un área límite dentro del polígono de soporte esto con el fin de que el sistema de control pueda actuar a tiempo y así estabilizar la caminata del bípedo. Mientras no se tenga un controlador para que el ZMP siempre este dentro del polígono de soporte, se tienen que elegir trayectorias que mantengan al ZMP en el lugar deseado para que de esta forma el robot bípedo no caiga.



5

Conclusiones y Trabajo a Futuro

En este capítulo se propondrá un esquema de controlador para el robot bípedo Scout, además de que se ofrecerán conclusiones respecto al proyecto de investigación



5.1 Conclusiones

El objetivo principal de este proyecto de investigación fue encontrar un modelo dinámico más rápido de computar que los modelos ya aplicados anteriormente al robot bípedo Scout. Se propuso un procedimiento que recurre al modelo cinemático, el cual nos aporta los datos que nos permitieron obtener las mejoras deseadas en el modelo dinámico. En este proyecto se muestra satisfactoriamente como aplicar los correctos modelos matemáticos para la cinemática que nos permiten llegar a la formulación de la dinámica propuesta para el robot bípedo, logrando que se cumpla el objetivo ya mencionado.

Se tenía contemplado al cálculo del ZMP como criterio de estabilidad durante la marcha del bípedo con la intención de ofrecer una base para generar un sistema de control. Este cálculo, en este proyecto fue muy útil para saber si la trayectoria propuesta para el COG era estable durante la caminata y se recurrió a él debido a que es el cálculo con más variedad de estudios previos y cuya información es extensa y de fácil acceso.

Uno dato interesante que se descubrió durante la realización de este proyecto es que el cálculo de la dinámica debe ser realizado de manera rápida por la computadora, ya que de no ser así, el controlador a diseñar tardaría demasiado tiempo en responder haciendo que el control estuviera fuera de línea, lo que hace que el robot no se caiga es la respuesta rápida del controlador, si el sistema tiene un control demasiado lento calculando la dinámica, este no podrá responder a tiempo a las perturbaciones externas ni a una mala selección de trayectorias del COG.

Los modelos mostrados de la cinemática y dinámica del robot bípedo Scout no solo pueden funcionar para este robot si no que son una excelente elección para todos los robots bípedos articulados con articulaciones de revolución, solo teniendo que adaptar los parámetros de Denavit-Hartenberg que son propios de cada robot. Esto se puede notar claramente debido a que las formulaciones de la cinemática son muy metódicas y solo difieren en los parámetros de entrada, lo mismo pasa con las formulaciones de la dinámica.

Durante la realización de este proyecto se presentaron algunas restricciones que no se tenían contempladas:

- Para el método de solución de la cinemática inversa se tienen que elegir los ángulos iniciales correctamente ya que de no ser así puede que la solución que obtengamos no sea la deseada, porque para cierto punto las soluciones pueden ser demasiadas, es posible además que el método nunca converja.
- Otra de las restricciones es que el modelo dinámico desprecia los efectos de Coriolis y centrífugo limitando la estabilidad de la marcha a velocidades pequeñas ya que estos términos toman más relevancia a velocidades grandes.



- El pie de apoyo debe estar siempre bien plantado en el suelo para garantizar que el ZMP este dentro del polígono de soporte, esto evita un movimiento natural de la marcha del ser humano (flexión del pie).
- Se tienen que proponer trayectorias muy parecidas a las que el ser humano usa para caminar en línea recta, la obtención de las mismas puede ser un problema bastante complejo.

5.2 Trabajo a Futuro

Una parte medular para hacer que el bípedo camine es controlar el ciclo de marcha, esto controlando los servomotores del robot bípedo para que cada uno llegue a la posición deseada, además de diseñar un controlador que modifique la posición del ZMP para que en el momento en que este cerca de salir del polígono de soporte el controlador lo compense y regrese el ZMP a un lugar más adecuado para la caminata.

Esta problemática se ve resuelta con el siguiente esquema de control (figura 5.1) que controla la posición de los servomotores y que no permite que el ZMP salga del polígono de soporte, para que este efecto tenga lugar se necesita también una trayectoria deseada del ZMP con el fin de que el controlador tenga con que comparar la trayectoria real, el objetivo de control que se seguiría es llevar el error de la comparación del ZMP a cero. (figura 5.1)

Como se aprecia en la figura 5.1 las zonas que están en rojo son partes que no se trabajaron en este proyecto de investigación tales como la generación de trayectorias de la cadera, el pie flotante y el ZMP deseado, tampoco se trabajó en el controlador de los servomotores ya que todo el trabajo fue hecho por simulación. Los bloques que se lograron fueron los de la cinemática inversa además del cálculo de la aceleración y velocidad del COG, así como las ecuaciones que posicionan al ZMP en el suelo.

Se deja como trabajo a futuro el diseño del bloque de control que ajusta el ZMP para que este criterio de estabilidad siempre este dentro del polígono de soporte, para ello se deben analizar las ecuaciones dinámicas basadas en el péndulo invertido 3D para saber si el sistema es estable o inestable y proponer controladores, también se pueden hacer uso de métodos heurísticos como redes neuronales, algoritmos genéticos, lógica difusa etc.

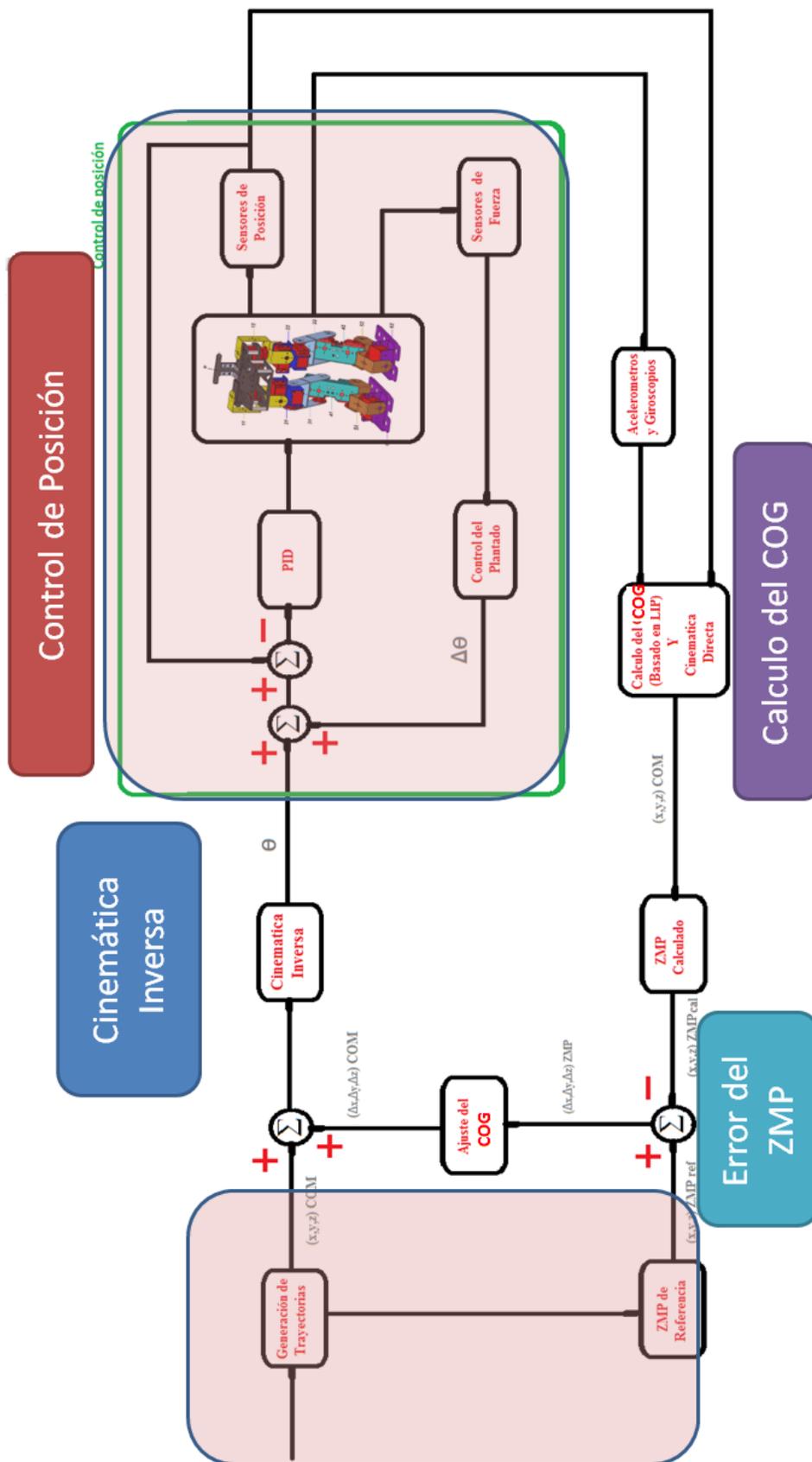


Figura 5.1. Esquema de control del robot bípedo Scout.



Anexo A



A.1 Especificaciones del Bípido Scout

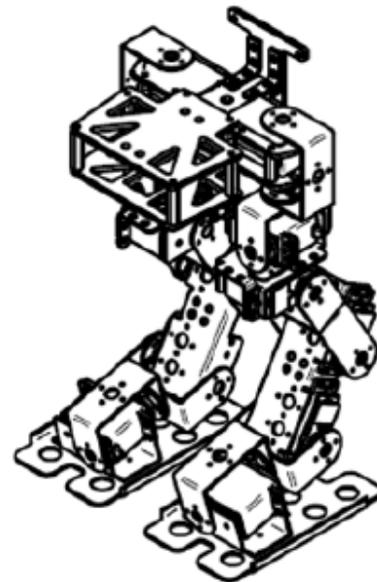
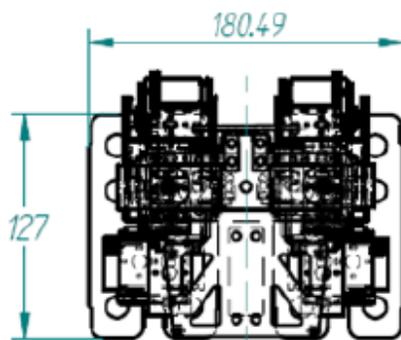
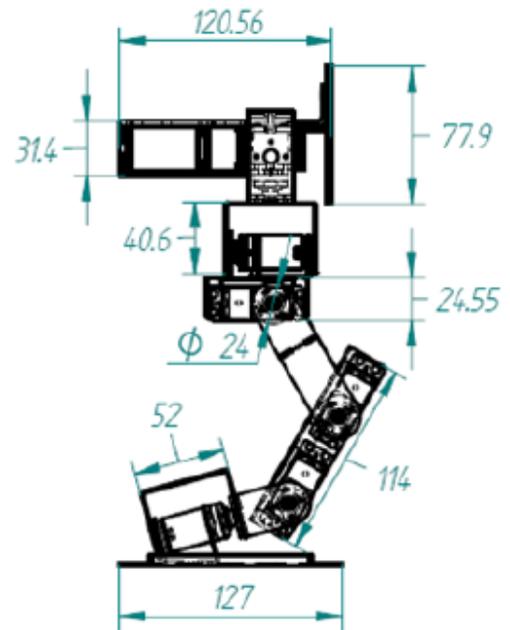
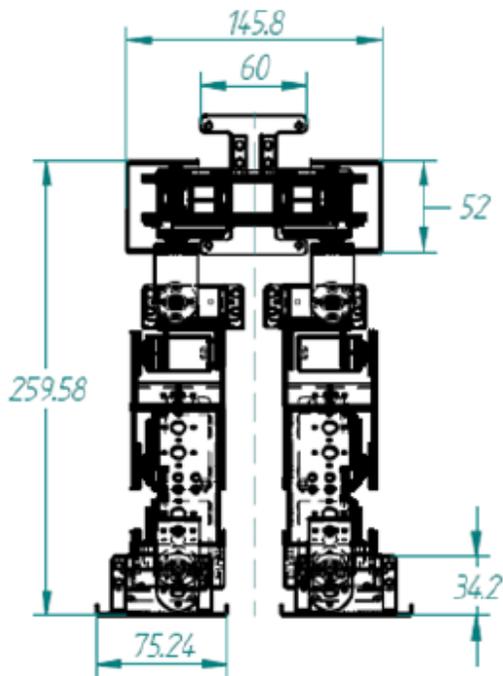
Todos los dibujos, mediciones y cálculos relacionados a los modelos CAD del robot bípido Scout fueron elaborados en Solid Edge Academic V16 (Narváez, 2010).

Tabla A.1. Lista de elementos

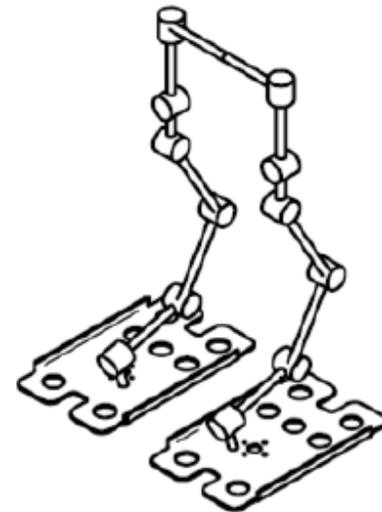
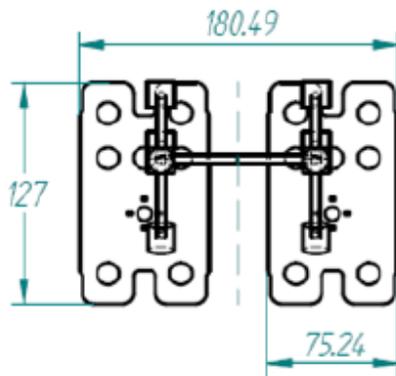
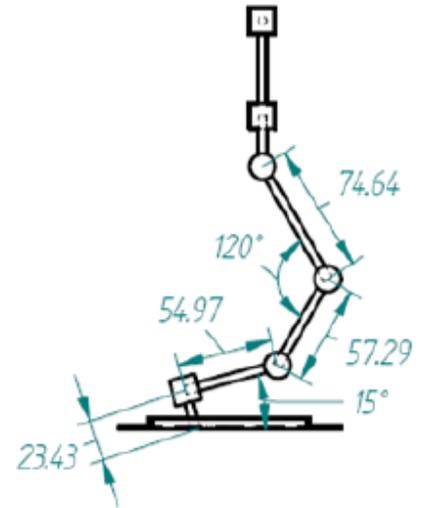
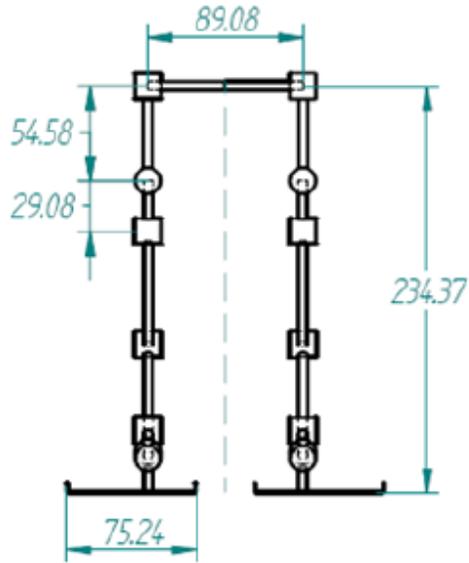
Elemento	Descripción	Cantidad	Material
	Soporte L	2	Aluminio $\rho = 2.6865 \text{ [g/cm}^3\text{]}$
	Separador hexagonal corto 4-40 x 0.750"	12	Nylon $\rho = 1.1756 \text{ [g/cm}^3\text{]}$
	Separador hexagonal largo 4-40 x 1.000"	4	Nylon
	Separador hexagonal M/F 4-40 x 0.375"	4	Nylon
	Remache plástico	16	Nylon
	Buje	12	Acero $\rho = 7.75 \text{ [g/cm}^3\text{]}$
	Tornillo corto 4-40 x 0.250"	8	Acero
	Tornillo largo 4-40 x 0.375"	24	Acero
	Servomotor programable Hitec HS-5475HB Velocidad de operación: 4.55[rad/s] @ 4.8[V] Torque: 0.432 [N.m] @ 4.8[V] Masa: 40 [g] Tamaño: 38.8 x 19.8 x 36 [mm]	12	Varios



Elemento	Descripción	Cantidad	Material
	Lexan 01	1	Lexan $\rho = 1.1789 \text{ [g/cm}^3\text{]}$
	Lexan 02	1	Lexan
	Lexan 03	1	Lexan
	Lexan 04	1	Lexan
	Lexan 05	1	Lexan
	Soporte para servomotor	2	Nylon $\rho = 1.1756 \text{ [g/cm}^3\text{]}$
	Soporte C corto	10	Aluminio $\rho = 2.6865 \text{ [g/cm}^3\text{]}$
	Soporte C largo	2	Aluminio
	Soporte multipropósito	4	Aluminio
	Soporte dual	2	Aluminio
	Soporte para tobillo	2	Aluminio
	Pie	2	Aluminio



ocnaar	ESCALA 1:4	mm
	Bípido Scout	



ocnaar	ESCALA 1:4	mm
	Simplificación del Bípodo Scout	



A.2 Parámetros Geométricos

$$l_1 = 44.54 \times 10^{-3} [m]$$

$$l_2 = 54.58 \times 10^{-3} [m]$$

$$l_3 = 29.08 \times 10^{-3} [m]$$

$$l_4 = 74.60 \times 10^{-3} [m]$$

$$l_5 = 57.29 \times 10^{-3} [m]$$

$$l_6 = 54.97 \times 10^{-3} [m]$$

$$a = 23.06 \times 10^{-3} [m]$$

Estas longitudes pertenecen a cada uno de los eslabones del robot de la pierna derecha e izquierda; son las mismas longitudes ya que el robot en ese aspecto es simétrico, cabe señalar que las longitudes empiezan desde la cadera hasta el suelo, por lo que el parámetro "a" es el eslabón que se refiere a la unión del último grado de libertad con el suelo.

A.3 Masa de los Eslabones

$$m_1 = 162.08 \times 10^{-3} [kg]$$

$$m_2 = 23.12 \times 10^{-3} [kg]$$

$$m_3 = 102.27 \times 10^{-3} [kg]$$

$$m_4 = 26.67 \times 10^{-3} [kg]$$

$$m_5 = 103.57 \times 10^{-3} [kg]$$

$$m_6 = 23.12 \times 10^{-3} [kg]$$

$$m_a = 89.48 \times 10^{-3} [kg]$$

La masa total del bípedo es:

$$m_T = 898.54 \times 10^{-3} [kg]$$



Anexo B



B.1 Programa “BPDPIA2”.- Este programa resuelve la cinemática directa e inversa de la cadera referenciada al pie de apoyo del robot bípedo Scout. (Programado en Matlab).

```
function [VCx,VCy,VCz,ACx,ACy,ACz,Th]=BPDPIA2 ()

    clf
    lol=1;

    for c=0:1:1

        dx=0.01;
        o=44.54;
        ll=0;

        if c==1
            o=-44.54;
            lol=lol+1;
        end

        g=40;
        e=24;
        h=0;
        k=24;

        g2=40;
        e2=24;
        h2=-37;
        k2=-235+e2;
        r=[0 0 0 1]';

        t(1)=(0*pi)/180;
        t(2)=(45*pi)/180;
        t(3)=(60*pi)/180;
        t(4)=(-30*pi)/180;
        t(5)=(0*pi)/180;
        t(6)=(0*pi)/180;
        t(7)=(0*pi)/180;
        t(8)=(0*pi)/180;
        t(9)=(30*pi)/180;
        t(10)=(-60*pi)/180;
        t(11)=(-45*pi)/180;
        t(12)=(90*pi)/180;

        Th(lol,1)=t(1);
        Th(lol,2)=t(2);
        Th(lol,3)=t(3);
        Th(lol,4)=t(4);
        Th(lol,5)=t(5);
        Th(lol,6)=t(6);
        Th(lol,7)=t(7);
        Th(lol,8)=t(8);
        Th(lol,9)=t(9);
        Th(lol,10)=t(10);
        Th(lol,11)=t(11);
        Th(lol,12)=t(12);
```



```
for l=360:-5:0

    if l<360

        l1=l1+1;
        lol=lol+1;

        A1=0;
        A2=0;
        A3=0;
        A4=pi/2;
        A5=0;
        A6=pi;

        a1=0;
        a2=57.29;
        a3=74.64;
        a4=29.08;
        a5=54.58;
        a6=0;

        d1=-54.97;
        d2=0;
        d3=0;
        d4=0;
        d5=0;
        d6=0;

        T1=[cos(t(1)) -cos(A1)*sin(t(1)) sin(A1)*sin(t(1)) a1*cos(t(1));...
            sin(t(1)) cos(A1)*cos(t(1)) -sin(A1)*cos(t(1)) a1*sin(t(1));...
            0 sin(A1) cos(A1) d1;...
            0 0 0 1];
        T2=[cos(t(2)) -cos(A2)*sin(t(2)) sin(A2)*sin(t(2)) a2*cos(t(2));...
            sin(t(2)) cos(A2)*cos(t(2)) -sin(A2)*cos(t(2)) a2*sin(t(2));...
            0 sin(A2) cos(A2) d2;...
            0 0 0 1];
        T3=[cos(t(3)) -cos(A3)*sin(t(3)) sin(A3)*sin(t(3)) a3*cos(t(3));...
            sin(t(3)) cos(A3)*cos(t(3)) -sin(A3)*cos(t(3)) a3*sin(t(3));...
            0 sin(A3) cos(A3) d3;...
            0 0 0 1];
        T4=[cos(t(4)) -cos(A4)*sin(t(4)) sin(A4)*sin(t(4)) a4*cos(t(4));...
            sin(t(4)) cos(A4)*cos(t(4)) -sin(A4)*cos(t(4)) a4*sin(t(4));...
            0 sin(A4) cos(A4) d4;...
            0 0 0 1];
        T5=[cos(t(5)) -cos(A5)*sin(t(5)) sin(A5)*sin(t(5)) a5*cos(t(5));...
            sin(t(5)) cos(A5)*cos(t(5)) -sin(A5)*cos(t(5)) a5*sin(t(5));...
            0 sin(A5) cos(A5) d5;...
            0 0 0 1];
        T6=[cos(t(6)) -cos(A6)*sin(t(6)) sin(A6)*sin(t(6)) a6*cos(t(6));...
            sin(t(6)) cos(A6)*cos(t(6)) -sin(A6)*cos(t(6)) a6*sin(t(6));...
            0 sin(A6) cos(A6) d6;...
            0 0 0 1];

        a=[1 0 0 -0;0 -1 0 -38.3578; 0 0 -1 -234.7737; 0 0 0 1];
        b=[1 0 0 0;0 1 0 0; 0 0 1 23.43; 0 0 0 1];
```



```
pto=a*rotx(-165)*b*rotx(90);
ptt1=pto*T1*roty(90);
ptt2=ptt1*T2;
ptt3=ptt2*T3;
ptt4=ptt3*T4;
ptt5=ptt4*T5*roty(-90)*rotz(90);
ptt6=ptt5*T6;

[VCx(11,1),VCy(11,1),VCz(11,1),ACx(11,1),ACy(11,1),ACz(11,1)]=VELACELBPD(r,t,o);

[p1,p2,p3,p4,p5,p6,p7,tt]=BPDPIF(1,ptt6,dx,h2,k2,g2,e2,o);

Th(101,1)=t(1);
Th(101,2)=t(2);
Th(101,3)=t(3);
Th(101,4)=t(4);
Th(101,5)=t(5);
Th(101,6)=t(6);
Th(101,7)=tt(1);
Th(101,8)=tt(2);
Th(101,9)=tt(3);
Th(101,10)=tt(4);
Th(101,11)=tt(5);
Th(101,12)=tt(6);

hold on
axis([-100 100,-100 100,-250 50]);
view(150,50)

plotbipedo2(a,pto,ptt1,ptt2,ptt3,ptt4,ptt5,ptt6);
[xx,yy,zz]=ellipse2(h,k,g,e);
plot3(xx,zz,yy)

plotbipedo3(ptt6,p1,p2,p3,p4,p5,p6,p7);
[xx,yy,zz]=ellipse(h2,k2,g2,e2);
plot3(yy,xx,zz)
plot3(-yy,xx,zz)

pause(.01)

if l>0
    clf
end
end

WW=[1 0 0 h-(g*sind(1));...
    0 1 0 k-(e*cosd(1));...
    0 0 1 0];
r=[WW(1,4) WW(2,4) WW(3,4) 1]';

for n=1:1000

    xp=t(1)+dx;
    zp=t(2)+dx;
    yp=t(3)+dx;
    rp=t(4)+dx;
```




$$J(9,4) = (F9PDA(t(1), t(2), t(3), rp, t(5), t(6), WW(3,3)) - F9PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,3))) / dx;$$

$$J(9,5) = (F9PDA(t(1), t(2), t(3), t(4), fp, t(6), WW(3,3)) - F9PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,3))) / dx;$$

$$J(9,6) = (F9PDA(t(1), t(2), t(3), t(4), t(5), wp, WW(3,3)) - F9PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,3))) / dx;$$

$$J(10,1) = (F10PDA(xp, t(2), t(3), t(4), t(5), t(6), WW(1,4)) - F10PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,4))) / dx;$$

$$J(10,2) = (F10PDA(t(1), zp, t(3), t(4), t(5), t(6), WW(1,4)) - F10PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,4))) / dx;$$

$$J(10,3) = (F10PDA(t(1), t(2), yp, t(4), t(5), t(6), WW(1,4)) - F10PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,4))) / dx;$$

$$J(10,4) = (F10PDA(t(1), t(2), t(3), rp, t(5), t(6), WW(1,4)) - F10PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,4))) / dx;$$

$$J(10,5) = (F10PDA(t(1), t(2), t(3), t(4), fp, t(6), WW(1,4)) - F10PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,4))) / dx;$$

$$J(10,6) = (F10PDA(t(1), t(2), t(3), t(4), t(5), wp, WW(1,4)) - F10PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,4))) / dx;$$

$$J(11,1) = (F11PDA(xp, t(2), t(3), t(4), t(5), t(6), WW(2,4)) - F11PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,4))) / dx;$$

$$J(11,2) = (F11PDA(t(1), zp, t(3), t(4), t(5), t(6), WW(2,4)) - F11PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,4))) / dx;$$

$$J(11,3) = (F11PDA(t(1), t(2), yp, t(4), t(5), t(6), WW(2,4)) - F11PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,4))) / dx;$$

$$J(11,4) = (F11PDA(t(1), t(2), t(3), rp, t(5), t(6), WW(2,4)) - F11PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,4))) / dx;$$

$$J(11,5) = (F11PDA(t(1), t(2), t(3), t(4), fp, t(6), WW(2,4)) - F11PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,4))) / dx;$$

$$J(11,6) = (F11PDA(t(1), t(2), t(3), t(4), t(5), wp, WW(2,4)) - F11PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,4))) / dx;$$

$$J(12,1) = (F12PDA(xp, t(2), t(3), t(4), t(5), t(6), WW(3,4)) - F12PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,4))) / dx;$$

$$J(12,2) = (F12PDA(t(1), zp, t(3), t(4), t(5), t(6), WW(3,4)) - F12PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,4))) / dx;$$

$$J(12,3) = (F12PDA(t(1), t(2), yp, t(4), t(5), t(6), WW(3,4)) - F12PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,4))) / dx;$$

$$J(12,4) = (F12PDA(t(1), t(2), t(3), rp, t(5), t(6), WW(3,4)) - F12PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,4))) / dx;$$

$$J(12,5) = (F12PDA(t(1), t(2), t(3), t(4), fp, t(6), WW(3,4)) - F12PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,4))) / dx;$$

$$J(12,6) = (F12PDA(t(1), t(2), t(3), t(4), t(5), wp, WW(3,4)) - F12PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,4))) / dx;$$

$$f(1) = F1PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,1));$$

$$f(2) = F2PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,1));$$

$$f(3) = F3PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,1));$$

$$f(4) = F4PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,2));$$

$$f(5) = F5PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,2));$$

$$f(6) = F6PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,2));$$

$$f(7) = F7PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,3));$$

$$f(8) = F8PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,3));$$

$$f(9) = F9PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(3,3));$$

$$f(10) = F10PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(1,4));$$

$$f(11) = F11PDA(t(1), t(2), t(3), t(4), t(5), t(6), WW(2,4));$$



```
f(12)=F12PDA(t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4));

ds=pinv(J)*(-f');

t(1)=t(1)+ds(1);
t(2)=t(2)+ds(2);
t(3)=t(3)+ds(3);
t(4)=t(4)+ds(4);
t(5)=t(5)+ds(5);
t(6)=t(6)+ds(6);

    if (abs(f(1))<1.0e-3&&abs(f(2))<1.0e-3&&abs(f(3))<1.0e-
3&&abs(f(4))<1.0e-3&&abs(f(5))<1.0e-3&&abs(f(6))<1.0e-3&&abs(f(7))<1.0e-
3&&abs(f(8))<1.0e-3&&abs(f(9))<1.0e-3&&abs(f(10))<1.0e-3&&abs(f(11))<1.0e-3&&abs(f(12))<1.0e-3)
        break
    end
end
end
end
end
```

B.2 Programa “BPDPIF”.- Este programa resuelve la cinemática directa e inversa del pie flotante referenciado a la cadera del robot bípedo Scout. (Programado en Matlab).

```
function[pto,ptt2,ptt3,ptt4,ptt5,ptt6,ptt7,t]=BPDPIF(l,a,dx,h,k,g,e,o)

clf

t(1)=(0*pi)/180;
t(2)=(0*pi)/180;
t(3)=(30*pi)/180;
t(4)=(-60*pi)/180;
t(5)=(-45*pi)/180;
t(6)=(90*pi)/180;

WW=[ 1 0 0 44.54;...
0 -1 0 h-(g*sind(l));...
0 0 -1 k-(e*cosd(l))];

for n=1:1000

    xp=t(1);
    zp=t(2)+dx;
    yp=t(3)+dx;
    rp=t(4)+dx;
    fp=t(5)+dx;
    wp=t(6)+dx;
    J(1,1)=(F1PIF(a(1,4),a(2,4),a(3,4),xp,t(2),t(3),t(4),t(5),t(6),WW(1,1))-
F1PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,1)))/dx;
    J(1,2)=(F1PIF(a(1,4),a(2,4),a(3,4),t(1),zp,t(3),t(4),t(5),t(6),WW(1,1))-
```




```
F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,4)))/dx;
J(10,4)=(F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),rp,t(5),t(6),WW(1,4))-
F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,4)))/dx;
J(10,5)=(F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),fp,t(6),WW(1,4))-
F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,4)))/dx;
J(10,6)=(F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),wp,WW(1,4))-
F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,4)))/dx;
```

```
J(11,1)=(F11PIF(a(1,4),a(2,4),a(3,4),xp,t(2),t(3),t(4),t(5),t(6),WW(2,4))-
F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4)))/dx;
J(11,2)=(F11PIF(a(1,4),a(2,4),a(3,4),t(1),zp,t(3),t(4),t(5),t(6),WW(2,4))-
F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4)))/dx;
J(11,3)=(F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4))-
F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4)))/dx;
J(11,4)=(F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),rp,t(5),t(6),WW(2,4))-
F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4)))/dx;
J(11,5)=(F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),fp,t(6),WW(2,4))-
F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4)))/dx;
J(11,6)=(F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),wp,WW(2,4))-
F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4)))/dx;
```

```
J(12,1)=(F12PIF(a(1,4),a(2,4),a(3,4),xp,t(2),t(3),t(4),t(5),t(6),WW(3,4))-
F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4)))/dx;
J(12,2)=(F12PIF(a(1,4),a(2,4),a(3,4),t(1),zp,t(3),t(4),t(5),t(6),WW(3,4))-
F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4)))/dx;
J(12,3)=(F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),yp,t(4),t(5),t(6),WW(3,4))-
F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4)))/dx;
J(12,4)=(F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),rp,t(5),t(6),WW(3,4))-
F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4)))/dx;
J(12,5)=(F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4))-
F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4)))/dx;
J(12,6)=(F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),wp,WW(3,4))-
F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4)))/dx;
```

```
f(1)=F1PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,1));
f(2)=F2PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,1));
f(3)=F3PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,1));
f(4)=F4PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,2));
f(5)=F5PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,2));
f(6)=F6PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,2));
f(7)=F7PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,3));
f(8)=F8PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,3));
f(9)=F9PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,3));
f(10)=F10PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(1,4));
f(11)=F11PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(2,4));
f(12)=F12PIF(a(1,4),a(2,4),a(3,4),t(1),t(2),t(3),t(4),t(5),t(6),WW(3,4));
```

```
ds=pinv(J)*(-f');
```

```
t(1)=t(1)+ds(1);
t(2)=t(2)+ds(2);
t(3)=t(3)+ds(3);
t(4)=t(4)+ds(4);
t(5)=t(5)+ds(5);
t(6)=t(6)+ds(6);
```



```
if (abs(f(1))<1.0e-9&&abs(f(2))<1.0e-9&&abs(f(3))<1.0e-
9&&abs(f(4))<1.0e-9&&abs(f(5))<1.0e-9&&abs(f(6))<1.0e-9&&abs(f(7))<1.0e-9&&abs(f(8))<1.0e-
9&&abs(f(9))<1.0e-9&&abs(f(10))<1.0e-9&&abs(f(11))<1.0e-9&&abs(f(12))<1.0e-9)
    break
end
end
A1=0;
A2=pi/2;
A3=0;
A4=0;
A5=0;
A6=0;

a1=0;
a2=29.08;
a3=74.64;
a4=57.29;
a5=54.97;
a6=23.43;

d1=54.58;
d2=0;
d3=0;
d4=0;
d5=0;
d6=0;

T1=[cos(t(1)) -cos(A1)*sin(t(1)) sin(A1)*sin(t(1)) a1*cos(t(1));...
    sin(t(1)) cos(A1)*cos(t(1)) -sin(A1)*cos(t(1)) a1*sin(t(1));...
    0 sin(A1) cos(A1) d1;...
    0 0 0 1];
T2=[cos(t(2)) -cos(A2)*sin(t(2)) sin(A2)*sin(t(2)) a2*cos(t(2));...
    sin(t(2)) cos(A2)*cos(t(2)) -sin(A2)*cos(t(2)) a2*sin(t(2));...
    0 sin(A2) cos(A2) d2;...
    0 0 0 1];
T3=[cos(t(3)) -cos(A3)*sin(t(3)) sin(A3)*sin(t(3)) a3*cos(t(3));...
    sin(t(3)) cos(A3)*cos(t(3)) -sin(A3)*cos(t(3)) a3*sin(t(3));...
    0 sin(A3) cos(A3) d3;...
    0 0 0 1];
T4=[cos(t(4)) -cos(A4)*sin(t(4)) sin(A4)*sin(t(4)) a4*cos(t(4));...
    sin(t(4)) cos(A4)*cos(t(4)) -sin(A4)*cos(t(4)) a4*sin(t(4));...
    0 sin(A4) cos(A4) d4;...
    0 0 0 1];
T5=[cos(t(5)) -cos(A5)*sin(t(5)) sin(A5)*sin(t(5)) a5*cos(t(5));...
    sin(t(5)) cos(A5)*cos(t(5)) -sin(A5)*cos(t(5)) a5*sin(t(5));...
    0 sin(A5) cos(A5) d5;...
    0 0 0 1];
T6=[cos(t(6)) -cos(A6)*sin(t(6)) sin(A6)*sin(t(6)) a6*cos(t(6));...
    sin(t(6)) cos(A6)*cos(t(6)) -sin(A6)*cos(t(6)) a6*sin(t(6));...
    0 sin(A6) cos(A6) d6;...
    0 0 0 1];

b=[1 0 0 0;0 1 0 0; 0 0 1 0; 0 0 0 1];
```



```
pto=a*b*rotx(180);  
ptt2=pto*T1*rotx(90);  
ptt3=ptt2*rotx(90)*T2;  
ptt4=ptt3*T3;  
ptt5=ptt4*T4;  
ptt6=ptt5*T5*roty(90);  
ptt7=ptt6*T6*rotx(90)*roty(90)*rotx(-15);
```

end

B.3 Programa “EcuacionesBipeDF”.- Este programa obtiene las ecuaciones de la cinemática directa del pie flotante con respecto a la cadera. (Programado en Matlab).

```
function [Pos]=EcuacionesBipeDF()  
  
t1=sym('t1');  
t2=sym('t2');  
t3=sym('t3');  
t4=sym('t4');  
t5=sym('t5');  
t6=sym('t6');  
a2=sym('a2');  
a3=sym('a3');  
a4=sym('a4');  
a5=sym('a5');  
a6=sym('a6');  
d1=sym('d1');  
df=sym('df');  
df1=sym('df1');  
df2=sym('df2');  
df3=sym('df3');  
F=sym('F');  
  
T1=[cos(t1) -sin(t1) 0 0;...  
    sin(t1)  cos(t1) 0 0;...  
    0 0 1 d1;...  
    0 0 0 1];  
T2=[cos(t2) 0  sin(t2) a2*cos(t2);...  
    sin(t2) 0 -cos(t2) a2*sin(t2);...  
    0 1 0 0;...  
    0 0 0 1];  
T3=[cos(t3) -sin(t3) 0 a3*cos(t3);...  
    sin(t3)  cos(t3) 0 a3*sin(t3);...  
    0 0 1 0;...  
    0 0 0 1];  
T4=[cos(t4) -sin(t4) 0 a4*cos(t4);...  
    sin(t4)  cos(t4) 0 a4*sin(t4);...  
    0 0 1 0;...  
    0 0 0 1];  
T5=[cos(t5) -sin(t5) 0 a5*cos(t5);...  
    sin(t5)  cos(t5) 0 a5*sin(t5);...  
    0 0 1 0;...  
    0 0 0 1];
```



```
0 0 0 1];
T6=[cos(t6) -sin(t6) 0 a6*cos(t6);...
    sin(t6)  cos(t6) 0 a6*sin(t6);...
    0 0 1 0;...
    0 0 0 1];

a=[1 0 0 df1;0 1 0 df2; 0 0 1 df3; 0 0 0 1];
b=[1 0 0 df;0 1 0 0; 0 0 1 0; 0 0 0 1];
c=[1 0 0 0;0 cos(F) -sin(F) 0;0 sin(F) cos(F) 0;0 0 0 1];

Pos=b*rotx(180)*T1*rotx(90)*rotz(90)*T2*T3*T4*T5*roty(90)*T6*rotx(90)*roty(90)*c;

end
```

B.4 Programa “EcuacionesBipeIA”.- Este programa obtiene las ecuaciones de la cinemática directa de la cadera con respecto al pie de apoyo. (Programado en Matlab).

```
function [Pos]=EcuacionesBipeIA()

t1=sym('t1');
t2=sym('t2');
t3=sym('t3');
t4=sym('t4');
t5=sym('t5');
t6=sym('t6');
a2=sym('a2');
a3=sym('a3');
a4=sym('a4');
a5=sym('a5');
a6=sym('a6');
d1=sym('d1');
df=sym('df');
df1=sym('df1');
df2=sym('df2');
df3=sym('df3');
F=sym('F');

T1=[cos(t1) -sin(t1) 0 0;...
    sin(t1)  cos(t1) 0 0;...
    0 0 1 d1;...
    0 0 0 1];

T2=[cos(t2) -sin(t2) 0 a2*cos(t2);...
    sin(t2)  cos(t2) 0 a2*sin(t2);...
    0 0 1 0;...
    0 0 0 1];

T3=[cos(t3) -sin(t3) 0 a3*cos(t3);...
    sin(t3)  cos(t3) 0 a3*sin(t3);...
    0 0 1 0;...
    0 0 0 1];

T4=[cos(t4) 0 sin(t4) a4*cos(t4);...
    sin(t4) 0 -cos(t4) a4*sin(t4);...
```



```
0 1 0 0;...
0 0 0 1];
T5=[cos(t5) -sin(t5) 0 a5*cos(t5);...
sin(t5) cos(t5) 0 a5*sin(t5);...
0 0 1 0;...
0 0 0 1];
T6=[cos(t6) sin(t6) 0 a6*cos(t6);...
sin(t6) -cos(t6) 0 a6*sin(t6);...
0 0 -1 0;...
0 0 0 1];

a=[1 0 0 df1;0 -1 0 df2; 0 0 -1 df3; 0 0 0 1];
b=[1 0 0 0;0 1 0 0; 0 0 1 df; 0 0 0 1];
c=[1 0 0 0;0 cos(F) -sin(F) 0;0 sin(F) cos(F) 0;0 0 0 1];

Pos=a*c*b*rotx(90)*T1*roty(90)*T2*T3*T4*T5*roty(-90)*rotz(90)*T6;

end
```

B.5 Programa “VELACELBPD”.- Este programa obtiene la solución de la velocidad y aceleración lineal del COG del bípedo Scout. (Programado en Matlab).

```
function [VPICx,VPICy,VPICz,APICx,APICy,APICz]=VELACELBPD(r,t,o)

Q=[0 -1 0 0;1 0 0 0;0 0 0 0;0 0 0 0];
va=[2 2 2 2 2 2];
aa=[0.01 0.01 0.01 0.01 0.01 0.01];
r=r*0.001;

A1=0;
A2=0;
A3=0;
A4=pi/2;
A5=0;
A6=pi;

a1=0;
a2=0.05729;
a3=0.07464;
a4=0.02908;
a5=0.05458;
a6=o;

d1=-0.05497;
d2=0;
d3=0;
d4=0;
d5=0;
d6=0;
```



```

T1=[cos(t(1)) -cos(A1)*sin(t(1)) sin(A1)*sin(t(1)) a1*cos(t(1));...
    sin(t(1)) cos(A1)*cos(t(1)) -sin(A1)*cos(t(1)) a1*sin(t(1));...
    0 sin(A1) cos(A1) d1;...
    0 0 0 1];
T2=[cos(t(2)) -cos(A2)*sin(t(2)) sin(A2)*sin(t(2)) a2*cos(t(2));...
    sin(t(2)) cos(A2)*cos(t(2)) -sin(A2)*cos(t(2)) a2*sin(t(2));...
    0 sin(A2) cos(A2) d2;...
    0 0 0 1];
T3=[cos(t(3)) -cos(A3)*sin(t(3)) sin(A3)*sin(t(3)) a3*cos(t(3));...
    sin(t(3)) cos(A3)*cos(t(3)) -sin(A3)*cos(t(3)) a3*sin(t(3));...
    0 sin(A3) cos(A3) d3;...
    0 0 0 1];
T4=[cos(t(4)) -cos(A4)*sin(t(4)) sin(A4)*sin(t(4)) a4*cos(t(4));...
    sin(t(4)) cos(A4)*cos(t(4)) -sin(A4)*cos(t(4)) a4*sin(t(4));...
    0 sin(A4) cos(A4) d4;...
    0 0 0 1];
T5=[cos(t(5)) -cos(A5)*sin(t(5)) sin(A5)*sin(t(5)) a5*cos(t(5));...
    sin(t(5)) cos(A5)*cos(t(5)) -sin(A5)*cos(t(5)) a5*sin(t(5));...
    0 sin(A5) cos(A5) d5;...
    0 0 0 1];
T6=[cos(t(6)) -cos(A6)*sin(t(6)) sin(A6)*sin(t(6)) a6*cos(t(6));...
    sin(t(6)) cos(A6)*cos(t(6)) -sin(A6)*cos(t(6)) a6*sin(t(6));...
    0 sin(A6) cos(A6) d6;...
    0 0 0 1];

```

```

a=[1 0 0 -o*0.001;0 -1 0 -0.0383578; 0 0 -1 -0.2347737; 0 0 0 1];
b=[1 0 0 0;0 1 0 0; 0 0 1 0.02343; 0 0 0 1];

```

```

pto=a*rotx(-165)*b*rotx(90);
ptt1=pto*T1*roty(90);
ptt2=ptt1*T2;
ptt3=ptt2*T3;
ptt4=ptt3*T4;
ptt5=ptt4*T5*roty(-90)*rotz(90);
ptt6=ptt5*T6;

```

```

VPIC=( (pto*Q*ptt1*ptt2*ptt3*ptt4*ptt5*ptt6*va(1)) + (pto*ptt1*Q*ptt2*ptt3*ptt4*ptt5*ptt6*va(
2)) + (pto*ptt1*ptt2*Q*ptt3*ptt4*ptt5*ptt6*va(3)) + (pto*ptt1*ptt2*ptt3*Q*ptt4*ptt5*ptt6*a(4))
+ (pto*ptt1*ptt2*ptt3*ptt4*Q*ptt5*ptt6*va(5)) + (pto*ptt1*ptt2*ptt3*ptt4*ptt5*Q*ptt6*va(6)) ) *
r;

```

```

VPICx=VPIC(1,1);
VPICy=VPIC(2,1);
VPICz=VPIC(3,1);

```

```

APIC=( (pto*Q*Q*ptt1*ptt2*ptt3*ptt4*ptt5*ptt6*va(1)*aa(1)) + (pto*ptt1*Q*Q*ptt2*ptt3*ptt4*ptt
5*ptt6*va(2)*aa(2)) + (pto*ptt1*ptt2*Q*Q*ptt3*ptt4*ptt5*ptt6*va(3)*aa(3)) + (pto*ptt1*ptt2*ptt
3*Q*Q*ptt4*ptt5*ptt6*va(4)*aa(4)) + (pto*ptt1*ptt2*ptt3*ptt4*Q*Q*ptt5*ptt6*va(5)*aa(5)) + (pto
*ptt1*ptt2*ptt3*ptt4*ptt5*Q*Q*ptt6*va(6)*aa(6)) ) *r;

```

```

APICx=APIC(1,1);
APICy=APIC(2,1);
APICz=APIC(3,1);

```

end



B.6 Programa “plotbipedo”.- Este programa dibuja al robot bípedo Scout de manera unifilar, dibujando también los sistemas de coordenadas de cada GDL. (Programado en Matlab).

```
function []=plotbipedo(a,pto,ptt2,ptt3,ptt4,ptt5,ptt6,ptt7)

grid on

line([-300 300],[0 0],[0 0],'color','r');
line([0 0],[-300 300],[0 0],'color','b');
line([0 0],[0 0],[-300 300],'color','g');

l=15;

line([0+a(1,4) (a(1,1)*1)+a(1,4)],[0+a(2,4) (a(2,1)*1)+a(2,4)],[0+a(3,4)
(a(3,1)*1)+a(3,4)], 'color','r', 'linewidth',3)
line([0+a(1,4) (a(1,2)*1)+a(1,4)],[0+a(2,4) (a(2,2)*1)+a(2,4)],[0+a(3,4)
(a(3,2)*1)+a(3,4)], 'color','b', 'linewidth',3)
line([0+a(1,4) (a(1,3)*1)+a(1,4)],[0+a(2,4) (a(2,3)*1)+a(2,4)],[0+a(3,4)
(a(3,3)*1)+a(3,4)], 'color','g', 'linewidth',3)

line([0+pto(1,4) (pto(1,1)*1)+pto(1,4)],[0+pto(2,4)
(pto(2,1)*1)+pto(2,4)],[0+pto(3,4) (pto(3,1)*1)+pto(3,4)], 'color','r', 'linewidth',3)
line([0+pto(1,4) (pto(1,2)*1)+pto(1,4)],[0+pto(2,4)
(pto(2,2)*1)+pto(2,4)],[0+pto(3,4) (pto(3,2)*1)+pto(3,4)], 'color','b', 'linewidth',3)
line([0+pto(1,4) (pto(1,3)*1)+pto(1,4)],[0+pto(2,4)
(pto(2,3)*1)+pto(2,4)],[0+pto(3,4) (pto(3,3)*1)+pto(3,4)], 'color','g', 'linewidth',3)

line([0+ptt2(1,4) (ptt2(1,1)*1)+ptt2(1,4)],[0+ptt2(2,4)
(ptt2(2,1)*1)+ptt2(2,4)],[0+ptt2(3,4) (ptt2(3,1)*1)+ptt2(3,4)], 'color','r', 'linewidth',3)
line([0+ptt2(1,4) (ptt2(1,2)*1)+ptt2(1,4)],[0+ptt2(2,4)
(ptt2(2,2)*1)+ptt2(2,4)],[0+ptt2(3,4) (ptt2(3,2)*1)+ptt2(3,4)], 'color','b', 'linewidth',3)
line([0+ptt2(1,4) (ptt2(1,3)*1)+ptt2(1,4)],[0+ptt2(2,4)
(ptt2(2,3)*1)+ptt2(2,4)],[0+ptt2(3,4) (ptt2(3,3)*1)+ptt2(3,4)], 'color','g', 'linewidth',3)

line([0+ptt3(1,4) (ptt3(1,1)*1)+ptt3(1,4)],[0+ptt3(2,4)
(ptt3(2,1)*1)+ptt3(2,4)],[0+ptt3(3,4) (ptt3(3,1)*1)+ptt3(3,4)], 'color','r', 'linewidth',3)
line([0+ptt3(1,4) (ptt3(1,2)*1)+ptt3(1,4)],[0+ptt3(2,4)
(ptt3(2,2)*1)+ptt3(2,4)],[0+ptt3(3,4) (ptt3(3,2)*1)+ptt3(3,4)], 'color','b', 'linewidth',3)
line([0+ptt3(1,4) (ptt3(1,3)*1)+ptt3(1,4)],[0+ptt3(2,4)
(ptt3(2,3)*1)+ptt3(2,4)],[0+ptt3(3,4) (ptt3(3,3)*1)+ptt3(3,4)], 'color','g', 'linewidth',3)

line([0+ptt4(1,4) (ptt4(1,1)*1)+ptt4(1,4)],[0+ptt4(2,4)
(ptt4(2,1)*1)+ptt4(2,4)],[0+ptt4(3,4) (ptt4(3,1)*1)+ptt4(3,4)], 'color','r', 'linewidth',3)
line([0+ptt4(1,4) (ptt4(1,2)*1)+ptt4(1,4)],[0+ptt4(2,4)
(ptt4(2,2)*1)+ptt4(2,4)],[0+ptt4(3,4) (ptt4(3,2)*1)+ptt4(3,4)], 'color','b', 'linewidth',3)
line([0+ptt4(1,4) (ptt4(1,3)*1)+ptt4(1,4)],[0+ptt4(2,4)
(ptt4(2,3)*1)+ptt4(2,4)],[0+ptt4(3,4) (ptt4(3,3)*1)+ptt4(3,4)], 'color','g', 'linewidth',3)

line([0+ptt5(1,4) (ptt5(1,1)*1)+ptt5(1,4)],[0+ptt5(2,4)
(ptt5(2,1)*1)+ptt5(2,4)],[0+ptt5(3,4) (ptt5(3,1)*1)+ptt5(3,4)], 'color','r', 'linewidth',3)
line([0+ptt5(1,4) (ptt5(1,2)*1)+ptt5(1,4)],[0+ptt5(2,4)
(ptt5(2,2)*1)+ptt5(2,4)],[0+ptt5(3,4) (ptt5(3,2)*1)+ptt5(3,4)], 'color','b', 'linewidth',3)
line([0+ptt5(1,4) (ptt5(1,3)*1)+ptt5(1,4)],[0+ptt5(2,4)
(ptt5(2,3)*1)+ptt5(2,4)],[0+ptt5(3,4) (ptt5(3,3)*1)+ptt5(3,4)], 'color','g', 'linewidth',3)
```



```
line([0+ptt6(1,4) (ptt6(1,1)*1)+ptt6(1,4)], [0+ptt6(2,4)
(ptt6(2,1)*1)+ptt6(2,4)], [0+ptt6(3,4) (ptt6(3,1)*1)+ptt6(3,4)], 'color', 'r', 'linewidth', 3)
line([0+ptt6(1,4) (ptt6(1,2)*1)+ptt6(1,4)], [0+ptt6(2,4)
(ptt6(2,2)*1)+ptt6(2,4)], [0+ptt6(3,4) (ptt6(3,2)*1)+ptt6(3,4)], 'color', 'b', 'linewidth', 3)
line([0+ptt6(1,4) (ptt6(1,3)*1)+ptt6(1,4)], [0+ptt6(2,4)
(ptt6(2,3)*1)+ptt6(2,4)], [0+ptt6(3,4) (ptt6(3,3)*1)+ptt6(3,4)], 'color', 'g', 'linewidth', 3)

line([0+ptt7(1,4) (ptt7(1,1)*1)+ptt7(1,4)], [0+ptt7(2,4)
(ptt7(2,1)*1)+ptt7(2,4)], [0+ptt7(3,4) (ptt7(3,1)*1)+ptt7(3,4)], 'color', 'r', 'linewidth', 3)
line([0+ptt7(1,4) (ptt7(1,2)*1)+ptt7(1,4)], [0+ptt7(2,4)
(ptt7(2,2)*1)+ptt7(2,4)], [0+ptt7(3,4) (ptt7(3,2)*1)+ptt7(3,4)], 'color', 'b', 'linewidth', 3)
line([0+ptt7(1,4) (ptt7(1,3)*1)+ptt7(1,4)], [0+ptt7(2,4)
(ptt7(2,3)*1)+ptt7(2,4)], [0+ptt7(3,4) (ptt7(3,3)*1)+ptt7(3,4)], 'color', 'g', 'linewidth', 3)

line([a(1,4) pto(1,4)], [a(2,4) pto(2,4)], [a(3,4)
pto(3,4)], 'color', 'k', 'linewidth', 3);%linea ab
line([pto(1,4) ptt2(1,4)], [pto(2,4) ptt2(2,4)], [pto(3,4)
ptt2(3,4)], 'color', 'k', 'linewidth', 3);%linea ab
line([ptt2(1,4) ptt3(1,4)], [ptt2(2,4) ptt3(2,4)], [ptt2(3,4)
ptt3(3,4)], 'color', 'k', 'linewidth', 3);%linea bc
line([ptt3(1,4) ptt4(1,4)], [ptt3(2,4) ptt4(2,4)], [ptt3(3,4)
ptt4(3,4)], 'color', 'k', 'linewidth', 3);%linea bc
line([ptt4(1,4) ptt5(1,4)], [ptt4(2,4) ptt5(2,4)], [ptt4(3,4)
ptt5(3,4)], 'color', 'k', 'linewidth', 3);%linea bc
line([ptt5(1,4) ptt6(1,4)], [ptt5(2,4) ptt6(2,4)], [ptt5(3,4)
ptt6(3,4)], 'color', 'k', 'linewidth', 3);%linea bc
line([ptt6(1,4) ptt7(1,4)], [ptt6(2,4) ptt7(2,4)], [ptt6(3,4)
ptt7(3,4)], 'color', 'k', 'linewidth', 3);%linea bc

end
```

B.7 Programa “ellipse”.- Este programa dibuja una elipse con el fin de poder observar como el bípedo si sigue una trayectoria elíptica. (Programado en Matlab).

```
function [xx,yy,zz]=ellipse(h,k,v,e)

o=-pi:0.001:pi;

for n=1:6284
    yy(1:n)=0;
end
xx=v*cos(o)+h;
zz=e*sin(o)+k;

end
```



B.8 Programa “rotx”.- Este programa indica una rotación en el eje “X” de cualquier sistema de coordenadas. (Programado en Matlab).

```
function [rx]=rotx(thetax)
rx=[1 0 0 0;0 cosd(thetax) -sind(thetax) 0;0 sind(thetax) cosd(thetax) 0;0 0 0 1];
end
```

B.9 Programa “roty”.- Este programa indica una rotación en el eje “Y” de cualquier sistema de coordenadas. (Programado en Matlab).

```
function [ry]=roty(thetay)
ry=[cosd(thetay) 0 sind(thetay) 0; 0 1 0 0; -sind(thetay) 0 cosd(thetay) 0;0 0 0 1];
end
```

B.10 Programa “rotz”.- Este programa indica una rotación en el eje “Z” de cualquier sistema de coordenadas. (Programado en Matlab).

```
function [rz]=rotz(thetaz)
rz=[cosd(thetaz) -sind(thetaz) 0 0; sind(thetaz) cosd(thetaz) 0 0; 0 0 1 0;0 0 0 1];
end
```



B.11 Programa “Torques”.- Este programa calcula el torque necesario de cada articulación para cada punto de la trayectoria propuesta anteriormente. (Programado en Matlab).

```
function [RA1,RA2,RA3,RA4,RA5,RA6]=torques(t,thp,thpp)

l=[0.05497 0.05729 0.07464 0.02908 0.05458 0.04454];
m=[0.1 0.1 0.1 0.1 0.1 0.1];
s=[-1(1)/2 -1(2)/2 -1(3)/2 -1(4)/2 -1(5)/2 -1(6)/2];
Z=[0 0 1]';

R1=[cos(t(1)) sin(t(1)) 0; sin(t(1)) -cos(t(1)) 0; 0 0 -1]';
R2=[cos(t(2)) 0 sin(t(2)); sin(t(2)) 0 -cos(t(2)); 0 1 0]';
R3=[cos(t(3)) 0 sin(t(3)); sin(t(3)) 0 -cos(t(3)); 0 1 0]';
R4=[cos(t(4)) -sin(t(4)) 0; sin(t(4)) cos(t(4)) 0; 0 0 1]';
R5=[cos(t(5)) -sin(t(5)) 0; sin(t(5)) cos(t(5)) 0; 0 0 1]';
R6=[cos(t(6)) 0 sin(t(6)); sin(t(6)) 0 -cos(t(6)); 0 1 0]';

g=9.8062;
RV1=[0 0 g]';
RV2=[0 g 0]';
RV3=[g 0 0]';
RV4=[g 0 0]';
RV5=[g 0 0]';
RV6=[0 0 0]';

OR6=R1'*R2'*R3'*R4'*R5'*R6';
OR5=R1'*R2'*R3'*R4'*R5';
OR4=R1'*R2'*R3'*R4';
OR3=R1'*R2'*R3';
OR2=R1'*R2';
OR1=R1';

RW1=R1*(Z*thp(1));
RW2=R2*(RW1+Z*thp(2));
RW3=R3*(RW2+Z*thp(3));
RW4=R4*(RW3+Z*thp(4));
RW5=R5*(RW4+Z*thp(5));
RW6=R6*(RW5+Z*thp(6));

RWp1=R1*(Z*thpp(1));
RWp2=R2*(RWp1+(Z*thpp(2)))+(RW1.*(Z*thp(2)));
RWp3=R3*(RWp2+(Z*thpp(3)))+(RW2.*(Z*thp(3)));
RWp4=R4*(RWp3+(Z*thpp(4)))+(RW3.*(Z*thp(4)));
RWp5=R5*(RWp4+(Z*thpp(5)))+(RW4.*(Z*thp(5)));
RWp6=R6*(RWp5+(Z*thpp(6)))+(RW5.*(Z*thp(6)));

RVp1=(RWp1.*[1(1) 0 0]')+(RW1.*(RW1.*[1(1) 0 0]'))+(OR1'*RV1);
RVp2=(RWp2.*[1(2) 0 0]')+(RW2.*(RW2.*[1(2) 0 0]'))+(OR2'*RV2);
RVp3=(RWp3.*[1(3) 0 0]')+(RW3.*(RW3.*[1(3) 0 0]'))+(OR3'*RV3);
RVp4=(RWp4.*[1(4) 0 0]')+(RW4.*(RW4.*[1(4) 0 0]'))+(OR4'*RV4);
RVp5=(RWp5.*[1(5) 0 0]')+(RW5.*(RW5.*[1(5) 0 0]'))+(OR5'*RV5);
RVp6=(RWp6.*[1(6) 0 0]')+(RW6.*(RW6.*[1(6) 0 0]'))+(OR6'*RV6);

RVp1=(RWp1.*[1(1) 0 0]')+(RW1.*(RW1.*[1(1) 0 0]'))+(R1*RV1);
```



$$\begin{aligned}
RVp2 &= (RWp2 \cdot [1(2) \ 0 \ 0]') + (RW2 \cdot (RW2 \cdot [1(2) \ 0 \ 0]')) + (R2 \cdot RVp1); \\
RVp3 &= (RWp3 \cdot [1(3) \ 0 \ 0]') + (RW3 \cdot (RW3 \cdot [1(3) \ 0 \ 0]')) + (R3 \cdot RVp2); \\
RVp4 &= (RWp4 \cdot [1(4) \ 0 \ 0]') + (RW4 \cdot (RW4 \cdot [1(4) \ 0 \ 0]')) + (R4 \cdot RVp3); \\
RVp5 &= (RWp5 \cdot [1(5) \ 0 \ 0]') + (RW5 \cdot (RW5 \cdot [1(5) \ 0 \ 0]')) + (R5 \cdot RVp4); \\
RVp6 &= (RWp6 \cdot [1(6) \ 0 \ 0]') + (RW6 \cdot (RW6 \cdot [1(6) \ 0 \ 0]')) + (R6 \cdot RVp5); \\
RVp7 &= (RWp7 \cdot [1(7) \ 0 \ 0]') + (RW7 \cdot (RW7 \cdot [1(7) \ 0 \ 0]')) + (R7 \cdot RVp6);
\end{aligned}$$

$$\begin{aligned}
RA1 &= (RWp1 \cdot [s(1) \ 0 \ 0]') + (RW1 \cdot (RW1 \cdot [s(1) \ 0 \ 0]')) + RVp1; \\
RA1 &= RA1(3, 1); \\
RA2 &= (RWp2 \cdot [s(2) \ 0 \ 0]') + (RW2 \cdot (RW2 \cdot [s(2) \ 0 \ 0]')) + RVp2; \\
RA2 &= RA2(3, 1); \\
RA3 &= (RWp3 \cdot [s(3) \ 0 \ 0]') + (RW3 \cdot (RW3 \cdot [s(3) \ 0 \ 0]')) + RVp3; \\
RA3 &= RA3(3, 1); \\
RA4 &= (RWp4 \cdot [s(4) \ 0 \ 0]') + (RW4 \cdot (RW4 \cdot [s(4) \ 0 \ 0]')) + RVp4; \\
RA4 &= RA4(3, 1); \\
RA5 &= (RWp5 \cdot [s(5) \ 0 \ 0]') + (RW5 \cdot (RW5 \cdot [s(5) \ 0 \ 0]')) + RVp5; \\
RA5 &= RA5(3, 1); \\
RA6 &= (RWp6 \cdot [s(6) \ 0 \ 0]') + (RW6 \cdot (RW6 \cdot [s(6) \ 0 \ 0]')) + RVp6; \\
RA6 &= RA6(3, 1);
\end{aligned}$$

$$\begin{aligned}
RF7 &= m(7) \cdot RA7; \\
RF6 &= (R7' \cdot RF7) + m(6) \cdot RA6; \\
RF5 &= (R6' \cdot RF6) + m(5) \cdot RA5; \\
RF4 &= (R5' \cdot RF5) + m(4) \cdot RA4; \\
RF3 &= (R4' \cdot RF4) + m(3) \cdot RA3; \\
RF2 &= (R3' \cdot RF3) + m(2) \cdot RA2; \\
RF1 &= (R2' \cdot RF2) + m(1) \cdot RA1;
\end{aligned}$$

$$\begin{aligned}
I1 &= [(m(1) \cdot 1(1)^2) / 3 \ 0 \ 0; 0 \ (m(1) \cdot 1(1)^2) / 3 \ 0; 0 \ 0 \ (m(1) \cdot 1(1)^2) / 3]; \\
I2 &= [(m(2) \cdot 1(2)^2) / 3 \ 0 \ 0; 0 \ (m(2) \cdot 1(2)^2) / 3 \ 0; 0 \ 0 \ (m(2) \cdot 1(2)^2) / 3]; \\
I3 &= [(m(3) \cdot 1(3)^2) / 3 \ 0 \ 0; 0 \ (m(3) \cdot 1(3)^2) / 3 \ 0; 0 \ 0 \ (m(3) \cdot 1(3)^2) / 3]; \\
I4 &= [(m(4) \cdot 1(4)^2) / 3 \ 0 \ 0; 0 \ (m(4) \cdot 1(4)^2) / 3 \ 0; 0 \ 0 \ (m(4) \cdot 1(4)^2) / 3]; \\
I5 &= [(m(5) \cdot 1(5)^2) / 3 \ 0 \ 0; 0 \ (m(5) \cdot 1(5)^2) / 3 \ 0; 0 \ 0 \ (m(5) \cdot 1(5)^2) / 3]; \\
I6 &= [(m(6) \cdot 1(6)^2) / 3 \ 0 \ 0; 0 \ (m(6) \cdot 1(6)^2) / 3 \ 0; 0 \ 0 \ (m(6) \cdot 1(6)^2) / 3]; \\
I7 &= [(m(7) \cdot 1(7)^2) / 3 \ 0 \ 0; 0 \ (m(7) \cdot 1(7)^2) / 3 \ 0; 0 \ 0 \ (m(7) \cdot 1(7)^2) / 3];
\end{aligned}$$

$$\begin{aligned}
N7 &= (OR7' \cdot I7 \cdot OR7 \cdot RWp7) + (RW7 \cdot (OR7' \cdot I7 \cdot OR7 \cdot RW7)); \\
N6 &= (OR6' \cdot I6 \cdot OR6 \cdot RWp6) + (RW6 \cdot (OR6' \cdot I6 \cdot OR6 \cdot RW6)); \\
N5 &= (OR5' \cdot I5 \cdot OR5 \cdot RWp5) + (RW5 \cdot (OR5' \cdot I5 \cdot OR5 \cdot RW5)); \\
N4 &= (OR4' \cdot I4 \cdot OR4 \cdot RWp4) + (RW4 \cdot (OR4' \cdot I4 \cdot OR4 \cdot RW4)); \\
N3 &= (OR3' \cdot I3 \cdot OR3 \cdot RWp3) + (RW3 \cdot (OR3' \cdot I3 \cdot OR3 \cdot RW3)); \\
N2 &= (OR2' \cdot I2 \cdot OR2 \cdot RWp2) + (RW2 \cdot (OR2' \cdot I2 \cdot OR2 \cdot RW2)); \\
N1 &= (OR1' \cdot I1 \cdot OR1 \cdot RWp1) + (RW1 \cdot (OR1' \cdot I1 \cdot OR1 \cdot RW1));
\end{aligned}$$

$$\begin{aligned}
RN7 &= ([1(7) \ 1(7) \ 1(7)]' + [s(7) \ s(7) \ s(7)]') \cdot (m(7) \cdot RA7) + N7; \\
RN6 &= (R6' \cdot RN7) + ([1(6) \ 1(6) \ 1(6)]' \cdot (R6 \cdot RF7)) + ([1(6) \ 1(6) \ 1(6)]' + [s(6) \ s(6) \ s(6)]') \cdot (m(6) \cdot RA6) + N6; \\
RN5 &= (R5' \cdot RN6) + ([1(5) \ 1(5) \ 1(5)]' \cdot (R5 \cdot RF6)) + ([1(5) \ 1(5) \ 1(5)]' + [s(5) \ s(5) \ s(5)]') \cdot (m(5) \cdot RA5) + N5; \\
RN4 &= (R4' \cdot RN5) + ([1(4) \ 1(4) \ 1(4)]' \cdot (R4 \cdot RF5)) + ([1(4) \ 1(4) \ 1(4)]' + [s(4) \ s(4) \ s(4)]') \cdot (m(4) \cdot RA4) + N4; \\
RN3 &= (R3' \cdot RN4) + ([1(3) \ 1(3) \ 1(3)]' \cdot (R3 \cdot RF4)) + ([1(3) \ 1(3) \ 1(3)]' + [s(3) \ s(3) \ s(3)]') \cdot (m(3) \cdot RA3) + N3; \\
RN2 &= (R2' \cdot RN3) + ([1(2) \ 1(2) \ 1(2)]' \cdot (R2 \cdot RF3)) + ([1(2) \ 1(2) \ 1(2)]' + [s(2) \ s(2) \ s(2)]') \cdot (m(2) \cdot RA2) + N2; \\
RN1 &= (R1' \cdot RN2) + ([1(1) \ 1(1) \ 1(1)]' \cdot (R1 \cdot RF2)) + ([1(1) \ 1(1) \ 1(1)]' + [s(1) \ s(1) \ s(1)]') \cdot (m(1) \cdot RA1) + N1;
\end{aligned}$$



```
RN7=([l(7) l(7) l(7)]'+[s(7) s(7)
s(7)]') .* (m(7) .*RA7)+(((1/12)*m(7)*l(7)^2)*(thpp(1)+thpp(2)+thpp(3)+thpp(4)+thpp(5)+thpp(6)+thpp(7)));
RN6=(R6'*RN7)+([l(6) l(6) l(6)]' .* (R6*RF7))+([l(6) l(6) l(6)]'+[s(6) s(6)
s(6)]') .* (m(6) .*RA6)+(((1/12)*m(6)*l(6)^2)*(thpp(1)+thpp(2)+thpp(3)+thpp(4)+thpp(5)+thpp(6)));
RN5=(R5'*RN6)+([l(5) l(5) l(5)]' .* (R5*RF6))+([l(5) l(5) l(5)]'+[s(5) s(5)
s(5)]') .* (m(5) .*RA5)+(((1/12)*m(5)*l(5)^2)*(thpp(1)+thpp(2)+thpp(3)+thpp(4)+thpp(5)));
RN4=(R4'*RN5)+([l(4) l(4) l(4)]' .* (R4*RF5))+([l(4) l(4) l(4)]'+[s(4) s(4)
s(4)]') .* (m(4) .*RA4)+(((1/12)*m(4)*l(4)^2)*(thpp(1)+thpp(2)+thpp(3)+thpp(4)));
RN3=(R3'*RN4)+([l(3) l(3) l(3)]' .* (R3*RF4))+([l(3) l(3) l(3)]'+[s(3) s(3)
s(3)]') .* (m(3) .*RA3)+(((1/12)*m(3)*l(3)^2)*(thpp(1)+thpp(2)+thpp(3)));
RN2=(R2'*RN3)+([l(2) l(2) l(2)]' .* (R2*RF3))+([l(2) l(2) l(2)]'+[s(2) s(2)
s(2)]') .* (m(2) .*RA2)+(((1/12)*m(2)*l(2)^2)*(thpp(1)+thpp(2)));
RN1=(R1'*RN2)+([l(1) l(1) l(1)]' .* (R1*RF2))+([l(1) l(1) l(1)]'+[s(1) s(1)
s(1)]') .* (m(1) .*RA1)+(((1/12)*m(1)*l(1)^2)*(thpp(1)));
```

```
T(7)=(RN7')*(R7*Z);
T(6)=(RN6')*(R6*Z);
T(5)=(RN5')*(R5*Z);
T(4)=(RN4')*(R4*Z);
T(3)=(RN3')*(R3*Z);
T(2)=(RN2')*(R2*Z);
T(1)=(RN1')*(R1*Z);
```

end



Anexo C



C.1 Algebra de matrices

Una matriz A (o $A_{m \times n}$) de orden m por n es una ordenación rectangular de números reales o complejos (llamados elementos) dispuestos en m filas y n columnas, esto se puede apreciar más claramente en la ecuación M.1

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (\text{M.1})$$

Mientras que no especifiquemos lo contrario, supondremos que A es una matriz real. Las matrices que sólo tienen una columna o una fila se tratan como vectores.

La transpuesta de una matriz A , escrita A^T , se define como la matriz cuyo número de columnas es idéntico al número de filas de A como se observa en la ecuación M.2.

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{1n} & a_{2n} & \dots & a_{nm} \end{bmatrix} \quad (\text{M.2})$$

En particular, la transpuesta de una matriz columna es una matriz fila y viceversa.

C.2 Suma de matrices

Dos matrices A y B del mismo orden se pueden sumar (o restar) obteniéndose una matriz C del mismo orden (ecuación M.3) sumando sus elementos correspondientes.

$$A + B = C \quad \text{o} \quad a_{ij} + b_{ij} = c_{ij} \quad \text{para todo } i, j \quad (\text{M.3})$$

C.3 Multiplicación de matrices

El producto de un escalar por una matriz se lleva a cabo multiplicando todos los elementos de A por él escalar como se aprecia en M.4.



$$kA = Ak = [ka_{ij}] \quad (M.4)$$

Dos matrices pueden multiplicarse solamente si son conformables. Es decir si $AB = C$, el número de columna de A deberá ser igual al número de fila de B y la matriz resultante C tendrá los números de fila y columna iguales a los de A y B , respectivamente (ecuación M.5).

$$(A_{m \times n})(B_{n \times p}) = C_{m \times p} \quad o \quad c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \quad (M.5)$$

El producto de matrices no es conmutativo aun si las matrices son multiplicables. A y B son matrices cuadradas de orden n , en general como se ve en la ecuación M.6.

$$AB \neq BA \quad (M.6)$$

C.4 Matrices adjuntas e inversas

Si A es una matriz cuadrada y A_{ij} es el adjunto de a_{ij} en $|A|$, la transpuesta de la matriz formada por los adjuntos A_{ij} se llama matriz adjunta A (ecuación M.7).

$$[A_{ij}]^T = [A_{ji}] \quad i, j = 1, 2, \dots, n \quad (M.7)$$

A veces la matriz adjunta de A se escribe $\text{adj } A$.

La inversa de una matriz cuadrada no singular A , A^{-1} , es la matriz adjunta de A dividida por el determinante de A , esto se aprecia mejor en la ecuación M.8.

$$A^{-1} = \frac{[A_{ij}]^T}{|A|} = \frac{\text{adj } A}{|A|} \quad (M.8)$$

En general, una matriz 2×2

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$



Tiene una matriz inversa

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$$

C.5 Traza de una matriz

La traza de una matriz cuadrada A de orden n es la suma de los elementos de su diagonal principal, esto es representado en la ecuación M.9.

$$\text{Traza de } A \equiv \text{Tr}(A) = \sum_{i=1}^n a_{ii} \quad (\text{M.9})$$



Bibliografía



-
- (H. Kazerooni, 2005), “*Exoskeletons for human power augmentation,*” in *Proc. IEEE Conf. Intelligent Robots and Systems*, Edmonton, Canada, pp. 3459–3464.
 - (Chevallereau et al (2009)) C. Chevallereau, Guy Bessonnet, G. Abba, and Y. Aoustin. *Bipedal Robots: Modeling, Design and Walking Synthesis*. John Wiley & Sons, Gran Bretaña, 2009.
 - (Kajita et al 2008) Shuuji Kajita and Bernard Espiau. Legged robots. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 361–378. Springer, EUA.
 - (Chevallereau et al (2007)) Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. Control and Automation. CRC Press, EUA, 2007.
 - (Fu, et al, 1988). K.S.Fu, R.C.Gonzalez, C.S.G.Lee, “*Robotica, Control, Detección, Visión e Inteligencia*”, McGraw-Hill, ISBN 0-07-022625-3.
 - (Narong and Manukid, 2009) School of Engineering and Technology, “Balancing Control of AITleg Exoskeleton Using ZMP based FLC “, *International Journal of Advanced Robotic Systems*, Vol. 6, No. 4.
 - (Kajita et al 2005) Shuuji Kajita, Mitsuharu Morisawa, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, Kiyoshi Fujiwara, and Hirohisa Hirukawa. Biped walking pattern generator allowing auxiliary zmp control. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2993–2999, Octubre 2006.
 - (Westervelt, Grizzle, Chevallereau & Ho Choi, 2007) Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. Control and Automation. CRC Press, EUA.
 - (Larriva and Vele, 2006) Ing. José Larriva Vásquez, Ing. Oscar Vele Guallpa, “*Estudio, Diseño y Contrucción de un Robot Bipedo Experimental: Analisis y Control*”, Universidad Politécnica Salesiana 2006.
 - (Dollar & Herr, 2008) Dollar & Herr, *Lower extremities exoeskeleton and active orthosis: Challenges and State- of-the-Art.*, IEEE, 2008.



-
- (Igual, 1996) Igual, Muñoz & Aramburu, Fisioterapia General: Cinesiterapia, Primera edición, España, 1996.
 - (Honda-Worldwide, 2010) Honda-Worldwide. Asimo, Marzo 2010. URL [Http://hunolab.co.kr/HUBOFX-1_Specification.php](http://hunolab.co.kr/HUBOFX-1_Specification.php).
 - (Kawada-Industries-Inc., 2009) Kawada-Industries-Inc. Humanoid robot hrp-3 promet mk-ii, Noviembre 2009. URL <http://global.kawada.jp/mechatronics/hrp3.html>.
 - (Spong et al., 2006) Mark W. Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. Robot Modeling and Control, John Wiley and Sons, EUA.
 - (Vukobratovic et al (1975)) M. Vukobratovic, B. Borovac, D. Surla and D. Stokic. *Biped Locomotion: Dynamics, Stability, Control and application*. Springer Verlag.
 - (Narváez, 2010) Octavio Narváez Aroche, Modelos Cinemático y Dinámico de un Robot Bípedo de Doce grados de Libertad Internos, UNAM.