

UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

Control basado en el ZMP usando bases de Groebner para la marcha de un robot humanoide.

T E S I S

Para obtener el grado en:

MAESTRO EN ROBÓTICA

Presenta:

Ing. Rocío Ambrocio Delgado

Director de Tesis:

Dr. Alberto Elías Petrilli Barceló

Co-Director de Tesis:

Dra. Esther Lugo González

Huajuapán de León, Oaxaca, México, Marzo de 2019

Tesis presentada en Marzo de 2019
ante los siguientes sinodales:

Dr. José Anibal Arias Aguilar.
Dr. Rosebet Miranda Luna.
Dr. Fermín Hugo Ramírez Leyva.
Dr. Heriberto Casarrubias Vargas.

Director de Tesis:
Dr. Alberto Elías Petrilli Barceló.

Co-Director de Tesis:
Dra. Esther Lugo González.

*A Lidia,
Con todo mi amor y cariño,
por que sin ti esto no hubiera sido posible,
¡Te amo, mamá!*

Agradecimientos

Gracias a Dios por darme la gran oportunidad de concluir mis estudios de maestría y por poner en mi camino personas maravillosas que, directa o indirecta participaron con sus valiosos consejos y apoyo en el desarrollo de este trabajo de tesis.

A la señora Lidia, por darme la oportunidad de seguir estudiando. Por todos los consejos que me has dado durante todo este tiempo, tu gran paciencia y tu amor incondicional. Por todo esto y muchas cosas más, gracias mamá.

A mi hermana Lizbeth, por que sin tu consejo no me hubiera animado a realizar la gran aventura que fue estudiar un posgrado. Gracias por estar a mi lado y apoyarme en mis estudios.

Gracias a mis hermanos Teresita de Jesús, Eduardo y Jatziry, por compartir maravillosos momentos llenos de alegrías. Gracias a mis sobrinos Hannia, Miguel y Juan por su amor incondicional y darme ánimos cuando más lo necesito.

A Francisco Javier Espinosa García, por todas las enseñanzas que me has dado. Te agradezco infinitamente toda la paciencia, los consejos y por todos los momentos que hemos compartido juntos. Aunque no nos hemos conocido por mucho tiempo, has dejado una huella muy grande en mi vida.

A mis directores de tesis, el Dr. Alberto Elías Petrilli Barceló y la Dra. Esther Lugo González por todo el tiempo y dedicación que han puesto a la realización de este trabajo. A los sinodales, el Dr. José Aníbal Arias Aguilar, el Dr. Rosebet Mirando Luna, el Dr. Fermín Hugo Ramirez Leyva y el Dr. Heriberto Casarrubias Vargas por todas las recomendaciones y exigencias para que el trabajo fuera excelente.

A mi amigo Levid por su apoyo, gran amistad y por todos los momentos que hemos vivido juntos. A los chicos que están en el laboratorio, Magdiel, Omar, Roberto, Daniel y Oscar por su ayuda y consejos durante el tiempo que estuve en la maestría, el tiempo compartido y sobre todo por su amistad.

A la Universidad Tecnológica de la Mixteca por darme la oportunidad de realizar mis estudios de maestría y realizar este trabajo de tesis. A todos mis profesores de maestría, por su enseñanza, dedicación y sabiduría. Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el apoyo económico brindado para mis estudios de maestría.

A Todos
¡GRACIAS!

Índice

Agradecimientos	VII
Resumen	1
1. Introducción	3
1.1. Estado del Arte	3
1.2. Planteamiento del problema	9
1.3. Motivación	10
1.4. Hipótesis	10
1.5. Objetivos	10
1.5.1. Objetivo general	10
1.5.2. Objetivos específicos	10
1.6. Metas	11
1.7. Metodología	11
1.7.1. Análisis de requerimientos	12
1.7.2. Obtención de la información	12
1.7.3. Desarrollo del algoritmo de marcha	12
1.7.4. Implementación	12
1.7.5. Pruebas	12
1.7.6. Resultados	12
2. Marco Teórico	13
2.1. Cinemática	13
2.1.1. Convención de Denavit-Hertenberg	13
2.1.2. Método geométrico	14
2.2. Dinámica de mecanismos	15
2.3. Locomoción bípeda	16
2.4. Criterios de estabilidad	18
2.4.1. Punto de momento cero (ZMP)	18
2.4.2. Modo de péndulo invertido linealizado 3D y Punto de momento cero . .	20
2.5. Control difuso	21
2.6. El robot Nao	22

2.6.1. Brazo izquierdo	23
2.6.2. Brazo derecho	25
2.6.3. Pierna izquierda	25
2.6.4. Pierna derecha	26
3. Teoría de las bases de Groebner	27
3.1. Conceptos preliminares	27
3.1.1. Conjuntos	27
3.1.2. Anillos	28
3.1.3. Ideal	29
3.1.4. Polinomios	30
3.1.5. Órdenes monomiales	31
3.2. Bases de Groebner	32
3.2.1. Algoritmo de Buchberger	34
4. Desarrollo del algoritmo de marcha	37
4.1. Cinemática inversa estimada con Bases de Groebner	37
4.1.1. Brazos del robot Nao	38
4.1.2. Piernas del robot Nao	42
4.2. Generador del patrón de marcha usando el criterio de estabilidad ZMP	46
4.2.1. Generación de trayectorias	47
4.3. Control de posición del robot cuando está en reposo	49
4.4. Control de posición de la cadera en el ciclo de marcha	50
4.5. Control del equilibrio del robot Nao en el ciclo de marcha	53
5. Pruebas y resultados	57
5.1. Control de posición del robot cuando está en reposo	59
5.2. Control de posición de la cadera en el ciclo de marcha	66
5.3. Control del equilibrio del robot Nao en el ciclo de marcha	68
5.4. Implementación del algoritmo de marcha en el robot Nao	71
6. Conclusiones y Trabajos Futuros	73
6.1. Conclusiones	73
6.2. Trabajos Futuros	74
Bibliografía	75
A. Masas del robot Nao	79
B. Artículo	93
C. Cinemática inversa de un manipulador de dos grados de libertad	113

Índice de figuras

1.1.	Ejemplos de robots bípedos [1]	4
1.2.	Robot WABIAN-2R desarrollado por la Universidad de Waseda [2].	5
1.3.	Robot HRP-2 desarrollado por Kawada Industries [3].	6
1.4.	Primer prototipo del bípedo ESBiRRo [4].	6
1.5.	Robot humanoide COMAN [5].	7
1.6.	Robot humanoide Bioloid [6].	7
2.1.	Representación del modelo de un manipulador planar de 3 GDL [7].	15
2.2.	Fases y subfases de la marcha humana [8].	16
2.3.	Fases principales de la marcha humana [9].	17
2.4.	Modelo que representa la marcha humana basado en el péndulo invertido [10].	17
2.5.	Análisis de marcha [11].	17
2.6.	Planos y ejes de movimiento del ser humano [12].	18
2.7.	Definición del Punto de Momento Cero (ZMP) [1].	19
2.8.	Polígono de soporte [1].	19
2.9.	Representación del péndulo invertido en el área restringida [13].	21
2.10.	Cadenas cinemáticas y articulaciones del robot Nao [14].	23
2.11.	Posición cero y la coordenada base del robot Nao [14].	24
4.1.	Cadena para calcular la cinemática inversa de los brazos del robot Nao.	38
4.2.	Ángulo θ_3 del brazo izquierdo del robot Nao [14].	39
4.3.	Cadena cinemática de los brazos del robot Nao.	40
4.4.	Ecuaciones propuestas para calcular la cinemática inversa de los brazos del robot Nao.	40
4.5.	Ángulo θ_1 del brazo del robot Nao.	42
4.6.	Cadena cinemática para calcular la cinemática inversa de las piernas del robot Nao.	43
4.7.	Cadena cinemática de las piernas del robot Nao visto como eslabones.	44
4.8.	Ecuaciones propuestas para la estimación de la cinemática inversa de las piernas del robot Nao.	45
4.9.	Ángulo θ_6 de la pierna del robot Nao.	46
4.10.	Ángulo θ_5 de la pierna del robot Nao.	46
4.11.	Ejes de referencia del robot Nao.	47

4.12. Sincronización de los pies del robot durante la caminata siguiendo la trayectoria ZMP en y	48
4.13. Elevación del pie izquierdo del Nao.	48
4.14. Elevación del pie derecho del Nao.	49
4.15. Movimiento del brazo derecho para corregir estabilidad del robot.	54
4.16. Movimiento de las piernas y brazo izquierdo para corregir estabilidad del robot.	54
4.17. Posición final después de corregir la estabilidad del robot.	55
5.1. Ciclo de marcha del robot Nao simulado en WEBOTS.	58
5.2. Gráficas donde se muestra el comportamiento del control difuso en la cadera del robot en el plano sagital mientras está en reposo.	60
5.3. Gráficas donde se muestra el comportamiento del control difuso de las rodillas del robot en el plano sagital mientras está en reposo.	61
5.4. Gráficas donde se muestra el comportamiento del control difuso en los tobillos del robot en el plano sagital mientras está en reposo.	62
5.5. Gráficas donde se muestra el comportamiento del control difuso en la cadera del robot en el plano frontal mientras está en reposo.	63
5.6. Gráficas donde se muestra el comportamiento del control difuso en los tobillos del robot en el plano frontal mientras está en reposo.	64
5.7. Gráficas donde se muestra el comportamiento del control difuso cuando el robot se encuentra inclinado mientras está en reposo.	65
5.8. Gráficas donde se muestra el comportamiento del control difuso en la trayectoria ZMP en x	66
5.9. Gráficas donde se muestra el comportamiento del control difuso en la trayectoria ZMP en y	67
5.10. Posición x del giroscopio en el robot Nao mientras camina.	68
5.11. Movimiento del tobillo derecho para corregir la estabilidad del robot.	69
5.12. Movimiento del tobillo izquierdo para corregir la estabilidad del robot.	69
5.13. Movimiento del tobillo derecho en el robot mientras está en la fase de balanceo.	70
5.14. Movimiento del tobillo izquierdo en el robot mientras está en la fase de balanceo.	70
5.15. Ciclo de marcha implementado en el robot Nao.	71
A.1. Centro de masas y masa del torso del robot Nao [14].	80
A.2. Centro de masas y masa del cuello del robot Nao [14].	80
A.3. Centro de masas y masa de la cabeza del robot Nao [14].	81
A.4. Centro de masas y masa del hombro derecho del robot Nao [14].	81
A.5. Centro de masas y masa del hombro izquierdo del robot Nao [14].	82
A.6. Centro de masas y masa del brazo derecho del robot Nao [14].	82
A.7. Centro de masas y masa del brazo izquierdo del robot Nao [14].	83
A.8. Centro de masas y masa del codo derecho del robot Nao [14].	83
A.9. Centro de masas y masa del codo izquierdo del robot Nao [14].	83
A.10. Centro de masas y masa del antebrazo derecho del robot Nao [14].	84

A.11. Centro de masas y masa del antebrazo izquierdo del robot Nao [14].	84
A.12. Centro de masas y masa de la mano derecha del robot Nao [14].	85
A.13. Centro de masas y masa de la mano izquierda del robot Nao [14].	85
A.14. Centro de masas y masa de la pelvis derecha del robot Nao [14].	86
A.15. Centro de masas y masa de la pelvis izquierda del robot Nao [14].	86
A.16. Centro de masas y masa de la cadera derecha del robot Nao [14].	87
A.17. Centro de masas y masa de la cadera izquierda del robot Nao [14].	87
A.18. Centro de masas y masa del muslo derecho del robot Nao [14].	88
A.19. Centro de masas y masa del muslo izquierdo del robot Nao [14].	88
A.20. Centro de masas y masa de la tibia derecha del robot Nao [14].	89
A.21. Centro de masas y masa de la tibia izquierda del robot Nao [14].	89
A.22. Centro de masas y masa del tobillo derecho del robot Nao [14].	90
A.23. Centro de masas y masa del tobillo derecho del robot Nao [14].	90
A.24. Centro de masas y masa del pie derecho del robot Nao [14].	91
A.25. Centro de masas y masa del pie izquierdo del robot Nao [14].	91
C.1. Manipulador de dos grados de libertad.	113
C.2. Manipulador de dos grados de libertad.	114

Índice de Tablas

1.1. Métodos para calcular la cinemática [15].	8
2.1. Dimensiones del robot Nao.	24
2.2. Parámetros de Denavit-Hartenberg del brazo izquierdo del robot Nao.	25
2.3. Parámetros de Denavit-Hartenberg del brazo derecho del robot Nao.	25
2.4. Parámetros de Denavit-Hartenberg de la pierna izquierda del robot Nao.	26
2.5. Parámetros de Denavit-Hartenberg de la pierna derecha del robot Nao.	26

Resumen

En este trabajo se presentan el desarrollo y validación de un algoritmo de marcha para el robot Nao de Aldebaran Robotics. Se inicia por la introducción, en la cual se menciona el estado del arte referente a la aplicación de algoritmos para tener la estabilidad y equilibrio en la marcha de los robots humanoides. En el planteamiento del problema y la motivación, se describe la problemática que se tiene en los algoritmos de marcha sobre pasto artificial y como es que se tomó este tema en consideración. Con lo anterior se plantea la hipótesis, el objetivo general y los objetivos específicos.

Para lograr que el robot camine de una forma considerada estable se propuso hacer uso de la teoría de las bases de Groebner, el criterio de estabilidad ZMP y el control difuso en el diseño del algoritmo de marcha. La metodología que se siguió para realizar este trabajo es en cascada.

El algoritmo de marcha inicia calculando la cinemática inversa con la teoría de las bases de Groebner a partir de una trayectoria ZMP propuesta. Además, cuenta con 4 controladores difusos: el que verifica que esté en una posición adecuada antes de dar un paso, el que estabiliza al robot cuando este tiene muchas oscilaciones, el que verifica la trayectoria ZMP y el que controla la posición del tobillo. Todo esto se simuló en el software Webots y se implementó en el robot Nao validando el algoritmo marcha. Las pruebas se realizaron sobre pasto artificial y sobre una superficie plana, demostrándose que el robot mantuvo una marcha considerada estable. Por último, se presentan las conclusiones y trabajos futuros.

Capítulo 1

Introducción

Los robots que han sido diseñados para tener el aspecto físico y movimientos muy parecidos al del ser humano son llamados humanoides. Estos robots tienen características antropomórficas con las cuales son capaces de interactuar con humanos, alcanzar objetos guiados por la visión, tener una marcha bípeda o utilizar herramientas.

En los últimos años, la investigación sobre la marcha de los robots tipo humanoide ha cobrado gran importancia entre los investigadores y la industria, por ejemplo el robot HRP-2 desarrollado por Kawada Industries [3] y ASIMO de HONDA^R [16] que pueden caminar sobre superficies irregulares.

Las investigaciones se han centrado en que un robot de tipo humanoide sea capaz de realizar caminatas largas y estables similares a la caminata humana, así como conocer la relación que existe entre el tobillo y el pie del robot. Su enfoque no solo es en robots en tamaño real sino también en robots a escala, por ejemplo, mRobo de TOSY Robotics [17], Nao de Aldebaran robotics [18] y Bioloid de ROBOTICS [6], por mencionar algunos.

La primera organización que usó la investigación de los robots de tipo humanoide con fines educativos usando el fútbol soccer fué la RoboCup. El objetivo de la RoboCup es, para mediados del siglo XXI, desarrollar un equipo de robots humanoides completamente autónomos que puedan ganar, de acuerdo con las reglas oficiales de la FIFA, contra el ganador de la copa mundial de fútbol de ese año [19]. Actualmente las competencias de fútbol se hacen con los robots Nao de Aldebaran robotics.

1.1. Estado del Arte

A través del tiempo el ser humano ha tenido el deseo de crear máquinas que sean muy similares a él. Por esta razón se han creado infinidad de máquinas llamadas robots. A las máquinas que más se parecen al ser humano se les conoce como robots humanoides.

El primer robot considerado humanoide fue el WABOT-1 [20], mostrado en la figura 1.1a, diseñado por Ichiro Kato de la Universidad de Waseda en 1973. WABOT-1 puede reconocer objetos mediante visión artificial, entiende el lenguaje hablado, habla con una voz artificial, manipula objetos con sus dos manos y puede caminar en dos piernas, aunque todavía es un sistema primitivo, es el primer robot que se consideró humanoide. En 1984, se da a conocer al WABOT-2 [20], el cual toca el piano, también diseñado por Ichiro Kato como se muestra en la figura 1.1b. El WABOT-2 fue el primer paso en el desarrollo de un robot personal".

Pasados varios años del lanzamiento del WABOT-2, se da a conocer el robot P2 en 1996, mostrado en la figura 1.1c, con una altura de 180 cm y un peso de 210 kg, fue el primer robot humanoide que realizó una marcha con suficiente estabilidad teniendo en su cuerpo una computadora y la batería de alimentación. En 1997, Honda da a conocer el P3, mostrado en la figura 1.1d, [1].

En el 2000 se da a conocer a ASIMO [16] mostrado en la figura 1.1e, con una altura de 120 cm y un peso de 43 kg. ASIMO es considerado el primer robot humanoide que pudo interactuar directamente con los humanos. El ASIMO presentado en 2011 es capaz de tomar decisiones de forma autónoma y cambiar su comportamiento dependiendo del entorno en el que se encuentre. Puede correr, brincar en una o dos piernas, correr en reversa y caminar sobre superficies irregulares.

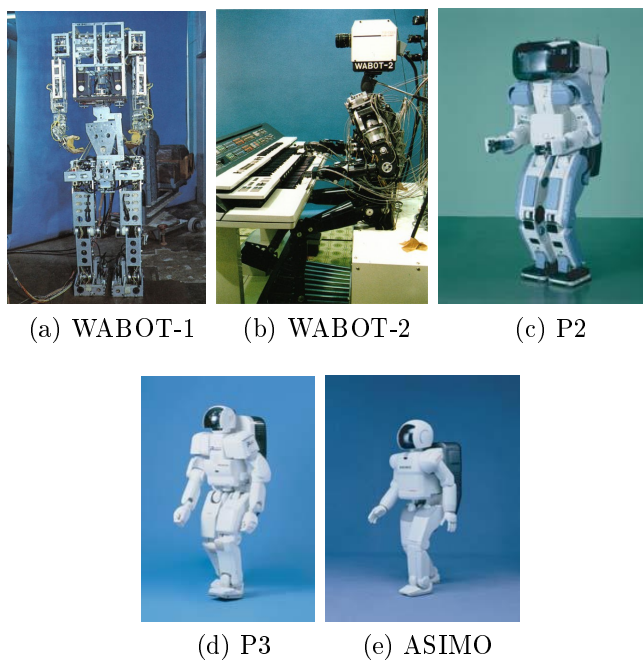


Figura 1.1: Ejemplos de robots bípedos [1]

En el 2008, Kondo hace simulaciones de la marcha de un robot humanoide bípedo para ayudar a las personas de la tercera edad o con discapacidad motora. La simulación se hizo

en el robot WABIAN-2R (WAseda BIpedal humANoid No. 2 Refined) mostrado en la figura 1.2, el cual puede caminar con las rodillas estiradas, tener contacto de talón y movimiento de los dedos del pie [21] como lo hace un humano. El robot WABIAN-2R tiene 6 GDL en cada pierna, 1 GDL pasivo en cada pie, 2 GDL en la cintura, 2 GDL en el tronco, 7 GDL en cada brazo, 3 GDL en cada mano y 3 GDL en el cuello. Las dimensiones de los eslabones están basados en una mujer adulta.

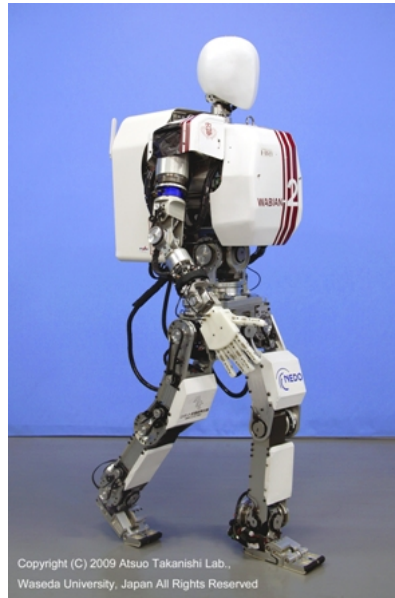


Figura 1.2: Robot WABIAN-2R desarrollado por la Universidad de Waseda [2].

Otro ejemplo es HRP-2 con una altura de 154 cm y un peso de 58 kg desarrollado por Kawada Industries mostrado en la figura 1.3, en el cual se ha experimentado que camine sobre superficies irregulares y que se levante del piso en caso de caída [3].

Se han propuesto una gran variedad de formas en las cuales se resuelve la forma en que la marcha del robot humanoide sea lo más parecido a la marcha humana, algunos de estos trabajos proponen para el seguimiento de una trayectoria controles basados en Lyapunov para el desplazamiento del pie y el movimiento del tobillo[22] o mediante control por imágenes para simular el ciclo de la marcha humana [23]. Otra propuesta es, mediante patrones de aprendizaje que el robot conozca la trayectoria que va a seguir [24].

Algunos investigadores se centran en el diseño de la unión de los eslabones que representan la rodilla humana [25], así como el comportamiento de la pierna en conjunto con la cadera y la cintura del robot [26].

Un diseño de bioingeniería es el robot ESBiRRo, mostrado en la figura 1.4, cuyo objetivo es lograr la generación de una marcha eficiente y robusta con el uso de estrategias biomiméticas de reacción ante perturbaciones, para esto utiliza la aproximación del ciclo límite [4].



Figura 1.3: Robot HRP-2 desarrollado por Kawada Industries [3].



Figura 1.4: Primer prototipo del bípido ESBiRRo [4].

Con el robot COMAN mostrado en la figura 1.5, se han centrado en hacer una caracterización genérica de la marcha del mismo, optimizando la forma en como posiciona el pie en el piso. Se le agregó un movimiento en la cadera para estabilizar al humanoide cuando camina [5].

No solo estos proyectos son llevados a cabo con el diseño de robots humanoides de tamaño real, también se han hecho robots antropomórficos a pequeña escala por empresas dedicadas a la robótica humanoide con fines de lucro, teniendo así a los robots Nao de Aldebaran robotics [18] y Bioloid de ROBOTIS [6].

Con Bioloid mostrado en la figura 1.6, se han hecho varias pruebas en su marcha. Se han sacado patrones de marcha 3D de tipo cicloidal para humanoides [27] y se han aplicado a este



Figura 1.5: Robot humanoide COMAN [5].

robot. También se ha ocupado la cinemática del Bioloid para simular su comportamiento con fines educativos [28].



Figura 1.6: Robot humanoide Bioloid [6].

En el Instituto Tecnológico de la Laguna en Coahuila, se están llevando a cabo experimentos de marcha con el robot bípedo ITLag que fue construido en el mismo. El ITLag cuenta con 5 grados de libertad, tiene una altura de 131 cm y un peso de 92 kg. El movimiento del robot solo es en el plano sagital [29].

La mayoría de los algoritmos de marcha para robots humanoides solo están diseñados con base a la dinámica del mismo en el plano sagital, esto provoca que el equilibrio se pierda fácilmente ante alguna perturbación y el robot caiga. Así mismo, algunos algoritmos de marcha solo se quedan en simulación, no se llegan a implementar en el robot. En otros casos, las pruebas de los algoritmos se llevan en ambientes controlados, ya sea en superficies planas o para subir escaleras.

Para desarrollar un algoritmo de marcha es necesario conocer la cinemática directa e inversa del robot bípedo. Existen diferentes métodos para hacer esto, entre ellos: a) Newton-Raphson, eliminación algebraica, técnicas de optimización, Jacobiano transpuesto, Jacobiano inverso, Denavit-Hartenberg y teoría de las bases de Groebner. En la tabla 1.1 se muestra una comparativa entre los métodos mencionados en donde se observa que la mejor opción para resolver la cinemática directa e inversa es con la teoría de las bases de Groebner, ya que es fácil resolver los conjuntos polinomiales que describen la cinemática.

Tabla 1.1: Métodos para calcular la cinemática [15].

Método	Ventajas	Desventajas
Newton-Raphson	Simple y sencillo	Iterativo y de cálculos lentos, falla cuando está cerca de una singularidad, converge a una sola solución y falla con sistemas que tienen muchos GDL
Eliminación algebraica	Funciona con mas de 6 GDL	No garantiza una solución
Técnicas de optimización	No se calculan matrices inversas	Falla para un mínimo local y no es factible para encontrar una solución global
Jacobiano transpuesto	Es rápido para estructuras pequeñas	Falla cuando está cerca de una singularidad y está condicionado.
Jacobiano inverso	Es rápido para estructuras pequeñas y hace cálculos en tiempo real	Alto consumo computacional y falla cuando esta cerca de una singularidad.
Denavit-Hartenberg	Funciona con mas de 6 GDL	Tiene errores en tiempo de ejecución, no encuentra todas las soluciones y en ocasiones no encuentra ninguna solución.
Teoría de las bases de Groebner	Es simple y sencillo, no tiene cálculo complejos, hace cálculos en tiempo real, encuentra todas las posibles soluciones y clasifica las singularidades	Tiene errores en tiempo de ejecución.

Por otro lado, existen otros criterios de estabilidad como FRI (Foot Rotation Indicator point) que indica la inestabilidad en la postura del robot [30] y Lyapunov [22]. El criterio de estabilidad más utilizado en los algoritmos de marcha es el ZMP (Zero Moment Point) [1], el cual asegura que el robot se mantendrá en equilibrio durante la caminata ya que el centro de masas del robot siempre está dentro del polígono de soporte, que es una envolvente convexa que considera todos los puntos de contacto entre el robot y el suelo. Con base en esto se utilizará el ZMP en esta tesis.

Aunque se han desarrollado varios algoritmos de marcha y estrategias para mantener el equilibrio del robot, como se mencionó, en las competencias donde se utilizan los robots Nao aun existen casos en que se necesita avanzar rápido y las vibraciones en él se vuelven muy grandes, haciendo que pierda el equilibrio o que sus pies se atoren en el pasto artificial, ya que cambia la superficie dependiendo del material con que está fabricado además de que algunas zonas de la alfombra están mas desgastadas que otras, por lo que es necesario proponer un algoritmo que resuelva dicha problemática.

1.2. Planteamiento del problema

Actualmente, la mayoría de los robots humanoides no pueden caminar de forma adecuada sobre superficies planas, estos tienden a tropezar y caer. Esto es un gran problema ya que los ambientes en los que se desenvuelve el ser humano tienen superficies irregulares que dificultan la marcha del robot.

Para que un robot humanoide sea útil para el ser humano, debe ser capaz de desplazarse en cualquier tipo de superficie. Por ello se busca que este tipo de robots pueda adaptarse a diferentes entornos.

Los robots humanoides a pequeña escala como el Bioloid y Nao, son utilizados en competencias de fútbol. Uno de los problemas que se presenta en la caminata sobre pasto artificial es que los pies del robot se atorán con las irregularidades del pasto, lo cual provoca que caigan constantemente durante el partido.

Estas caídas en la mayoría de los casos descalibran los sensores del humanoide hasta llegar a dañarlo. Por esta razón se requiere de un algoritmo de marcha que mantenga en equilibrio al robot bípedo mientras camina siguiendo una trayectoria.

En la planificación de trayectorias se toman en cuenta muchos factores, entre ellos, la cinemática inversa del robot. Con el cálculo de la cinemática inversa del robot es posible obtener la mayor cantidad de soluciones posibles para llegar a un punto determinado. Algunos métodos solo proporcionan soluciones locales, existen otros que dan soluciones globales, como las bases de Groebner. Las ventajas de usar bases de Groebner son: a) un método simple y directo, b) cálculos en tiempo real, c) encuentra todas las posibles soluciones y d) clasifica las singularidades [15].

Con base en la problemática se propone la creación de un algoritmo de caminata, el cual considere un algoritmo de control que permita mantener el equilibrio del robot durante la marcha sobre pasto artificial y superficies planas utilizando herramientas o metodologías como el criterio de estabilidad ZMP (Zero Moment Point) y las bases de Groebner para mejorar la caminata del robot.

1.3. Motivación

Los robots de tipo humanoide se han diseñado para realizar diferentes actividades, por ejemplo: trabajos de alto riesgo o que requieren alta precisión, interacción con personas para entretenimiento y ayudar a personas con discapacidad en sus actividades diarias. Los investigadores han utilizado estos robots para realizar actividades tan complejas para una máquina como el jugar fútbol o el ajedrez.

El fútbol soccer, motiva a buscar la solución a problemas específicos como la caminata sobre pasto artificial o en ambientes no controlados con superficies planas, como son el pasto natural y algunos tipos de suelos no fangosos.

Una de las organizaciones que realiza competencias de fútbol soccer para robots humanoides es la RoboCup, que año con año va cambiando las restricciones de tal forma que para mediados del siglo XXI, se pueda tener una competencia entre robots humanoides y futbolistas humanos.

La problemática de las competencias de fútbol soccer, no solo implica los algoritmos de marcha, también involucra otras actividades como: visión artificial, algoritmos de aprendizaje y sincronización entre los diferentes robots que conforman al equipo.

1.4. Hipótesis

Con el criterio de estabilidad ZMP, la teoría de las bases de Groebner y un control difuso, un robot bípedo puede caminar de una manera considerada estable en superficies planas con pasto artificial.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar e implementar en un robot humanoide bípedo Nao un controlador basado en el ZMP, usando bases de Groebner, para mantener una caminata estable en superficies planas con pasto artificial.

1.5.2. Objetivos específicos

1. Determinar la cinemática inversa del robot Nao empleando bases de Groebner para la generación de una trayectoria ZMP.
2. Usar técnicas de control difuso para mantener el equilibrio del robot cuando se encuentra de pie tanto en reposo como en movimiento.

3. Implementar un control difuso para minimizar la distancia entre la trayectoria calculada de la cadera y la trayectoria del ZMP.

1.6. Metas

Las metas que se quieren lograr con este trabajo de investigación son las siguientes:

1. Aplicar la teoría de las bases de Groebner para el cálculo de la cinemática inversa del robot Nao.
 - a) Piernas.
 - b) Brazos.
2. Hacer un algoritmo para la sincronización de las piernas y brazos del robot Nao.
3. Diseñar un control difuso para mantener el equilibrio del robot Nao cuando está de pie en reposo.
4. Simular el control difuso para mantener el equilibrio del robot Nao en reposo utilizando WEBOTS.
5. Implementar el control difuso para mantener el equilibrio del robot Nao cuando está de pie en reposo.
6. Aplicar el criterio de estabilidad ZMP para la generación de trayectorias.
7. Diseñar un generador de patrón de marcha utilizando el ZMP.
8. Diseñar un control difuso para controlar la posición de la cadera del robot Nao.
9. Simular el control difuso para controlar la posición de la cadera del robot Nao utilizando WEBOTS.
10. Implementar el control difuso para controlar la posición de la cadera del robot Nao.
11. Diseñar un control difuso para mantener en equilibrio al robot Nao durante la caminata sobre superficies planas con pasto artificial.
12. Simular el control difuso para mantener en equilibrio al robot Nao durante la caminata sobre superficies planas con pasto artificial utilizando WEBOTS.
13. Implementar el control difuso para mantener en equilibrio al robot Nao durante la caminata sobre superficies planas con pasto artificial.

1.7. Metodología

La metodología que se va a emplear en este trabajo es en cascada, esto es, realizar cada una de las actividades de forma ordenada.

1.7.1. Análisis de requerimientos

Los requerimientos que se necesitan para que el robot mantenga una caminata estable, son que la cancha con pasto artificial debe estar en una superficie plana y que no existan irregularidades significativas en la superficie. Además, el robot bípedo debe contar con actuadores en las piernas de tal forma que pueda existir movimiento en el plano frontal, así mismo se requiere de un giroscopio en la cintura del robot para tener una medición más precisa de la posición del robot.

1.7.2. Obtención de la información

La información de la posición de las piernas y del equilibrio del robot Nao se obtendrá de los sensores que se encuentran en las uniones de las piernas y del giroscopio de la cintura respectivamente.

1.7.3. Desarrollo del algoritmo de marcha

El algoritmo de marcha debe ser capaz de equilibrar al robot bípedo en cualquier instante de la caminata. La cinemática inversa se obtendrá con las bases de Groebner. Se tendrá un generador de patrón de marcha el cuál determinará la trayectoria ZMP que debe seguir el robot durante la caminata y las señales obtenidas de los servomotores se procesarán para obtener la posición de las piernas del robot. Con la señal del giroscopio en la cintura del robot se determinará si el robot está en equilibrio o no.

1.7.4. Implementación

Una vez que se han procesado las señales del giroscopio y de los servomotores del robot Nao, se implementarán los controles difusos tanto para el equilibrio del robot como para la posición de las piernas y que de esta forma el robot alcance los puntos de la trayectoria dada por el generador de patrón de marcha.

1.7.5. Pruebas

En esta etapa se probará que el algoritmo de marcha sea eficiente para los partidos de fútbol, así como demostrar que la teoría de las bases de Groebner es un método que facilita el cálculo de la cinemática inversa de sistemas complejos como son los robots bípedos. Las pruebas de los controles difusos para el equilibrio del robot y la posición de la cadera del mismo se harán primero en simulación y después se implementaran en el robot.

1.7.6. Resultados

En esta etapa se presentará el sistema que haya dado mejores resultados durante las pruebas, es decir, el algoritmo de marcha que mantenga al robot con una marcha considerada estable.

Capítulo 2

Marco Teórico

Para el diseño de un algoritmo de marcha es necesario conocer diferentes características del robot Nao, como son: la cinemática inversa y directa, el braceo y la relación que existe entre ambas piernas para generar un patrón de marcha.

2.1. Cinemática

La cinemática es la parte de la física que se encarga del estudio del movimiento de sistemas mecánicos sin considerar las fuerzas que lo producen. La cinemática se divide en: cinemática directa y cinemática inversa, las cuales pueden calcularse mediante el método de las bases de Groebner, Denavit-Hartenberg y haciendo uso de herramientas geométricas.

La cinemática directa consiste en analizar el movimiento del robot con respecto a un sistema cartesiano fijo relacionando las coordenadas generalizadas, los parámetros geométricos y las coordenadas cartesianas y de orientación del efector final.

La cinemática inversa determina si un robot puede alcanzar una posición determinada dadas las coordenadas generalizadas, los parámetros geométricos y de orientación en ángulos de Euler del efector final de un robot con respecto a un sistema de coordenadas fijo.

Calcular la cinemática inversa de un robot es más complejo de hacer que la cinemática directa. La cinemática directa siempre tiene solución, mientras que la cinemática inversa no siempre la tiene, puede ser que tenga más de una solución o que no tenga ninguna solución.

2.1.1. Convención de Denavit-Hartenberg

En 1955 Jaques Denavit y Richard S. Hartenberg presentaron un método con el cual se puede calcular la orientación y traslación de un robot. Con este método es fácil deducir las ecuaciones de cinemática directa [31].

La convención de Denavit-Hartenberg toma como referencia al diagrama de un robot en cadena cinemática abierta. Los parámetros que considera son: θ_i , d_i , a_i y α_i . Donde θ_i representa el ángulo entre los ejes x_{i-1} y x_i medido alrededor del eje z_{i-1} . d_i es la distancia del origen del sistema de referencia $i-1$ a la intersección de eje x_i con el eje z_{i-1} , su medición se realiza a lo largo del eje z_{i-1} . a_i es la distancia a lo largo del eje x_i desde el origen del sistema de referencia coordinado $i-1$ hasta la intersección del eje z_{i-1} con el eje x_i . α_i es el ángulo entre los ejes z_i y z_{i-1} respecto a un plano normal a x_i .

Los parámetros de Denavit-Hartenberg se sustituyen en la matriz de transformación homogénea, esta matriz puede ser directa o inversa. La matriz de transformación homogénea directa esta representada en 2.1 y la matriz de transformación homogénea inversa esta representada en 2.2.

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \cdot \sin\theta_i & \sin\alpha_i \cdot \sin\theta_i & a_i \cdot \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cdot \cos\theta_i & -\sin\alpha_i \cdot \sin\theta_i & a_i \cdot \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i \\ \cos\alpha_i \cdot \sin\theta_i & \cos\alpha_i \cdot \cos\theta_i & -\sin\alpha_i & -d_i \cdot \sin\alpha_i \\ \sin\alpha_i \cdot \sin\theta_i & \sin\alpha_i \cdot \cos\theta_i & \cos\alpha_i & d_i \cdot \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

En un robot humanoide bípedo, el modelo cinemático se analiza dividiéndolo en dos partes. La primera consiste en tomar el punto medio de la cadera como eslabón final y la planta del pie derecho como el eje de referencia. La segunda consiste en tomar el pie izquierdo como efector final y el punto medio de la cadera como eje de referencia [32].

2.1.2. Método geométrico

Para encontrar la solución geométrica de un robot, se trata de descomponer la geometría espacial del mismo en varios problemas de geometría plana. El método geométrico consiste en resolver los ángulos de las articulaciones utilizando herramientas trigonométricas.

Mientras más eslabones tiene el robot se complica la forma de calcular las ecuaciones de la cinemática. En muchos robots es fácil calcular los ángulos de las articulaciones utilizando las herramientas de la geometría plana.

Para un brazo de 3 GDL como el mostrado en la figura 2.1 que es planar se le puede aplicar directamente la geometría plana para encontrar una solución [7]. Esto es, considerar que es un triángulo sólido y de esta forma aplicar funciones trigonométricas como la *ley de cosenos* y *el teorema de Pitágoras* para encontrar la solución de los ángulos que debe tomar

el mecanismo para llegar a un punto determinado.

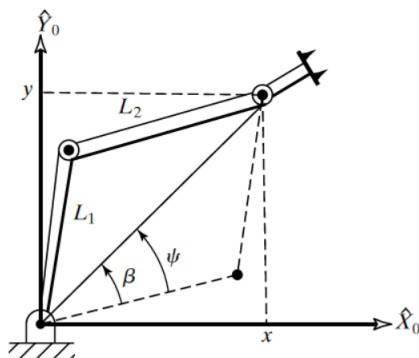


Figura 2.1: Representación del modelo de un manipulador planar de 3 GDL [7].

2.2. Dinámica de mecanismos

La dinámica de mecanismos considera las fuerzas requeridas para producir el movimiento de un cuerpo. En los mecanismos existen dos problemas a los que se les quiere dar solución. En el primer problema se tiene un punto conocido de una trayectoria y se desea encontrar el vector de momentos de torsión de las articulaciones. El segundo problema es determinar la forma en como se va a mover el mecanismo bajo la aplicación de un vector de momentos de torsión.

Con la dinámica de mecanismos se pretende conocer en los eslabones del robot: las aceleraciones tanto lineales como angulares, la distribución de masas y las fuerzas y momentos de torsión [7].

La aceleración en un cuerpo rígido es la segunda derivada de su vector de posición en cualquier instante de tiempo llamadas aceleración lineal y angular. La aceleración lineal se genera cuando existen movimientos lineales y la aceleración angular se genera cuando existen movimientos rotacionales.

La distribución de masas se conoce a partir de calcular el centro de masas (CoM), el cual es el promedio de la posición de todas las partes del mecanismo ponderadas de acuerdo a sus masas.

Cuando existe una torsión en el mecanismo se generan fuerzas de torsión y momentos de torsión. La torsión se genera cuando se le aplica un momento al mecanismo, tomando como momento el producto vectorial del vector de posición del punto donde se aplica la fuerza y la fuerza misma.

2.3. Locomoción bípeda

La locomoción bípeda es un fenómeno periódico, donde el ciclo de la marcha se puede definir como un intervalo entre dos eventos sucesivos referidos a una misma pierna con al menos un pie en fijo en el suelo.

La marcha humana se puede dividir en dos fases, la fase de *apoyo*, en la cuál el pie está en contacto con el piso, y la fase de *balanceo*, durante la cual el pie es aerotransportado [32]. Cada una de estas fases tiene un meta diferente, la fase de apoyo como su nombre lo dice, le da soporte al cuerpo manteniéndolo en una posición recta, mientras que la fase de balanceo es para avanzar y prepararse para el siguiente paso [8].

La distribución temporal de cada una de las fases es 60% para el apoyo y 40% para el balanceo. Considerando el contacto relativo que tienen los dos pies con el piso se puede segmentar la marcha en partes mas especificas, estas subfases son: Respuesta de carga, apoyo medio, apoyo final, pre balanceo, balanceo inicial, balanceo medio y balanceo final, como se muestra en la figura 2.2.

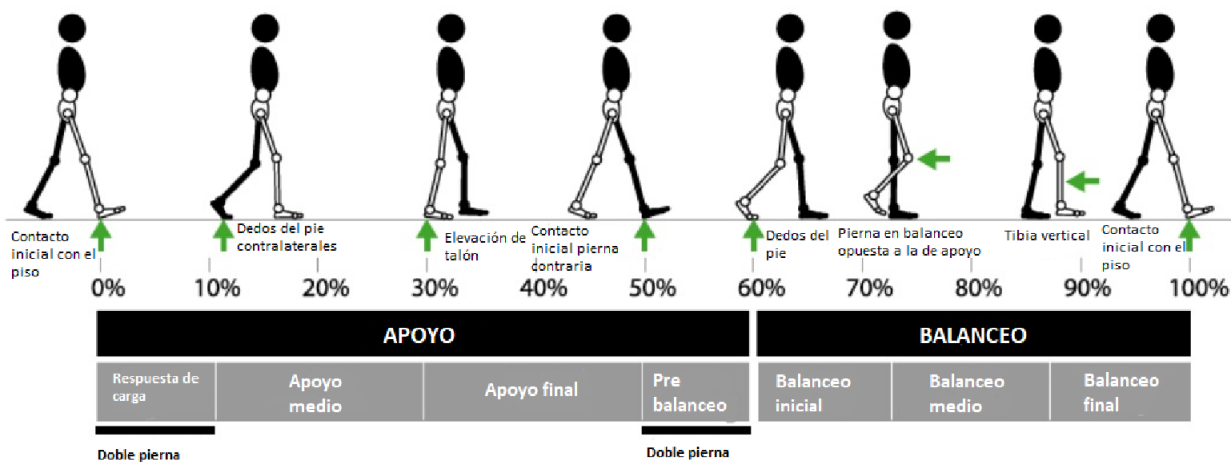


Figura 2.2: Fases y subfases de la marcha humana [8].

La marcha humana mostrada en la figura 2.2 se puede resumir en dos componentes, las cuales se consideran las más importantes de la marcha humana [9], estas son: fase de doble soporte (FSD) o fase de doble apoyo, y fase de soporte simple (FSS) o fase de apoyo sencillo, estas fases se pueden observar en la figura 2.3. Estas fases son las que se replican en la marcha de los robots bípedos.

La locomoción humana mecánicamente se puede ver como un péndulo invertido [10]. Las piernas se representan como estructuras rígidas que soportan una masa como se muestra en la figura 2.4.

La trayectoria descrita por el péndulo invertido puede describir de forma aproximada el

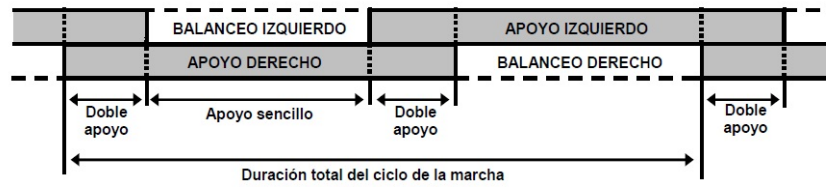


Figura 2.3: Fases principales de la marcha humana [9].

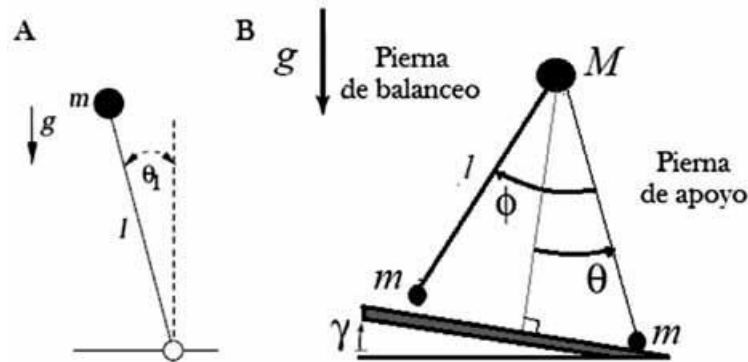


Figura 2.4: Modelo que representa la marcha humana basado en el péndulo invertido [10].

comportamiento del movimiento global del cuerpo humano, generando un movimiento semi-circular.

La planeación de trayectorias en los robots de tipo humanoide se hacen en los planos sagital y frontal para tener movimientos en 3D [11] como se muestra en el diagrama a bloques de la figura 2.5. La planeación en el plano sagital mostrado en la figura 2.6 se hace con el fin de imitar la locomoción humana. En el plano frontal mostrado en la figura 2.6 se tiene la dinámica del balanceo que en combinación con los movimientos en el plano sagital se tiene como resultado una marcha natural.

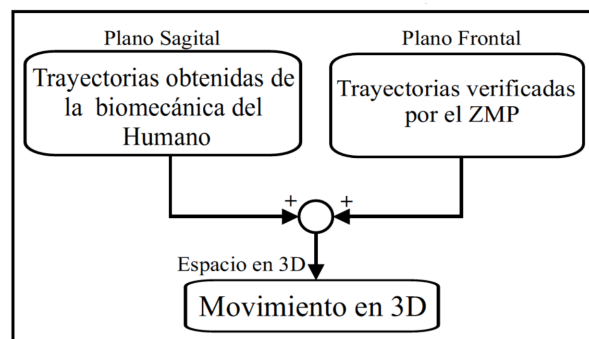


Figura 2.5: Análisis de marcha [11].

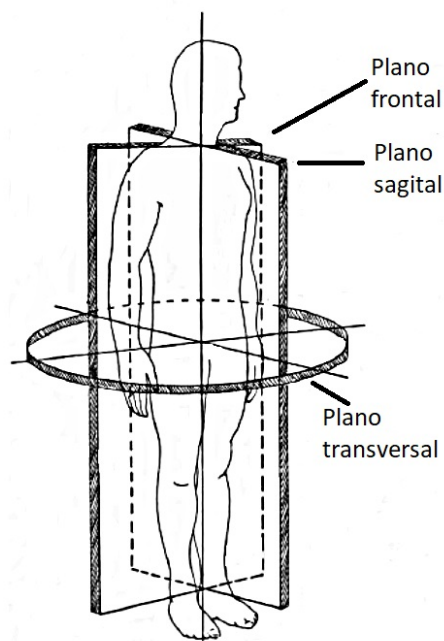


Figura 2.6: Planos y ejes de movimiento del ser humano [12].

En los robots bípedos existen dos tipos de algoritmos de caminata, los pasivos y los activos. La caminata pasiva consiste en que el robot se mueva con la inercia generada por el movimiento de la marcha, haciendo que el consumo de energía sea menor. La caminata activa hace uso de los actuadores que tiene el robot en las piernas.

2.4. Criterios de estabilidad

Para mantener el equilibrio del robot bípedo se usan criterios de estabilidad. Entre ellos se puede encontrar al *punto de momento cero* (*ZMP* por sus siglas en inglés *Zero Moment Point*), Modo de péndulo invertido linealizado 3D o 3D-LIMP por sus siglas en inglés *Three-Dimensional Linear Inverted Pendulum Mode*, criterio de estabilidad de contacto de pie [33], entre otros.

2.4.1. Punto de momento cero (ZMP)

Vukabratovic y Stepanenko en 1972 propusieron un método para que la marcha de un robot tenga estabilidad local. A este método lo llamaron *Zero Moment Point* (ZMP) [4] o *Punto de Momento Cero* en español. El ZMP mostrado en la figura 2.7 se define como el punto del suelo respecto al cual el momento creado por las fuerzas inerciales y gravitacionales no tiene componente en los ejes horizontales.

En los robots bípedos el único contacto con el medio ambiente se realiza mediante la

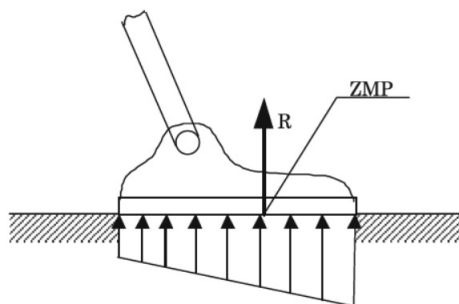


Figura 2.7: Definición del Punto de Momento Cero (ZMP) [1].

fuerza de fricción y la fuerza vertical generadas por el piso.

Para facilitar el análisis se puede descartar la parte del mecanismo que está por encima del tobillo del pie de apoyo y sustituir su influencia por una fuerza y un momento con tres componentes cada una, las cuales son las reacciones totales del robot con respecto al piso.

Todos los sensores de fuerza en el pie del robot deben estar en contacto con el piso, de no hacerlo el robot puede rotar y caer. Cuando uno o varios de los sensores de fuerza en el pie no está en contacto con el piso o si el robot no cuenta con sensores de fuerza en los pies, se emplea una estrategia de control para evitar que el robot caiga [34].

Para ejecutar la estrategia de control primero se debe verificar que el CoM (centro de masas) se encuentra dentro del polígono de soporte. La región que considera todos los puntos de contacto entre el robot y el suelo se le llama *polígono de soporte*. Matemáticamente el *polígono de soporte* esta definido por una envolvente convexa como se muestra en 2.8.

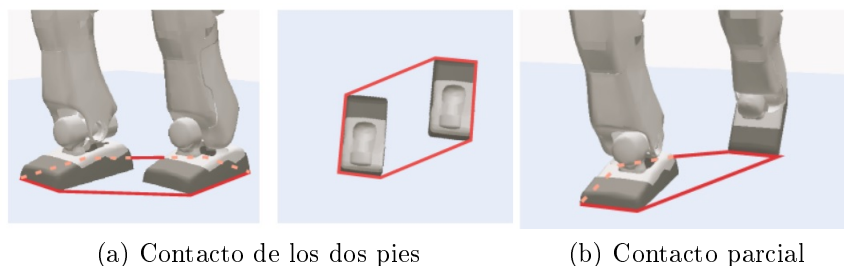


Figura 2.8: Polígono de soporte [1].

La relación que existe entre el ZMP y el polígono de soporte es muy importante. El ZMP solo existe dentro del polígono de soporte. Si el ZMP se encuentra fuera del polígono de soporte el robot caerá. Existen dos formas de analizar el ZMP, en 2D y en 3D.

Análisis en 2D

Se reemplazan las fuerzas distribuidas a lo largo del suelo por una fuerza y momento equivalente en un solo punto en el piso.

Análisis en 3D

Considerando que las fuerzas aplicadas al robot en movimiento están en 3D, la suma de las componentes verticales y horizontales de la fuerza de reacción del piso deben ser sumadas al mismo tiempo. El punto (x,y) en donde se hace cero la suma de las componentes verticales y horizontales es el centro de presión.

2.4.2. Modo de péndulo invertido linealizado 3D y Punto de momento cero

El criterio 3D-LIMP [13] consiste en diseñar un control de posición en donde el centro de masas (CoM) se mueva en un plano definido arbitrariamente que tiene restricciones. Este método permite hacer dos controles, uno en el plano sagital (x, z) y otro en el plano frontal (x, y) [35]. Tomando como referencia el plano cartesiano mostrado en la figura 2.9 y fijando como dirección ordinal de la marcha al eje x . El plano restringido está representado por un vector normal $(k_x, k_y, -1)$ y la intersección de z con z_c como en la ecuación 2.3.

$$z = k_x x + k_y y + z_c \quad (2.3)$$

Si el plano restringido es paralelo al plano (x, y) , entonces $k_x = k_y = 0$. Así, la dinámica del péndulo está dada como se muestra en las ecuaciones 2.4 y 2.5.

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} \tau_x \quad (2.4)$$

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{m z_c} \tau_y \quad (2.5)$$

donde m es la masa del péndulo, g es la gravedad y τ_x, τ_y son los torques generados alrededor de los ejes x y y respectivamente. Aunque $k_x, k_y \neq 0$, utilizando la restricción en los torques de entrada que se muestran en la ecuación 2.6 se puede tener la misma dinámica.

$$\tau_x x + \tau_y y = 0 \quad (2.6)$$

Con la restricción $k_x = k_y = 0$ se puede calcular el ZMP con las ecuaciones 2.7 y 2.8.

$$p_x = -\frac{\tau_y}{mg} \quad (2.7)$$

$$p_y = -\frac{\tau_x}{mg} \quad (2.8)$$

Sustituyendo las ecuaciones 2.8 y 2.7 en 2.4 y 2.5 respectivamente, se tiene que

$$p_y = y - \frac{z_c \ddot{y}}{g} \quad (2.9)$$

$$p_x = x - \frac{z_c \ddot{x}}{g} \quad (2.10)$$

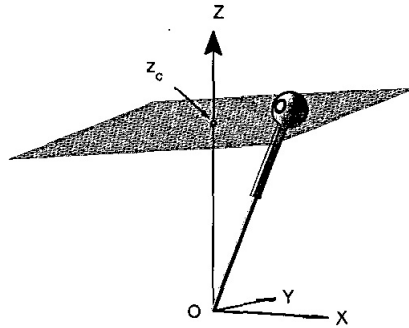


Figura 2.9: Representación del péndulo invertido en el área restringida [13].

2.5. Control difuso

Para hacer uso del control difuso es necesario conocer la lógica difusa. Esta es una metodología para obtener una salida con información de entrada ambigua, imprecisa o incompleta. La lógica difusa modela la forma de pensar de un humano para tomar decisiones basadas en información incompleta.

La lógica difusa trabaja con conjuntos difusos. Estos conjuntos están definidos por sus funciones de pertenencia, la cual expresa la distribución de verdad de una variable [36].

Un conjunto difuso se puede definir matemáticamente al asignar a cada posible individuo un valor que representa su grado de pertenencia en el mismo. Este grado indica cuando el elemento es similar o compatible con el concepto representado por el conjunto difuso.

Un sistema difuso esta constituido por [36]:

Etapas de difusificación. Se transforman las variables controladas por proceso en variables llamadas variables de tipo lingüísticas. Como resultado se obtienen valores lingüísticos medidos.

Reglas. Contiene las reglas difusas que encierran el conocimiento necesario por la solución del problema de control. Son del tipo condicional, es decir, si ocurre algo entonces se toma

una acción.

Difusificador. Calcula las variables de salida a partir de las variables de entrada mediante las reglas y la inferencia difusa, entregando conjuntos difusos de salida.

Desdifusificador. El resultado de la inferencia difusa es convertido de un concepto lingüístico a una salida física.

El uso de la lógica difusa en los sistemas de control satisface uno o más de los siguientes objetivos [36]

1. Mejorar la robustez que se obtiene con los métodos clásicos de control lineal.
2. Diseño de control simplificado para modelos complejos.
3. Se obtiene una implementación simplificada.
4. Autonomía.
5. Adaptabilidad.
6. En el caso de control difuso, no es necesario un modelo matemático de la planta.

2.6. El robot Nao

El robot humanoide Nao de Aldebaran Robotics tiene 5 cadenas cinemáticas [14], estas son, la cabeza, sus dos brazos y sus dos piernas. Las articulaciones que tiene cada cadena cinemática mostradas en la figura 2.10 son:

- **Cabeza:** HeadYaw, HeadPitch
- **Brazo izquierdo:** LShoulderPitch, LShoulderRoll, LElbowYaw, LElbowRoll
- **Brazo derecho:** RShoulderPitch, RShoulderRoll, RElbowYaw, RElbowRoll
- **Pierna izquierda:** LHipYawPitch, LHipRoll, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll
- **Pierna derecha:** RHipYawPitch, RHipRoll, RHipPitch, RKneePitch, RAnklePitch, RAnkleRoll

Los nombres de las articulaciones se conforman de tres elementos. El primer elemento puede ser L o R, en donde L es left (izquierdo) y R es Right (derecha). El segundo elemento indica en que parte del robot se encuentra, este puede ser Head (cabeza), Shoulder (hombro), Elbow (codo), Hip (cadera), Knee (rodilla) o Ankle (tobillo). Y por último, el tercer elemento representa la dirección de movimiento. Este puede ser Roll (rotación en el eje x), Pitch (la

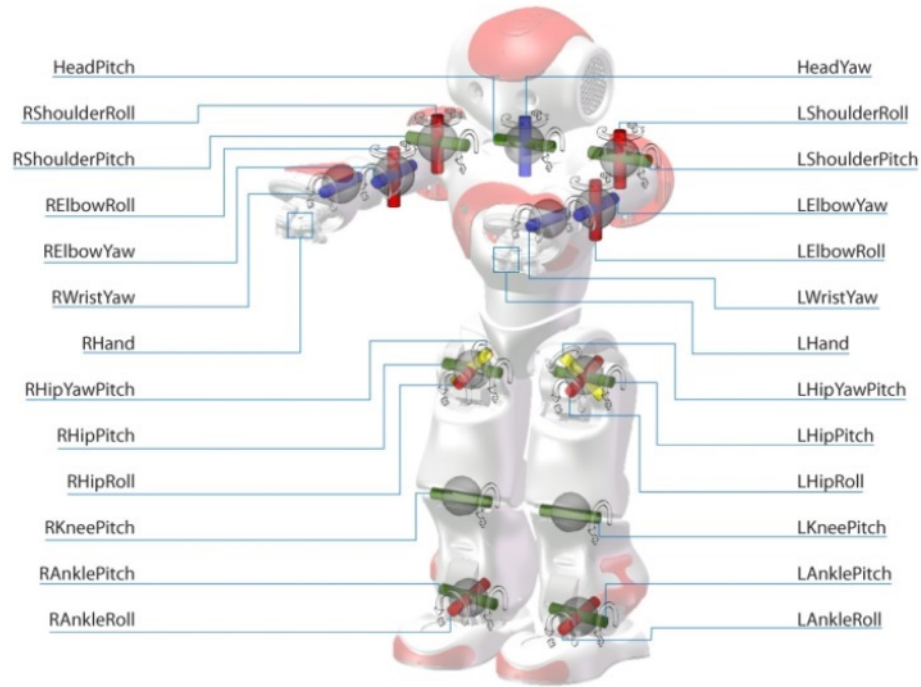


Figura 2.10: Cadenas cinemáticas y articulaciones del robot Nao [14].

rotación en el eje y) o Yaw (la rotación en el eje z).

Las dimensiones que tiene el robot Nao son las mostradas en la tabla 2.1, las cuales son muy importantes en el cálculo de la cinemática directa.

La coordenada base se toma en la parte central del torso como se muestra en la figura 2.11, al mismo tiempo se muestra la posición cero de todas las articulaciones del robot Nao [37].

2.6.1. Brazo izquierdo

La cadena cinemática del brazo izquierdo cuenta con 4 articulaciones. Los parámetros de Denavit-Hartenberg se muestran en la tabla 2.2 con los cuales se calculan las matrices de transformación homogénea. La matriz de transformación homogénea es la mostrada en 2.11 [37].

$$T_{base}^{fin} = A_{base}^0 T_0^1 T_1^2 T_2^3 T_3^4 R_z(\pi/2) A_4^{fin} \quad (2.11)$$

Tabla 2.1: Dimensiones del robot Nao.

Nombre	Longitud(mm)
NeckOffsetZ	126.50
ShoulderOffsetY	98.00
ElbowOffsetY	15.00
UpperArmLength	105.00
LowerArmLength	55.95
ShoulderOffsetZ	100.00
HandOffsetX	57.75
HipOffsetZ	85.00
HipOffsetY	50.00
ThighLength	100.00
TibiaLength	102.90
FootHeight	45.19
HandOffsetZ	12.31

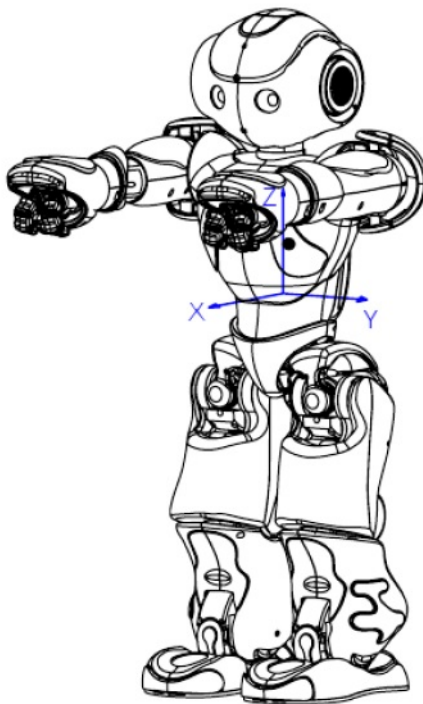


Figura 2.11: Posición cero y la coordenada base del robot Nao [14].

Tabla 2.2: Parámetros de Denavit-Hartenberg del brazo izquierdo del robot Nao.

Articulación	a_i	α_i	d_i	θ_i
Base	A(0, ShoulderOffsetY+ElbowOffsetY, ShoulderOffsetZ)			
LShoulderPitch	0	$-\pi/2$	0	θ_1
LShoulderRoll	0	$\pi/2$	0	$\theta_2 - \pi/2$
LElbowYaw	0	$-\pi/2$	UpperArmLength	$-\theta_3$
LElbowRoll	0	$\pi/2$	0	θ_4
Rotación	$R_z(\pi/2)$			
Efector final	A(HandOffsetX+LowerArmLength, 0, 0)			

2.6.2. Brazo derecho

La cadena cinemática del brazo derecho cuenta con 4 articulaciones, es simétrico al brazo izquierdo. Al final se le debe agregar una rotación extra llamada *Rotación fix* para corregir la orientación del brazo, ya que el eje Z está invertido. Los parámetros de Denavit-Hartenberg se muestran en la tabla 2.3 con los cuales se calculan las matrices de transformación homogénea. La matriz de transformación homogénea es la mostrada en A.6 [37].

$$T_{base}^{fin} = A_{base}^0 T_0^1 T_1^2 T_2^3 T_3^4 R_z(\pi/2) A_4^{fin} R_z(-\pi) \quad (2.12)$$

Tabla 2.3: Parámetros de Denavit-Hartenberg del brazo derecho del robot Nao.

Articulación	a_i	α_i	d_i	θ_i
Base	A(0, -ShoulderOffsetY-ElbowOffsetY, ShoulderOffsetZ)			
RShoulderPitch	0	$-\pi/2$	0	θ_1
RShoulderRoll	0	$\pi/2$	0	$\theta_2 + \pi/2$
RElbowYaw	0	$-\pi/2$	-UpperArmLength	θ_3
RElbowRoll	0	$\pi/2$	0	θ_4
Rotación	$R_z(\pi/2)$			
Efector final	A(-HandOffsetX-LowerArmLength, 0, 0)			
Rotation fix	$R_z(-\pi)$			

2.6.3. Pierna izquierda

La cadena cinemática de la pierna izquierda del Nao tiene 6 articulaciones. La tabla 2.4 muestra los parámetros de Denavit-Hartenberg y la matriz de transformación final se muestra en 2.13 [37].

$$T_{base}^{fin} = A_{base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z(\pi) R_y(-\pi/2) A_6^{fin} \quad (2.13)$$

Tabla 2.4: Parámetros de Denavit-Hartenberg de la pierna izquierda del robot Nao.

Articulación	a_i	α_i	d_i	θ_i
Base	A(0, HipOffsetY, -HipOffsetZ)			
LHipYawPitch	0	$-3\pi/4$	0	$\theta_1 - \pi/2$
LHipRoll	0	$-\pi/2$	0	$\theta_2 + \pi/4$
LHipPitch	0	$\pi/2$	0	θ_3
LKneePitch	-ThighLength	0	0	θ_4
LAnklePitch	-TibiaLength	0	0	θ_5
LAnkleRoll	0	$-\pi/2$	0	θ_6
Rotación	$R_z(\pi) R_y(\pi/2)$			
Efecto final	A(0, 0, -FootHeight)			

2.6.4. Pierna derecha

La cadena cinemática de la pierna derecha del Nao tiene 6 articulaciones, es simétrica a la pierna izquierda. La diferencia entre ambas piernas radica en las distancias y rotaciones en el eje Y. La tabla 2.5 muestra los parámetros de Denavit-Hartenberg y la matriz de transformación final se muestra en 2.14 [37].

$$T_{base}^{fin} = A_{base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z(\pi) R_y(-\pi/2) A_6^{fin} \quad (2.14)$$

Tabla 2.5: Parámetros de Denavit-Hartenberg de la pierna derecha del robot Nao.

Articulación	a_i	α_i	d_i	θ_i
Base	A(0, -HipOffsetY, -HipOffsetZ)			
RHipYawPitch	0	$-\pi/4$	0	$\theta_1 - \pi/2$
RHipRoll	0	$-\pi/2$	0	$\theta_2 - \pi/4$
RHipPitch	0	$\pi/2$	0	θ_3
RKneePitch	-ThighLength	0	0	θ_4
RAnklePitch	-TibiaLength	0	0	θ_5
RAnkleRoll	0	$-\pi/2$	0	θ_6
Rotación	$R_z(\pi) R_y(\pi/2)$			
Efecto final	A(0, 0, -FootHeight)			

Capítulo 3

Teoría de las bases de Groebner

La teoría de las bases de Groebner, nombrado así por Bruno Buchberger en su tesis doctoral en honor a Wolfgang Groebner, provee una aproximación uniforme a la solución de muchos problemas expresados en términos de conjuntos de polinomios multivariados [38].

La teoría de las bases de Groebner se utilizan para resolver varios problemas, por ejemplo: Ideales Fermat, demostración de teoremas geométricos, teorías invariantes, sistemas de ecuaciones polinomiales [39], ecuaciones diferenciales parciales, funciones hipergeométricas, estadística y álgebra no conmutativa [38].

En la ingeniería se han utilizado las bases de Groebner para calcular la cinemática inversa de algunos robots, como es un brazo manipulador de 2 grados de libertad (GDL) en la tesis doctoral de Kimberly Kendricks [40] y en el ciclo de marcha humana para identificar a un individuo que lleve alguna carga en su cuerpo, esto con el fin de evitar ataques terroristas [15].

A lo largo del documento se denotará por $R = \mathbb{K}[x_1, \dots, x_n] = \mathbb{K}[x]$ al anillo de polinomios sobre el cuerpo \mathbb{K} . Un subconjunto no vacío $I \in R$ es un ideal de R si es cerrado bajo la suma de polinomios y es cerrado por la multiplicación de elementos de R .

3.1. Conceptos preliminares

Para entender las bases de Groebner se deben conocer algunos conceptos, como son: conjuntos, subconjuntos, anillo y funciones, entre otros conceptos.

3.1.1. Conjuntos

Cualquier colección de objetos como "los puntos de un segmento dado", "las rectas que pasan por un punto en el espacio ordinario" o "los números naturales menores que 10" se dice un *conjunto* o *clase*. Los puntos, las rectas y los números se dirán *elementos* de los conjuntos respectivos. Por lo general, se denotan con letras mayúsculas a los conjuntos y con letras mi-

núsculas a los elementos cualesquiera [41].

Definición 3.1. Sea A un conjunto dado y sean p y q ciertos objetos. Si p es un elemento de A , se indicará esto escribiendo $p \in A$; si tanto p como q son elementos de A , se escribirá $p, q \in A$ en vez de $p \in A$ y $q \in A$; cuando q no es un elemento de A se escribe $q \notin A$.

Cuando dos conjuntos A y B constan de los mismos elementos, se dicen iguales y se denota $A = B$. Para indicar que A y B no son iguales, escribiremos $A \neq B$.

Definición 3.2. Sean A y B dos conjuntos. Decimos que B es un subconjunto de A , si cada elemento de B es también un elemento de A [42].

Cuando B sea un subconjunto de A se denotará $B \subset A$. Así, $B \subset A$, si y solo si, $x \in B$ implica que $x \in A$. Si B no es un subconjunto de A se denota $B \not\subset A$.

3.1.2. Anillos

Un conjunto A no vacío \mathbb{Z} forma un anillo con respecto a las operaciones binarias de adición (+) y multiplicación (\cdot), si para cualquiera $a, b, c \in \mathbb{Z}$ satisfacen los siguientes puntos [42]:

1. La suma de dos elementos es conmutativa, es decir, si $a, b \in \mathbb{Z}$, entonces

$$a + b = b + a$$

2. La suma de dos elementos es asociativa, es decir, si $a, b, c \in \mathbb{Z}$, entonces

$$(a + b) + c = a + (b + c)$$

3. Existe en Z un elemento neutro para la suma, el 0. Es decir, si $a \in \mathbb{Z}$,

$$a + 0 = 0 + a = a$$

4. Para cada a en \mathbb{Z} existe en \mathbb{Z} su inverso aditivo que se denota por $-a$. Esto es,

$$a + (-a) = (-a) + a = 0$$

5. El producto de dos elementos es conmutativo, es decir, si $a, b \in \mathbb{Z}$ entonces

$$ab = ba$$

6. El producto de Z es asociativo, es decir, $a, b, c \in \mathbb{Z}$ entonces

$$(ab)c = a(bc)$$

7. Existe en \mathbb{Z} un elemento neutro para la multiplicación, el 1. Es decir, si $a \in \mathbb{Z}$

$$a1 = 1a = a$$

8. En \mathbb{Z} el producto distribuye a la suma, es decir, si $a, b, c \in \mathbb{Z}$ entonces

$$\begin{aligned} a(b + c) &= ab + ac \\ (a + b)c &= ac + bc \end{aligned}$$

Los conjuntos que cumplen los puntos anteriores, con las operaciones respectivas, constituyen un anillo conmutativo, con elemento unitario. Este tipo de conjuntos aparecen con mucha frecuencia en matemáticas [42].

De esta forma, podemos decir que el conjunto \mathbb{Z} de los números enteros, con las operaciones $+$ y \times forman un anillo.

Cuando un conjunto no cumple todos los puntos mencionados se les llama simplemente anillos. Los puntos que posiblemente no se cumplan son el 5 y el 7.

Un anillo en que la multiplicación sea conmutativa se llama anillo conmutativo (no cumple con el punto 7 y un anillo dotado de elemento neutro multiplicativo (elemento unidad) se llama anillo unitario (no cumple con el punto 5 [41]).

3.1.3. Ideal

Un ideal es un subconjunto de un anillo tal que para dos elementos $a - b$ y $a \cdot r$ (siendo r un elemento arbitrario del anillo) vuelven a ser elementos del subconjunto, como lo son a y b [43]. El ejemplo más común de un ideal es el generado por un número finito de polinomios [44].

Definición 3.3. Dado un subconjunto $I \subset \mathbb{K}[x_1, \dots, x_n]$ es un ideal si satisface:

1. $0 \in I$
2. Si $f, g \in I$, entonces $f + g \in I$
3. Si $F \in I$ y $h \in \mathbb{K}[x_1, \dots, x_n]$, entonces $hf \in I$

Consideremos un conjunto de polinomios $F = f_1, \dots, f_s$.

Lema 3.1. si $F = f_1, \dots, f_s \in \mathbb{K}[x_1, \dots, x_n]$, entonces F es un ideal de $\mathbb{K}[x_1, \dots, x_n]$, al que llamaremos *ideal generado por F* y denotaremos por

$$\langle f_1, \dots, f_s \rangle$$

3.1.4. Polinomios

Una gran parte del álgebra elemental llama polinomios a las expresiones

$$a_0 + a_1x + \cdots + a_nx^n \quad (3.1)$$

donde a_0, a_1, \dots, a_n son los coeficientes del polinomio, estos son números reales o complejos. Al símbolo x se le llama indeterminada y a_0, a_1x, \dots, a_nx^n son los términos del polinomio. Los coeficientes a_i pueden ser todos reales, todos racionales, todos irracionales o todos enteros. Los polinomios son conocidos dependiendo del tipo de coeficientes que tenga, por ejemplo, si un polinomio tiene coeficientes reales, entonces decimos que se trata de un polinomio con coeficientes reales. También, pueden considerarse polinomios cuyos coeficientes pertenecen a alguna estructura algebraica [42].

Un polinomio en x se puede considerar como una aplicación de un conjunto S (dominio de x) sobre un conjunto T (dominio de los valores del polinomio). En este caso *igualdad* implica idéntico a", así dos polinomios en x son iguales si tienen idéntica forma. Por ejemplo, $a + bx = c + dx$ si, y solamente si, $a = c$ y $b = d$ [41].

Formas polinómicas

Sea \mathbb{R} un anillo y sea x , que se llamará una *indeterminada*, un símbolo cualquiera que no pertenece a \mathbb{R} . Se entiende por polinomio en x sobre \mathbb{R} una expresión de la forma]

$$\alpha(x) = a_0x^0 + a_1x^1 + a_2x^2 + \cdots = \sum a_kx^k, a_i \in \mathbb{R} \quad (3.2)$$

en la que solamente un número finito de las a son diferentes de z , el elemento cero de \mathbb{R} . Dos polinomios en x sobre \mathbb{R} , $\alpha(x)$ tal como se acaba de definir, y

$$\beta(x) = b_0x^0 + b_1x^1 + b_2x^2 + \cdots = \sum b_kx^k, b_i \in \mathbb{R} \quad (3.3)$$

se dicen iguales, $\alpha(x) = \beta(x)$, siempre que $a_k = b_k$ para todos los valores de k .

Si en un polinomio como $\alpha(x)$ el coeficiente $a_n \neq z$, en tanto que todos los coeficientes de los términos que siguen son z , se dice que $\alpha(x)$ es de grado n y a_n se llama coeficiente dominante. Por ejemplo, el polinomio $a_0x^0 + a_1zx^1 + a_2zx^2 + \cdots$ es de grado cero con coeficiente dominante a_0 cuando $a_0 \neq z$ y no tiene grado (ni coeficiente dominante) si $a_0 = z$.

Polinomios mónicos

Sea \mathbb{R} un anillo con unidad u . Entonces $u = ux^0$ es la unidad de $\mathbb{R}[x]$ puesto que $ux^0 \cdot \alpha(x) = \alpha(x)$ para todo $\alpha(x) \in \mathbb{R}[x]$. Así que escribiendo $x = ux^1 = zx^0 + ux^1$, se tiene $x \in \mathbb{R}[x]$. Ahora bien, $a_k(x \cdot x \cdot x \cdots \text{con } k \text{ factores}) = a_kx^k \in \mathbb{R}[x]$ de modo que $\alpha(x) = a_0 + a_1x + a_2x^2 + \cdots$ podemos considerar el superíndice i en a_ix^i como un verdadero exponente, la yuxtaposición

en cualquier término $a_i x^i$ como multiplicación en el anillo (polinómica) y la conexión $+$ como adición en el anillo (polinómica)[41].

3.1.5. Órdenes monomiales

Si se examina a detalle el algoritmo de división y la eliminación Gaussianas, ordenar los términos de los polinomios es la clave en estos dos métodos. Al definir una división de polinomios de más de una variable, es importante establecer un orden en el conjunto de los monomios [44].

$$\mathbb{T}^n = \{x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \mid \alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n\} \quad (3.4)$$

Definición 3.4. Se define un orden monomial sobre \mathbb{T}^n como un orden total \prec que satisface:

1. $1 \prec x^\alpha$ para todo $x^\alpha \in \mathbb{T}^n$ distinto de 1.
2. Si $x^\alpha \prec x^\beta$ entonces $x^\gamma x^\alpha \prec x^\gamma x^\beta$ para todo $x^\gamma \in \mathbb{T}^n$.

Un orden monomial es compatible con la relación de la divisibilidad [45].

Proposición 3.1. Sea \prec un orden monomial sobre \mathbb{T}^n

1. Sean $x^\alpha, x^\beta \in \mathbb{T}^n$, si x^α divide a x^β entonces $x^\alpha \preceq x^\beta$.
2. Un orden monomial es un buen orden, esto es, para todo subconjunto $T \subseteq \mathbb{T}^n$ existe un $x^\alpha \in T$ tal que $x^\alpha \preceq x^\beta$ para todo $x^\beta \in T$.

Las definiciones 3.5 y 3.6 son dos de los órdenes monomiales más importantes:

Definición 3.5. (Orden lexicográfico). Sean $x^\alpha, x^\beta \in \mathbb{T}^n$ decimos que $x^\alpha \succ_{lex} x^\beta$ si la componente no nula más a la izquierda del vector $\alpha - \beta$ es positiva.

Definición 3.6. (Orden graduado reverso-lexicográfico). Sean $x^\alpha, x^\beta \in \mathbb{T}^n$ decimos que $x^\alpha \succ_{grevlex} x^\beta$ si $\sum_i \alpha_i > \sum_i \beta_i$ o si $\sum_i \alpha_i = \sum_i \beta_i$ entonces la componente no nula más a la derecha del vector $\alpha - \beta$ es negativo.

La proposición 3.2 extiende la división euclídea de polinomios en una sola variable al caso multivariable.

Proposición 3.2. Dado un orden monomial \succ en \mathbb{T}^n y sea f_1, \dots, f_n una s -upla ordenada de polinomios de R . Cada $f \in R$ puede ser expresada de la forma

$$f = u_1 f_1 + \cdots + u_s f_s + r \quad (3.5)$$

donde $u_i, r \in R$, tal que $r = 0$ o r es una combinación lineal de monomios no divisibles por los monomios $\{lt_\succ(f_i)\}_{i=1}^s$

El algoritmo de división multivariable da como resultado la proposición 3.2.

Entrada: f_1, \dots, f_s con $f_i \neq 0 (1 \leq i \leq s)$ y un orden monomial \prec

Salida: u_1, \dots, u_s, r tales que $f = u_1 f_1 + \dots + u_s f_s + r$, donde $r = 0$ o r es una combinación lineal de monomios no divisibles por los monomios $\{lt_{\succ}(f_i)\}_{i=1}^s$

- 1: $u_i \leftarrow 0, 1 \leq i \leq s, r \leftarrow 0, h \leftarrow f$
- 2: **mientras** $h \neq 0$ **hacer**
- 3: **si** existe un i tal que $lt_{\succ}(f_i)$ divide a $lt_{\succ}(h)$ **entonces**
- 4: $u_i \leftarrow u_i + \frac{lt_{\succ}(h)}{lt_{\succ}(f_i)}$
- 5: $h \leftarrow h - \frac{lt_{\succ}(h)}{lt_{\succ}(f_i)} f_i$
- 6: **si no**
- 7: $r \leftarrow r + lt_{\succ}(h)$
- 8: $h \leftarrow h - lt_{\succ}(h)$
- 9: **fin si**
- 10: **fin mientras**

3.2. Bases de Groebner

Las bases de Groebner siguen la siguiente estrategia: Dado un conjunto polinomial $F \subset K[x_1, \dots, x_n]$ y \mathbb{K} representa un campo de característica cero [38].

1. El conjunto polinomial F se transforma en otro conjunto polinomial G que tiene "propiedades agradables".
2. F y G son equivalentes, es decir, tienen el mismo principio.

El cálculo computacional en los polinomios multivariados no depende de la cantidad de polinomios que tengamos, sino de orden lexicográfico de los mismos. Para ordenar un conjunto de polinomios lexicográficamente, el elemento de menor grado en el polinomio debe estar en términos de una sola variable.

Este tipo de orden permite que se tenga un cálculo recursivo para las diferentes variables del conjunto de polinomios.

La teoría de las bases de Groebner es atractivo debido a diferentes factores, entre ellos se encuentran:

1. El problema principal puede ser resuelto de forma rápida si se conocen bien los polinomios a utilizar.
2. El algoritmo de Buchberger se puede comprender fácilmente.
3. Muchos problemas de las diferentes áreas de las matemáticas pueden ser resueltos mediante bases de Groebner.

Para comprender matemáticamente las Bases de Groebner se presentan las siguientes definiciones [45]:

Definición 3.7. Definimos un orden monomial sobre \mathbb{T}^n como un orden total \prec que satisface:

1. $1 \prec x^\alpha$ para todo $x^\alpha \in \mathbb{T}^n$ distinto de 1
2. Si $x^\alpha \prec x^\beta$ entonces $x^\gamma x^\alpha \prec x^\gamma x^\beta$ para todo $x^\gamma \in \mathbb{T}^n$

Definición 3.8. Diremos que $a = (a_1, \dots, a_n) \in \mathbb{T}^n$ es menor (respecto al orden lexicográfico) que $b = (b_1, b_2, \dots, b_n) \in \mathbb{T}^n$, y lo denotaremos por $a <_{lex} b$, si existe $i \in \{1, \dots, n-1\}$ tal que $a_j = b_j$, si $1 \leq j \leq i$, y $a_i < b_i$, o equivalentemente, si en la diferencia $b - a$ la primera coordenada distinta de cero de izquierda a derecha es positiva.

Por ejemplo, si tenemos $x_1 x_2 x_3^2 x_4^5$ y $x_1 x_2 x_3^3 x_4^2$, usando las definiciones 3.7 y 3.8 tenemos que

$$x_1 x_2 x_3^2 x_4^5 <_{lex} x_1 x_2 x_3^3 x_4^2 \quad (3.6)$$

ya que $(1, 1, 2, 5) <_{lex} (1, 1, 3, 2)$.

Definición 3.9. (Bases de Groebner). Dado un orden monomial \succ y un ideal $I \subset R$ se dice que el conjunto $f_1, \dots, f_s \subset R$ es una base de Groebner de I para \succ si se cumple

$$\langle lt_\succ(f_1), \dots, (f_s) \rangle = \langle lt_\succ(I) \rangle \quad (3.7)$$

donde $lt_\succ(I) = \{lt_\succ(f) \mid f \in I\}$ [45]

Las bases de Groebner son un sistema de generadores del ideal I , lo cual se demuestra a partir del teorema 3.1 [45].

Teorema 3.1. *Dado un orden monomial \succ y un ideal $I \subset R$ existe una base de Groebner de I , la cual es un sistema de generadores del ideal I .*

Demostración. El teorema de la base de Hilbert asegura que existe un sistema de generadores finito h_1, \dots, h_s para el ideal $\langle lt_\succ(I) \rangle$. Como $\langle lt_\succ(I) \rangle$ está generado por términos líderes de I se obtienen polinomios $g_1, \dots, g_s \in I$ tales que

$$h_i \in \langle lt_\succ(g_1), \dots, lt_\succ(g_s) \rangle, 1 \leq i \leq s$$

esto es

$$\langle lt_\succ(I) \rangle = \langle h_1, \dots, h_s \rangle \subset \langle lt_\succ(g_1), \dots, lt_\succ(g_s) \rangle \subset lt_\succ(I)$$

de donde se sigue que $\langle lt_\succ(I) \rangle = \langle g_1, \dots, g_s \rangle$ y por lo tanto el conjunto de polinomios g_1, \dots, g_s es una base de Groebner.

Proposición 3.3. Si $G = g_1, \dots, g_s$ es una base de Groebner del ideal $I \subset R$ entonces $f \in I$ si y sólo si el resto de la división multivariable de f por G es 0.

Proposición 3.4. Si $G = g_1, \dots, g_s$ es una base de Groebner del ideal $I \subset R$ para el orden monomial \succ y $f \in R$, entonces f puede ser escrito de forma única como

$$f = g + r \quad (3.8)$$

donde $g \in I$ y ningún término de r se puede dividir por $lt_\succ(g_i), 1 \leq i \leq s$.

3.2.1. Algoritmo de Buchberger

El algoritmo de Buchberger calcula la base de Groebner de un ideal dado por un sistema de generadores, todo esto a partir de los S-polinomios que son la combinación más simple de dos polinomios que cancelan sus términos líderes [45]. Los S-Polinomios dependen del orden monomial \succ que se elija.

Definición 3.10. (S-polinomio). Sean $f, g \in R$ dos polinomios no nulos y \succ un orden monomial. El S-polinomio de f y g es

$$S(f, g) = \frac{x^\gamma}{lt_\succ(f)} \cdot f - \frac{x^\gamma}{lt_\succ(g)} \cdot g \quad (3.9)$$

donde $x^\gamma = mcm\{lm_\succ(f), lm_\succ(g)\}$.

En la proposición 3.5 se aprecia que cualquier cancelación de términos líderes entre polinomios con el mismo multigrado es producto de una cancelación de S-polinomios.

Proposición 3.5. Dado un orden monomial \succ , supongamos que $f = \sum_{i=1}^s c_i f_i$, donde $c_i \in \mathbb{K}$ y $f_i \in R$ para $1 \leq i \leq s$; y el multigrado de f_i es $\delta = (\delta_1, \dots, \delta_n) \in \mathbb{Z}_{\geq 0}^n$ para todo $1 \leq i \leq s$. Si el multigrado de f es menor que δ entonces f es una combinación lineal con coeficientes en \mathbb{K} de los S-Polinomios.

$$S(f_j, f_k), \quad 1 \leq j, \quad k \leq s \quad (3.10)$$

Además cada S-Polinomio tiene un multigrado menor o igual que δ .

Con la proposición 3.5 podemos probar el criterio de Buchberger el cual es mostrado en la proposición 3.6. El criterio de Buchberger se utiliza para comprobar que un conjunto de polinomios multivariados es una base de Groebner.

Proposición 3.6. (Criterio de Buchberger). Sea \succ un orden monomial e I un ideal en R . Un sistema de generadores $G = \{g_1, \dots, g_s\}$ de I es una base de Groebner si y sólo si

$$\overline{S(g_i, g_j)}_\succ^G = 0, \quad 1 \leq i, j \leq s, i \neq j \quad (3.11)$$

El algoritmo de Buchberger calcula la base de Groebner de un ideal dado por un sistema de generadores, todo esto a partir de los S-polinomios son la combinación más simple de dos polinomios que cancela sus términos líderes.

El algoritmo de Buchberger es el siguiente:

Entrada: $F = f_1, \dots, f_s$ con $I = \langle \{f_i\}_{i=1}^s \rangle \neq \{0\}$ y \succ un orden monomial

Salida: Una base de Groebner para el ideal I en el orden monomial \succ

- 1: $G \leftarrow F, G' \leftarrow \{0\}$
- 2: **mientras** $G \neq G'$ **hacer**
- 3: $G' \leftarrow G$
- 4: **para** cada par $\{p, q\} \subset G'$ con $p \neq q$ **hacer**
- 5: $S \leftarrow \overline{S(p, q)}_{G'}$
- 6: **si** $S \neq 0$ **entonces**
- 7: $G \leftarrow G \cup \{S\}$
- 8: **fin si**
- 9: **fin para**
- 10: **fin mientras**

Capítulo 4

Desarrollo del algoritmo de marcha

Al diseñar un algoritmo de marcha se consideran varios aspectos, entre ellos se encuentran: la cinemática directa e inversa del robot, el criterio de estabilidad y los controles de posición.

La parte más importante del algoritmo de marcha es el cálculo de la cinemática inversa, sin esta no se pueden obtener los ángulos de las diferentes articulaciones del robot y por consiguiente no se podría seguir una trayectoria propuesta.

En este trabajo se hicieron 4 controles de posición. El primer control se emplea cuando el robot tiene los dos pies juntos como se indica en la figura 2.3. Antes de dar un paso, se debe verificar que el robot no se encuentre desbalanceado, si desde el principio no se tiene estabilidad en el robot, no se tendrá en el transcurso del paso.

Mientras el robot esté en el ciclo de marcha, se debe verificar que el CoM se encuentre dentro del polígono de soporte, esto es, que el ZMP coincida con el CoM. De esta tarea se encarga el segundo control de posición.

Por último se debe controlar que el robot no se caiga mientras da un paso, esto se dividió en dos etapas, cuando se encuentra en la fase de balanceo y cuando el robot en el transcurso del ciclo de marcha tiene grandes oscilaciones.

4.1. Cinemática inversa estimada con Bases de Groebner

La cinemática inversa de un manipulador se puede resolver con cálculos matemáticos que usan las bases de Groebner (para referencia ver el capítulo 3). Las bases de Groebner son de gran utilidad para calcular la cinemática inversa de los robots. El cálculo de la cinemática inversa de un robot es muy compleja, además de que la solución es exclusiva del robot.

Las bases de Groebner se calculan utilizando el algoritmo de Buchberger (para mayor información ver la sección C), el cual emplea álgebra computacional. El resultado no depende

del número de funciones que se tengan, sino del orden de las variables de las funciones. Para hacer uso de las bases de Groebner, las funciones deben tener un orden lexicográfico.

Las funciones al estar ordenadas, cuando se resuelven se hace de manera "amigable", debido a que las sustituciones se hacen de forma recursiva, encontrando las soluciones de todas las variables del problema.

Para el cálculo de la cinemática inversa del robot Nao se utilizaron las bases de Groebner, esto se hace a partir de proponer una posición final para los brazos y piernas del robot. Las funciones propuestas para calcular la cinemática inversa se propusieron utilizando el método geométrico (como referencia ver la sección 2.1).

4.1.1. Brazos del robot Nao

Los brazos del robot Nao se pueden ver como dos cadenas cinemáticas que trabajan en reflejo una de la otra, es decir, el brazo izquierdo es el reflejo del brazo derecho (ver la sección 2.6). En este caso se resolverá la cinemática inversa del brazo derecho. La obtención de las funciones a convertir a bases de Groebner son a partir del hombro a la muñeca como se muestra en la figura 4.1, donde $L_1 = 105$ mm y $L_2 = 113.7$ mm, de acuerdo al cuadro 2.1, L_1 corresponde a UpperArmLength y L_2 es la suma de LowerArmLength y HandOffsetX. Además, se propone que el ángulo $\theta_3 = 0$ como se muestra en la figura 4.2, que es la rotación en el codo, ya que en el algoritmo de marcha no se manipulan objetos.

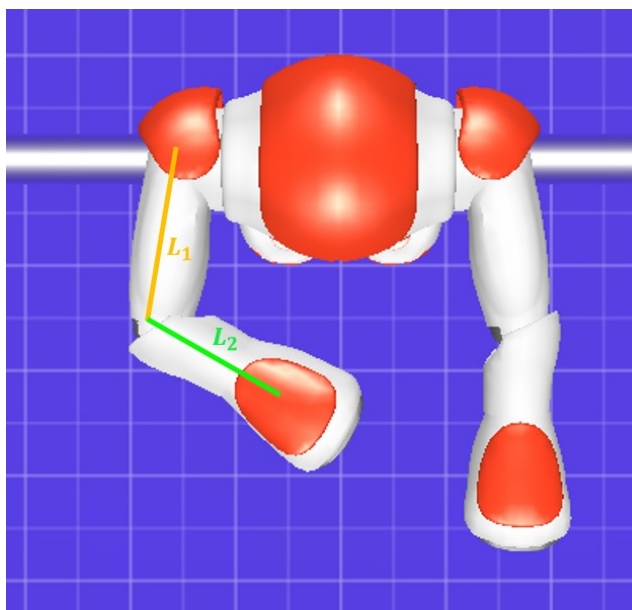


Figura 4.1: Cadena para calcular la cinemática inversa de los brazos del robot Nao.

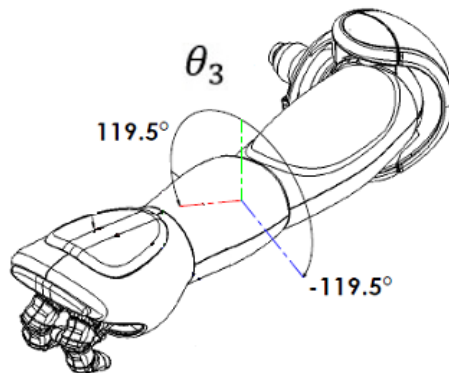


Figura 4.2: Ángulo θ_3 del brazo izquierdo del robot Nao [14].

La cadena cinemática del brazo derecho se puede ver como dos eslabones articulados como en la figura 4.3.

Los ángulos de la figura 4.3 se pueden representar como vectores como se muestra en la figura 4.4.

A partir de la figura 4.4 se obtienen las ecuaciones mostradas en 4.1 y 4.2. Estas ecuaciones resultan de aplicar identidades trigonométricas y de igualarlas con el vector $P = (p_y, p_z)$, donde el vector P representa la posición deseada en (p_y, p_z) , $s_2 = \sin(\theta_2)$, $c_2 = \cos(\theta_2)$, $s_4 = \sin(\theta_4)$ y $c_4 = \cos(\theta_4)$.

$$L_1 \cdot c_2 + L_2 \cdot s_4 - p_z = 0 \quad (4.1)$$

$$L_1 \cdot s_2 + L_2 \cdot c_4 - p_y = 0 \quad (4.2)$$

Para que el sistema de ecuaciones se pueda resolver, es necesario proponer otras dos ecuaciones (4.3 y 4.4) a partir de identidades trigonométricas:

$$c_2^2 + s_2^2 - 1 = 0 \quad (4.3)$$

$$c_4^2 + s_4^2 - 1 = 0 \quad (4.4)$$

Las ecuaciones de la 4.1 a la 4.4, se proponen como la solución a la cinemática inversa de los brazos del robot Nao. Estas ecuaciones se convierten a bases de Groebner con el software *Maple^TM*, con el orden lexicográfico c_4, c_2, s_4 y s_2 , se tienen 4 ecuaciones:

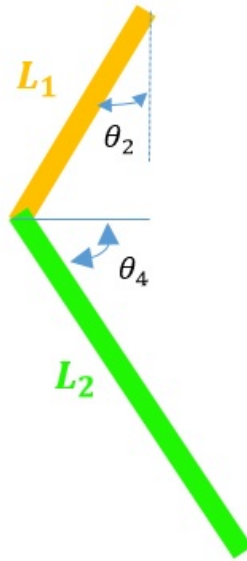


Figura 4.3: Cadena cinemática de los brazos del robot Nao.

$$\begin{aligned}
 & (4 \cdot L_1^2 \cdot p_y^2 + 4 \cdot L_1^2 \cdot p_z^2) \cdot s_2^2 \\
 & + (-4 \cdot L_1^3 \cdot p_y + 4 \cdot L_1 \cdot L_2^2 \cdot p_y - 4 \cdot L_1 \cdot p_y^3 - 4 \cdot L_1 \cdot p_y \cdot p_z^2) \cdot s_2 \\
 & + L_1^4 - 2 \cdot L_1^2 \cdot L_2^2 + 2 \cdot L_1^2 \cdot p_y^2 - 2 \cdot L_1^2 \cdot p_z^2 + L_2^4 - 2 \cdot L_2^2 \cdot p_y^2 - 2 \cdot L_2^2 \cdot p_z^2 \\
 & + p_y^4 + 2 \cdot p_y^2 \cdot p_z^2 + p_z^4 = 0
 \end{aligned} \tag{4.5}$$

$$-2 \cdot L_1 \cdot p_y \cdot s_2 + 2 \cdot L_2 \cdot p_z \cdot s_4 + L_1^2 - L_2^2 + p_y^2 - p_z^2 = 0 \tag{4.6}$$

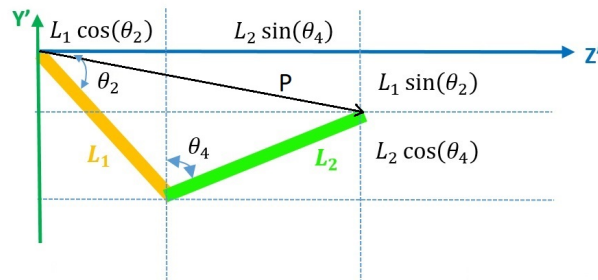


Figura 4.4: Ecuaciones propuestas para calcular la cinemática inversa de los brazos del robot Nao.

$$2 \cdot c_2 \cdot L_1 \cdot p_z + 2 \cdot L_1 \cdot p_y \cdot s_2 - L_1^2 + L_2^2 - p_y^2 - p_z^2 = 0 \quad (4.7)$$

$$c_4 \cdot L_2 + L_1 \cdot s_2 - p_y = 0 \quad (4.8)$$

Sustituyendo las variables $L_1 = 105$ y $L_2 = 113.7$ en las ecuaciones 4.5 a la 4.8, tenemos lo siguiente:

$$44100 \cdot (p_y^2 + p_z^2) \cdot s_2^2 - 420 \cdot (p_y^3 + p_y \cdot p_z^2) \cdot s_2 + 799129.8 \cdot p_y \cdot s_2 + 3620229.3 - 3805.38 \cdot p_y^2 - 47905.38 \cdot p_z^2 + p_y^4 + 2 \cdot p_y^2 \cdot p_z^2 + p_z^4 = 0 \quad (4.9)$$

$$p_y^2 - 210 \cdot s_2 \cdot p_y - p_z^2 + 227.40 \cdot p_z \cdot s_4 - 1902.69 = 0 \quad (4.10)$$

$$210 \cdot p_z \cdot c_2 - p_y^2 + 210 \cdot s_2 \cdot p_y - p_z^2 + 1902.69 = 0 \quad (4.11)$$

$$113.70 \cdot c_4 - p_y + 105 \cdot s_2 = 0 \quad (4.12)$$

Debido a que los senos nos proporcionan una solución signada de las variables a conocer, se toman 4.9 y 4.10. Las ecuaciones 4.11 y 4.12 son los cosenos de las variables a conocer, por lo que sería redundante resolverlas. Para conocer s_2 se utiliza la ecuación 4.13. Se toma la raíz negativa por las restricciones del robot. La solución positiva del polinomio cuadrático da una posición que el robot no puede alcanzar.

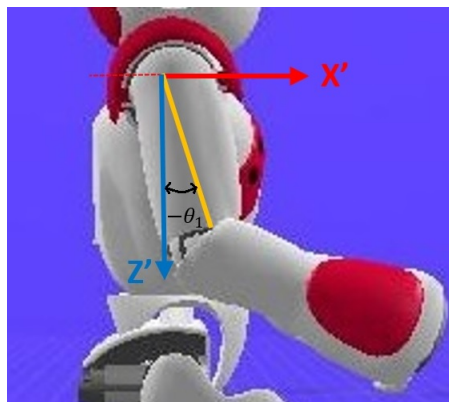
$$s_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (4.13)$$

Al sustituir los valores de a, b y c se tiene:

$$\begin{aligned} a &= 44100 \cdot (p_y^2 + p_z^2) \\ b &= -420 \cdot (p_y^3 + p_y \cdot p_z^2) + 799129.8 \cdot p_y \\ c &= 3620229.3 - 3805.38 \cdot p_y^2 - 47905.38 \cdot p_z^2 + p_y^4 + 2 \cdot p_y^2 \cdot p_z^2 + p_z^4 \end{aligned}$$

Al conocer el valor de s_2 se calcula s_4 para conocer los valores de θ_2 y θ_4 . El ángulo θ_1 se puede determinar a partir de la figura 4.5, en donde se puede observar que

$$\theta_1 = -atan(p_z, p_x) \quad (4.14)$$

Figura 4.5: Ángulo θ_1 del brazo del robot Nao.

Con estas ecuaciones se puede calcular la cinemática inversa de los brazos del robot Nao proponiendo un punto (p_x, p_y, p_z) que se encuentre dentro del área de trabajo. Los ángulos para el brazo izquierdo quedan de la siguiente forma:

$$\begin{aligned}
 \theta_{1I} &= \theta_{1D} \\
 \theta_{2I} &= -\theta_{2D} \\
 \theta_{3I} &= -\theta_{3D} \\
 \theta_{4I} &= -\theta_{4D}
 \end{aligned} \tag{4.15}$$

En donde $\theta_{1I} = \theta_1$ del brazo izquierdo, $\theta_{2I} = \theta_2$ del brazo izquierdo, $\theta_{3I} = \theta_3$ del brazo izquierdo, $\theta_{4I} = \theta_4$ del brazo izquierdo, $\theta_{1D} = \theta_1$ del brazo derecho, $\theta_{2D} = \theta_2$ del brazo derecho, $\theta_{3D} = \theta_3$ del brazo derecho y $\theta_{4D} = \theta_4$ del brazo derecho.

4.1.2. Piernas del robot Nao

Las piernas del robot Nao son simétricas (ver la sección 2.6 para mayor referencia), por lo que la estimación de la cinemática inversa se puede realizar solo a una de las dos cadenas. Las medidas que se considera para proponer las funciones a convertir a bases de Groebner van de la cadera al tobillo como se muestra en la figura 4.6, donde $L_1 = 100.0$ mm y $L_2 = 102.9$ mm. Además se propone $\theta_1 = 0$, debido a que no se requiere la rotación de cadera en el algoritmo de marcha.

Visto de otra forma, la cadena cinemática se forma por dos eslabones articulados como en la figura 4.7. Para obtener las funciones que describen el comportamiento de las articulaciones, los ángulos se pueden ver como en la figura 4.8.

Teniendo así las ecuaciones mostradas en 4.16 y 4.17. Resultantes de aplicar identidades trigonométricas y de igualar las ecuaciones mostradas en la figura 4.8 con el vector $P = (p_x, p_z)$,

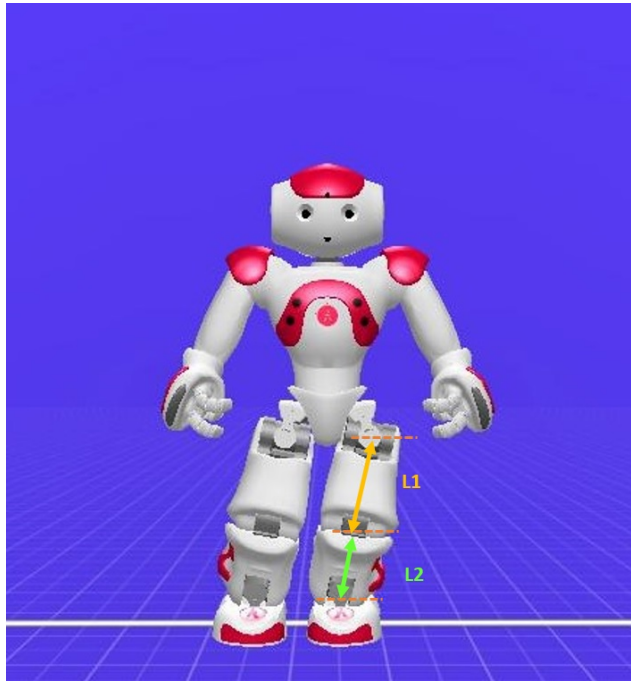


Figura 4.6: Cadena cinemática para calcular la cinemática inversa de las piernas del robot Nao.

en donde el vector P representa la posición deseada en (x, z) .

$$L_2 \cdot c_3 \cdot c_4 - L_2 \cdot s_3 \cdot s_4 + L_1 \cdot c_3 - p_z = 0 \quad (4.16)$$

$$L_2 \cdot c_3 \cdot s_4 + L_2 \cdot c_4 \cdot s_3 + L_1 \cdot s_3 - p_x = 0 \quad (4.17)$$

Estas ecuaciones aún no son suficientes. Para que el sistema de ecuaciones se pueda resolver se agregan las siguientes identidades trigonométricas:

$$c_3^2 + s_3^2 - 1 = 0 \quad (4.18)$$

$$c_4^2 + s_4^2 - 1 = 0 \quad (4.19)$$

Las ecuaciones de la 4.16 a la 4.19, donde $s_3 = \sin(\theta_3)$, $c_3 = \cos(\theta_3)$, $s_4 = \sin(\theta_4)$ y $c_4 = \cos(\theta_4)$, se proponen como un sistema de ecuaciones con el cual se puede resolver la cinemática inversa de las piernas. Estas ecuaciones se convierten a bases de Groebner con el software *Maple^TM*, siguiendo el orden lexicográfico c_4 , c_3 , s_4 y s_3 , se tienen 4 ecuaciones:

$$\begin{aligned} & (4 \cdot L_1^2 \cdot p_x^2 + 4 \cdot L_1^2 \cdot p_z^2) \cdot s_3^2 \\ & + (-4 \cdot L_1^3 \cdot p_x + 4 \cdot L_1 \cdot L_2^2 \cdot p_x - 4 \cdot L_1 \cdot p_x^3 - 4 \cdot L_1 \cdot p_x \cdot p_z^2) \cdot s_3 \\ & + L_1^4 - 2 \cdot L_1^2 \cdot L_2^2 + 2 \cdot L_1^2 \cdot p_x^2 - 2 \cdot L_1^2 \cdot p_z^2 + L_2^4 - 2 \cdot L_2^2 \cdot p_x^2 \\ & - 2 \cdot L_2^2 \cdot p_z^2 + p_x^4 + 2 \cdot p_x^2 \cdot p_z^2 + p_z^4 = 0 \end{aligned} \quad (4.20)$$

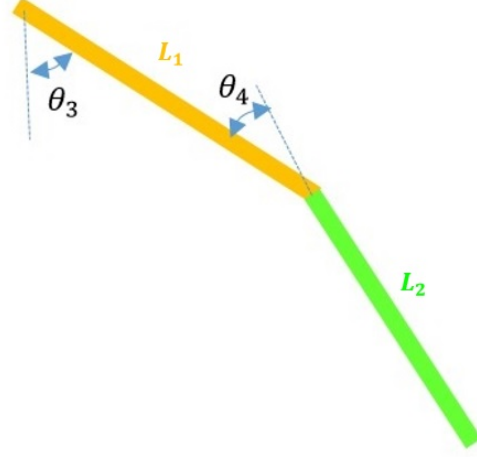


Figura 4.7: Cadena cinemática de las piernas del robot Nao visto como eslabones.

$$(2 \cdot L_1 \cdot p_x^2 + 2 \cdot L_1 \cdot p_z^2) \cdot s_3 + 2 \cdot L_2 \cdot s_4 \cdot L_1 \cdot p_z - L_1^2 \cdot p_x + L_2^2 \cdot p_x - p_x^3 - p_x \cdot p_z^2 = 0 \quad (4.21)$$

$$2 \cdot c_4 \cdot L_1 \cdot L_2 + L_1^2 + L_2^2 - p_x^2 - p_z^2 = 0 \quad (4.22)$$

$$2 \cdot c_3 \cdot L_1 \cdot p_z + 2 \cdot L_1 \cdot p_x \cdot s_3 - L_1^2 + L_2^2 - p_x^2 - p_z^2 = 0 \quad (4.23)$$

Al sustituir los valores de $L_1 = 100\text{mm}$ y $L_2 = 102.9\text{ mm}$ en las ecuaciones 4.20 a la 4.23, se tienen lo siguiente:

$$40000 \cdot (p_x^2 + p_z^2) \cdot s_3^2 - 400 \cdot (p_x^3 + p_x \cdot p_z^2) \cdot s_3 + 235364 \cdot p_x \cdot s_3 + 346226.3 - 1176.82 \cdot p_x^2 - 41176.82 \cdot p_z^2 + p_x^4 + 2 \cdot p_x^2 \cdot p_z^2 + p_z^4 = 0 \quad (4.24)$$

$$200 \cdot (p_x^2 + p_z^2) \cdot s_3 + 20580 \cdot p_z \cdot s_4 + 588.41 \cdot p_x - p_x^3 - p_x \cdot p_z^2 = 0 \quad (4.25)$$

$$-p_x^2 - p_z^2 + 20580 \cdot c_4 + 20588.41 = 0 \quad (4.26)$$

$$200 \cdot p_z \cdot c_3 - p_x^2 + 200 \cdot p_x \cdot s_3 - p_z^2 + 588.41 = 0 \quad (4.27)$$

De estas 4 ecuaciones se toman 4.24 y 4.25, ya que resolver las 4 ecuaciones es algo redundante. La ecuación 4.24 es un polinomio cuadrático, para resolverlo se utiliza la fórmula

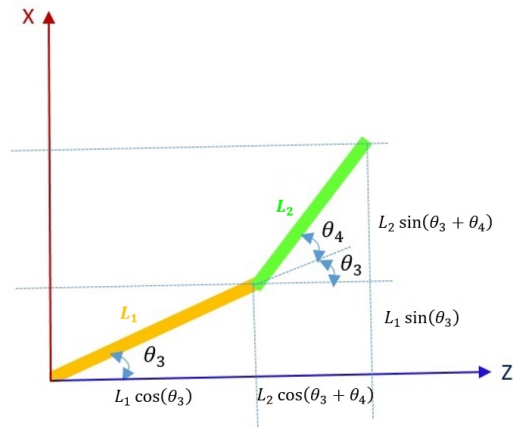


Figura 4.8: Ecuaciones propuestas para la estimación de la cinemática inversa de las piernas del robot Nao.

4.28.

$$s_3 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (4.28)$$

Al sustituir a, b y c de la fórmula 4.28 se tiene:

$$\begin{aligned} a &= 40000 \cdot (p_x^2 + p_z^2) \\ b &= -400 \cdot (p_x^3 + p_x \cdot p_z^2) + 235364 \cdot p_x \\ c &= 346226.3 - 1176.82 \cdot p_x^2 - 41176.82 \cdot p_z^2 + p_x^4 + 2 \cdot p_x^2 \cdot p_z^2 + p_z^4 \end{aligned}$$

Obtenida s_3 se calcula s_4 para conocer los valores de θ_3 y θ_4 .

Hasta el momento se conocen 3 ángulos, θ_1 , θ_3 y θ_4 . Los ángulos θ_2 y θ_6 se calculan a partir de la figura 4.9, en donde se puede observar que

$$\theta_6 = \text{atan}(p_y, p_z) \quad (4.29)$$

y que

$$\theta_2 = -\theta_6 \quad (4.30)$$

El ángulo θ_5 se obtiene a partir de θ_3 y θ_4 , como se muestra en la figura 4.10. Teniendo así:

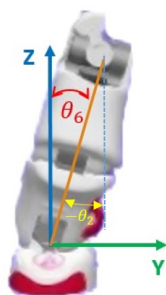


Figura 4.9: Ángulo θ_6 de la pierna del robot Nao.

$$\theta_5 = -(\theta_3 + \theta_4) \quad (4.31)$$

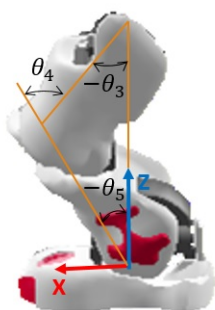


Figura 4.10: Ángulo θ_5 de la pierna del robot Nao.

Con estas ecuaciones se puede calcular la cinemática inversa de las piernas del robot Nao proponiendo un punto (p_x, p_y, p_z) que se encuentre dentro del área de trabajo.

4.2. Generador del patrón de marcha usando el criterio de estabilidad ZMP

El generador de marcha es el que se encarga de generar la trayectoria a seguir, dependiendo del movimiento que se necesite hacer, ya sea caminar en línea recta o girar en un mismo punto.

4.2.1. Generación de trayectorias

El diseño del generador de trayectorias considera dos cosas: los puntos de la trayectoria ZMP en el plano (x, y) como se muestra en la figura 4.11 y el tamaño del pie del robot, el cual mide 16.0 cm de longitud por 9.0 cm de anchura (ver la sección 2.4).

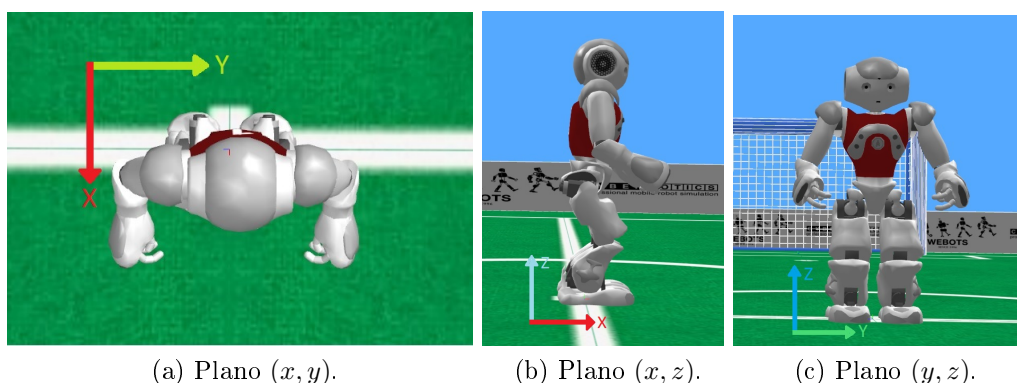


Figura 4.11: Ejes de referencia del robot Nao.

La trayectoria que debe seguir el robot Nao se generará mediante una función senoidal en el plano (x, y) con base en una distancia establecida x_d para hacer uso del criterio de estabilidad ZMP, debido a que una función senoidal proporciona un movimiento suave a comparación de una función escalón. Los puntos de la trayectoria deberán estar dentro del área del pie del robot en el plano (x, y) .

La trayectoria no solo se establece en un solo plano, la fase de balanceo (ver la sección 2.3) se hace en el plano (x, z) estableciendo la elevación z_e del pie con respecto al piso z_d . Esta trayectoria sigue una función sigmoide en z durante la fase de balanceo, como se muestra en la ecuación 4.32.

$$z(t) = z_d - \frac{z_e}{1 + e^{-t}} \quad (4.32)$$

El centro de masas ubicado en la cintura del robot sigue la trayectoria de la ecuación 4.33. La combinación de movimientos entre la pierna izquierda y la pierna derecha del robot en el eje y tiene la misma trayectoria, debido a que ambas están unidas en la cintura del robot.

$$y(x) = 50 \sin\left(\frac{\pi}{x_d} x\right) \quad (4.33)$$

La generación de la trayectoria mostrada en la figura 4.12 queda de la siguiente forma:

1. Cuando el pie derecho esté en el piso, al alcanzar una posición en el eje y igual a 50 mm, el pie izquierdo comienza la elevación para estar en la fase de balanceo como se muestra en la figura 4.13.

2. Cuando el pie izquierdo este en el piso, al alcanzar una posición en la dirección negativa del eje y igual a -50 mm, el pie derecho comienza la elevación para estar en la fase de balanceo como se muestra en la figura 4.14.

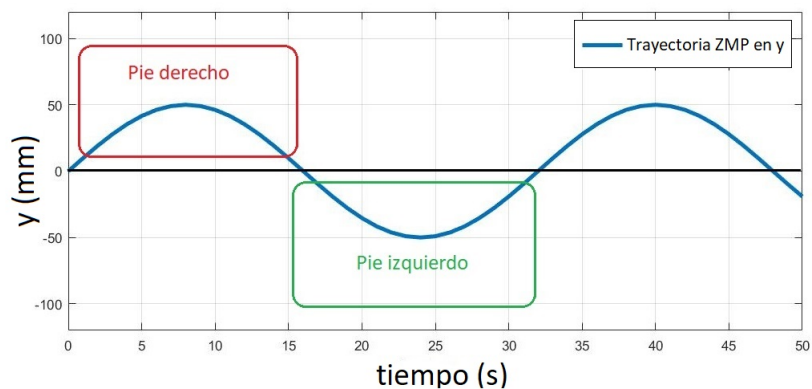


Figura 4.12: Sincronización de los pies del robot durante la caminata siguiendo la trayectoria ZMP en y .

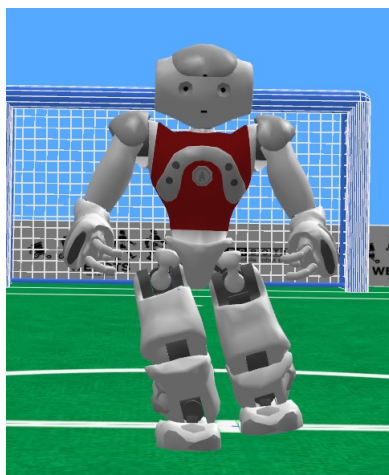


Figura 4.13: Elevación del pie izquierdo del Nao.

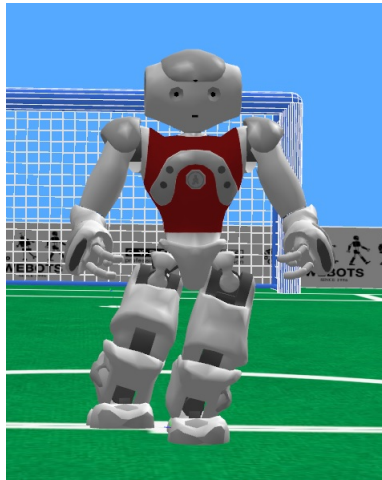


Figura 4.14: Elevación del pie derecho del Nao.

4.3. Control de posición del robot cuando está en reposo

Los robots bípedos existentes se han dividido en dos grupos: basados en el control de posición y en el control de fuerza o torque [46]. Se va a emplear un controlador difuso usando los servomotores del robot Nao. Con los servomotores y el giroscopio se obtiene el ángulo de la postura del cuerpo [47].

Este control se utilizará cuando el robot esté con los dos pies juntos en la fase de doble soporte (ver la sección 2.3), así como se muestra en la figura 2.3.

Usando la estrategia de tobillo [48] para mantener el balance cuando sus articulaciones no tengan los ángulos correspondientes para dar un paso o si el robot quedó sobre una superficie un tanto irregular, se proponen las siguientes características para el controlador difuso (ver la sección 2.5):

- Variables de entrada. Se tienen los ángulos de los servomotores y el giroscopio del robot.
- Variables de salida. Se tiene la posición de las articulaciones de las piernas.

Con esto se propone como variables lingüísticas para cada motor de ambas piernas:

- EP : Error positivo.
- EZ : Error cero.
- EN : Error negativo.

Estas variables dicen si el robot tiene un error en su posición cuando se habla de los servomotores o si está inclinado cuando se trata de la señal del giroscopio. El error en el giroscopio indica en x si hay una inclinación lateral hacia la derecha (error negativo EN) o si

hay una inclinación lateral hacia la izquierda (error positivo EP). Por otro lado, el error en y indica si está inclinado hacia adelante (error negativo EN) o si está inclinado hacia atrás (error positivo EP). A partir de esto, se establecen las salidas como:

- MP : Mover positivo.
- NM : No mover.
- MN : Mover negativo.

Con esto se obtienen las reglas de control:

- if EP, then MP.
- if EZ, then NM.
- if EN, then MN.

Los servomotores que se controlan en el plano sagital son los correspondientes a los ángulos *HipPitch, *KneePitch y *AnklePitch (θ_3 , θ_4 y θ_5 respectivamente). En el plano frontal se controlan los servomotores correspondientes a *HipRoll y *AnkleRoll (θ_2 y θ_6 respectivamente), en donde * se sustituye por R o L (para referencia ver la figura 2.10). De esta forma se tienen 10 ángulos a controlar, 6 en el plano sagital y 4 en el plano frontal.

El controlador revisa si alguno de los servomotores no alcanzó la posición correspondiente con un error de $\pm 2^\circ$. Si los servomotores alcanzaron la posición correspondiente, verifica que el giroscopio no esté inclinado ya sea hacia los lados, hacia enfrente o hacia atrás con un error de $\pm 2^\circ$.

El algoritmo del control es el siguiente:

Entrada: Se tiene los ángulos de los servomotores y el giroscopio del robot

Salida: Se tiene la posición de las articulaciones de las piernas

- 1: **mientras** El robot esté en reposo **hacer**
- 2: Obtener ángulos de los servomotores en ambas piernas y el giroscopio
- 3: Estimar error
- 4: **si** Existe error **entonces**
- 5: Aplicar las reglas de control
- 6: **fin si**
- 7: **fin mientras**

4.4. Control de posición de la cadera en el ciclo de marcha

Es posible hacer una aproximación ZMP usando los sensores de aceleración del robot Nao empleando las ecuaciones 4.34 y 4.35 [49]. Así, la posición dada por la aproximación ZMP estará dada por (p_x, p_y) (ver la sección 2.4).

$$p_x = X - \frac{Z_h}{g} \ddot{x} \quad (4.34)$$

$$p_y = Y - \frac{Z_h}{g} \ddot{y} \quad (4.35)$$

En donde:

- p_x : posición ZMP en x.
- p_y : posición ZMP en y.
- X : desplazamiento en x.
- Y : desplazamiento en y.
- Z_h : altura del CoM.
- g : gravedad.

Como variables principales para calcular el ZMP se tienen las aceleraciones traslacionales y rotacionales deseadas, las cuales no deberán mover el centro de masas (CoM) por ningún motivo, si llegase a pasar esto, el robot se desestabilizaría y caería.

Lo primero que tiene que hacer el control de posición es convertir las aceleraciones en la posición deseada de la cadera del robot, esto a partir de las ecuaciones 4.34 y 4.35. La posición deseada en z se mide a partir de dos sistemas de referencia que se encuentran en cada uno de los tobillos del robot Nao hacia la cadera y las posiciones en x y y se miden a partir del centro de masas que se encuentra en el torso del robot.

El control de posición calcula la aproximación ZMP a partir de puntos dados y verifica que estos puntos estén dentro del polígono de soporte [49] con un error de 5 mm. Dentro del polígono de soporte se puede elegir cualquier punto, ya que este garantiza una caminata estable.

Se proponen las siguientes características para el controlador difuso (ver la sección 2.5):

- Variables de entrada. La señal del giroscopio en el robot Nao.
- Variables de salida. La posición de las articulaciones de las piernas.

A partir de esto se propone como variables lingüísticas para determinar la posición del punto ZMP deseado:

- EPx : Error positivo en x (no llega al punto).
- EPy : Error positivo en y (no llega al punto).
- EZ : Error cero.

- EN_x : Error negativo en x (se pasa del punto).
- EN_y : Error negativo en y (se pasa del punto).

A partir de esto, se establecen las salidas como:

- MR_x : Mover rápido en x.
- MR_y : Mover rápido en y.
- NM : No mover.
- ML_x : Mover lento en x.
- ML_y : Mover lento en y.

Con esto se obtienen las reglas de control para la parte positiva de la trayectoria ZMP descrita por la ecuación 4.33:

- if EN_x, then MR_x.
- if EN_y, then MR_y.
- if EZ, then NM.
- if EP_x, then ML_x.
- if EP_y, then ML_y.

Las reglas de control para la parte negativa de la trayectoria ZMP descrita por la ecuación 4.33:

- if EN_x, then MR_x.
- if EN_y, then ML_y.
- if EZ, then NM.
- if EP_x, then ML_x.
- if EP_y, then MR_y.

Las reglas de control para la parte positiva y negativa descritas por la ecuación 4.33 validan las posiciones de todos los puntos que conforman el ciclo de marcha. Los puntos que se consideraron críticos se presentan cuando levanta alguno de sus pies, ya que en estos puntos se reduce considerablemente el polígono de soporte, caso contrario de cuando están los dos pies apoyados en el piso.

Como la caminata es hacia adelante, si la aceleración en x es mayor a cero entonces la posición alcanzada es menor a la posición deseada. Si va muy rápido no llega al punto deseado, por lo que tiene que ir más lento. Si la aceleración en x es menor a cero entonces la posición alcanzada es mayor a la posición deseada. Si va muy lento no llega al punto deseado, por lo que tiene que ir más rápido.

Para el caso en y se tienen dos posiciones, una positiva y otra negativa. En la parte positiva, si la aceleración en y es mayor a cero entonces la posición alcanzada es mayor a la posición deseada (error positivo) y si es menor a cero entonces la posición alcanzada es menor a la posición deseada (error negativo). En la parte negativa, si la aceleración en y es mayor a cero entonces la posición alcanzada es menor a la posición deseada (error positivo) y si es menor a cero entonces la posición alcanzada es mayor a la posición deseada (error negativo).

Los servomotores que se controlan son los correspondientes a los ángulos *HipRoll, *HipPitch, *KneePitch, *AnklePitch y *AnkleRoll($\theta_2, \theta_3, \theta_4, \theta_5$ y θ_6 respectivamente), en donde * se sustituye por R o L, teniendo así 10 posiciones a controlar.

El algoritmo del control es el siguiente:

1. MIENTRAS (esté en movimiento) HACER
 - a) Obtener posición (x,y) del ZMP deseado.
 - b) Estimar error de posición
 - c) Aplicar reglas de control
2. fin MIENTRAS

4.5. Control del equilibrio del robot Nao en el ciclo de marcha

Una marcha natural y estable es una combinación de movimientos en los planos frontal y sagital (ver las sección 2.3). El movimiento en el plano frontal se utiliza para la dinámica de balanceo y es en donde se verifica la trayectoria ZMP. En el plano sagital se establecen los movimientos obtenidos de la biomecánica del humano [11].

Para mantener en equilibrio al robot Nao mientras camina se emplea la estrategia de paso [48]. Cuando el robot tiene una gran oscilación ya sea en el plano frontal o sagital (para referencia ver la figura 4.11) mientras está en movimiento, se establece que de un paso lateral hacia la derecha.

Viendo al robot Nao como un péndulo invertido, se consideró para el diseño del algoritmo que si tiene mas de 5 oscilaciones fuera del rango $\pm 10^\circ$ corrigiera su estabilidad. Lo primero que hace es mover el brazo derecho como se muestra en la figura 4.15.

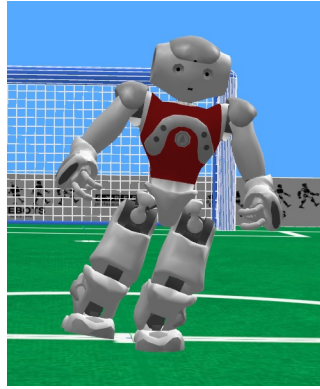


Figura 4.15: Movimiento del brazo derecho para corregir estabilidad del robot.

Una vez que mueve el brazo derecho, separa las piernas dando un paso lateral hacia la derecha hasta llegar a la posición mostrada en la figura 4.16 al mismo tiempo que mueve el brazo izquierdo, esto con el fin de tener un polígono de soporte con una mayor área. De esta forma quedan equilibradas las masas en el robot.

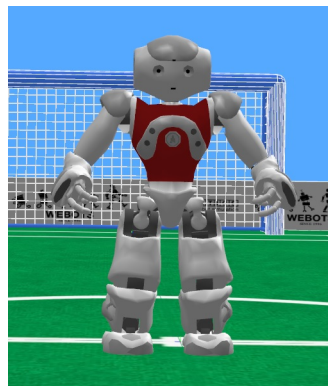


Figura 4.16: Movimiento de las piernas y brazo izquierdo para corregir estabilidad del robot.

Después de esperar 3 segundos, el robot regresa a la posición inicial como se muestra en la figura 4.17 para continuar su trayectoria.

El control difuso para el equilibrio en el ciclo de marcha del robot Nao tiene como características al giroscopio, que es la variable de entrada, y como variables de salida tiene a los servomotores de brazos y piernas del robot.

Con esto se propone como variables lingüísticas para los ángulos del giroscopio:

- EP_x : Error positivo en x.
- EP_y : Error positivo en y.
- EZ : Error cero.
- EN_x : Error negativo en x.

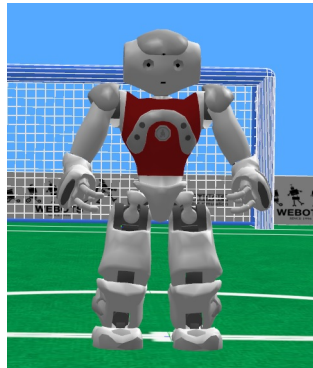


Figura 4.17: Posición final después de corregir la estabilidad del robot.

- EN_y : Error negativo en y .

Estas variables indican si el robot se encuentra dentro del rango deseado, es decir, si se encuentra en $-10^\circ \leq E_{px}, E_{py}, EN_x, EN_y \leq 10^\circ$. El error positivo en x (EP_x) del robot indica que el robot tiene una oscilación hacia el lado izquierdo y el error negativo en x (EN_x) que tiene una oscilación hacia el lado derecho. Por otro lado, el error positivo en y (EP_y) indica que el robot tiene una oscilación hacia atrás y el error negativo en y que tiene una oscilación hacia adelante. A partir de esto, se establecen las salidas como:

- M : Dar paso lateral hacia la derecha.
- NM : No mover.

Con esto se obtienen las reglas de control:

- if EN_x, then M.
- if EN_y, then M.
- if EZ, then NM.
- if EP_x, then M.
- if EP_y, then M.

Al levantar el pie, este no siempre se queda en una posición paralela al piso, si no que cae. Al caer el pie, como se mueve la pierna hacia adelante, se atora con las irregularidades del pasto. Para corregir esto, se diseñó un control que mantuviera el tobillo a un ángulo de -41° . Este ángulo se obtuvo a partir de la experimentación.

Este control tiene como características a las articulaciones RAnklePitch y LAnklePitch (como referencia ver la figura 2.10), que se encuentran en el plano sagital. Estos ángulos son las variables de entrada y como salida se tienen a los mismos ángulos.

A partir de esto se propone como variables lingüísticas para los ángulos del tobillo de cada pierna:

- EP : Error positivo.
- EZ : Error cero.
- EN : Error negativo.

Estas variables nos dicen si el tobillo se encuentra en una posición adecuada para avanzar hacia adelante. Con esto podemos dar la salida:

- MP : Mover positivo.
- NM : No mover.
- MN : Mover negativo.

Así, las reglas de control son:

- if EP, then MP.
- if EZ, then NM.
- if EN, then MN.

El algoritmo de control es el siguiente:

Entrada: Se tiene los ángulos a los servomotores del tobillo y al giroscopio

Salida: Se tiene la posición de las articulaciones de las piernas y brazos

- 1: **mientras** El robot esté en movimiento **hacer**
- 2: Obtener ángulos de los servomotores de los tobillos en ambas piernas y el giroscopio
- 3: Estimar error
- 4: **si** Existe error **entonces**
- 5: Aplicar las reglas de control
- 6: **fin si**
- 7: **fin mientras**

Capítulo 5

Pruebas y resultados

El ciclo de marcha es una combinación de movimientos entre las piernas y los brazos. A partir de la experimentación, se observó que el braceo no es de suma importancia durante el ciclo de marcha del robot. En este caso los brazos del robot Nao se dejaron fijos, en una posición que este cerca del torso del mismo para evitar momentos de torsión. Así mismo, la velocidad con la cual se mueven los servomotores del robot Nao es baja, para evitar que el robot se desestabilice. Un ciclo de marcha se ve como el mostrado en la figura 5.1.

La marcha mostrada en la figura 5.1 se describe como:

1. Al empezar la marcha se tiene una posición recta (figura 5.1a).
2. La cadera del robot se desplaza en la dirección positiva del eje y siguiendo una trayectoria dada por la ecuación 4.33 (figura 5.1b).
3. Al alcanzar la cresta de la ecuación 4.33, es decir, una vez que llegue a la posición máxima deseada en y , el robot levantará el pie derecho (figura 5.1c).
4. A continuación se empieza a elevar el pie derecho hasta una distancia z_e , al mismo tiempo este se desplaza hacia adelante mientras se deja fija la pierna izquierda (figura 5.1d).
5. Al alcanzar la distancia deseada en x , se baja el pie derecho (figura 5.1e).
6. La cadera del robot se mueve en la dirección negativa del eje y siguiendo la trayectoria de la ecuación 4.33 (figura 5.1f).
7. Al llegar a la posición máxima deseada en $-y$ se eleva el pie izquierdo (figura 5.1g).
8. Mientras el pie izquierdo este elevado una distancia z_e , este se desplaza hacia adelante hasta alcanzar la misma posición del pie derecho (figura 5.1h).
9. El robot baja el pie izquierdo y se desplaza en la dirección positiva del eje y hasta llegar a la posición inicial (figura 5.1i).

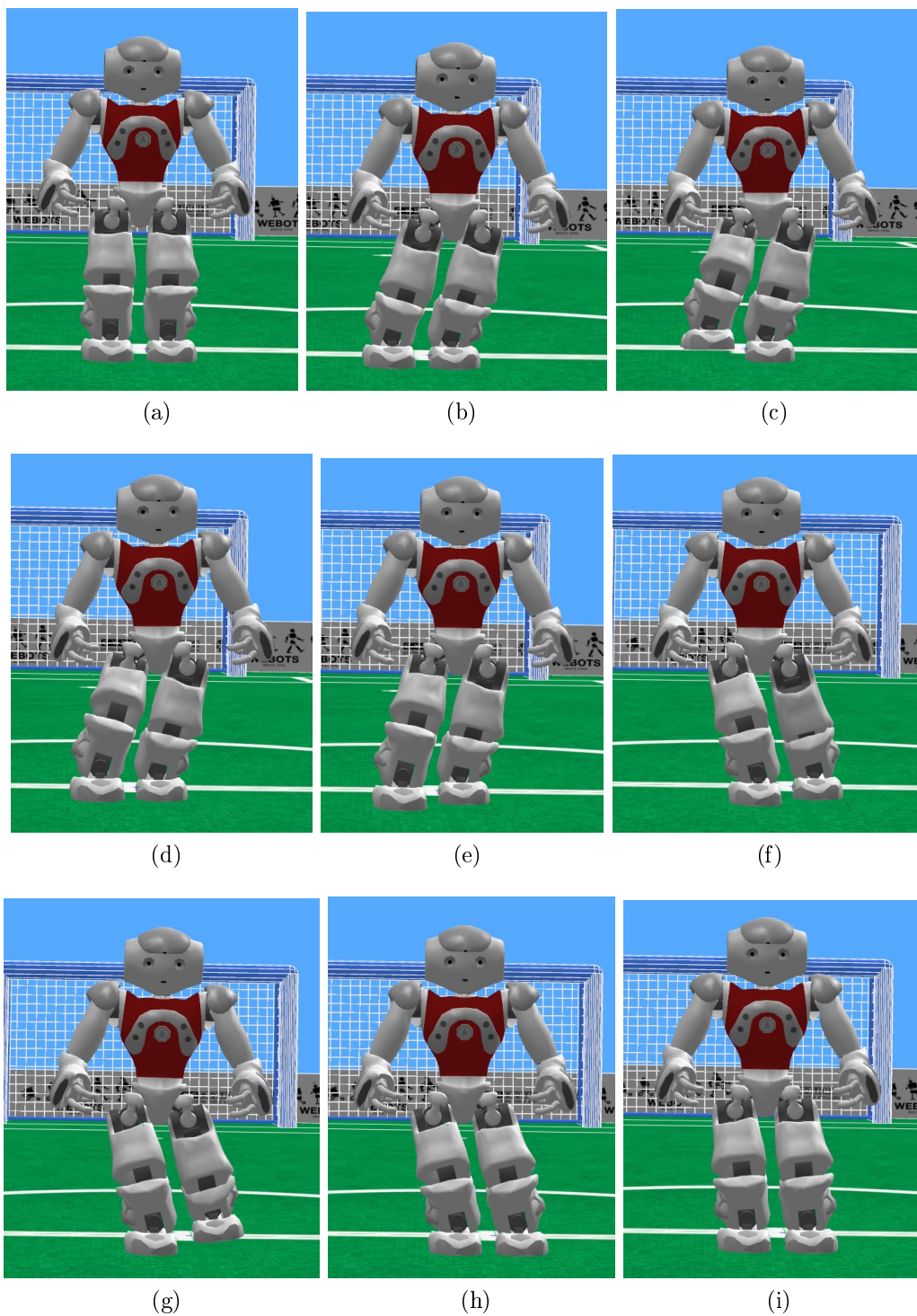


Figura 5.1: Ciclo de marcha del robot Nao simulado en WEBOTS.

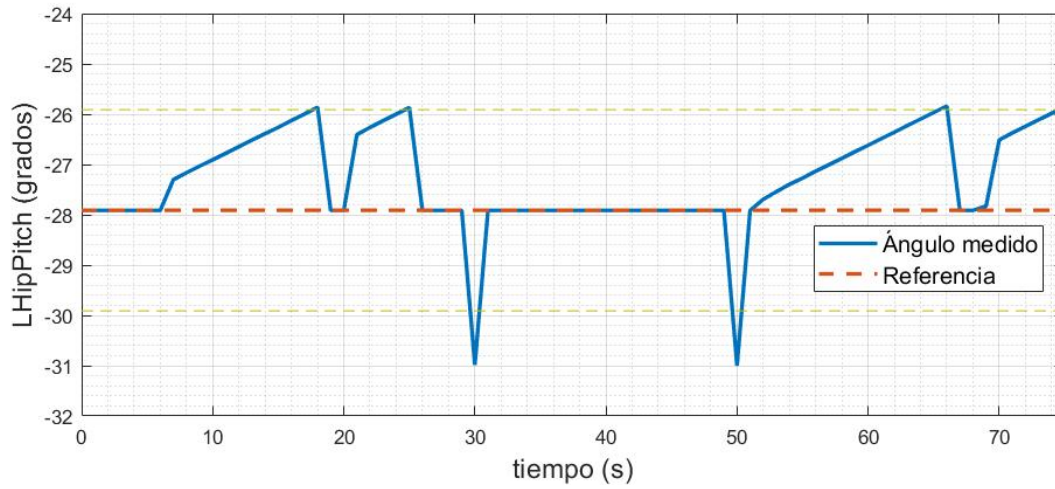
Durante el ciclo de marcha, se utilizan varios controladores para mantener estable al robot, ya sea cuando el robot está en reposo o cuando está en movimiento. Los controladores difusos que se utilizaron para generar la simulación de la marcha y para implementarlos en el robot Nao son:

1. El que verifica la posición del robot antes de que empiece un nuevo paso. Debido a que el ciclo de marcha se constituye de dar de uno a varios pasos, si el robot no se encuentra en una posición adecuada, es más probable que este caiga.
2. El que permite comprobar que los pies no rocen con el pasto en la fase de balanceo, esto con el fin de evitar tropiezos. El robot levanta el pie de la pierna que está en la fase de balanceo, con esto se comprueba que el robot no roza el pasto, evitando que se atore la pierna oscilante.
3. El que estabiliza el robot por medio de un paso, cuando este tiene muchas oscilaciones mientras está en movimiento.
4. El que verifica que el robot está siguiendo la trayectoria ZMP propuesta. En caso de no seguir esta trayectoria, el controlador corrige la posición del robot.

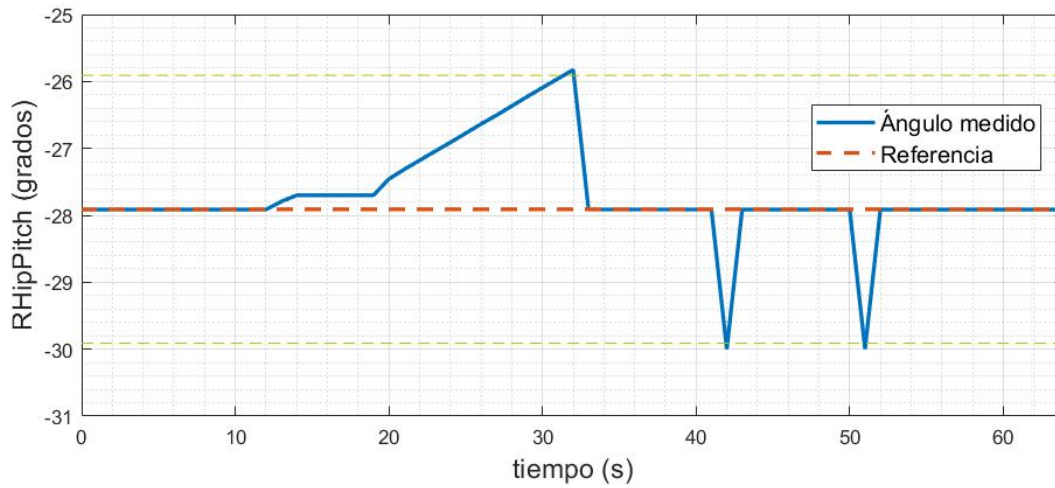
5.1. Control de posición del robot cuando está en reposo

Cuando el robot va a empezar un paso, se encuentra con los dos pies juntos. En este punto se verifica que el robot se encuentre en una posición considerada estable para empezar a moverse siguiendo el algoritmo que se muestra en la sección 4.3. El monitoreo de los ángulos de cada pierna se lleva al inicio de cada paso, en caso de estar en reposo el monitoreo de los ángulos de las articulaciones de las piernas se realiza cada 0.5 segundos. Si llegase a existir un error en la posición en las articulaciones del robot, el controlador cuando está en reposo el robot lo corrige en el plano sagital como se muestra en las figuras 5.2, 5.3 y 5.4, evitando que se caiga hacia adelante o que se vaya hacia atrás.

Por ejemplo, en la articulación LHipPitch el ángulo de referencia es -27.91° . Tomando los datos de la figura 5.2a, se tiene como entrada un ángulo de -25.86° , calculando el error entre el ángulo de referencia y el de entrada se tiene que $e = -27.91^\circ - (-25.86^\circ) = -2.05^\circ$. Este error es menor a -2° y le corresponde la variable lingüística *error negativo (EN)*, por lo que se aplica la regla del control difuso *if EN, then MN*. En donde *MN* es *mover negativo*.

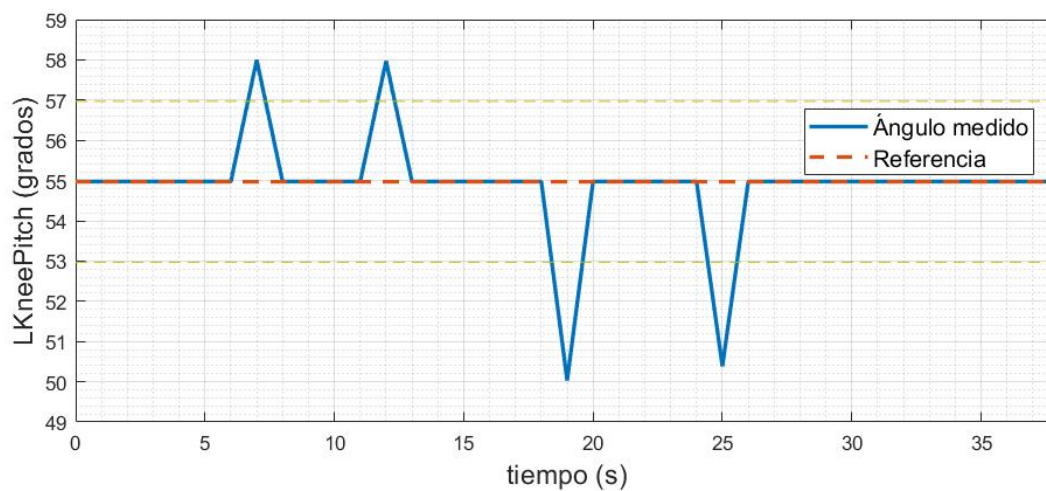


(a) Articulación LHipPitch correspondiente al ángulo θ_3 de la pierna izquierda

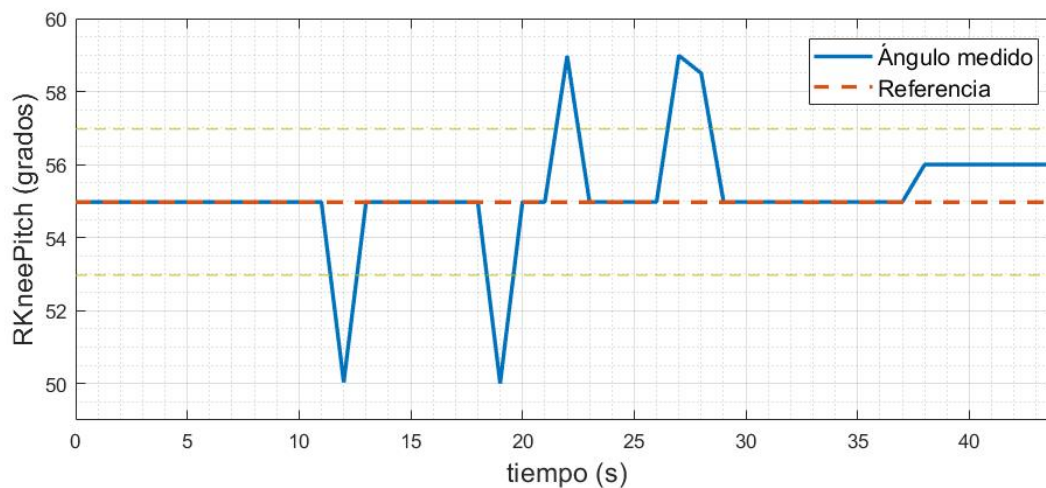


(b) Articulación RHipPitch correspondiente al ángulo θ_3 de la pierna derecha

Figura 5.2: Gráficas donde se muestra el comportamiento del control difuso en la cadera del robot en el plano sagital mientras está en reposo.

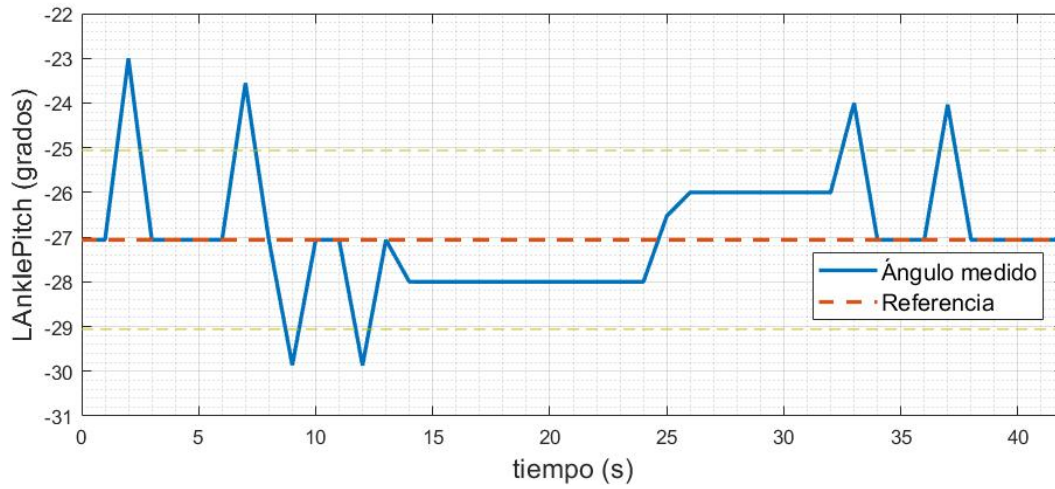


(a) Articulación LKneePitch correspondiente al ángulo θ_4 de la pierna izquierda

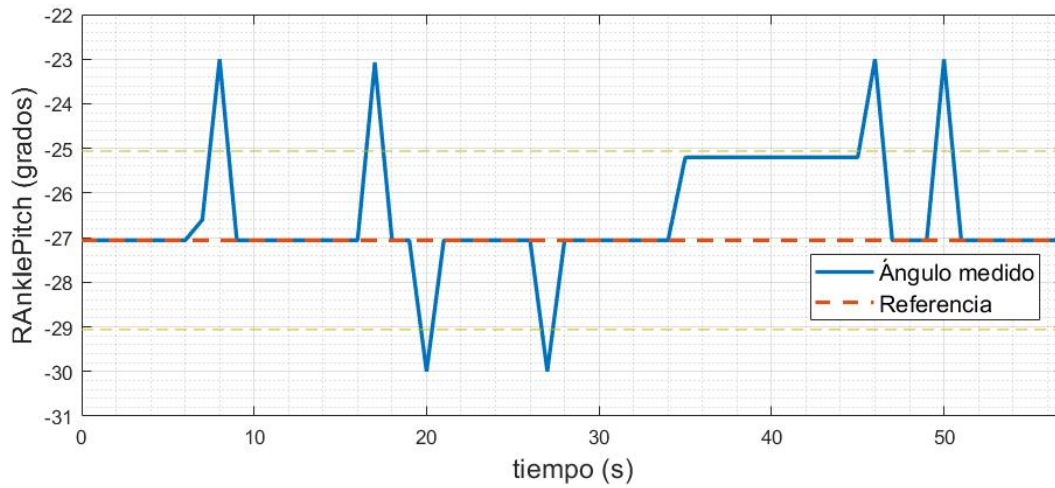


(b) Articulación RKneePitch correspondiente al ángulo θ_4 de la pierna derecha

Figura 5.3: Gráficas donde se muestra el comportamiento del control difuso de las rodillas del robot en el plano sagital mientras está en reposo.



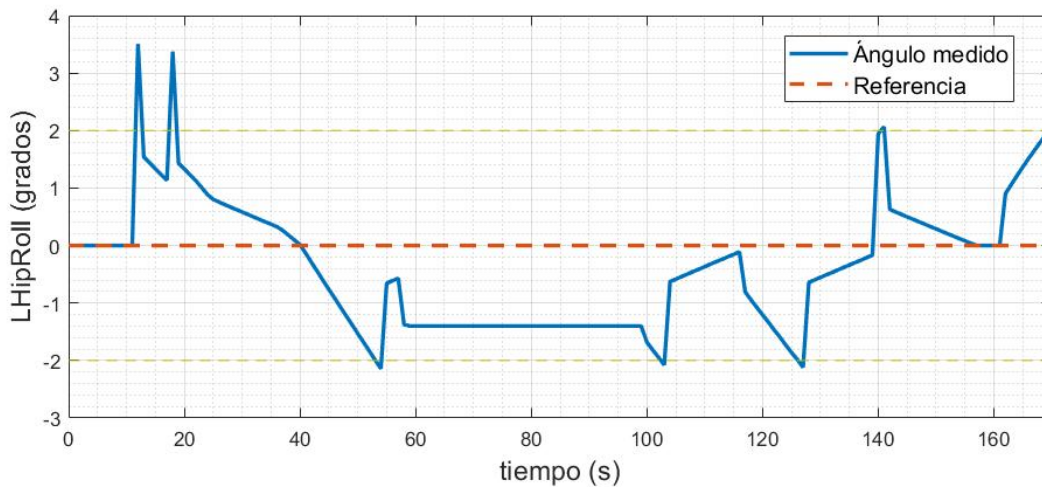
(a) Articulación LAnklePitch correspondiente al ángulo θ_5 de la pierna izquierda



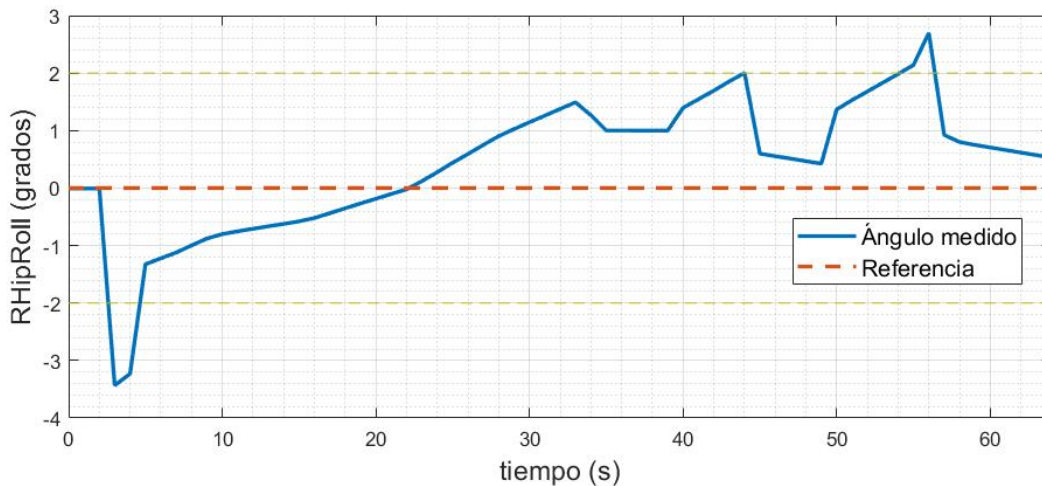
(b) Articulación RAnklePitch correspondiente al ángulo θ_5 de la pierna derecha

Figura 5.4: Gráficas donde se muestra el comportamiento del control difuso en los tobillos del robot en el plano sagital mientras está en reposo.

Además de los errores que pueden ocurrir en el plano sagital, se tienen los errores en el plano frontal. Estos pueden provocar que el robot caiga hacia alguno de los lados. Al igual que en el plano sagital, los servomotores son monitoreados cada segundo y en caso de existir algún error se corrige como se muestra en las figuras 5.5 y 5.6.

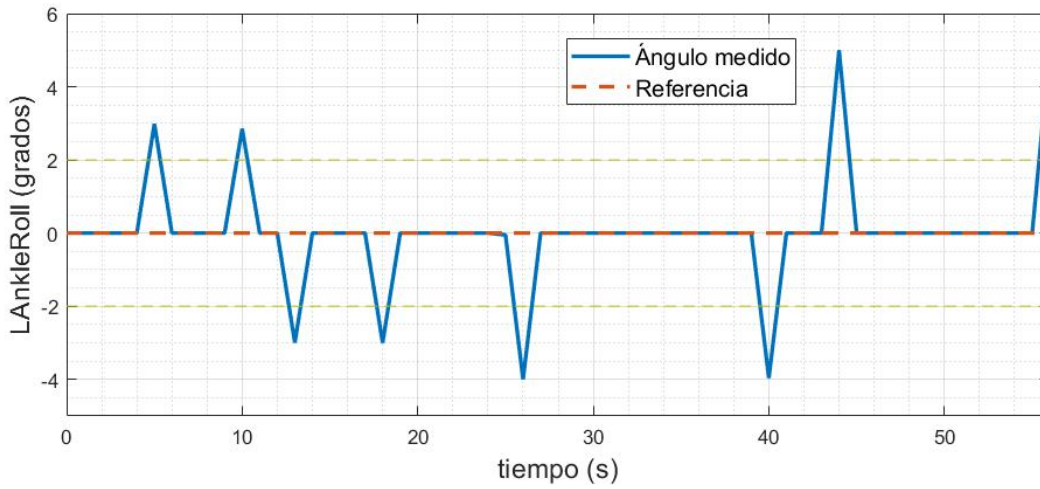


(a) Articulación LHipRoll correspondiente al ángulo θ_2 de la pierna izquierda

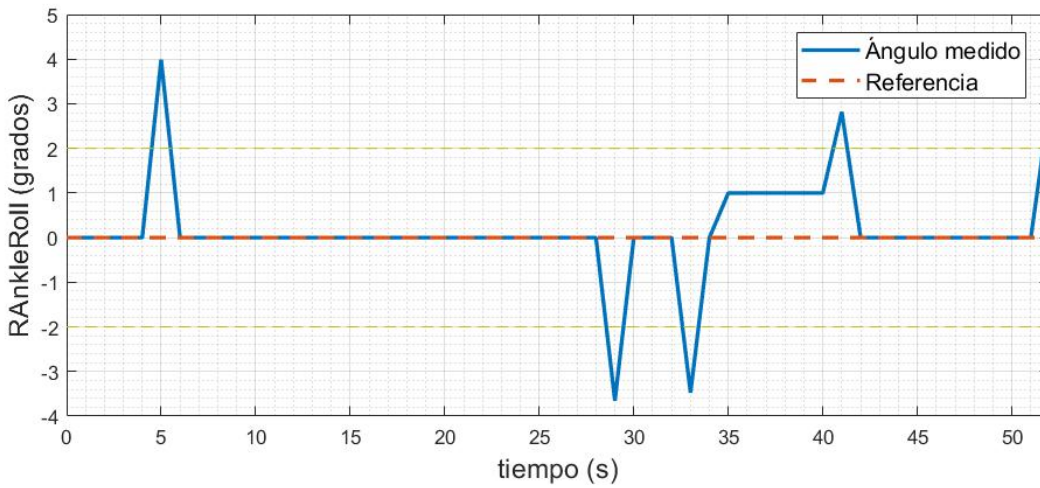


(b) Articulación RHipRoll correspondiente al ángulo θ_2 de la pierna izquierda

Figura 5.5: Gráficas donde se muestra el comportamiento del control difuso en la cadera del robot en el plano frontal mientras está en reposo.



(a) Articulación LAnkleRoll correspondiente al ángulo θ_6 de la pierna izquierda

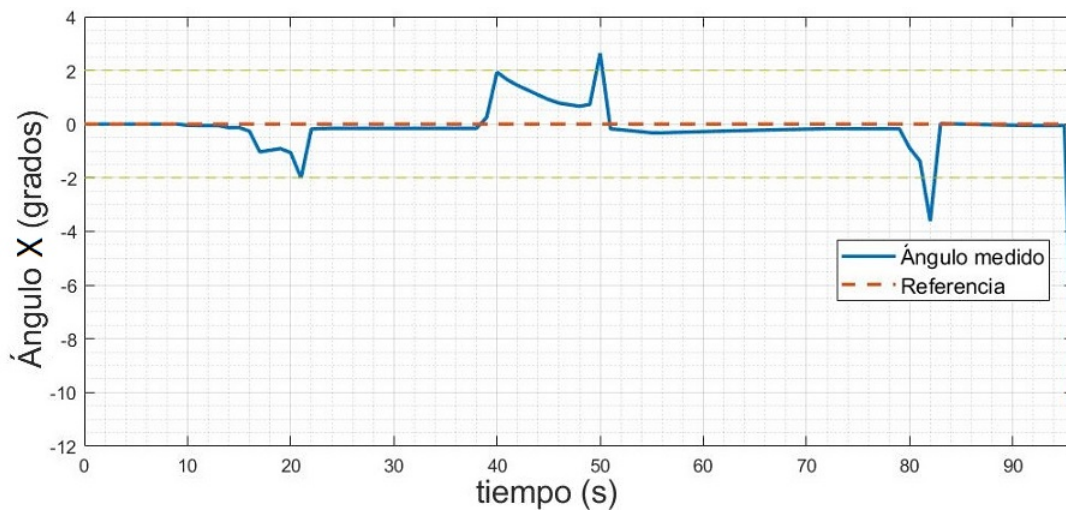


(b) Articulación RAnkleRoll correspondiente al ángulo θ_6 de la pierna izquierda

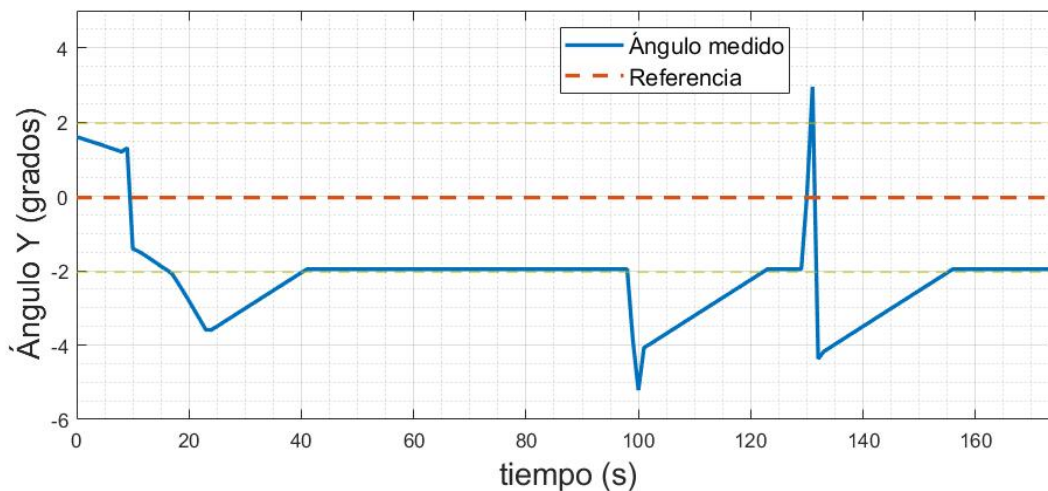
Figura 5.6: Gráficas donde se muestra el comportamiento del control difuso en los tobillos del robot en el plano frontal mientras está en reposo.

En caso de que los servomotores tengan la posición correspondiente cuando el robot está en reposo, se verifica la posición del giroscopio, si el giroscopio está inclinado, se corrige la posición como se muestra en la figura 5.7, los ángulos medidos por el giroscopio deben estar siempre entre -2° y 2° .

En la figura 5.7a, que es la posición del giroscopio en el robot en el eje x , se tiene como referencia un ángulo de 0° . Tomando como ángulo de entrada 2.637° y calculando el error entre el ángulo de referencia y el de entrada se tiene que $e = 0^\circ - 2.637^\circ = -2.637^\circ$. Este error sobrepasa el permitido en el rango de -2° a 2° y le corresponde a la variable lingüística *error negativo (EN)*, por lo que se aplica la regla de control *if EN, then mover negativo (MN)*.



(a) Ángulo medido por el giroscopio correspondiente a x

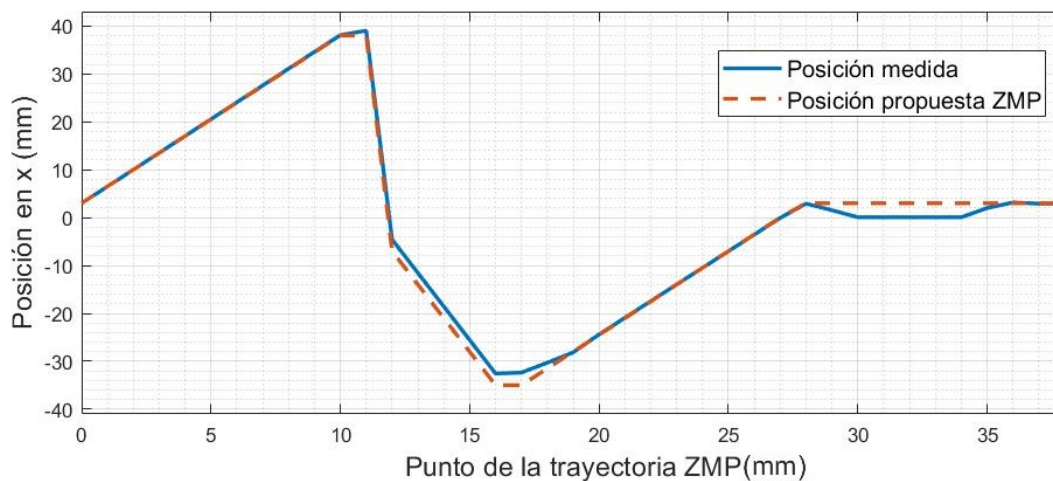


(b) Ángulo medido por el giroscopio correspondiente a y

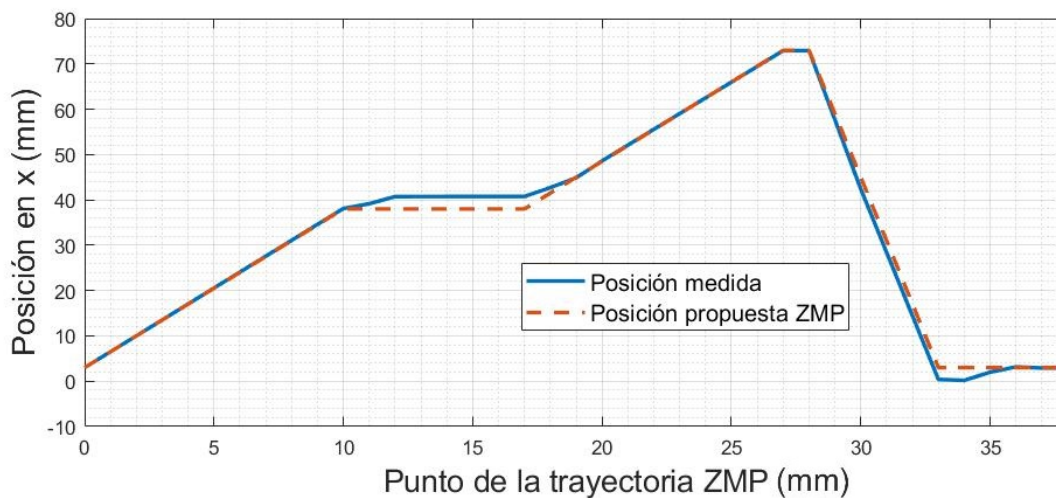
Figura 5.7: Gráficas donde se muestra el comportamiento del control difuso cuando el robot se encuentra inclinado mientras está en reposo.

5.2. Control de posición de la cadera en el ciclo de marcha

La posición ZMP se calcula a partir de la aceleración en x y la aceleración en y . Estas aceleraciones las proporciona el IMU del robot Nao. La trayectoria ZMP en x y y se puede observar en las figuras 5.8 y 5.9 respectivamente, en donde se aprecia que el error entre la posición propuesta y la calculada con las ecuaciones 4.34 y 4.35 es muy pequeña. Además, se comprueba que el control ZMP corrige la posición del robot en los puntos críticos como se muestra en la sección 4.4. Estos puntos son cuando se va a levantar ya sea el pie derecho o el pie izquierdo.



(a) Trayectoria ZMP en el eje x de la pierna derecha



(b) Trayectoria ZMP en el eje x de la pierna izquierda

Figura 5.8: Gráficas donde se muestra el comportamiento del control difuso en la trayectoria ZMP en x .

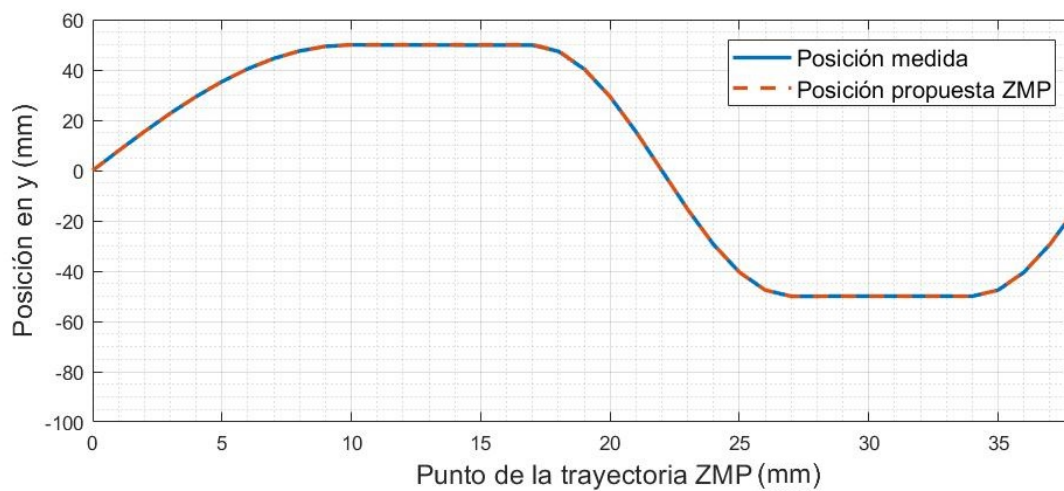
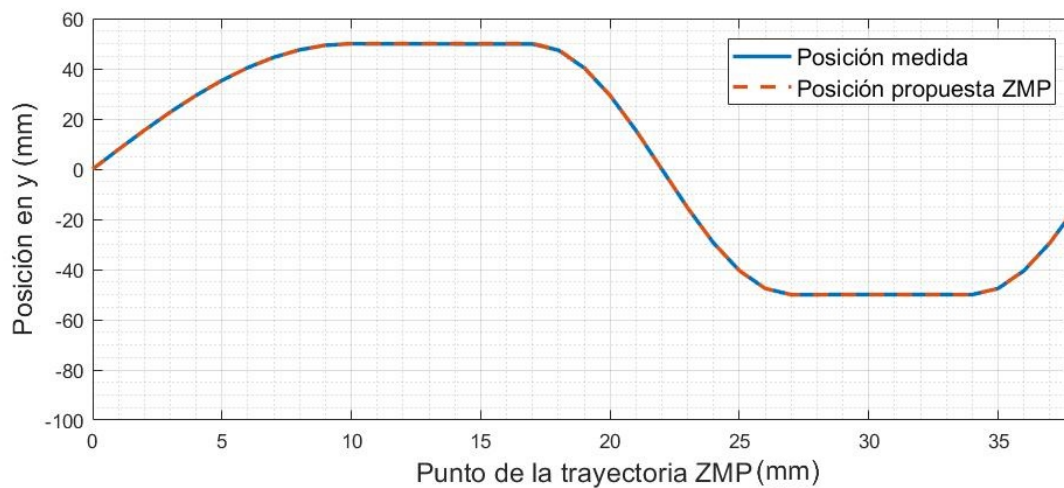
(a) Trayectoria ZMP en el eje y de la pierna derecha(b) Trayectoria ZMP en el eje y de la pierna izquierda

Figura 5.9: Gráficas donde se muestra el comportamiento del control difuso en la trayectoria ZMP en y .

5.3. Control del equilibrio del robot Nao en el ciclo de marcha

Cuando el robot está en movimiento, por las irregularidades del pasto, se presentan oscilaciones en el robot. Si estas oscilaciones llegan a ser demasiado grandes, se implementa un control difuso que corrige estas oscilaciones dando un paso lateral (ver la sección 4.5). Por cada punto de la trayectoria propuesta se revisa la posición del giroscopio, si esta excede los $\pm 10^\circ$ se considera una oscilación grande.

En la figura 5.10 se aprecian oscilaciones grandes en el robot. Después de 5, el robot da un paso lateral en los 80 segundos de la figura 5.10 y se corrige la estabilidad del robot. Las oscilaciones son más pequeñas.

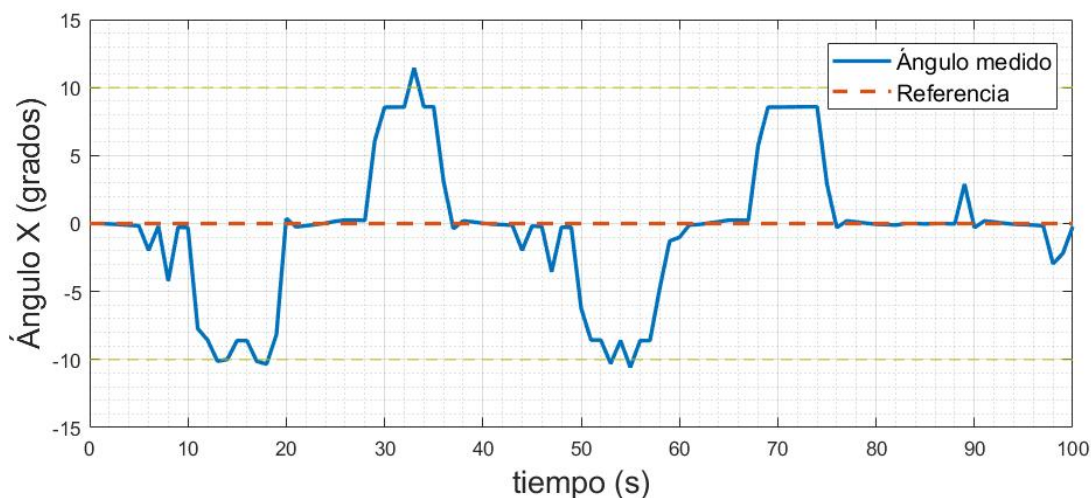


Figura 5.10: Posición x del giroscopio en el robot Nao mientras camina.

Al aplicar el control de equilibrio en el robot mientras está en movimiento, se verifica que el tobillo del robot se encuentre en una posición de -41° . Esta posición se obtuvo a partir de la experimentación en el robot. Cuando el robot levanta alguno de los pies, estos llegan a rozar con el pasto, haciendo que el robot tropiece. Este controlador difuso permite levantar el pie mientras el robot se encuentre en la fase de balanceo. La posición del tobillo se corrige como se muestra en las figuras 5.11 y 5.12.

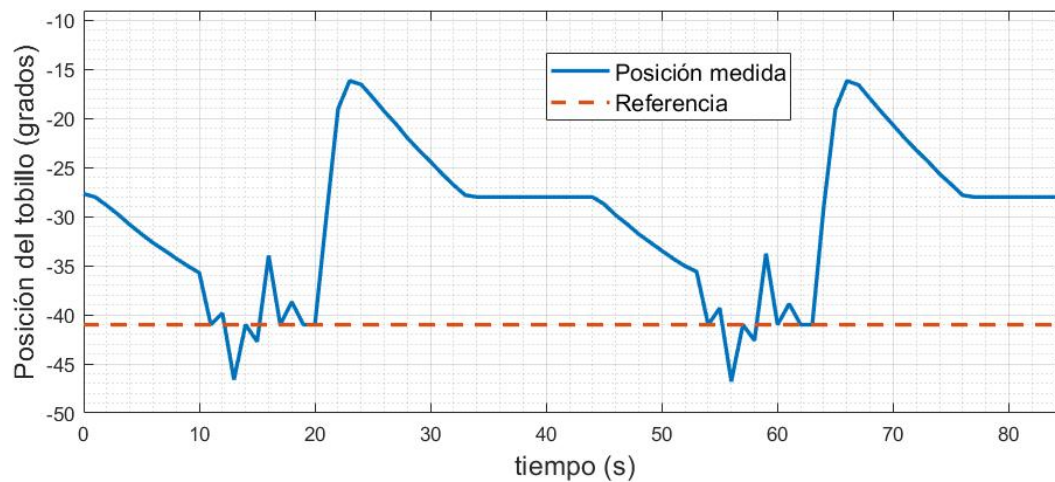


Figura 5.11: Movimiento del tobillo derecho para corregir la estabilidad del robot.

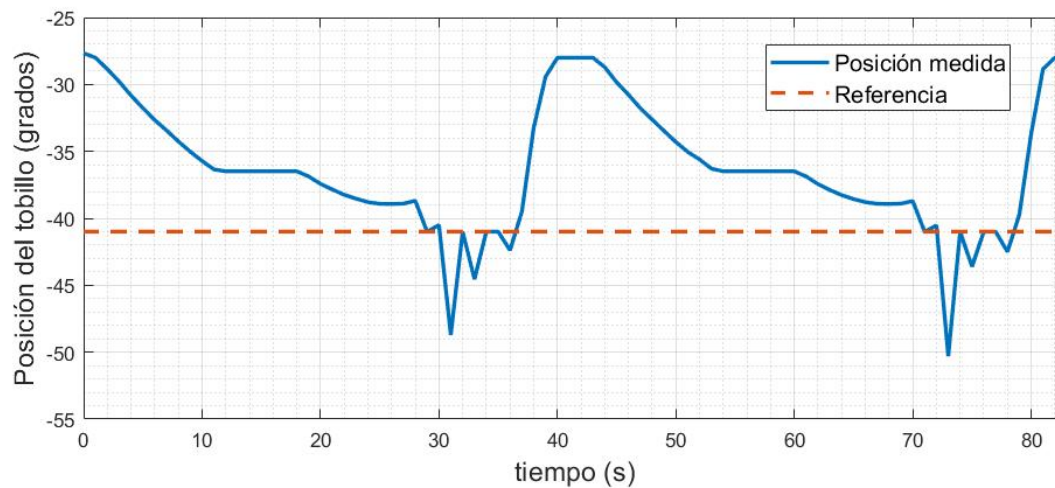


Figura 5.12: Movimiento del tobillo izquierdo para corregir la estabilidad del robot.

En la figura 5.11 se muestra el comportamiento del tobillo derecho en el plano sagital durante dos pasos. Cada paso está compuesto por 38 puntos, del punto 11 al 17 se levanta el pie derecho. Por ejemplo, el control difuso para mantener en equilibrio al robot tiene como referencia -41° y se tiene una entrada de -46.61° , calculado el error tenemos que $e = -41^\circ - (-46.61^\circ) = 5.61^\circ$. Este error es positivo y le corresponde la variable lingüística *EP*, por lo que la regla de control a aplicar es *if EP, then MP*.

La corrección en la posición del tobillo visto en el robot se aprecia en la punta de los pies como en las figuras 5.14 y 5.13.

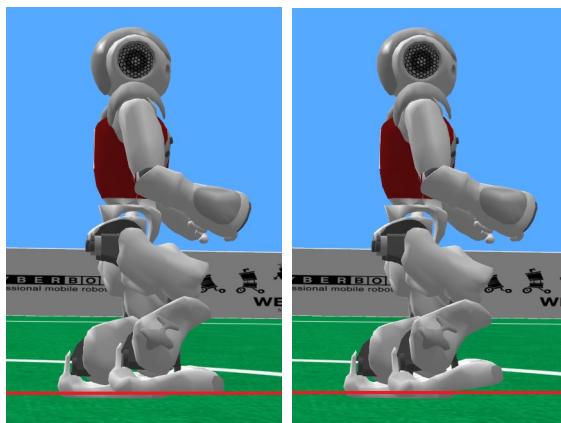


Figura 5.13: Movimiento del tobillo derecho en el robot mientras está en la fase de balanceo.

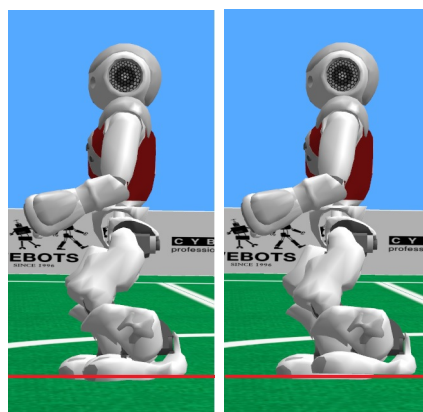


Figura 5.14: Movimiento del tobillo izquierdo en el robot mientras está en la fase de balanceo.

5.4. Implementación del algoritmo de marcha en el robot Nao

El algoritmo de marcha (incluyendo los controladores difusos) se implementó en el robot Nao, dando un buen resultado cuando camina sobre pasto artificial con una duración de 34 segundos por cada paso, como se muestra en la figura 5.15. Además, aquí se puede observar que comparando la figura 5.1 con la 5.15 los movimientos son iguales, y se valida el funcionamiento del sistema de control implementado en las articulaciones de las piernas, ya que sin este podría llegar a caerse el robot.

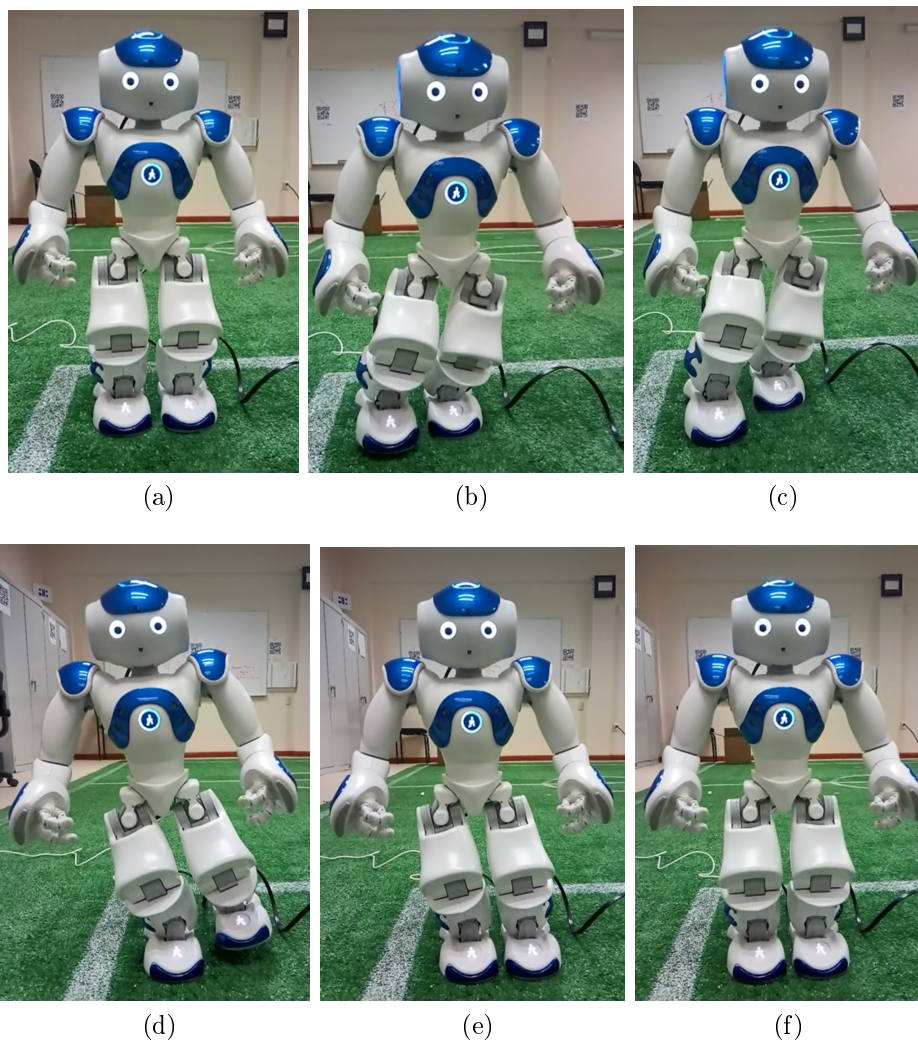


Figura 5.15: Ciclo de marcha implementado en el robot Nao.

Capítulo 6

Conclusiones y Trabajos Futuros

6.1. Conclusiones

Se buscan herramientas que puedan facilitar el cálculo de la cinemática inversa de cualquier mecanismo o robot, debido a que es muy difícil de obtener. Mientras más grados de libertad (GDL) tenga un robot, el cálculo de la cinemática inversa se complica más. En este trabajo, se propuso utilizar la teoría de la bases de Groebner.

La teoría de las bases de Groebner, proporciona todas las posibles soluciones que tiene la cinemática inversa de un robot, no importando el número de grados de libertad que este tenga. Respetando las restricciones de movimiento del robot se seleccionan las ecuaciones que se van a utilizar para calcular la cinemática inversa.

Además de proponer la ecuaciones que describen la cinemática directa del robot para calcular la cinemática inversa del mismo, se debe establecer un orden lexicográfico para obtener la solución. Si no existe un orden lexicográfico adecuado, tal vez, no se pueda hacer un cambio de base y por consiguiente, no se obtendrá una solución del problema.

Al implementar las ecuaciones calculadas con la teoría de las bases de Groebner en el algoritmo de marcha para el robot Nao, se observó que siguió los puntos ZMP propuestos, con lo cual se valida que el cálculo de la cinemática inversa fue correcto, obteniendo con esto una marcha en línea recta hacia adelante.

En la práctica, se observó que la sincronización de brazos y piernas no afectaba mucho el comportamiento del robot en el ciclo de marcha. Esto, debido a que la masa de los brazos del robot es mayor en los brazos [14] y su centro de masas (CoM) se encuentra pegado al torso, como se muestra en la figura A.10 (para referencia ver el Anexo A de este trabajo). Por esto, se decidió no aplicar dicha sincronización.

El robot Nao tiene una caminata lenta con el algoritmo de marcha diseñado en este trabajo de tesis, en comparación con otros algoritmos de marcha como la función de caminata en NAOqi [14]. Una de las posibles causas por las que el robot tiene un avance lento es el número de puntos que tiene la trayectoria, debido al tiempo de interpolación (0.8 s). En este caso no se puede comparar los tiempos de interpolación ya que NAOqi es arquitectura cerrada.

Además del criterio de estabilidad ZMP, se propusieron cuatro controladores difusos (el que verifica la posición antes de dar el paso, para comprobar que los pies no rocen el pasto en la fase de oscilación, para estabilizar al robot si hay muchas oscilaciones y para verificar que el robot siga la trayectoria ZMP), los cuales comprueban que el robot tenga estabilidad, no solo cuando está en movimiento, sino que también cuando el robot se encuentra en reposo.

Analizando los resultados obtenidos de la IMU del robot Nao, se pudo determinar que los controladores estaban funcionando correctamente, de esta forma el robot se mantiene en equilibrio cuando está caminando.

Con este trabajo se puede comprobar que utilizando el criterio de estabilidad ZMP, la teoría de las bases de Groebner y un control difuso, un robot bípedo puede caminar de una manera considerada estable en superficies planas con pasto artificial.

6.2. Trabajos Futuros

El criterio de estabilidad propuesto en este trabajo dio buenos resultados, sin embargo, se requiere de hacer algunas mejoras en la implementación. Una de estas mejoras sería cambiar la trayectoria ZMP en el plano (x, y) , en lugar de usar una trayectoria senoidal, cambiarla por una trayectoria sigmoidea. Una trayectoria sigmoidea evitaría el exceso de movimiento en la cadera del robot. También, agregar una trayectoria en z , en lugar de que la cadera se mantenga fija, que siga una trayectoria curva.

La trayectoria ZMP en este trabajo, se realizó con la obtención de las aceleraciones en (x, y) . Los controles de las aceleraciones en los mecanismos son muy imprecisos. Por lo que se propone utilizar los sensores de las plantas de los pies.

Además del criterio de estabilidad ZMP existen otros como el criterio de estabilidad de contacto de pie [33], este criterio verifica que no exista rotación en los tobillos del robot. Se propone hacer una comparativa de métodos, para conocer cual de estos proporciona una mejor estabilidad en el robot Nao.

Bibliografía

- [1] S. Kajita, K. Harada, H. Hirukawa, and K. Yokoi, *Introduction to humanoid robotics*. Springer, 2014.
- [2] Takanishi-Laboratory, “Biped humanoid robot wabian-2r.” <http://www.takanishi.mech.waseda.ac.jp/top/research/wabian/>, Oct. 2014. Accesado 10-08-2017.
- [3] KAWADA, “Humanoid robot hrp-2 "promet".” <http://global.kawada.jp/mechatronics/hrp2.html>, 2016. Accesado 15-08-2017.
- [4] J. Álvaro Gallego, A. F. Cordero, J. C. Moreno, E. A. Turowska, and J. L. Pons, “Síntesis de la marcha robótica mediante la aproximación del ciclo límite: El robor bípedo esbirro,” 08 2017.
- [5] C. Zhou, X. Wang, Z. Li, and N. Tsagarakis, “Overview of gait synthesis for the humanoid coman,” *Journal of Bionic Engineering*, vol. 14(2017), pp. 15–25, 2017.
- [6] ROBOTIS, “Robotis kits - bioloid series - robotis.” <http://www.robotis.us/bioloid-1/>, Sept. 2017. Accesado 15-08-2017.
- [7] J. J. Craig, *Robótica*. Pearson educación, tercera edición, 2006.
- [8] D. Torricelli, J. Gonzalez, M. Weckx, R. Jiménez-Fabián, B. Vanderborght, M. Sartori, S. Dosen, D. Farina, D. Lefeber, and J. L. Pons, “Human-like compliant locomotion: state of the art of robotic implementations,” *Bioinspiration & Biomimetics*, vol. 11, 2016.
- [9] P. M. V. Luna, A. C. i Fabregat, R. L. Jorge, M. C. Clavijo, C. S. Gracia, E. V. Herrero, J. V. H. Fuentes, J. M. P. Pastor, and J. J. S. Lacuesta, *Biomecánica de la marcha humana normal y patológica*. Instituto de Biomecánica de Valencia, 2009.
- [10] C. Cifuentes, F. M. M.sc., and E. R. MD, “Análisis teórico y computacional de la marcha normal y patológica: una revisión,” *Med*, vol. 18, no. 2, pp. 182–196, 2010.
- [11] D. A. Vite-Téllez, C. A. Cruz-Villar, and A. Rodríguez-Angeles, “Diseño e integración de un robot bípedo,” *Asociación de México de Control Automático*, 2009.
- [12] M. Nordin and V. Frankel, *Biomecánica básica del sistema musculoesquelético*. McGRAW HILL-INTERAMERICANA, tercera edición, 2004.

- [13] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 1620–1626, 09 2003.
- [14] Aldebaran, "Nao documentation." http://doc.aldebaran.com/2-1/home_nao.html, 2017. Accesado 10-10-2017.
- [15] A. Barki, "An inverse kinematic approach using groebner basis theory applied to gait cycle analysis," *Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio*, pp. 9–13, 2013.
- [16] HONDA, "Asimo – the honda worldwide asimo." <http://world.honda.com/ASIMO/>, 2016. Accesado 10-08-2017.
- [17] T. R. JSC, "mrobo - tosy-robots for life." <http://www.tosy.com/index.php/mrobo/>, 2016. Accesado 15-08-2017.
- [18] S. Robotics, "Discover nao, the little humanoid robot from softbank robotics." <https://www.ald.softbankrobotics.com/en/robots/nao>, Sept. 2017. Accesado 15-08-2017.
- [19] ROBOCUP, "Robocup objective." <http://www.robocup.org/objective>, 2016. Accesado 10-08-2017.
- [20] H. R. I. Waseda University, "Humanoid history -wabot-." http://www.humanoid.waseda.ac.jp/booklet/kato_2.html, Feb. 2007. Accesado 15-08-2017.
- [21] H. Kondo, Y. Ogura, H. Aikawa, A. Morishima, J. Shimizu, H. ok Lim, and A. Takanishi, "Application of biped humanoid robot to motion simulation for elderly and disabled people," *Gerontechnology*, 2008.
- [22] A. Zamani and P. A. Bhounsule, "Foot placement and ankle push-off control for the orbital stabilization of bipedal robots," 2017.
- [23] M. A. L. Silva, I. Dorna, and J. Bauer, "Simulación robótica del ciclo de marcha humana," 2014.
- [24] F. M. Rivas, J. M. Cañas, and J. González, "Aprendizaje automático de modos de caminar para un robot humanoide," 2011.
- [25] Q. Yuan and I.-M. Chen, "Human-like walking of humanoid robot based on biped kinematics and captured motion of human," *World Congress in Mechanism and Machine Science*, pp. 19–25, June 2011.
- [26] G. Gini, U. Scarfogliero, and M. Folgheraiter, "Human-oriented biped robot design: Insights into the development of a truly anthropomorphic leg," *IEEE International Conferences on Robotics and Automation*, pp. 2910–2915, 2007.

- [27] L. E. Arias, L. I. Olvera, J. A. Pámanes, and J. V. Núñez, “Patrón de marcha 3d de tipo cicloidal para humanoides y su aplicación al robot bioloid,” *Revista Iberoamericana de Ingeniería Mecánica*, vol. 18, pp. 03–22, Aug. 2014.
- [28] V. Núñez, A. Sapiens, D. Rodríguez, and V. Rodríguez, “Modelo vrml interactivo de un robot humanoide bioloid,” *Ciencias de la Ingeniería y Tecnología Handbook T-III*, vol. 3, pp. 51–63, Aug. 2013.
- [29] V. de León-Gómez, J. A. P. G., and V. Santibañez, “Experimentos de caminado del robot bípedo del itlag,” *Memorias del XVI Congreso Latinoamericano de Control Automático, CLCA 2014*, 2014.
- [30] A. Goswami, “Postural stability of biped robots and the foot-rotation indicator (fri) point,” *The International Journal of Robotics Research*, vol. 18, pp. 523–533, June 1999.
- [31] F. Reyes, *Robótica. Control de robots manipuladores*. Alfaomega Grupo Editor, S.A. de C.V., primera edición, Mar. 2011.
- [32] A. Rodríguez-Ángeles, C. A. Cruz-Villar, and D. A. Vite-Téllez, “Generación de trayectorias para marcha semiestática de un robot bípedo: diseño y pruebas experimentales,” *Ingeniería Revista Académica*, vol. 15, no. 2, pp. 57–68, 2011.
- [33] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, “A universal stability criterion of the foot contact of legged robots - adios zmp,” *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 05 2006.
- [34] M. Vukobratovic and B. Borovac, “Zero-moment point — thirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.
- [35] Y. Fujimoto and A. Kawamura, “The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation,” *Proceedings of the 2001 IEEE/SRJ International Conference on Intelligent Robots and Systems*, pp. 239–246, 2001.
- [36] P. Ponce-Cruz, *Inteligencia artificial con aplicaciones a la ingeniería*. Alfaomega Grupo Editor, primera edición, 2010.
- [37] N. Kofinas, “Forward and inverse kinematics for the nao humanoid robot,” 2012.
- [38] R. Moreno-Díaz, B. Buchberger, and J. L. Freire, *Groebner Bases: A Short Introduction for Systems Theorists*, vol. 2178. EUROCAST 2001: Computer Aided Systems Theory — EUROCAST 2001, Springer, 2001.
- [39] B. Buchberger and F. Winkler., *Groebner Bases and Applications*. Cambridge University Press, 1998.
- [40] K. Kendricks, “Solving the inverse kinematic robotics problem: A comparison study of the denavit-hartenberg matrix and groebner basis theory,” 2007.

- [41] F. A. JR., *Álgebra moderna*. McGraw-Hill, 2003.
- [42] H. Cárdenas, E. Lluís, F. Raggi, and F. Tomás, *Álgebra superior: conjuntos y combinatoria, introducción al álgebra lineal, estructuras numéricas, polinomios y ecuaciones*. Trillas, 2004.
- [43] W. Ströbl, *Diccionarios Rioduero. Matemática*. Ediciones Rioduero, 1980.
- [44] D. Cox, J. Little, and D. O’Shea, *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2007.
- [45] E. Martínez-Moro, C. Munuera-Gómez, and D. Ruano-Benito, *bases de groebner: aplicaciones a la codificación algebraica*. Editorial Texto, 2007.
- [46] J. Kim, H. Kwak, H. Lee, K. Seo, B. Lim, M. Lee, J. Lee, , and K. Roh, “Balancing control of a biped robot,” *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2756–2761, 2012.
- [47] K. Matsumoto and A. Kawamura, “The direction of a biped robot using gyro sensor feedback,” *IEEE International Workshop on Advanced Motion Control*, pp. 137–142, 2010.
- [48] H. Ono, T. Sato, and K. Ohnishi, “Balance recovery of ankle strategy: Using knee joint for biped robot,” *IEEE-ISAS 2011*, pp. 236–241, 2011.
- [49] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa, “Biped walking pattern generator allowing auxiliary zmp control,” *IEEE*, pp. 2993–2999, 2006.
- [50] Aldebaran, “H25 - masses.” http://doc.aldebaran.com/2-1/family/nao_h25/masses_h25.html, 2018. Accesado 20-03-2018.
- [51] Maplesoft, “Groebner.” <https://www.maplesoft.com/support/help/maple/view.aspx?path=Groebner>, 2017. Accesado 06-10-2017.
- [52] Maplesoft, “Maple.” <https://www.addlink.es/productos/maple>, 2017. Accesado 08-11-2017.

Apéndice A

Masas del robot Nao

En este anexo se detallan para cada parte sólida del robot Nao H-25 [50], la masa y el centro de masas. La masa está dada en Kg y el centro de masa (CoM) en m.

$Masa(kg)$

$$CoM(S) = \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}_{(o,R)} \quad (m)$$

El centro de masas está descrito a partir de un sistema de referencia local en el sólido a analizar (S)(o, R). Todos los sólidos (S) y el sistema de coordenadas local se describen en relación con la postura cero: de pie con las piernas rectas y los brazos apuntando hacia delante como se muestra en la figura 2.11. La masa total del Nao es de $masa = 5.305350006kg$.

Torso

(o, R) está localizado en: torso.

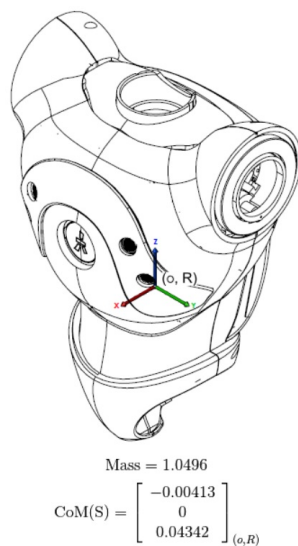


Figura A.1: Centro de masas y masa del torso del robot Nao [14].

cabeza

Cuello

(o, R) está localizado en: HeadYaw.

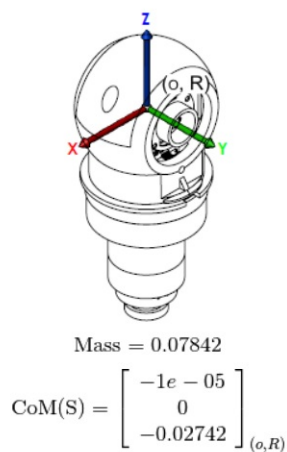


Figura A.2: Centro de masas y masa del cuello del robot Nao [14].

Cabeza

(o, R) está localizado en: HeadPitch.

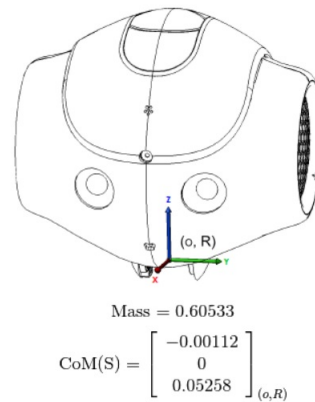


Figura A.3: Centro de masas y masa de la cabeza del robot Nao [14].

Brazos

Hombro derecho

(o, R) está localizado en: RShoulderPitch.

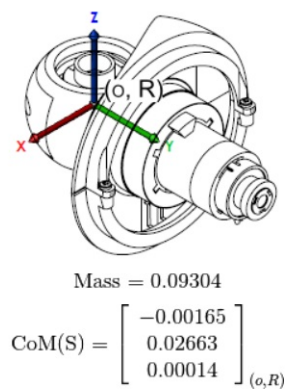


Figura A.4: Centro de masas y masa del hombro derecho del robot Nao [14].

Hombro izquierdo

(o, R) está localizado en: LShoulderPitch.

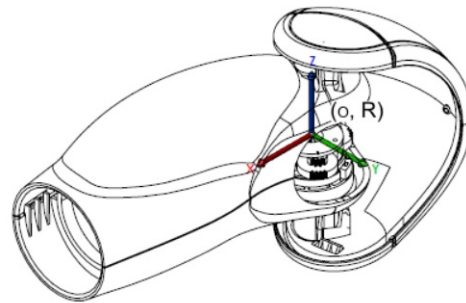
$$\text{Mass} = 0.09304$$

$$\text{CoM(S)} = \begin{bmatrix} -0.00165 \\ -0.02663 \\ 0.00014 \end{bmatrix}_{(o,R)}$$

Figura A.5: Centro de masas y masa del hombro izquierdo del robot Nao [14].

Brazo derecho

(o, R) está localizado en: RShoulderRoll.



$$\text{Mass} = 0.15777$$

$$\text{CoM(S)} = \begin{bmatrix} 0.02455 \\ -0.00563 \\ 0.0033 \end{bmatrix}_{(o,R)}$$

Figura A.6: Centro de masas y masa del brazo derecho del robot Nao [14].

Brazo izquierdo

(o, R) está localizado en: LShoulderRoll.

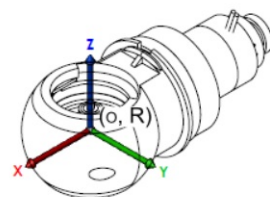
$$\text{Mass} = 0.15777$$

$$\text{CoM}(S) = \begin{bmatrix} 0.02455 \\ 0.00563 \\ 0.0033 \end{bmatrix}_{(o,R)}$$

Figura A.7: Centro de masas y masa del brazo izquierdo del robot Nao [14].

Codo derecho

(o, R) está localizado en: RElbowYaw.



$$\text{Mass} = 0.06483$$

$$\text{CoM}(S) = \begin{bmatrix} -0.02744 \\ 0 \\ -0.00014 \end{bmatrix}_{(o,R)}$$

Figura A.8: Centro de masas y masa del codo derecho del robot Nao [14].

Codo izquierdo

(o, R) está localizado en: LElbowYaw.

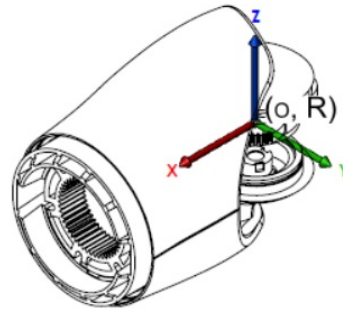
$$\text{Mass} = 0.06483$$

$$\text{CoM}(S) = \begin{bmatrix} -0.02744 \\ 0 \\ -0.00014 \end{bmatrix}_{(o,R)}$$

Figura A.9: Centro de masas y masa del codo izquierdo del robot Nao [14].

Antebrazo derecho

(o, R) está localizado en: RElbowRoll.



$$\text{Mass} = 0.07761$$

$$\text{CoM}(S) = \begin{bmatrix} 0.02556 \\ -0.00281 \\ 0.00076 \end{bmatrix}_{(o,R)}$$

Figura A.10: Centro de masas y masa del antebrazo derecho del robot Nao [14].

Antebrazo izquierdo

(o, R) está localizado en: LElbowRoll.

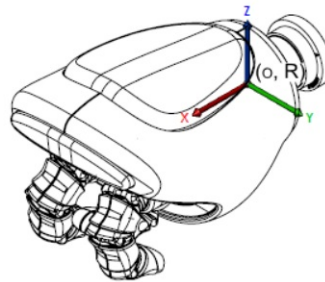
$$\text{Mass} = 0.07761$$

$$\text{CoM}(S) = \begin{bmatrix} 0.02556 \\ 0.00281 \\ 0.00076 \end{bmatrix}_{(o,R)}$$

Figura A.11: Centro de masas y masa del antebrazo izquierdo del robot Nao [14].

Mano derecha

(o, R) está localizado en: RWristYaw.



$$\text{Mass} = 0.18533$$

$$\text{CoM(S)} = \begin{bmatrix} 0.03434 \\ 0.00088 \\ 0.00308 \end{bmatrix}_{(o,R)}$$

Figura A.12: Centro de masas y masa de la mano derecha del robot Nao [14].

Mano izquierda

(o, R) está localizado en: LWristYaw.

$$\text{Mass} = 0.18533$$

$$\text{CoM(S)} = \begin{bmatrix} 0.03434 \\ -0.00088 \\ 0.00308 \end{bmatrix}_{(o,R)}$$

Figura A.13: Centro de masas y masa de la mano izquierda del robot Nao [14].

Piernas

Pelvis derecha

(o, R) está localizado en: RHipYawPitch.

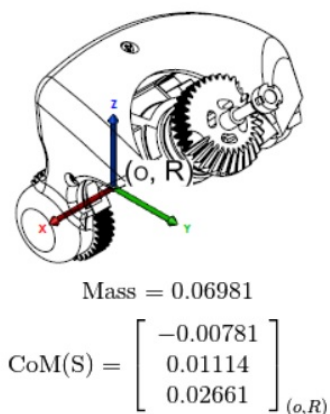


Figura A.14: Centro de masas y masa de la pelvis derecha del robot Nao [14].

Pelvis izquierda

(o, R) está localizado en: LHipYawPitch.

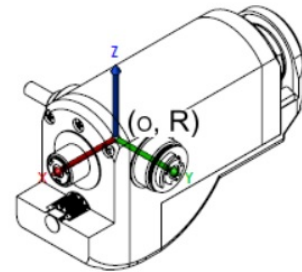
$$\text{Mass} = 0.06981$$

$$\text{CoM}(S) = \begin{bmatrix} -0.00781 \\ -0.01114 \\ 0.02661 \end{bmatrix}_{(o,R)}$$

Figura A.15: Centro de masas y masa de la pelvis izquierda del robot Nao [14].

Cadera derecha

(o, R) está localizado en: RHipRoll.



$$\text{Mass} = 0.14053$$

$$\text{CoM}(S) = \begin{bmatrix} -0.01549 \\ -0.00029 \\ -0.00515 \end{bmatrix}_{(o,R)}$$

Figura A.16: Centro de masas y masa de la cadera derecha del robot Nao [14].

Cadera izquierda

(o, R) está localizado en: LHipRoll.

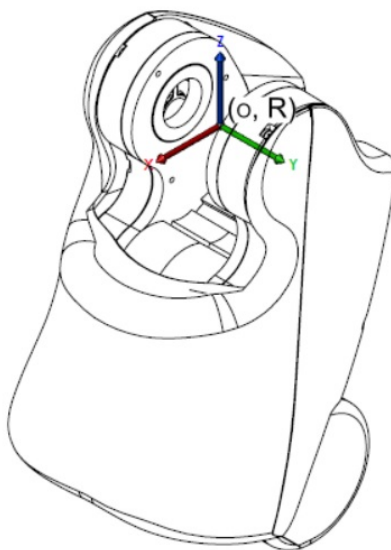
$$\text{Mass} = 0.14053$$

$$\text{CoM}(S) = \begin{bmatrix} -0.01549 \\ 0.00029 \\ -0.00515 \end{bmatrix}_{(o,R)}$$

Figura A.17: Centro de masas y masa de la cadera izquierda del robot Nao [14].

Muslo derecho

(o, R) está localizado en: RHipPitch.



$$\text{Mass} = 0.38968$$

$$\text{CoM}(S) = \begin{bmatrix} 0.00138 \\ -0.00221 \\ -0.05373 \end{bmatrix}_{(o,R)}$$

Figura A.18: Centro de masas y masa del muslo derecho del robot Nao [14].

Muslo izquierdo

(o, R) está localizado en: LHipPitch.

$$\text{Mass} = 0.38968$$

$$\text{CoM}(S) = \begin{bmatrix} 0.00138 \\ 0.00221 \\ -0.05373 \end{bmatrix}_{(o,R)}$$

Figura A.19: Centro de masas y masa del muslo izquierdo del robot Nao [14].

Tibia derecha

(o, R) está localizado en: RKneePitch.

$$\begin{aligned} \text{Mass} &= 0.30142 \\ \text{CoM(S)} &= \begin{bmatrix} 0.00453 \\ -0.00225 \\ -0.04936 \end{bmatrix}_{(o,R)} \end{aligned}$$

Figura A.20: Centro de masas y masa de la tibia derecha del robot Nao [14].

Tibia izquierda

(o, R) está localizado en: LKneePitch.

$$\begin{aligned} \text{Mass} &= 0.30142 \\ \text{CoM(S)} &= \begin{bmatrix} 0.00453 \\ 0.00225 \\ -0.04936 \end{bmatrix}_{(o,R)} \end{aligned}$$

Figura A.21: Centro de masas y masa de la tibia izquierda del robot Nao [14].

Tobillo derecho

(o, R) está localizado en: RAnklePitch.

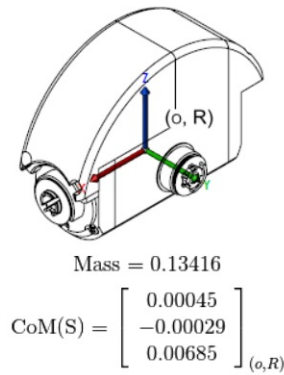


Figura A.22: Centro de masas y masa del tobillo derecho del robot Nao [14].

Tobillo izquierdo

(o, R) está localizado en: LAnklePitch.

$$\text{Mass} = 0.13416$$

$$\text{CoM(S)} = \begin{bmatrix} 0.00045 \\ 0.00029 \\ 0.00685 \end{bmatrix}_{(o,R)}$$

Figura A.23: Centro de masas y masa del tobillo derecho del robot Nao [14].

Pie derecho

(o, R) está localizado en: RAnkleRoll.

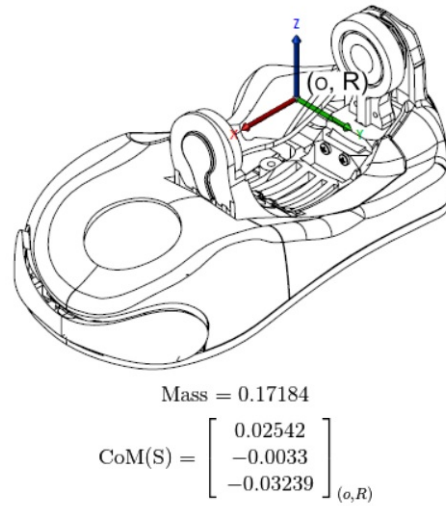


Figura A.24: Centro de masas y masa del pie derecho del robot Nao [14].

Pie izquierdo

(o, R) está localizado en: LAnkleRoll.

Mass = 0.17184

$$\text{CoM}(S) = \begin{bmatrix} 0.02542 \\ 0.0033 \\ -0.03239 \end{bmatrix}_{(o,R)}$$

Figura A.25: Centro de masas y masa del pie izquierdo del robot Nao [14].

Apéndice B

Artículo

Se participó en la *3rd International Conference on Mathematical Modelling (CIMM)*, que se realizó en la Universidad Tecnológica de la Mixteca ubicada en la ciudad de Huajuapán de León, los días 4 y 5 de octubre del 2018.

Después de la participación en el congreso, se entregó un artículo que se titula *Teoría de las bases de Groebner aplicado al cálculo de la cinemática inversa de un robot bípedo*, el cuál fue revisado y aprobado por un comité para su publicación en un libro con registro ISBN.

Este artículo aún se encuentra en el proceso de publicación, por lo que a continuación se pone en este trabajo de tesis sin las páginas correspondientes al libro.

Capítulo 1

Teoría de las bases de Groebner aplicado al cálculo de la cinemática inversa de un robot bípedo

Rocío Ambrocio Delgado¹
Heriberto Casarrubias Vargas²
Esther Lugo González³
Alberto E. Petrilli Barceló⁴

Abstract: In this paper, we give an analysis of the inverse kinematics of the biped robot NAO for the development of a walking algorithm. Using the Groebner Bases theory, the NAO robot's joints angles are estimated to follow the desired trajectory. With the *MapleTM* software, which is a program oriented to the solution of mathematical problems, the Groebner bases are estimated from different proposed equations, which describe the NAO robot behavior at rest. The proposed equations that will be transformed to Groebner bases are obtained from the forward kinematics of the Nao robot and will be used for joint angles calculation from a proposed point. As a result, is presented the trajectory generation using the ZMP stability criterion.

Keywords: inverse kinematics, Groebner Bases theory, walking algorithm.

Resumen: En el presente trabajo se realiza un análisis de la cinemática inversa del robot bípedo NAO para el desarrollo de un algoritmo de marcha. Usando la teoría de las Bases de Groebner se calculan los ángulos de las articulaciones que debe tomar el robot NAO para

¹rocio.ambrocio.delgado@gmail.com. Estudiante de la maestría en Robótica, Universidad Tecnológica de la Mixteca

²hcasarrubiasv@uaemex.mx. Centro Universitario UAEM Valle de México, Universidad Autónoma del Estado de México

³elugog@mixteco.utm.mx. Instituto de Electrónica y Mecatrónica, Universidad Tecnológica de la Mixteca

⁴petrilli@mixteco.utm.mx. División de Estudios de Posgrado, Universidad Tecnológica de la Mixteca

seguir la trayectoria deseada. Con el software *MapleTM*, que es un programa orientado a la solución de problemas matemáticos, se calcula la base de Groebner a partir de diferentes ecuaciones propuestas, las cuales describen el comportamiento del robot NAO en reposo. Las ecuaciones propuestas que serán transformadas a bases de Groebner son obtenidas a partir de la cinemática directa del robot Nao y serán utilizadas para el cálculo de los ángulos de las articulaciones a partir de un punto propuesto. Como resultado se presenta la generación de una trayectoria utilizando el criterio de estabilidad ZMP.

Palabras clave: cinemática inversa, teoría de las Bases de Groebner, algoritmo de marcha.

1.1. Introducción

La cinemática es la parte de la física que se encarga del estudio del movimiento de sistemas mecánicos sin considerar las fuerzas que lo producen [11]. La cinemática se divide en: cinemática directa y cinemática inversa, las cuales pueden calcularse mediante el método de las bases de Groebner, Denavit-Hartenberg y haciendo uso de herramientas geométricas.

La cinemática directa consiste en analizar el movimiento del robot con respecto a un sistema cartesiano fijo relacionando las coordenadas generalizadas, los parámetros geométricos y las coordenadas cartesianas y de orientación del efector final.

La cinemática inversa determina si un robot puede alcanzar una posición determinada dadas las coordenadas generalizadas, los parámetros geométricos y de orientación en ángulos de Euler del efector final de un robot con respecto a un sistema de coordenadas fijo.

La cinemática directa siempre tiene solución, mientras que la cinemática inversa no siempre la tiene, puede ser que tenga más de una solución o que no tenga ninguna solución [11].

Para obtener cálculos más exactos en la cinemática inversa de un manipulador, pueden utilizarse las bases de Groebner.

1.2. Teoría de las Bases de Groebner

La teoría de la bases de Groebner, nombrado así por Bruno Buchberger en su tesis doctoral en honor a Wolfgang Groebner, provee una aproximación uniforme

a la solución de muchos problemas expresados en términos de conjuntos de polinomios multivariados [2].

El cálculo computacional en los polinomios multivariados no depende de la cantidad de polinomios que se tengan, sino del orden lexicográfico de los mismos. Para ordenar un conjunto de polinomios lexicográficamente, el elemento de menor grado en el polinomio debe estar en términos de una sola variable.

Este tipo de orden permite que se tenga un cálculo recursivo para las diferentes variables del conjunto de polinomios.

La teoría de las bases de Groebner es atractiva debido a diferentes factores [7], entre ellos se encuentran:

1. El problema principal puede ser resuelto de forma rápida si se conocen bien los polinomios a utilizar.
2. El algoritmo de Buchberger se puede comprender fácilmente.
3. Muchos problemas de las diferentes áreas de las matemáticas pueden ser resueltos mediante bases de Groebner.

Para comprender matemáticamente las Bases de Groebner se presentan las siguientes definiciones [9]:

Definición 1 *Definimos un orden monomial sobre \mathbb{T}^n como un orden total \prec que satisface:*

1. $1 \prec x^\alpha$ para todo $x^\alpha \in \mathbb{T}^n$ distinto de 1
2. Si $x^\alpha \prec x^\beta$ entonces $x^\gamma x^\alpha \prec x^\gamma x^\beta$ para todo $x^\gamma \in \mathbb{T}^n$

Definición 2 *Diremos que $a = (a_1, \dots, a_n) \in \mathbb{T}^n$ es menor (respecto al orden lexicográfico) que $b = (b_1, b_2, \dots, b_n) \in \mathbb{T}^n$, y lo denotaremos por $a <_{lex} b$, si existe $i \in \{1, \dots, n-1\}$ tal que $a_j = b_j$, si $1 \leq j \leq i$, y $a_i < b_i$, o equivalentemente, si en la diferencia $b - a$ la primera coordenada distinta de cero de izquierda a derecha es positiva.*

Por ejemplo, si tenemos $x_1 x_2 x_3^2 x_4^5$ y $x_1 x_2 x_3^3 x_4^2$, usando las definiciones 1 y 2 tenemos que

$$x_1 x_2 x_3^2 x_4^5 <_{lex} x_1 x_2 x_3^3 x_4^2 \quad (1.2.1)$$

ya que $(1, 1, 2, 5) <_{lex} (1, 1, 3, 2)$.

A lo largo del documento se denotará por $R = \mathbb{K}[x_1, \dots, x_n] = \mathbb{K}[x]$ al anillo de polinomios sobre el cuerpo \mathbb{K} . Un subconjunto no vacío $I \in R$ es un ideal de R si es cerrado bajo la suma de polinomios y es cerrado por la multiplicación de elementos de R .

Definición 3 (*Bases de Groebner*). Dado un orden monomial \succ y un ideal $I \subset R$ se dice que el conjunto $f_1, \dots, f_s \subset R$ es una base de Groebner de I para \succ si se cumple

$$\langle \text{lt}_\succ(f_1), \dots, \text{lt}_\succ(f_s) \rangle = \langle \text{lt}_\succ(I) \rangle \quad (1.2.2)$$

donde $\text{lt}_\succ(I) = \text{lt}_\succ(f) | f \in I$ [9]

Las bases de Groebner son un sistema de generadores del ideal I , lo cual se demuestra a partir del teorema 1.2.1 [9].

Teorema 1.2.1 Dado un orden monomial \succ y un ideal $I \subset R$ existe una base de Groebner de I , la cual es un sistema de generadores del ideal I .

Para comprobar que un conjunto de polinomios multivariados es una base de Groebner se utiliza el criterio de Buchberger el cual es mostrado en la proposición 1.2.2.

Proposición 1.2.2 (*Criterio de Buchberger*). Sea \succ un orden monomial e I un ideal en R . Un sistema de generadores $G = \{g_1, \dots, g_s\}$ de I es una base de Groebner si y sólo si

$$\overline{S(g_i, g_j)}_\succ^G = 0, 1 \leq i, j \leq s, i \neq j \quad (1.2.3)$$

1.3. Algoritmo de Buchberger

El algoritmo de Buchberger calcula la base de Groebner de un ideal dado por un sistema de generadores, todo esto a partir de los S-polinomios que son la combinación más simple de dos polinomios que cancelan sus términos líderes [9].

Definición 4 (*S-polinomio*). Sean $f, g \in R$ dos polinomios no nulos y \succ un orden monomial. El S-polinomio de f y g es

$$S(f, g) = \frac{x^\gamma}{\text{lt}_\succ(f)} \cdot f - \frac{x^\gamma}{\text{lt}_\succ(g)} \cdot g \quad (1.3.1)$$

donde $x^\gamma = \text{mcm}\{\text{lm}_\succ(f), \text{lm}_\succ(g)\}$.

El algoritmo de Buchberger es el siguiente:

ENTRADA: $F = f_1, \dots, f_s$ con $I = \langle \{f_i\}_{i=1}^s \rangle \neq \{0\}$ y \succ un orden monomial.

SALIDA: Una base de Groebner para el ideal I en el orden monomial \succ .

1. $G \leftarrow F, G' \leftarrow \{0\}$
2. **MIENTRAS** $G \neq G'$ **HACER**
3. $G' \leftarrow G$
4. **PARA** cada par $\{p, q\} \subset G'$ con $p \neq q$ **HACER**
5. $S \leftarrow \overline{S(p, q)}^{G'}$
6. **SI** $S \neq 0$ **HACER**
7. $G \leftarrow G \cup \{S\}$
8. **fin SI**
9. **fin PARA**
10. **fin MIENTRAS**

1.4. Punto de momento cero (ZMP)

Vukabratovic y Stepanenko en 1972 propusieron un método para que la marcha de un robot tenga estabilidad local. A este método lo llamaron *Zero Moment Point* (ZMP) [5] o *Punto de Momento Cero* en español. El ZMP mostrado en la Figura 1.1 se define como el punto del suelo respecto al cual el momento creado por las fuerzas inerciales y gravitacionales no tiene componente en los ejes horizontales.

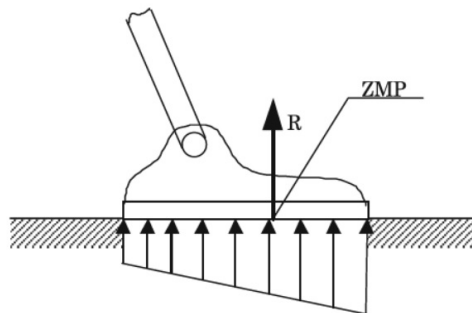


Figura 1.1: Definición del Punto de Momento Cero (ZMP) [6].

En los robots bípedos el único contacto con el medio ambiente se realiza mediante la fuerza de fricción y la fuerza vertical generadas por el piso.

Para facilitar el análisis se puede descartar la parte del mecanismo que está por encima del tobillo del pie de apoyo y sustituir su influencia por una fuerza y

un momento con tres componentes cada una, las cuales son las reacciones totales del robot con respecto al piso.

Todos los sensores de fuerza en el pie del robot deben estar en contacto con el piso, de no hacerlo el robot puede rotar y caer. Cuando uno o varios de los sensores de fuerza en el pie no están en contacto con el piso o si el robot no cuenta con sensores de fuerza en los pies, se emplea una estrategia de control para evitar que el robot caiga [12].

Para ejecutar la estrategia de control primero se debe verificar que el CoM (centro de masas) se encuentra dentro del polígono de soporte. La región que considera todos los puntos de contacto entre el robot y el suelo se le llama *polígono de soporte*. Matemáticamente el *polígono de soporte* esta definido por una envolvente convexa como se muestra en la Figura 1.2.

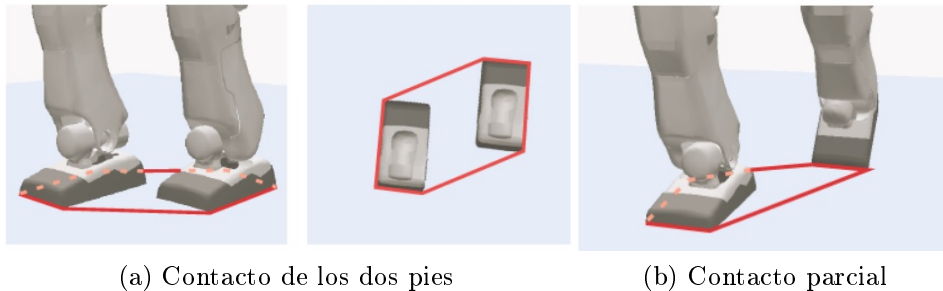


Figura 1.2: Polígono de soporte [6].

La relación que existe entre el ZMP y el polígono de soporte es muy importante. El ZMP solo existe dentro del polígono de soporte. Si el ZMP se encuentra fuera del polígono de soporte el robot caerá.

1.5. Método geométrico

Para encontrar la solución geométrica de un robot, se trata de descomponer la geometría espacial del mismo en varios problemas de geometría plana. El método geométrico consiste en encontrar la cinemática inversa de un robot a partir de identidades trigonométricas.

En muchos robots es fácil calcular los ángulos de las articulaciones utilizando las herramientas de la geometría plana.

Para un brazo de 3 Grados de Libertad (GDL) como el mostrado en la Figura 1.3 que es planar, se le puede aplicar directamente la geometría plana para encontrar una solución [3]. Esto es, considerar que es un triángulo sólido y de esta

forma aplicar funciones trigonométricas como la *ley de cosenos* y el *teorema de Pitágoras* para encontrar la solución de los ángulos que debe tomar el mecanismo para llegar a un punto determinado.

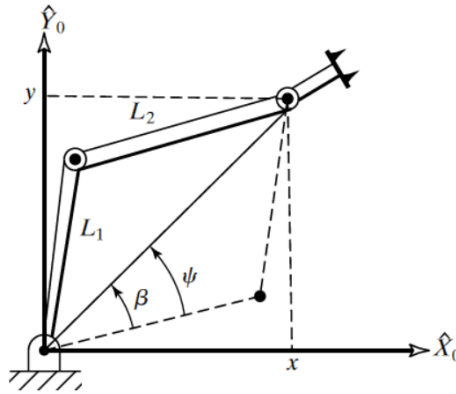


Figura 1.3: Representación del modelo de un manipulador planar de 3 GDL [3].

1.6. El robot Nao

El robot humanoide Nao de Aldebaran Robotics tiene 5 cadenas cinemáticas [10], estas son: la cabeza, sus dos brazos y sus dos piernas. Las articulaciones que tiene cada cadena cinemática, mostradas en la Figura 1.4, son:

- **Cabeza:** HQ1, HQ2
- **Brazo izquierdo:** LSQ1, LSQ2, LEQ3, LEQ4
- **Brazo derecho:** RSQ1, RSQ2, REQ3, REQ4
- **Pierna izquierda:** LHQ1, LHQ2, LHQ3, LKQ4, LAQ5, LAQ6
- **Pierna derecha:** RHQ1, RHQ2, RHQ3, RKQ4, RAQ5, RAQ6

Las dimensiones que tiene el robot Nao son las mostradas en el cuadro 1.1, las cuales son muy importantes en el cálculo de la cinemática directa.

La coordenada base se toma en la parte central del torso como se muestra en la Figura 1.5, al mismo tiempo se muestra la posición inicial de todas las articulaciones del robot Nao [8].

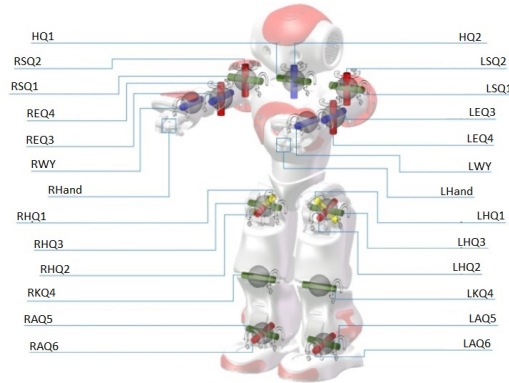


Figura 1.4: Cadenas cinemáticas y articulaciones del robot Nao [10].

1.7. Obtención de la cinemática inversa del robot Nao

Para el cálculo de la cinemática inversa del robot Nao se utilizaron las bases de Groebner, esto se hace a partir de proponer una posición final para las piernas del robot. Las funciones para calcular la cinemática inversa se propusieron utilizando el método geométrico.

1.7.1. Brazos del robot Nao

Los brazos del robot Nao se pueden ver como dos cadenas cinemáticas que trabajan en reflejo una de la otra, es decir, el brazo izquierdo es el reflejo del brazo derecho. En este caso se resolverá la cinemática inversa del brazo derecho. La obtención de las funciones a convertir a bases de Groebner son a partir del hombro a la muñeca como se muestra en la Figura 1.6, donde $L_1 = 105$ mm y $L_2 = 113,7$ mm, de acuerdo al cuadro 1.1, L_1 corresponde a UpperArmLength y L_2 es la suma de LowerArmLength y HandOffsetX. Además, se propone que el ángulo $\theta_3 = 0$, que es la rotación en el codo, ya que en el algoritmo de marcha no se manipulan objetos.

La cadena cinemática del brazo derecho se puede ver como dos eslabones articulados como en la Figura 1.7.

Los ángulos de la Figura 1.7 se pueden representar como vectores como se muestra en la Figura 1.8, en donde $q_2 = \theta_2$ y $q_4 = \theta_4$.

A partir de la Figura 1.8 se obtienen las ecuaciones mostradas en 1.7.1 y 1.7.2. Estas ecuaciones resultan de aplicar identidades trigonométricas y de igualarlas con el vector $P = [p_y, p_z]$, donde el vector P representa la posición deseada en (y, z) .

Nombre	Longitud(mm)
NeckOffsetZ	126.50
ShoulderOffsetY	98.00
ElbowOffsetY	15.00
UpperArmLength	105.00
LowerArmLength	55.95
ShoulderOffsetZ	100.00
HandOffsetX	57.75
HipOffsetZ	85.00
HipOffsetY	50.00
ThighLength	100.00
TibiaLength	102.90
FootHeight	45.19
HandOffsetZ	12.31

Cuadro 1.1: Dimensiones del robot Nao.

$$L_1 \cdot c_2 + L_2 \cdot s_4 - p_z = 0 \quad (1.7.1)$$

$$L_1 \cdot s_2 + L_2 \cdot c_4 - p_y = 0 \quad (1.7.2)$$

Para que el sistema de ecuaciones se pueda resolver, es necesario proponer otras dos ecuaciones (1.7.3 y 1.7.4) a partir de identidades trigonométricas:

$$c_2^2 + s_2^2 - 1 = 0 \quad (1.7.3)$$

$$c_4^2 + s_4^2 - 1 = 0 \quad (1.7.4)$$

Las ecuaciones de la 1.7.1 a la 1.7.4, donde $s_2 = \sin(\theta_2)$, $c_2 = \cos(\theta_2)$, $s_4 = \sin(\theta_4)$ y $c_4 = \cos(\theta_4)$, se proponen como la solución a la cinemática inversa de los brazos del robot Nao. Estas ecuaciones se convierten a bases de Groebner con el software *Maple^TM*, con el orden lexicográfico c_4 , c_2 , s_4 y s_2 , se tienen 4 ecuaciones:

$$44100 \cdot (p_y^2 + p_z^2) \cdot s_2^2 - 420 \cdot (p_y^3 + p_y \cdot p_z^2) \cdot s_2 + 799129,8 \cdot p_y \cdot s_2 + 3620229,3 - 3805,38 \cdot p_y^2 - 47905,38 \cdot p_z^2 + p_y^4 + 2 \cdot p_y^2 \cdot p_z^2 + p_z^4 = 0 \quad (1.7.5)$$

$$p_y^2 - 210 \cdot s_2 \cdot p_y - p_z^2 + 227,40 \cdot p_z \cdot s_4 - 1902,69 = 0 \quad (1.7.6)$$

$$210 \cdot p_z \cdot c_2 - p_y^2 + 210 \cdot s_2 \cdot p_y - p_z^2 + 1902,69 = 0 \quad (1.7.7)$$

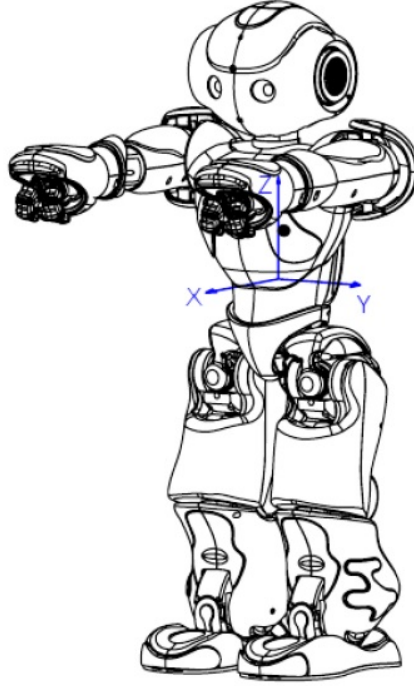


Figura 1.5: Posición inicial y la coordenada base del robot Nao [10].

$$113,70 \cdot c_4 - p_y + 105 \cdot s_2 = 0 \quad (1.7.8)$$

De estas 4 ecuaciones se toman 1.7.5 y 1.7.6. Para conocer s_2 se utiliza la ecuación 1.7.9. Se toma la raíz negativa por las restricciones del robot. La solución positiva del polinomio cuadrático da una posición que el robot no puede alcanzar.

$$s_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (1.7.9)$$

Al sustituir los valores de a, b y c se tiene:

$$a = 44100 \cdot (p_y^2 + p_z^2)$$

$$b = -420 \cdot (p_y^3 + p_y \cdot p_z^2) + 799129,8 \cdot p_y$$

$$c = 3620229,3 - 3805,38 \cdot p_y^2 - 47905,38 \cdot p_z^2 + p_y^4 + 2 \cdot p_y^2 \cdot p_z^2 + p_z^4$$

Al conocer el valor de s_2 se calcula s_4 para conocer los valores de θ_2 y θ_4 . El ángulo θ_1 se puede determinar a partir de la Figura 1.9.

en donde se puede observar que

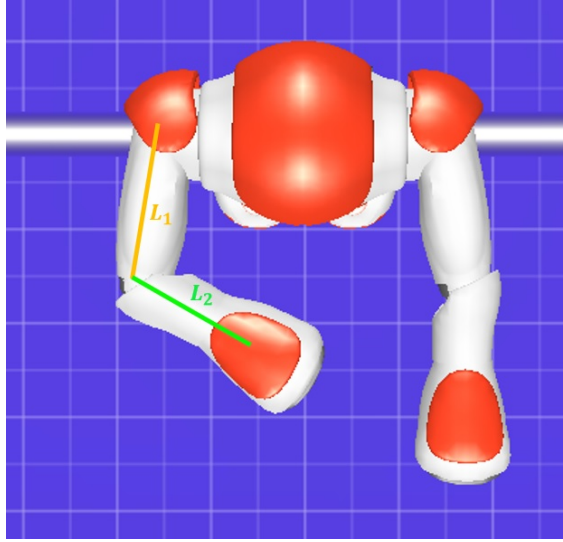


Figura 1.6: Cadena para calcular la cinemática inversa de los brazos del robot Nao.

$$\theta_1 = -atan(p_z, p_x) \quad (1.7.10)$$

Con estas ecuaciones se puede calcular la cinemática inversa de los brazos del robot Nao proponiendo un punto (p_x, p_y, p_z) que se encuentre dentro del área de trabajo. Los ángulos para el brazo izquierdo quedan de la siguiente forma:

$$\begin{aligned} \theta_{1I} &= \theta_{1D} \\ \theta_{2I} &= -\theta_{2D} \\ \theta_{3I} &= -\theta_{3D} \\ \theta_{4I} &= -\theta_{4D} \end{aligned} \quad (1.7.11)$$

En donde $\theta_{1I} = \theta_1$ del brazo izquierdo, $\theta_{2I} = \theta_2$ del brazo izquierdo, $\theta_{3I} = \theta_3$ del brazo izquierdo, $\theta_{4I} = \theta_4$ del brazo izquierdo, $\theta_{1D} = \theta_1$ del brazo derecho, $\theta_{2D} = \theta_2$ del brazo derecho, $\theta_{3D} = \theta_3$ del brazo derecho y $\theta_{4D} = \theta_4$ del brazo derecho.

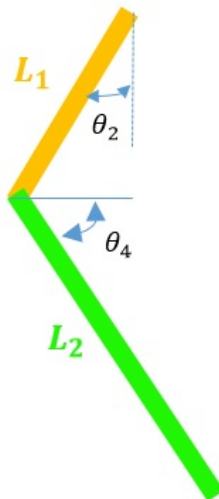


Figura 1.7: Cadena cinemática de los brazos del robot Nao.

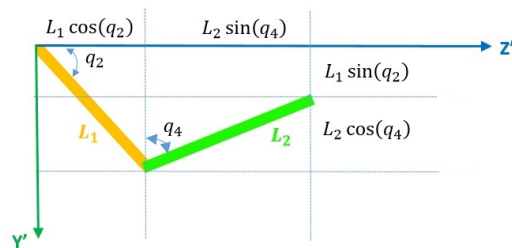


Figura 1.8: Ecuaciones propuestas para calcular la cinemática inversa de los brazos del robot Nao.

1.7.2. Piernas del robot Nao

Las piernas del robot Nao son simétricas, por lo que la estimación de la cinemática inversa se puede realizar solo a una de las dos cadenas. Las medidas que se considera para proponer las funciones a convertir a bases de Groebner van de la cadera al tobillo como se muestra en la Figura 1.10, donde $L_1 = 100.0$ mm y $L_2 = 102.9$ mm, de acuerdo al cuadro 1.1, L_1 corresponde a ThighLength y L_2 a TibiaLength. Además se propone $\theta_1 = 0$, debido a que no se requiere la rotación de cadera en el algoritmo de marcha.

Visto de otra forma, la cadena cinemática se forma por dos eslabones articulados como en la Figura 1.11. Para obtener las funciones que describen el comportamiento de las articulaciones, los ángulos se pueden ver como en la Figura 1.12.

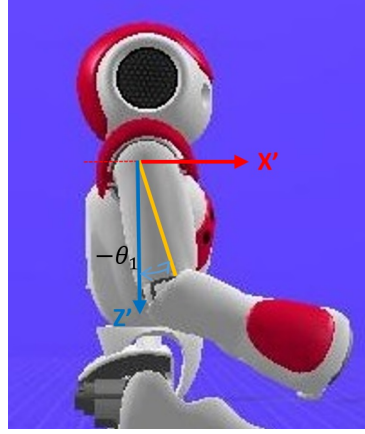


Figura 1.9: Ángulo θ_1 del brazo del robot Nao.

Teniendo así las ecuaciones mostradas en 1.7.12 y 1.7.13. Resultantes de aplicar identidades trigonométricas y de igualar las ecuaciones mostradas en la Figura 1.12 con el vector $P=[p_x, p_z]$, en donde el vector P representa la posición deseada en (x, z) .

$$L_2 \cdot c_3 \cdot c_4 - L_2 \cdot s_3 \cdot s_4 + L_1 \cdot c_3 - p_z = 0 \quad (1.7.12)$$

$$L_2 \cdot c_3 \cdot s_4 + L_2 \cdot c_4 \cdot s_3 + L_1 \cdot s_3 - p_x = 0 \quad (1.7.13)$$

Estas ecuaciones aún no son suficientes. Para que el sistema de ecuaciones se pueda resolver se agregan las siguientes identidades trigonométricas:

$$c_3^2 + s_3^2 - 1 = 0 \quad (1.7.14)$$

$$c_4^2 + s_4^2 - 1 = 0 \quad (1.7.15)$$

Las ecuaciones de la 1.7.12 a la 1.7.15, donde $s_3 = \sin(\theta_3)$, $c_3 = \cos(\theta_3)$, $s_4 = \sin(\theta_4)$ y $c_4 = \cos(\theta_4)$, se proponen como un sistema de ecuaciones con el cual se puede resolver la cinemática inversa de las piernas. Estas ecuaciones se convierten a bases de Groebner con el software *Maple^TM*, siguiendo el orden lexicográfico c_4, c_3, s_4 y s_3 , se tienen 4 ecuaciones:

$$\begin{aligned} &40000 \cdot (p_x^2 + p_z^2) \cdot s_3^2 + -400 \cdot (p_x^3 + p_x \cdot p_z^2) \cdot s_3 + \\ &235364 \cdot p_x \cdot s_3 + 346226,3 - 1176,82 \cdot p_x^2 - \\ &41176,82 \cdot p_z^2 + p_x^4 + 2 \cdot p_x^2 \cdot p_z^2 + p_z^4 = 0 \end{aligned} \quad (1.7.16)$$

$$200 \cdot (p_x^2 + p_z^2) \cdot s_3 + 20580 \cdot p_z \cdot s_4 + 588,41 \cdot p_x - p_x^3 - p_x \cdot p_z^2 = 0 \quad (1.7.17)$$

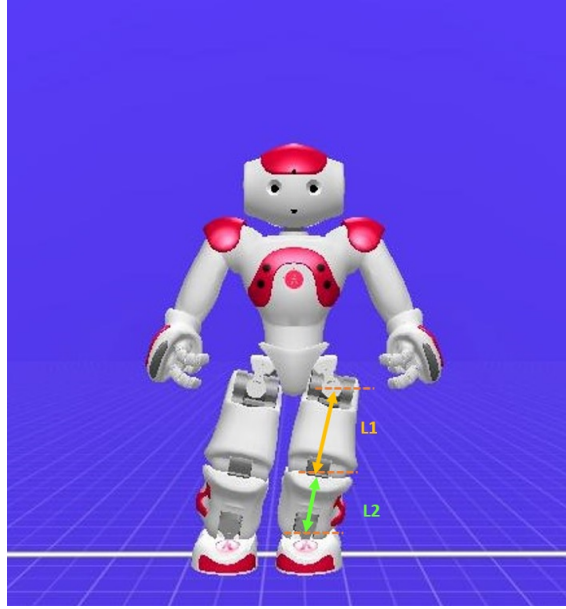


Figura 1.10: Cadena para calcular la cinemática inversa de las piernas del robot Nao.

$$-p_x^2 - p_z^2 + 20580 \cdot c_4 + 20588,41 = 0 \quad (1.7.18)$$

$$200 \cdot p_z \cdot c_3 - p_x^2 + 200 \cdot p_x \cdot s_3 - p_z^2 + 588,41 = 0 \quad (1.7.19)$$

De estas 4 ecuaciones se toman 1.7.16 y 1.7.17, ya que resolver las 4 ecuaciones es algo redundante. La ecuación 1.7.16 es un polinomio cuadrático del cuál se tomará la raíz negativa debido a las restricciones en el movimiento del robot. La solución positiva del polinomio cuadrático da como resultado una posición que no se puede alcanzar. Para resolverlo la ecuación 1.7.16 se utiliza la fórmula 1.7.20.

$$s_3 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (1.7.20)$$

Al sustituir a, b y c de la fórmula 1.7.20 se tiene:

$$a = 40000 \cdot (p_x^2 + p_z^2)$$

$$b = -400 \cdot (p_x^3 + p_x \cdot p_z^2) + 235364 \cdot p_x$$

$$c = 346226,3 - 1176,82 \cdot p_x^2 - 41176,82 \cdot p_z^2 + p_x^4 + 2 \cdot p_x^2 \cdot p_z^2 + p_z^4$$

Obtenida s_3 se calcula s_4 para conocer los valores de θ_3 y θ_4 .

Hasta el momento se conocen 3 ángulos, θ_1 , θ_3 y θ_4 . Los ángulos θ_2 y θ_6 se calculan a partir de la Figura 1.13, en donde se puede observar que

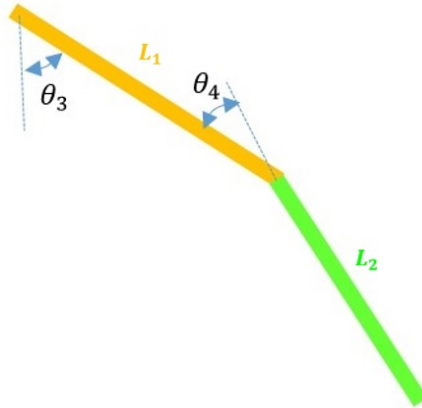


Figura 1.11: Cadena cinemática de las piernas del robot Nao visto como eslabones.

$$\theta_6 = \text{atan}(p_y, p_z) \quad (1.7.21)$$

y que

$$\theta_2 = -\theta_6 \quad (1.7.22)$$

El ángulo θ_5 se obtiene a partir de θ_3 y θ_4 , como se muestra en la Figura 1.14. Teniendo así:

$$\theta_5 = -(\theta_3 + \theta_4) \quad (1.7.23)$$

Con estas ecuaciones se puede calcular la cinemática inversa de las piernas del robot Nao proponiendo un punto (p_x, p_y, p_z) que se encuentre dentro del área de trabajo.

1.8. Generación de trayectorias

El diseño del generador de trayectorias considera dos cosas: los puntos de la trayectoria ZMP en el plano (x, y) y el tamaño del pie del robot, el cual mide 16.0 cm de longitud por 9.0 cm de anchura.

La trayectoria que debe seguir el robot Nao se generará mediante la función senoidal de la ecuación 1.8.1 en el plano (x, y) , con base en una distancia establecida x_d para hacer uso del criterio de estabilidad ZMP, donde x toma valores entre $\frac{-x_d}{2}$ y $\frac{x_d}{2}$. Una función senoidal proporciona movimientos suaves en el robot

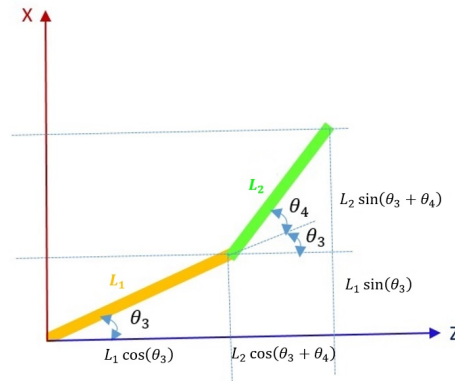


Figura 1.12: Ecuaciones propuestas para la estimación de la cinemática inversa de las piernas del robot Nao.

a comparación de una función escalón. Los puntos de la trayectoria deberán estar dentro del área del pie del robot en el plano (x, y) .

La trayectoria no solo se establece en un solo plano, la fase de oscilación se hace en el plano (x, z) proponiendo la elevación del pie con respecto al piso z_d .

La combinación de movimientos entre la pierna izquierda y la pierna derecha del robot en el eje 'y' tiene la misma trayectoria mostrada en la ecuación 1.8.1, debido a que el efector final es el centro de masas que se encuentra en la cintura del robot como se muestra en la Figura 1.5.

$$y(x) = 50 \sin\left(\frac{\pi}{x_d} x\right) \quad (1.8.1)$$

La generación de la trayectoria queda de la siguiente forma:

1. Cuando el pie derecho este en el piso, al alcanzar una posición en el eje y igual a 50 mm, el pie izquierdo comienza la elevación para estar en la fase de oscilación.
2. Cuando el pie izquierdo este en el piso, al alcanzar una posición en el eje y igual a -50 mm, el pie derecho comienza la elevación para estar en la fase de oscilación.

El ciclo de marcha es una combinación de movimientos entre las piernas y los brazos. En este caso los brazos del robot Nao se dejaron fijos, en una posición que este cerca del torso del mismo para evitar momentos de torsión, así mismo la velocidad con la cual se mueven los servomotores del robot Nao es muy baja, para evitar los momentos inerciales. Un ciclo de marcha se ve como el mostrado

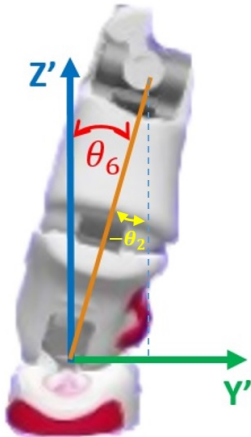


Figura 1.13: Ángulo θ_6 de la pierna del robot Nao.

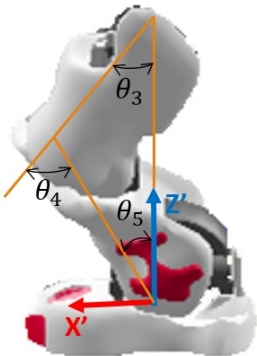


Figura 1.14: Ángulo θ_5 de la pierna del robot Nao.

en la Figura 1.15.

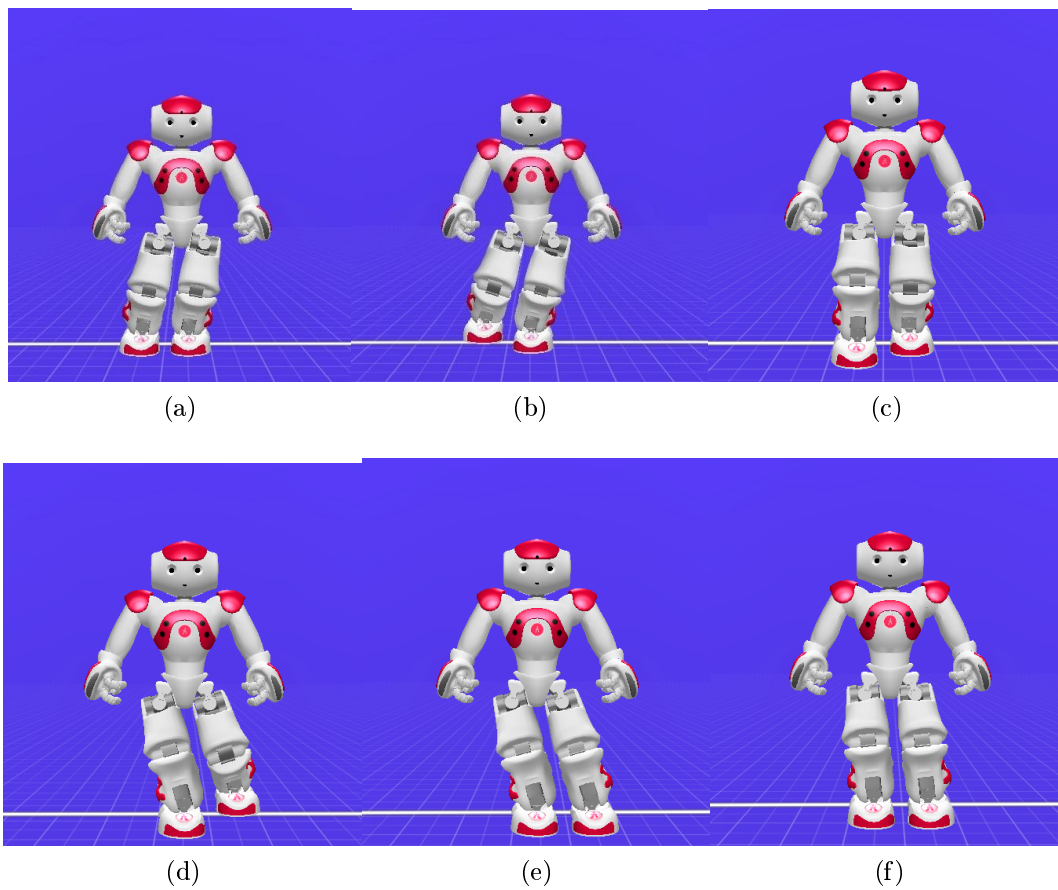


Figura 1.15: Ciclo de marcha del robot Nao simulado en WEBOTS.

1.9. Conclusiones

El cálculo de la cinemática inversa de cualquier tipo de mecanismo o robot es muy difícil de hacer. Por esta razón se buscan herramientas que puedan facilitar este tipo de cálculos. En este caso se propuso resolver la cinemática inversa del robot Nao utilizando la teoría de las bases de Groebner.

Respetando las restricciones de movimiento del robot se seleccionan las ecuaciones que se va a utilizar para calcular la cinemática inversa. Usando la teoría de las bases de Groebner se pueden encontrar todas las posibles soluciones para los ángulos de las articulaciones a partir de dichas ecuaciones estableciendo un orden lexicográfico.

Si se compara la teoría de las bases de Groebner con otros métodos, como lo son el método geométrico y los Algoritmos Genéticos [1], la teoría de las bases de Groebner proporciona todas las soluciones a diferencia de los métodos mencionados que solo encuentran soluciones particulares.

Al aplicar las ecuaciones calculadas con la teoría de las bases de Groebner en el algoritmo de marcha para el robot Nao, el cual siguió los puntos propuestos, esto quiere decir que el cálculo de la cinemática inversa es correcta y por consiguiente se tuvo una marcha adecuada en línea recta hacia adelante.

Apéndice C

Cinemática inversa de un manipulador de dos grados de libertad

En esta sección se presenta la cinemática inversa de un manipulador de dos grados de libertad que se muestra en la figura C.1, la cual se calcula a partir del método geométrico y la teoría de la bases de Groebner.

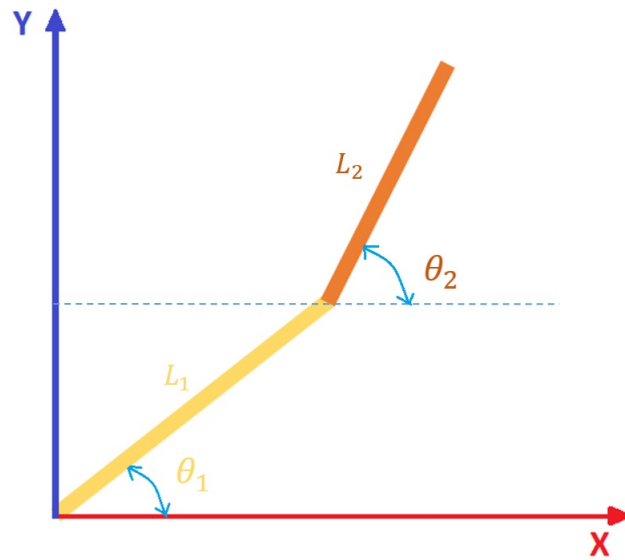


Figura C.1: Manipulador de dos grados de libertad.

Con base en la figura C.2, se calcula la cinemática directa utilizando el método geométrico.

Como se observa en la figura C.2, con la cinemática directa, se llega a un punto determinado. Las funciones trigonométricas que proyectan las distancias L_1 y L_2 en los ejes x y y , se igualan al punto $P(p_x, p_y)$. A partir de esto se obtienen las ecuaciones C.1 y C.2.

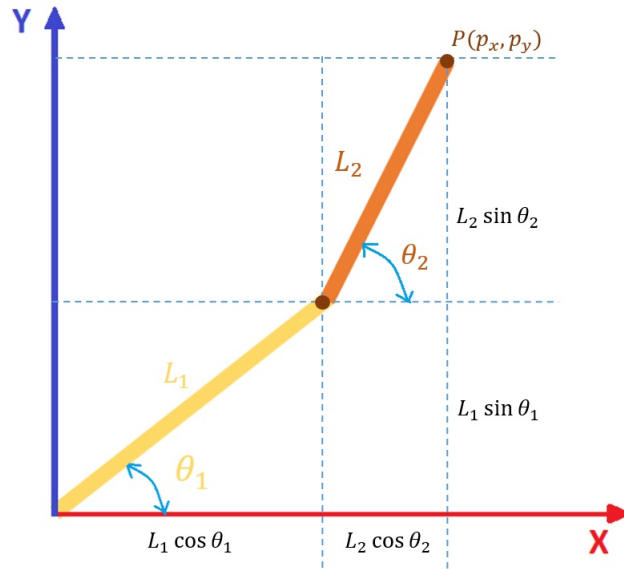


Figura C.2: Manipulador de dos grados de libertad.

$$L_1 \cdot \cos_1 + L_2 \cdot \cos_2 - p_x = 0 \quad (\text{C.1})$$

$$L_1 \cdot \sin_1 + L_2 \cdot \sin_2 - p_y = 0 \quad (\text{C.2})$$

Tomando en cuenta la cinemática directa del brazo manipulador para el cálculo de la cinemática inversa del mismo, se tienen dos variables a conocer, θ_1 y θ_2 . El algoritmo de Buchberger al utilizar álgebra computacional, no permite variables de punto flotante, por lo que solo se dejan expresadas. En este caso, las variables se nombraron de la siguiente forma: $s_1 = \sin \theta_1$, $c_1 = \cos \theta_1$, $s_2 = \sin \theta_2$, $c_2 = \cos \theta_2$. Sustituyendo estas en las ecuaciones C.1 y C.2 se obtiene:

$$L_1 \cdot c_1 + L_2 \cdot c_2 - p_x = 0 \quad (\text{C.3})$$

$$L_1 \cdot s_1 + L_2 \cdot s_2 - p_y = 0 \quad (\text{C.4})$$

Al hacer la sustitución, se obtienen más variables que ecuaciones (4 variables y dos ecuaciones), por esta razón, el sistema no se puede resolver.

El método de la teoría de las Bases de Groebner establece que para completar el sistema se propongan las ecuaciones faltantes a partir de identidades trigonométricas. Se proponen las ecuaciones C.5 y C.6 para completar el sistema de ecuaciones y convertirlas a Bases de

Groebner.

$$s_1^2 + c_1^2 - 1 = 0 \quad (\text{C.5})$$

$$s_2^2 + c_2^2 - 1 = 0 \quad (\text{C.6})$$

Para convertir las ecuaciones C.3 a la C.6 a Bases de Groebner, se emplea el comando *Groebner*[51] definido en el software *Maple*TM, el cual ejecuta el algoritmo de Buchberger que se muestra en el capítulo 3.

El software *Maple*TM es una herramienta matemática estándar para el cálculo analítico que permite realizar álgebra simbólica, cálculo numérico, resolver ecuaciones diferenciales, gráficos y animaciones [52].

Estas ecuaciones se capturan en *Maple*TM con el orden lexicográfico c_2, c_1, s_2, s_1 de la siguiente manera:

$$F := \{c1^2 + s1^2 - 1, c3^2 + s3^2 - 1, L1 * c1 + L2 * c2 - px, L1 * s1 + L2 * s2 - py\}$$

with(Groebner) :

$$G := \text{Basis}(F, \text{plex}(c2, c1, s2, s1))$$

*Maple*TM da como resultado las ecuaciones C.7 - C.10:

$$\begin{aligned} & (4 \cdot L_1^2 \cdot p_x^2 + 4 \cdot L_1^2 \cdot p_y^2) \cdot s_1^2 \\ & + (-4 \cdot L_1^3 \cdot p_y + 4 \cdot L_1 \cdot L_2^2 \cdot p_y - 4 \cdot L_1 \cdot p_x^2 \cdot p_y - 4 \cdot L_1 \cdot p_y^3) \cdot s_1 \\ & + L_1^4 - 2 \cdot L_1^2 \cdot L_2^2 - 2 \cdot L_1^2 \cdot p_x^2 + 2 \cdot L_1^2 \cdot p_y^2 + L_2^4 \\ & - 2 \cdot L_2^2 \cdot p_x^2 - 2 \cdot L_2^2 \cdot p_y^2 + p_x^4 + 2 \cdot p_x^2 \cdot p_y^2 + p_y^4 = 0 \end{aligned} \quad (\text{C.7})$$

$$L_1 \cdot s_1 + L_2 \cdot s_2 - p_y = 0 \quad (\text{C.8})$$

$$2 \cdot L_1 \cdot c_1 \cdot p_x + 2 \cdot L_1 \cdot p_y \cdot s_1 - L_1^2 + L_2^2 - p_x^2 - p_y^2 = 0 \quad (\text{C.9})$$

$$-2 \cdot L_1 \cdot p_y \cdot s_1 + 2 \cdot L_2 \cdot c_2 \cdot p_x + L_1^2 - L_2^2 - p_x^2 + p_y^2 = 0 \quad (\text{C.10})$$

De las ecuaciones en Bases de Groebner se eligen C.7 y C.8, que son las que calculan s_1 y s_2 para conocer θ_1 y θ_2 respectivamente. Se eligen las variables senoidales debido a que esta función trigonométrica es signada.

Para conocer el valor de s_1 se observa que la ecuación C.7 es de segundo grado y se puede resolver con la fórmula C.11.

$$s_1 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (\text{C.11})$$

En donde:

$$a = 4 \cdot L_1^2 \cdot p_x^2 + 4 \cdot L_1^2 \cdot p_y^2$$

$$b = -4 \cdot L_1^3 \cdot p_y + 4 \cdot L_1 \cdot L_2^2 \cdot p_y - 4 \cdot L_1 \cdot p_x^2 \cdot p_y - 4 \cdot L_1 \cdot p_y^3$$

$$c = L_1^4 - 2 \cdot L_1^2 \cdot L_2^2 - 2 \cdot L_1^2 \cdot p_x^2 + 2 \cdot L_1^2 \cdot p_y^2 + L_2^4 - 2 \cdot L_2^2 \cdot p_x^2 - 2 \cdot L_2^2 \cdot p_y^2 + p_x^4 + 2 \cdot p_x^2 \cdot p_y^2 + p_y^4$$

La raíz a utilizar para calcular el valor de s_1 depende del área de trabajo del manipulador. Una vez que se conoce el valor de s_1 , se puede calcular el valor de s_2 como se muestra en la ecuación C.12.

$$s_2 = \frac{p_y - L_1 \cdot s_1}{L_2} \quad (\text{C.12})$$

Con los valores de s_1 y s_2 , se obtienen los valores de θ_1 y θ_2 sacando el inverso del seno de las ecuaciones C.11 y C.12, como se muestra en las ecuaciones C.13 y C.14.

$$\theta_1 = \arcsin(s_1) \quad (\text{C.13})$$

$$\theta_2 = \arcsin(s_2) \quad (\text{C.14})$$

Por ejemplo, si $L_1 = 250$ mm, $L_2 = 200$ mm, $p_x = 400$ mm y $p_y = 100$ mm y se sustituyen en las ecuaciones C.7 y C.8, se tienen las ecuaciones C.15 y C.16 respectivamente.

$$42500000000 \cdot s_1^2 - 19250000000 \cdot s_1 - 2943750000 = 0 \quad (\text{C.15})$$

$$250 \cdot s_1 + 200 \cdot s_2 - 100 = 0 \quad (\text{C.16})$$

Calculando el valor de s_1 a partir de la ecuación C.11, se tienen dos resultados:

$$s_1 = 0.57367893 \quad (\text{C.17})$$

$$s_1 = -0.120737754 \quad (\text{C.18})$$

El resultado para calcular s_2 depende del área de trabajo del manipulador. En este caso se utiliza la raíz positiva, que da como resultado C.17. Sustituyendo C.17 en la ecuación C.8, se tiene que:

$$s_2 = -0.217098663 \quad (\text{C.19})$$

Despejando θ_1 y θ_2 de C.17 y C.19, se tiene que

$$\theta_1 = 0.610990365 \text{ rad}$$

$$\theta_2 = -0.218841258 \text{ rad}$$