



UNIVERSIDAD AUTÓNOMA DE
SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

CENTRO DE INVESTIGACIÓN Y
ESTUDIOS DE POSGRADO

CONTROL DEL CENTRO DE MASA
DE ROBOTS HUMANOIDES PARA
TAREAS DE LOCOMOCIÓN

TESIS

Que para obtener el grado de:
Maestro en Ingeniería Mecánica
Opción: Mecatrónica y Sistemas Mecánicos

Presenta:

Ing. Hugo Edgardo Oliva Moreno

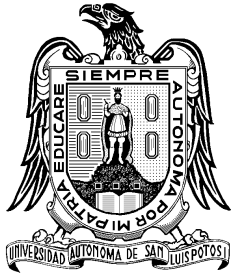
Asesor:

**Dr. Mauro Eduardo Maya
Méndez**

San Luis Potosí, S.L.P.

Febrero de 2018





Universidad Autónoma de San Luis Potosí

Facultad de Ingeniería

Centro de Investigación y Estudios de Posgrado



Maestría en Ingeniería Mecánica

Opción: Mecatrónica y Sistemas Mecánicos

CONTROL DEL CENTRO DE MASA DE ROBOTS HUMANOIDES PARA TAREAS DE LOCOMOCIÓN

Presenta:

Ing. Hugo Edgardo Oliva Moreno

Sinodales:

Dr. Mauro Eduardo Maya Méndez

Asesor

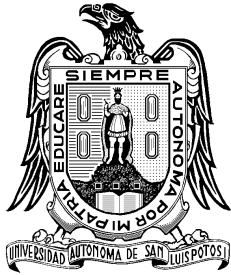
Dr. Juan Antonio Cárdenas Galindo

Sinodal

Dr. Héctor Méndez Azúa

Sinodal

San Luis Potosí, S.L.P., febrero de 2018



Universidad Autónoma de San Luis Potosí

Facultad de Ingeniería

Centro de Investigación y Estudios de Posgrado



Aclaración

El presente trabajo que lleva por título

"Control del centro de masa de robots humanoides para tareas de locomoción"

Se realizó en el periodo de enero de 2016 a febrero de 2018 bajo la dirección del Dr. Mauro Eduardo Maya Méndez.

Originalidad

Por este medio aseguro que he realizado este documento de tesis para fines académicos sin ayuda indebida de terceros y sin utilizar otros medios más que los indicados.

Las referencias e información tomadas directa o indirectamente de otras fuentes se han definido en el texto como tales y se ha dado el debido crédito a las mismas.

El autor exime a la UASLP de las opiniones vertidas en este documento y asume la responsabilidad total del mismo.

Este documento no ha sido sometido como tesis a ningún otra institución nacional o internacional en forma parcial o total.

Se autoriza a la UASLP para que divulgue este documento de tesis para fines académicos.

Hugo Edgardo Oliva Moreno

Agradecimientos

Primeramente, agradezco a mi familia, quienes son mi soporte y me han motivado a continuar desarrollándome profesional y académicamente.

También agradezco a mis compañeros Rebeca, Brenda, Stefanny y Carlos por el tiempo compartido y la amistad que logramos formar durante el programa. A Robert y Antonio mis compañeros de clase y cubículo por las interminables pláticas y bromas que hicieron mucho más ameno este proceso.

Agradezco a la Facultad de Ingeniería de la UASLP y a CONACyT por financiar mi estancia y los recursos utilizados para el desarrollo de este trabajo de tesis. Al Dr. Mauro Maya, por todo su tiempo y su valiosa asesoría para el desarrollo de este trabajo.

Resumen

En la presente tesis se desarrollan diversas estrategias de control que permiten obtener estabilidad mecánica de un robot humanoide tanto cuando éste se encuentra de posición estática de pie, como durante tareas de marcha a bajas velocidades. Las estrategias se basan en la cinemática inversa del centro de masa y cinemática inversa de robots redundantes.

Se presenta el modelado cinemático directo e inverso de un humanoide de 18 grados de libertad, haciendo hincapié en las consideraciones necesarias para hacer el modelado del humanoide.

Un robot humanoide presenta una gran variedad de complicaciones tanto en su control como en su modelado. En general un humanoide está constituido de un gran número de eslabones y juntas, por lo que, normalmente son considerados como robots redundantes o incluso hiper redundantes. La redundancia del humanoide implica que existen soluciones infinitas para su cinemática inversa y obliga a usar métodos iterativos para obtener alguna solución aproximada de la misma. Sumado al problema anterior, cuando se considera una tarea de marcha de un humanoide, se debe considerar una tarea adicional que mantenga el equilibrio del robot. Para poses estáticas y movimientos en baja velocidad, donde la inercia es despreciable, el equilibrio está dado por la proyección del centro de masa del humanoide sobre el polígono de apoyo de éste.

Dada la naturaleza de los problemas descritos, en este trabajo se busca controlar el centro de masa del robot humanoide por medio de diversos algoritmos, con la intención de obtener resultados eficientes, tanto en tiempo de ejecución, como en precisión en el posicionamiento del centro de masa para su futura aplicación práctica.

Además, hablando de un prototipo físico, estos cuentan con un rango limitado de movimiento para cada junta, lo cual se debe de tener en cuenta dentro de los algoritmos para evitar obtener poses no realizables y auto-colisiones. Estas restricciones afectan a los algoritmos, ya que en vez de buscar una de las soluciones infinitas que resuelva los problemas

se debe de buscar una solución que cumpla sus objetivos, manteniendo a las juntas dentro de sus rangos operativos, propiciando un aumento en el tiempo de ejecución y el error en el posicionamiento del centro de masa.

Finalmente, se generan trayectorias para el centro de masa y el pie en vuelo, con la finalidad de obtener un trayecto de marcha conformado de una serie de poses estáticas, que cumplan con los criterios de estabilidad. Se implementan las estrategias multi-tarea de vector aumentado de tareas y control jerárquico para ejecutar correctamente las trayectorias del centro de masa y pie en vuelo previamente mencionadas.



Universidad Autónoma
de San Luis Potosí

14 de diciembre de 2017

**ING. HUGO EDGARDO OLIVA MORENO
P R E S E N T E.**

En atención a su solicitud de Temario, presentada por el **Dr. Mauro Eduardo Maya Méndez**, Asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de **Maestro en Ingeniería Mecánica con Orientación Terminal en Mecatrónica y Sistemas Mecánicos**, me es grato comunicarle que en la Sesión del H. Consejo Técnico Consultivo celebrada el día 14 de diciembre del presente, fue aprobado el Temario propuesto:

TEMARIO:

“Control del centro de masa de robots humanoides para tareas de locomoción”

Introducción

1. Antecedentes de robótica humanoide.
2. Modelado de robots humanoides.
3. Control del centro de masa para robots humanoides.
4. Aplicación de control de centro de masa a tareas de caminado de un robot humanoide.

Conclusiones.

Bibliografía.

Anexos.

“MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO”

A T E N T A M E N T E

**M. I. JORGE ALBERTO PÉREZ GONZÁLEZ
DIRECTOR.**

UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
DIRECCIÓN



FACULTAD DE
INGENIERÍA

Copia. Archivo.
*etn.

Av. Manuel Nava 8
Zona Universitaria • CP 78290
San Luis Potosí, S.L.P.
tel. (444) 826 2330 al39
fax (444) 826 2336
www.uaslp.mx

Índice General

Capítulos	Página
Introducción.	1
1. Antecedentes de robótica humanoide	5
1.1. Robots	5
1.2. Robots humanoides	6
1.3. Estado del arte de la robótica humanoide	7
1.3.1. Robots humanoides en la actualidad	9
1.3.2. Robots "do it yourself"	16
1.4. Estabilidad de un robot humanoide	18
1.4.1. Estabilidad estática	19
1.4.2. Estabilidad dinámica	19
1.5. Elementos básicos del modelado de robots	21
1.5.1. Cinemática directa	26

1.5.2.	Cinemática inversa	28
1.5.3.	Robots redundantes	29
1.5.4.	Cinemática directa del centro de masa	31
1.5.5.	Cinemática inversa del centro de masa	32
1.6.	Pseudo-inversa	34
1.6.1.	Pseudo-inversa ponderada	35
1.6.2.	Pseudo-inversa amortiguada	35
1.6.3.	Pseudo-inversa ponderada-amortiguada	36
1.7.	Algoritmos límite de junta	37
1.7.1.	Uso de ponderación para restringir límites en las juntas	38
1.8.	Péndulo Invertido Lineal	40
1.8.1.	Péndulo Invertido Lineal 3D	42
2.	Modelado de robots humanoides	45
2.1.	Modelado de robots humanoides	45
2.2.	Cinemática directa de un robot humanoide	47
2.2.1.	Cinemática directa respecto a la cadera	47
2.2.2.	Cinemática directa respecto a la planta del pie	52
2.3.	Cinemática inversa de un robot humanoide	55
2.3.1.	Cinemática inversa de la pierna	55
2.3.2.	Cinemática inversa del brazo	58
2.3.3.	Cinemática inversa del cuerpo completo	59
2.4.	Generación de trayectorias cinemáticas	61
2.5.	Conclusión del capítulo	65
3.	Control del centro de masa para robots humanoides	67

3.1.	Generación de poses estables	67
3.1.1.	Obtención de distribución de masas objetivo	68
3.2.	Obtención de configuración articular	70
3.2.1.	División por Extremidades (DE)	71
3.2.2.	Manipulación Simultánea de masas principales(MS)	73
3.2.3.	Manipulación del Centro de Masa Total (MCMT)	74
3.3.	Pruebas de rendimiento	75
3.4.	Conclusiones del capítulo	80
4.	Aplicación de control de centro de masa a tareas de caminado de un robot humanoide	83
4.1.	Generación de trayectorias de marcha	83
4.1.1.	Generación de trayectoria por <i>splines</i> cúbicas	84
4.1.2.	Generación de trayectoria por Péndulo Invertido Lineal 3D	84
4.2.	Definición de tareas	85
4.2.1.	Posicionamiento del centro de masa	85
4.2.2.	Posicionamiento y orientación del pie en vuelo	86
4.2.3.	Orientación del torso	86
4.3.	Marcha	87
4.3.1.	Múltiples tareas como vector aumentado de tareas	87
4.3.2.	Múltiples tareas como control jerárquico	92
4.3.3.	Prioridad primaria a posicionamiento del centro de masa, la orientación del pie en vuelo y orientación de cadera	97
4.4.	Adición de límite de juntas a algoritmos	101
4.4.1.	Múltiples tareas como vector aumentado y límites de junta	101
4.4.2.	Control jerárquico y límites de junta	105

4.5. Tiempos de ejecución	113
4.6. Conclusiones del capítulo	114
Conclusiones.	115
Bibliografía	119
Anexos	I
A. Límites de junta	III
B. Matriz de Pesos	v

Lista de Figuras

1.	Robot YuMi (tomado de [ABB, 2018]).	1
2.	Robot iCUB (tomado de [Metta <i>et al.</i> , 2010]).	2
3.	Caída del Equipo IHMC ROBOTICS durante DARPA ROBOTICS CHALLENGE 2015.	3
1.1.	Tipos de Robots.	6
1.2.	WABOT y WABOT-2.	8
1.3.	Parámetros para generar trayectorias de caminado (tomado de [Huang <i>et al.</i> , 1999]).	9
1.4.	Robot Atlas (tomado de [DYNAMICS, 2017a]).	10
1.5.	Robot Atlas caminando sobre superficie irregular (tomado de [Wiedebach <i>et al.</i> , 2016]).	11
1.6.	Robot Petman (tomado de [Velasco, 2011]).	11
1.7.	Robot HRP-02 (tomado de [KAWADA INDUSTRIES, 2017]).	12
1.8.	Serie de robots HRP.	13
1.9.	Robot WABIAN 2/RIII (tomado de [Ogura <i>et al.</i> , 2006]).	14

1.10. "Se ha tomado un enfoque paso a paso, comenzando con una caminata básica y estática, pasando por una caminata inteligente ajustable automáticamente, finalizando con la capacidad de trabajar para un ser humano" (tomado de [Hirose y Ogawa, 2007]).	15
1.11. Robots HONDA P2, P3 y ASIMO (tomado de [Fitzpatrick <i>et al.</i> , 2016]).	15
1.12. Robot HUBO dentro de una prueba de DARPA robotics challenge (tomado de [Wang <i>et al.</i> , 2014]).	16
1.13. Robot HOVIS Lite (tomado de [Robotics, 2017b]).	17
1.14. Robot Bioloid Premium (tomado de [Shop, 2017]).	18
1.15. Comparación entre el COM y ZMP (tomado de [Kajita <i>et al.</i> , 2014]).	18
1.16. Ciclo de marcha.	21
1.17. Vector de fuerza de reacción del suelo durante el caminado (tomado de [Vaughan, 2003]).	21
1.18. Representación de rotaciones de Euler zyz (tomado de [Spong <i>et al.</i> , 2006]).	23
1.19. Traslación entre marcos de referencia.	24
1.20. Configuración articular obtenida a partir de algoritmo de Cinemática inversa del centro de masa (CmICOM) para una cadena cinemática de 4 Grado de libertad (DOF).	33
1.21. Configuración articular obtenida a partir de algoritmo de CmICOM para una cadena cinemática de 8 DOF.	34
1.22. Comportamiento de función $ \partial H(\theta)/\partial \theta_i $	39
1.23. Péndulo Invertido Lineal (tomado de [Kajita <i>et al.</i> , 2014]).	40
1.24. Péndulo Invertido Lineal 3D (tomado de [Kajita <i>et al.</i> , 2014]).	42
2.1. Robot BIOLOID PREMIUM (vista frontal y lateral).	46
2.2. Modelo simplificado de robot Bioloid premium.	47
2.3. Representación de las juntas del brazo izquierdo del robot humanoide.	49
2.4. Representación de las juntas del brazo derecho del robot humanoide.	49

2.5.	Representación de las juntas de las piernas.	51
2.6.	Esquema geométrico de una pierna del robot humanoide.	56
2.7.	Esquema geométrico del brazo izquierdo del robot humanoide.	58
2.8.	Algoritmo utilizado para la generación de trayectorias cinemáticas.	62
2.9.	Pose obtenida a partir del algoritmo de cinemática inversa para el robot completo.	63
2.10.	Trayectoria cinemática generada para un paso.	64
3.1.	Distribución de masas principales m_k	70
3.2.	Algoritmo de División por Extremidades (DE).	73
3.3.	Algoritmo de Manipulación Simultánea.	74
3.4.	Algoritmo de Manipulación del Centro de Masa Total.	76
3.5.	Errores en simulación de los algoritmos de posicionamiento del COM.	78
3.6.	Tiempos de ejecución resultantes durante las simulaciones de los algoritmos de posicionamiento del COM.	78
3.7.	Poses obtenidas con el algoritmo MCMT y valores de $\lambda = 0.1$ y $\alpha = 0.1$	79
3.8.	Poses obtenidas con el algoritmo MSAV y valores de $\lambda = 0.5$ y $\alpha = 0.1$	79
3.9.	Poses obtenidas con el algoritmo DEAV y valores de $\lambda = 0.5$ y $\alpha = 0.1$	80
4.1.	Secuencia de movimientos generados para un paso usando la estrategia de VAT.	89
4.2.	Trayectoria de pie en vuelo para VAT.	90
4.3.	Trayectoria del COM para VAT.	90
4.4.	Error de posicionamiento de pie en vuelo para VAT.	91
4.5.	Error de posicionamiento del COM para VAT.	91
4.6.	Secuencia de movimientos generados para un paso usando la estrategia de CJ con prioridad en el posicionamiento del pie.	94
4.7.	Trayectoria de pie en vuelo para CJ con prioridad en el posicionamiento del pie.	95

4.8. Trayectoria del COM para CJ con prioridad en el posicionamiento del pie.	95
4.9. Error de posicionamiento de pie en vuelo para CJ con prioridad en el posicionamiento del pie.	96
4.10. Error de posicionamiento del COM para CJ con prioridad en el posicionamiento del pie.	96
4.12. Trayectoria de pie en vuelo para CJ con prioridad en el posicionamiento del COM.	98
4.13. Trayectoria del COM para CJ con prioridad en el posicionamiento del COM.	98
4.14. Error de posicionamiento de pie en vuelo para CJ con prioridad en el posicionamiento del COM.	99
4.15. Error de posicionamiento del COM para CJ con prioridad en el posicionamiento del COM.	99
4.11. Secuencia de movimientos generados para un paso usando la estrategia de CJ con prioridad en el posicionamiento del COM.	100
4.17. Trayectoria de pie en vuelo para VATL.	102
4.18. Trayectoria del COM para VATL.	102
4.19. Error de posicionamiento de pie en vuelo para VATL.	103
4.20. Error de posicionamiento del COM para VATL.	103
4.16. Secuencia de movimientos generados para un paso usando la estrategia de VATL.	104
4.21. Secuencia de movimientos generados para un paso usando la estrategia de CJL con prioridad en el posicionamiento del pie.	106
4.22. Trayectoria de pie en vuelo para CJL con prioridad en el posicionamiento del pie.	107
4.23. Trayectoria del COM para CJL con prioridad en el posicionamiento del pie.	107
4.24. Error de posicionamiento de pie en vuelo para CJL con prioridad en el posicionamiento del pie.	108
4.25. Error de posicionamiento del COM para CJL con prioridad en el posicionamiento del pie.	108

4.26. Secuencia de movimientos generados para un paso usando la estrategia de CJL con prioridad en el posicionamiento del COM.	110
4.27. Trayectoria de pie en vuelo para CJL con prioridad en el posicionamiento del COM.	111
4.28. Trayectoria del COM para CJL con prioridad en el posicionamiento del COM.	111
4.29. Error de posicionamiento de pie en vuelo para CJL con prioridad en el posicionamiento del COM.	112
4.30. Error de posicionamiento del COM para CJL con prioridad en el posicionamiento del COM.	112

Lista de Tablas

2.2.1. Parámetros D-H de los brazos.	50
2.2.2. Parámetros D-H de las piernas.	52
2.2.3. Parámetros D-H de la pierna de apoyo.	53
3.3.1. Valores de las constantes λ y α usados en las simulaciones de los algoritmos de posicionamiento del COM.	76
3.3.2. Errores de posicionamiento (mm) de las simulaciones de los algoritmos de posicionamiento del COM.	77
3.3.3. Tiempos de ejecución (ms) de las simulaciones de los algoritmos de posicionamiento del COM.	77
A.1. Ángulos límite de robot Bioloid Premium en los brazos.	III
A.2. Ángulos límite de robot Bioloid Premium en las piernas.	III
B.1. Pesos asignados a las juntas del robot para las simulaciones.	V

Abreviaciones

COM Centro de masa

DOF Grado de libertad

ZMP Punto de cero momento

ASIMO *Advanced Step in Innovative Mobility*

D-H Denavit-Hartenberg

CmD Cinemática directa

CmI Cinemática inversa

CmDCOM Cinemática directa del centro de masa

CmICOM Cinemática inversa del centro de masa

HRP *Humanoid Robotics Project*

AIST *Agency of Industrial Science and Technology*

MITI *Ministry availability of International Trade and Industry*

JRA *Joint Range Availability*

KHR *KAIST Humanoid Robot*

WABOT Waseda Robot

DIY *Do It Yourself*

MIT *Massachusetts Institute of Technology*

KAIST *Korean Agency of Industrial Science and Technology*

DE División por Extremidades

MS Manipulación Simultanea

DEAV División por Extremidades con Amortiguamiento Variable

MSAV Manipulación Simultánea con Amortiguamiento Variable

MCMT Manipulación del Centro de Masa Total

VAT Vector Aumentado de Tareas

VATL Vector Aumentado de Tareas con Límites

CJ Control Jerárquico

CJL Control Jerárquico con Límites

IHMC *Institute for Human and Machine Cognition*

PIL Péndulo Invertido Lineal

Introducción

La robótica en general está experimentando actualmente un crecimiento constante . La creación de robots bio inspirados, el uso de robots redundantes, los robots colaborativos, la robótica humanoide, etc., son ejemplos de los desarrollos presentes en la última década. Estas nuevas tecnologías poco a poco se han ido incorporando a los ámbitos industrial y social. El caso más llamativo es el robot YuMi de la empresa ABB [ABB, 2018]; éste es un robot de dos brazos redundantes y además tiene las características de un robot colaborativo, pudiendo trabajar en un entorno junto con seres humanos. El robot YuMi se puede observar en la figura 1.

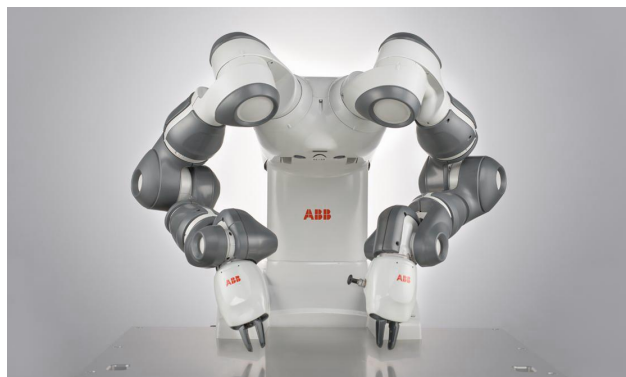


Figura 1. Robot YuMi (tomado de [ABB, 2018]).

La robótica humanoide ha tenido un creciente desarrollo e interés en los últimos treinta años. Las mejoras como la miniaturización de componentes, la mayor capacidad de cálculo de

las computadoras actuales e incluso las tecnologías de manufactura aditiva, han propiciado la creación de más y mejores modelos de robots humanoides. Actualmente existen diversos robots humanoides que captan la atención del público, debido a su capacidad de movimiento y sus habilidades para interactuar con el entorno. Dentro de los robots más avanzados se encuentra ATLAS de Boston Dynamics [DYNAMICS, 2017a], capaz de adaptarse a entornos no definidos y hacer diversas tareas, como detectar y cargar objetos.

Otra de las tendencias dentro de la robótica humanoide, son las técnicas conocidas como "Machine learning" e inteligencia artificial, con los que se espera que los robots sean capaces de aprender un comportamiento a partir de la experiencia y repetición de diversas situaciones. Uno de los ejemplos más claros es el robot iCUB [Metta *et al.*, 2010, Kormushev *et al.*, 2010], el cual ha sido diseñado como un entorno de prueba para las nuevas tendencias de aprendizaje por lenguaje natural. El robot iCUB se puede observar en la figura 2.

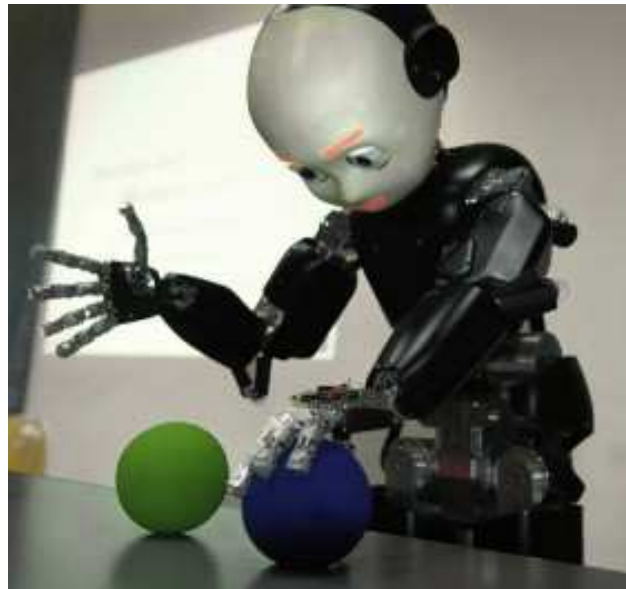


Figura 2. Robot iCUB (tomado de [Metta *et al.*, 2010]).

A pesar de los avances en robótica humanoide a nivel mundial, aún hay problemas que resolver, entre ellos destaca la estabilidad mecánica de un humanoide(bípedo) tanto en poses estáticas como durante la marcha. El ejemplo más claro de este problema se presentó durante el concurso "DARPA ROBOTICS CHALLENGE 2015", en donde una gran cantidad de robots bípedos perdieron el equilibrio durante la marcha o incluso de manera abrupta mientras se encontraban de pie [Limer, 2015, Guizzo y Ackerman, 2015]. En la figura 3 se puede ver al robot del equipo *Institute for Human and Machine Cognition* (IHMC) ROBOTICS cayendo durante un trayecto del concurso.



(a) Robot IHMC ROBOTICS (tomado de [Limer, 2015]).



(b) Robot IHMC ROBOTICS (tomado de [Limer, 2015]).

Figura 3. Caída de Equipo IHMC ROBOTICS durante DARPA ROBOTICS CHALLENGE 2015.

El desarrollo de robots humanoide en México, es escaso. En general no existe un desarrollo a destacar de un humanoide propio dentro del país, sin embargo, se sabe que al menos existen 80 robots Nao hasta el año 2016 [Conacyt, 2016], estos son usados principalmente para divulgación científica. Entre las instituciones conocidas que tienen líneas de investigación asociadas a la robótica humanoide, se encuentran: Cinvestav Unidad Saltillo, el Instituto Tecnológico de la Laguna y la Universidad Politécnica de Sinaloa.

Dadas las necesidades mostradas, en este trabajo de tesis se busca crear estrategias de control que permitan obtener estabilidad mecánica de un robot humanoide durante la operación de marcha. Los objetivos tanto generales como particulares de este trabajo quedan definidos como:

Objetivo general

Desarrollar estrategias de control del centro de masa de un robot humanoide para la operación de marcha.

Objetivos particulares

- Modelado de un robot humanoide.
- Análisis de trayectorias cinemáticas.
- Desarrollo de estrategias de control para estabilizar trayectorias de marcha para diferentes arquitecturas de robots humanoides.
- Validación en simulación de los controles desarrollados.

Esta tesis está estructurada como sigue:

- **Capítulo 1: Antecedentes de robótica humanoide.** En este capítulo se presentan los diversos avances existentes dentro de la robótica humanoide , además, se hace un breve recordatorio múltiples conceptos básicos utilizados tanto en el modelado como en el control de robots.
 - **Capítulo 2: Modelado de robots humanoides.** Se presentan varias consideraciones necesarias para modelar un robot humanoide de forma cinemática y se desarrolla el modelo cinemático directo e inverso de un humanoide de 18 DOFs.
 - **Capítulo 3: Control del centro de masa para robots humanoides.** Se desarrollan tres algoritmos para manipular el centro de masas de un robot humanoide de n DOF. Los algoritmos se basan en el cálculo de la cinemática inversa del centro de masa y cinemática inversa de robots redundantes.
 - **Capítulo 4: Aplicación de control de centro de masa a tareas de caminado de un robot humanoide.** Se generan trayectorias para el centro de masa y el pie en fase de vuelo. Además, se aplican estrategias de control de múltiples tareas durante la operación de marcha, con la finalidad de lograr las trayectorias generadas
-

Antecedentes de robótica humanoide

La curiosidad y deseo por construir mecanismos que reproduzcan los movimientos humanos y posteriormente aplicarlos a alguna tarea, han estado presentes desde la antigüedad. El estudio del cuerpo humano y sus movimientos se remonta a los primeros médicos de la historia, teniendo un avance lento hasta llegar al presente. Actualmente es posible hacer capturas de movimientos basados en imágenes, tomadas durante una acción de un ser humano y posteriormente reproducirlas en un robot humanoide existente. Otro de los avances presentes en la actualidad tiene que ver con la capacidad de movimiento en los robots humanoides actuales derivado de los progresos en la miniaturización de los componentes tanto mecánicos como electrónicos del robot, lo que, aunque dificulta su análisis, logra dotar al robot con la capacidad de realizar acciones que hace 50 años eran imposibles.

Dentro de este capítulo se hace una recapitulación de la historia y diversos avances en la robótica humanoide. Además, se tratan conceptos teóricos de suma importancia para la estabilidad de un humanoide durante la ejecución de tareas. Posteriormente se tratan las metodologías usadas en el análisis de robots y robots redundantes.

1.1. Robots

El concepto de Robot ha ido evolucionando tomando como una de las definiciones más aceptadas "Un robot es un manipulador funcional reprogramable, capaz de mover material, piezas, herramientas o dispositivos especializados mediante movimientos variables programados, con el fin de realizar tareas diversas" propuesta por el *Robot Institute of America* en 1979. Esta definición está totalmente orientada a robots industriales ya que en la actualidad el ámbito industrial es donde los robots tienen un mayor uso.

Los robots basados en mecanismos seriales son los más comunes, sin embargo, los robots

paralelos están ganando terreno debido a sus mejores prestaciones, tanto en velocidad como en su relación entre carga-peso. En la figura 1.1(a) se puede observar un robot serial y en la figura 1.1(b) un robot paralelo, los dos pertenecientes a la marca ABB.



(a) Robot serial ABB IRB2400 (tomado de [ROBOTRADERS, 2017]).

(b) Robot paralelo ABB IRB 360 (tomado de [Robotics, 2017a]).

Figura 1.1. Tipos de Robots.

Además de los robots industriales antes mencionados existen una gran cantidad de robots con diferentes configuraciones, formas y tamaños, buscando resolver desde la colocación de una pieza sobre alguna plataforma hasta poder volar, nadar o caminar de forma autónoma, siendo los humanoides los más complejos. La robótica humanoide busca representar un cuerpo humano como un robot y que éste sea capaz de realizar la mayoría de tareas de un ser humano.

1.2. Robots humanoides

Los robots humanoides son como se definen en [Kajita *et al.*, 2014] "Un robot humanoide es un robot que tiene una forma humana", es decir, son aquellos robots que tienen una forma parecida a un cuerpo humano. Además, existen otras definiciones o clasificaciones donde se incluye la inteligencia y capacidad cognitiva del robot [Eaton, 2015], sin embargo, debido al carácter de este trabajo, no serán tomadas en cuenta.

Existen muchas justificaciones para buscar la forma humana en un robot, desde la sustitución directa de un humano por un robot para la ejecución de tareas peligrosas, hasta su uso en espectáculos y entretenimiento. Una de las principales ventajas que provee un modelo humanoide es que el entorno en el que se desarrolla no tendrá que ser modificado, pudiendo

desarrollar sus actividades en un ambiente diseñado para personas y además manipular artefactos, herramientas y maquinaria humana [Kemp *et al.*, 2008]. Otro de los factores beneficiosos que favorecen a los robots humanoides es su aceptación entre las personas.

En cuanto a movilidad un robot humanoide en teoría es igual de versátil que un cuerpo humano, es decir, es capaz de adaptarse y realizar una gran cantidad de tareas, como caminar, sujetar objetos, brincar, subir escaleras, nadar, agacharse, etc. Una de sus principales cualidades es el hecho de contar con dos piernas dando una clara ventaja sobre los robots con ruedas al desplazarse sobre terrenos irregulares. Las piernas permiten al robot humanoide obtener una mejor adaptación del cuerpo sobre el terreno de desplazamiento.

1.3. Estado del arte de la robótica humanoide

A pesar de que el concepto de robot y humanoide son relativamente modernos, existen diversas aproximaciones previas en la creación de éstos. En la época medieval se crearon diversos desarrollos alrededor del mundo de mecanismos con formas humanas, siendo de los más conocidos el "caballero mecánico" de Da Vinci en 1495 [Yates *et al.*, 2011, Da Vinci, 2012] o el mecanismo con forma humanoide desarrollado por Al-Jazari [Mechanics, 2010].

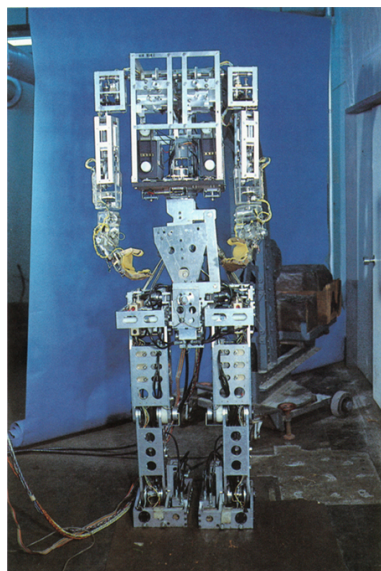
Los intentos por reproducir el caminado humano tiene sus inicios cerca del siglo XIX. Uno de los ejemplos más destacados es el mecanismo conocido como *STEAM MAN* [Dederick, 1868]; este desarrollo consistía en una máquina de vapor conectada a un sistema de válvulas, capaces de reproducir el movimiento de las piernas de un ser humano.

La palabra robot desde sus orígenes ha estado asociada a la idea de una máquina en forma humana teniendo su primera aparición en el año de 1925 en [Čapek, 1925] donde el término robot se usa para describir máquinas con forma humanoide usadas para la manufactura barata y en [Asimov, 2004] se continúa con esta idea.

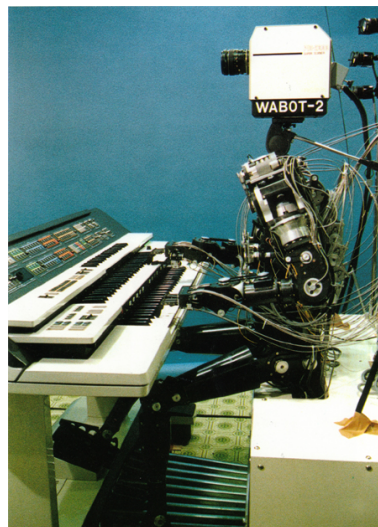
Uno de los avances más significativos fue la aplicación del punto de soporte (*support point*) [Borelli, 2012] al control de mecanismos con piernas. En 1969 el Prof. Vukobratovic fue el primero en aplicarlo a este tipo de mecanismos y lo renombró como *Zero-Moment-Point* (Punto de cero momento (ZMP)) [Vukobratovic y Juricic, 1969, Vukobratović y Borovac, 2004]; este punto es un indicador de vital importancia, ya que en él se concentran las fuerzas que actúan sobre todo el mecanismo y, por lo tanto, las reacciones dinámicas que presenta el suelo sobre el pie de apoyo.

En 1979 dentro de la Universidad de Waseda en Japón se creó el primer robot

antropomórfico a escala real en el mundo con el nombre de Waseda Robot (WABOT) [Humanoid Robotics Institute, 2008]. En 1987 se presenta WABOT-2 el cual incorpora manos antropomorfas y brazos de 7 DOF, capaz de tocar el teclado de forma autónoma [Sugano y Kato, 1987].



(a) WABOT (tomado de [Humanoid Robotics Institute, 2008]).



(b) WABOT-2 (tomado de [Humanoid Robotics Institute, 2008]).

Figura 1.2. WABOT y WABOT-2.

En 1984 se presentan los robots bípedos con el primer control activo de balance llamados BIPER-1, 2, 3, 4 y 5 [Miura y Shimoyama, 1984], estos robots eran estáticamente inestables, pero con un sistema de control, eran capaces de realizar un caminado dinámicamente estable.

En 1980 se fundó el *Leg Laboratory* en *Carnegie Mellon University* y posteriormente se trasladó al *Massachusetts Institute of Technology (MIT)* en 1986; dentro de éste fueron creados el robot *Spring Turkey* [Pratt *et al.*, 1997] y *Spring Flamingo* [Pratt y Pratt, 1998]; los robots contaban con cadera, rodillas actuadas y restricciones que evitaban su desplazamiento lateral y rotaciones sobre sí mismos.

En [Kajita y Tani, 1996] se propone el uso del modelo de péndulo invertido con una junta prismática para el control de un robot bípedo de 6 DOF. En [Huang *et al.*, 1999] se sugiere la generación de trayectorias definiendo diversos parámetros en el movimiento del pie mostrados en la figura 1.3, donde q_b representa el ángulo de despegue del pie, H_{a0} la altura máxima del pie, q_f el ángulo de aterrizaje y D_s la distancia del paso, a partir de éstos se usan *splines*

para generar una trayectoria con una transición suave entre las fases de caminado. Además, se propone un algoritmo para encontrar las trayectorias óptimas de las juntas y de la cadera que aseguren el mayor margen de estabilidad para el ZMP.

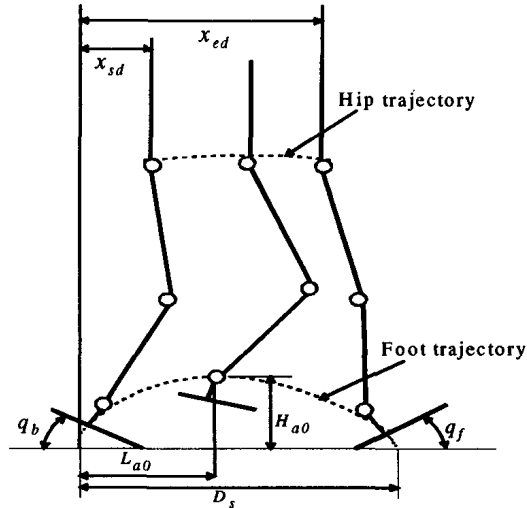


Figura 1.3. Parámetros para generar trayectorias de caminado (tomado de [Huang *et al.*, 1999]).

En [Kajita *et al.*, 2003] se hace uso del *3D Linear Inverted Pendulum Mode* y se calcula la generación de trayectorias de caminado basados en el uso del esquema de *Preview control* a la ubicación de ZMP. En este caso se propone una trayectoria del ZMP y por medio de *Preview control* se obtiene una trayectoria para el Centro de masa (COM) del robot humanoide.

1.3.1. Robots humanoides en la actualidad

El desarrollo de robots humanoides ha ido en aumento en la última década ya que la mayoría de las tecnologías involucradas en su creación han evolucionado a pasos agigantados. Existen diversas empresas, universidades y centros de investigación pioneros en la investigación de humanoides alrededor del mundo, dentro de las cuales destacan *Boston Dynamics*, *HONDA*, *Agency of Industrial Science and Technology (AIST)*, *Carnegie Mellon University*, *Waseda University*, etc. A continuación se presentan los desarrollos más destacados sobre robótica humanoide.

Boston Dynamics *Boston Dynamics* [DYNAMICS, 2017b] es una empresa especializada en el desarrollo de robots inspirados en animales, cuentan con diversos trabajos destacados

como los robots Atlas, Petman, Bigdog, WildCat, etc. En cuanto a robots humanoides los robots Petman y Atlas son considerados actualmente dentro de los robots con mayor movilidad, capaces de adaptarse a terrenos irregulares.

ATLAS El robot ATLAS es considerado como el robot humanoide con más movilidad en la actualidad, capaz de mover todo su cuerpo simultáneamente y mantener su equilibrio a pesar de tener una superficie de contacto pequeña con el suelo. ATLAS cuenta con una actuación electrohidráulica y distribuida mediante 30 **DOF**, éste fue presentado durante el concurso *DARPA Robotics* en 2013 y ha tenido posteriores actualizaciones con su versión más reciente en 2016. Actualmente ATLAS es capaz de caminar por más de 30 minutos sin la intervención humana y sin contar con un apoyo completo sobre la superficie [Wiedebach *et al.*, 2016], puede levantar 11 kg y cuenta con varios sistemas de visión incorporados. En la figura 1.4 se puede observar el robot Atlas con su configuración actual y en la figura 1.5 se puede contemplar su marcha sobre superficies irregulares.

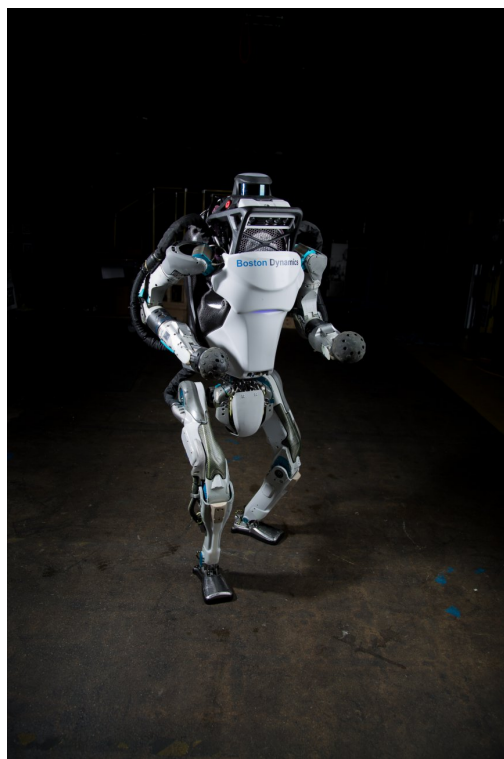


Figura 1.4. Robot Atlas (tomado de [DYNAMICS, 2017a]).

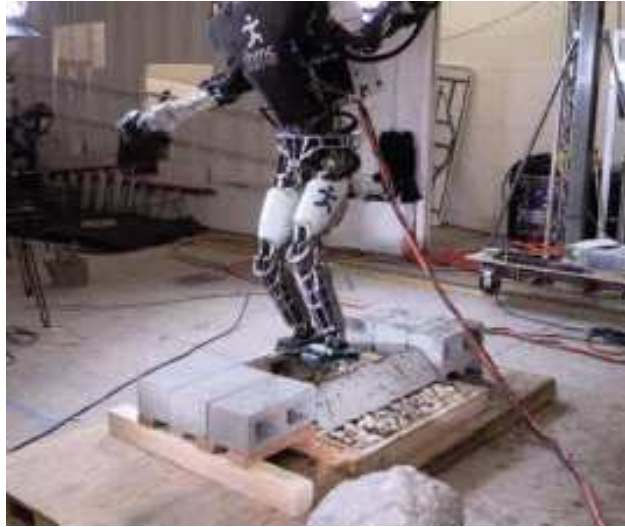


Figura 1.5. Robot Atlas caminando sobre superficie irregular (tomado de [Wiedebach *et al.*, 2016]).

PETMAN PETMAN es un robot antropomórfico con un peso de 80kg, una capacidad de carga extra de 23kg y una altura de 140 cm diseñado con la finalidad de probar ropa resistente a reactivos químicos, cuenta con actuación hidráulica y es capaz de reproducir la fuerza y velocidades de un ser humano mientras está vistiendo ropa, incluyendo zapatos [Nelson *et al.*, 2012]. El robot PETMAN ha demostrado ser dinámicamente estable durante la ejecución de un caminado estilo *heel-to-toe* obteniendo velocidades de 4.8 km/h. En la figura 1.6 se observa la estructura del robot PETMAN sujeta por arneses para evitar posibles caídas.



Figura 1.6. Robot Petman (tomado de [Velasco, 2011]).

HRP El *Humanoid Robotics Project* (HRP) [Inoue *et al.*, 2000] es un proyecto originalmente financiado por la AIST y *Ministry availability of International Trade and Industry* (MITI)

de Japón con la finalidad de crear una plataforma para la investigación de la Robótica humanoide. El proyecto consistía en la construcción de un prototipo físico incluyendo control remoto y su equivalente virtual. En la actualidad existen varios modelos derivados de este proyecto teniendo especial relevancia las versiones [HRP-2](#), [HRP-3](#) y [HRP-4](#).

HRP-02 El robot [HRP-2](#) fue desarrollado en 2004 como evolución de la primer versión de los modelos [HRP](#). El robot cuenta con 30 [DOF](#), tiene una altura de 1.539 m y una masa de 58 kg, puede caminar a una velocidad de 2.5 km/h, las dimensiones del robot fueron basadas en cuerpos femeninos japoneses [[Kaneko et al., 2004](#)]. El robot es capaz de evadir colisiones y caminar sobre terreno irregular [[Stasse et al., 2008](#)]. En la figura 1.7 se puede observar al robot [HRP-02](#).



Figura 1.7. Robot [HRP-02](#) (tomado de [[KAWADA INDUSTRIES, 2017](#)]).

HRP-03 El robot [HRP-3](#) mejora características del robot [HRP-2](#), cuenta con 40 [DOF](#), tiene una altura de 1.6 m y una masa de 68 kg; las principales diferencias con [HRP-02](#) es que este robot está diseñado para moverse fuera de entornos controlados [[Kaneko et al., 2008](#)] y el aumento en su movilidad. En la figura 1.8(a) se observa el robot [HRP-03](#).

HRP-04 El robot [HRP-04](#) es una mejora directa del robot [HRP-2](#) en cuestiones estéticas, ya que éste es un robot mucho más esbelto y con menor masa, logrando reducir los costes de producción y su consumo energético. el robot tiene una altura de 1.5 m y una masa de 39 kg, 19 kg más ligero que [HRP-2](#) además de contar con 34 [DOF](#) también superando a

su predecesor [Kaneko *et al.*, 2011]. En la figura 1.8(b) se observa el robot HRP-04.



(a) Robot HRP-03 (tomado de [Kaneko *et al.*, 2008]).

(b) Robot HRP-04 (tomado de [Kaneko *et al.*, 2011]).

Figura 1.8. Serie de robots HRP.

WABIAN La serie de robots WABIAN fueron desarrollados en la Universidad de WASEDA en Japón dentro de los laboratorios Kato/Takanishi [Laboratory, 2015]. Los robots tienen como finalidad ser una plataforma de simulación del movimiento humano. En la figura 1.9 se puede observar el modelo WABIAN 2RIII el cual cuenta con 41 DOF distribuidos de la siguiente manera: 6 DOF por pierna, 1 DOF por pie, 7 DOF por brazo, 2DOF en la cadera, 2DOF en el tronco, 3 DOF por mano y 3 DOF en el cuello. El robot WABIAN 2RIII tiene una altura de 1.5 m y una masa de 64 Kg.



Figura 1.9. Robot WABIAN 2/RIII (tomado de [Ogura *et al.*, 2006]).

HONDA HONDA es una empresa de tecnología japonesa, entre otras cosas pionera en el desarrollo de robots caminantes y posteriormente humanoide. El robot humanoide más conocido mundialmente es fabricado por esta empresa y tiene el nombre de *Advanced Step in Innovative Mobility* (ASIMO).

HONDA ha desarrollado robots bípedos desde 1986, teniendo un avance progresivo descrito en [Hirose y Ogawa, 2007]. Los desarrollos de HONDA comenzaron con los modelos E0, 1, 2, 3, 4 y 5, estos están compuestos solo por piernas y se pueden observar en el periodo de tiempo del año 1986 a 1991 de la figura 1.10. La serie de robots E se centra en la estabilización dinámica y estática del caminado bípedo. En los modelos P1, 2 y 3 se añaden brazos y torso obteniendo modelos antropomorfos como se observan en el lapso de tiempo del año 1993 a 2003 en la figura 1.10. La serie P implementan un control para la estabilización del caminado y añade habilidades para la comunicación e interacción con el entorno del robot.

En el año 2003 fue presentado el robot ASIMO, éste integra los desarrollos anteriores de HONDA pero haciendo una reducción de tamaño y peso. Además, se añaden características de comunicación e interacción con las personas. ASIMO cuenta con 57 DOF totales repartidos de la siguiente manera: 6 DOF por pierna, 7 DOF por brazo, 13 DOF por mano, 2 DOF en la cadera y 3 DOF en la cabeza. ASIMO puede caminar a una velocidad de 2.7 km/hr y correr a 7 km/hr [HONDA, 2017]. En la figura 1.11 se puede ver a los robots P2, P3 y ASIMO, donde se percibe una diferencia significativa de tamaño entre ASIMO y los robots de la serie P.

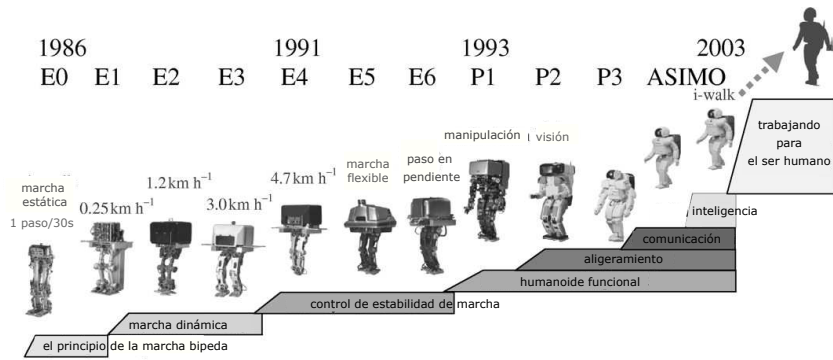


Figura 1.10. "Se ha tomado un enfoque paso a paso, comenzando con una caminata básica y estática, pasando por una caminata inteligente ajustable automáticamente, finalizando con la capacidad de trabajar para un ser humano" (tomado de [Hirose y Ogawa, 2007]).

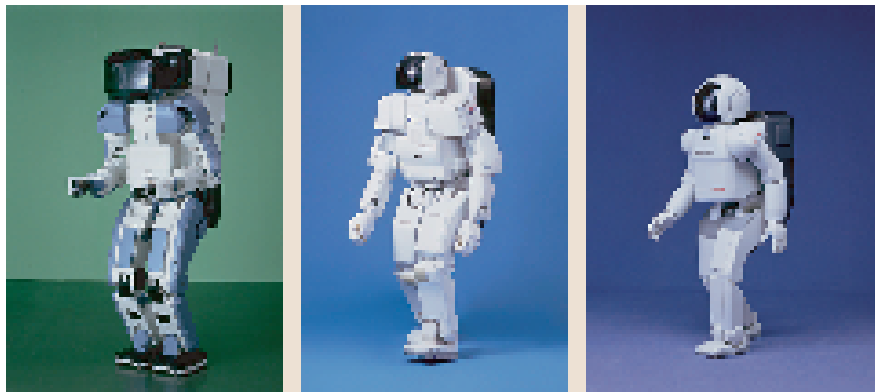


Figura 1.11. Robots HONDA P2, P3 y ASIMO (tomado de [Fitzpatrick *et al.*, 2016]).

KAIST El *Korean Agency of Industrial Science and Technology* (**KAIST**) es el equivalente coreano del instituto **AIST** en el cual se desarrollan una gran cantidad de proyectos dedicados a la robótica, entre ellos el diseño y construcción de robots humanoides y bípedos especialmente enfocados al uso en rescate de personas.

HUBO La serie de humanoides *KAIST Humanoid Robot* (**KHR**) conocido popularmente como HUBO, es desarrollada por el **KAIST** contando con varios modelos destacando el **KHR-03** desarrollado en 2005 [Park *et al.*, 2005]. El robot cuenta con un total de 41 **DOF** distribuidas entre sus extremidades, con una altura de 125 cm, masa de 55 kg y puede caminar con una velocidad de hasta 1.25 Km/h.

En el 2014 se presentó una actualización de **KHR-03** el cual concursó en el *DARPA robotics*

challenge obteniendo la victoria. En la figura 1.12 se observa el robot HUBO caminando sobre superficies irregulares dentro del concurso *DARPA robotics challenge*.



Figura 1.12. Robot HUBO dentro de una prueba de DARPA robotics challenge (tomado de [Wang *et al.*, 2014]).

1.3.2. Robots "do it yourself"

Los robots *Do It Yourself* (DIY) (ó hágalo usted mismo) generalmente se obtienen como kits de robótica donde es necesario ensamblar las partes para obtener una arquitectura deseada. Las principales ventajas de estos kits, es su bajo costo en comparación a los modelos de tamaño real y la capacidad que tienen para reconfigurarse y tomar diversas formas como humanoides, hexápodos, cuadrúpedos, etc. Además, son fáciles de modificar pudiendo crear piezas externas al kit e integrarlas a éste. Existe una gran variedad de kits con estas características, en este trabajo se utilizan los kits Hovis Lite y Bioloid Premium descritos a continuación.

1.3.2.1. HOVIS

El kit de robótica HOVIS Lite permite crear un robot humanoide de 16 DOF mostrado en la figura 1.13. Donde los DOF del robot se distribuyen como sigue: 5 DOF por pierna y 3 DOF por brazo. Este robot tiene como propósito ser utilizado de forma educacional, por lo que cuenta con un entorno de programación didáctico y fácil de usar. El kit cuenta con los motores DRS-0101 de Dongbu robot y un micro controlador ATmega 128; tiene la capacidad de añadir 4 motores más y llegar a los 20 DOF, además de poder agregar una gran variedad de sensores [robot, 2017]. En la figura 1.13 se puede observar el kit armado con una

configuración de 16 DOF.

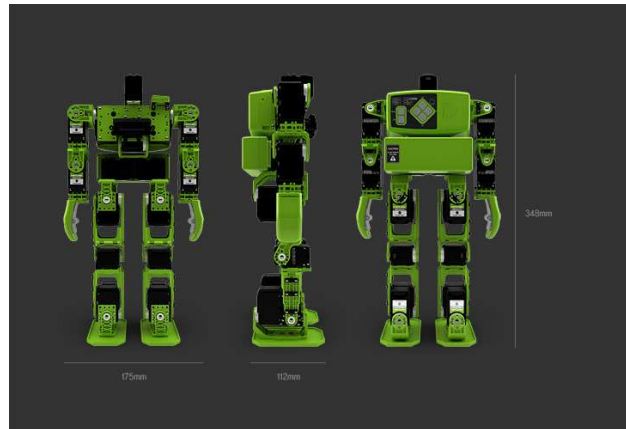


Figura 1.13. Robot HOVIS Lite (tomado de [Robotics, 2017b]).

1.3.2.2. Bioloid Premium

El robot Bioloid Premium es fabricado por la empresa Robotis, cuenta con 18 DOF dándole una movilidad equiparable a un modelo de mayores prestaciones. Los motores que se incluye son los AX-12A teniendo retroalimentación de posición, velocidad, par, temperatura de operación, etc. El robot cuenta con el controlador STM32F103RE el cual es un ARM Cortex-M3 con 512 Kbytes Flash, y una velocidad del CPU de 72 MHz [STMicroelectronics, 2017]. El robot incluye un conjunto de software que permite una programación fácil, así como una utilidad para capturar movimientos y poses que posteriormente se pueden usar como rutinas predefinidas.

Otra de las alternativas de programación es el uso del lenguaje C; éste permite un total acceso a los registros de los motores, así como cambiar la frecuencia de actualización del estado del robot, lo cual es preferible en los desarrollos presentados en este trabajo. En la figura 1.14 se puede observar el robot Bioloid Premium.



Figura 1.14. Robot Bioid Premium (tomado de [Shop, 2017]).

1.4. Estabilidad de un robot humanoide

La estabilidad mecánica de un humanoide permitirá realizar movimientos sin que el robot pierda el equilibrio y caiga. Para lograr la estabilidad mecánica en un humanoide, se deben considerar dos conceptos, la posición del **COM** en un estado estático es decir cuando el robot no se mueve y la posición del **ZMP** para un estado dinámico es decir cuando el robot está en movimiento. Uno de los aspectos a considerar es que cuando se presenta movimientos a bajas velocidades donde la inercia es despreciable, el criterio del **COM** puede ser usado para analizar la estabilidad de un robot. En la figura 1.15 se ilustra la ubicación del **COM** y el **ZMP** durante los casos de estabilidad estática y dinámica.

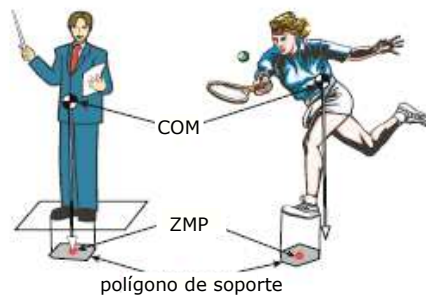


Figura 1.15. Comparación entre el **COM** y **ZMP** (tomado de [Kajita *et al.*, 2014]).

1.4.1. Estabilidad estática

La estabilidad estática de un humanoide se logra posicionando en todo momento el COM dentro del polígono de soporte. El polígono de soporte se define como el conjunto convexo más pequeño que incluye todos los puntos de contacto existentes entre el suelo [Kajita *et al.*, 2014].

1.4.1.1. Centro de masa (COM)

Consideremos un sistema con n partículas cada una con masa m_j y posición \mathbf{x}_j con $j = 1 \dots n$. El centro de masa físicamente representa la posición promedio (\mathbf{x}) ponderada del sistema de partículas y se calcula como:

$$\mathbf{x} = \frac{\sum_{j=1}^n m_j \mathbf{x}_j}{M} \quad (1.1)$$

con $M = \sum_{j=1}^n m_j$ es la masa total del sistema. En el caso de que el sistema de partículas se encuentre sobre un campo gravitacional uniforme, el COM será coincidente con el centro de gravedad [Meirovitch, 2010]. Dado que un cuerpo rígido puede ser considerado como un sistema de partículas, la definición anterior aplica tanto para cuerpos rígidos como para sistemas de cuerpos rígidos.

1.4.2. Estabilidad dinámica

La estabilidad dinámica se obtiene manteniendo el ZMP dentro del polígono de apoyo del robot en todo momento.

1.4.2.1. Punto de cero momento (ZMP)

El punto de cero momento ZMP fue definido en 1969 [Vukobratovic y Juricic, 1969], su importancia radica en que antes de él no existía un criterio que permitiera determinar la estabilidad dinámica de un humanoide, por lo que se dice que, a partir de su formulación, comenzó el desarrollo del control de humanoides.

Para casos en 2D donde solo se analizan movimientos sobre el plano sagital, el ZMP es el punto sobre el polígono de apoyo donde el momento generado por las fuerzas de reacción del suelo es igual a cero. Para casos 3D donde existen movimientos en el espacio, el ZMP es

el punto sobre el polígono de apoyo donde la componente horizontal del momento generado por las fuerzas de reacción del suelo es igual a cero [Kajita *et al.*, 2014].

Las ecuaciones para el cálculo de **ZMP** se presentan a continuación

$$p_x = \frac{Mgx + p_z \dot{P}_x - \dot{L}_y}{Mg + \dot{P}_z} \quad (1.2)$$

$$p_y = \frac{Mgy + p_z \dot{P}_y + \dot{L}_x}{Mg + \dot{P}_z} \quad (1.3)$$

donde p_x , p_y y p_z son los componentes \hat{i} , \hat{j} y \hat{k} del **ZMP** con $p_z = 0$ si el robot se está moviendo sobre una superficie plana, x y y son los componentes \hat{i} y \hat{j} del **COM**, $\dot{\mathbf{P}}$ es la derivada en el tiempo del momentum y \dot{P}_x , \dot{P}_y y \dot{P}_z sus componentes \hat{i} , \hat{j} y \hat{k} , $\dot{\mathbf{L}}$ es la derivada en el tiempo del momentum angular y \dot{L}_x y \dot{L}_y sus componentes \hat{i} y \hat{j} , M la masa total del cuerpo y g la gravedad.

1.4.2.2. Marcha

El ciclo de marcha en bípedos es un movimiento periódico y está compuesto por dos fases, soporte simple y doble soporte. La fase de soporte simple comienza cuando uno de los pies se desprende del suelo, por lo que un solo pie está en contacto con la superficie, mientras que el otro pie se balancea para generar un desplazamiento. La fase de doble soporte es la transición entre dos fases de soporte simple, comienza cuando el pie que se está balanceando tiene contacto con la superficie y finaliza cuando el pie contrario se despega de ella. En esta fase los dos pies tienen contacto con la superficie y por lo general tiene una duración mucho menor que la fase de soporte simple. El ciclo de marcha inicia cuando uno de los pies hace contacto con el suelo y finaliza cuando el pie que inicio el ciclo vuelve a hacer contacto con el suelo, como se observa en la figura 1.16.

Una de las actividades más complejas de ejecutar para un robot humanoide es la marcha, ya que requiere de la ejecución de diversas tareas simultáneamente. Para un desplazamiento quasi-estático, es decir un movimiento a baja velocidad, es al menos necesario el control de posición del **COM** y el movimiento del pie de balanceo. Para un desplazamiento dinámico, es decir una marcha común en un ser humano, es necesario el control de posición del **ZMP** y el movimiento del pie.

En la figura 1.17 se puede observar la fuerza de reacción del suelo sobre el pie de apoyo

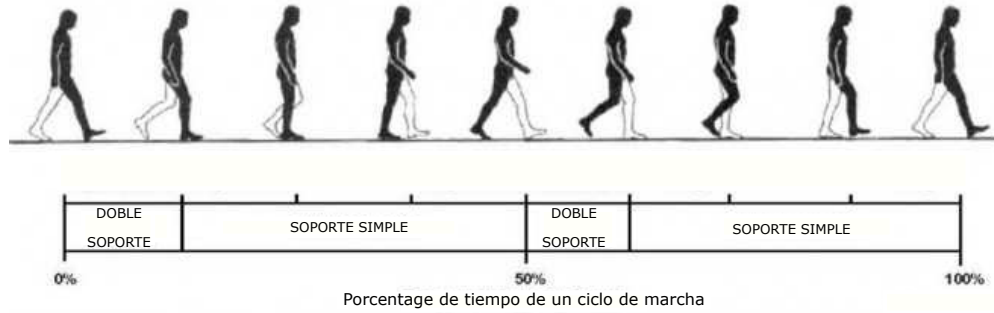


Figura 1.16. Ciclo de marcha.

del caminante.

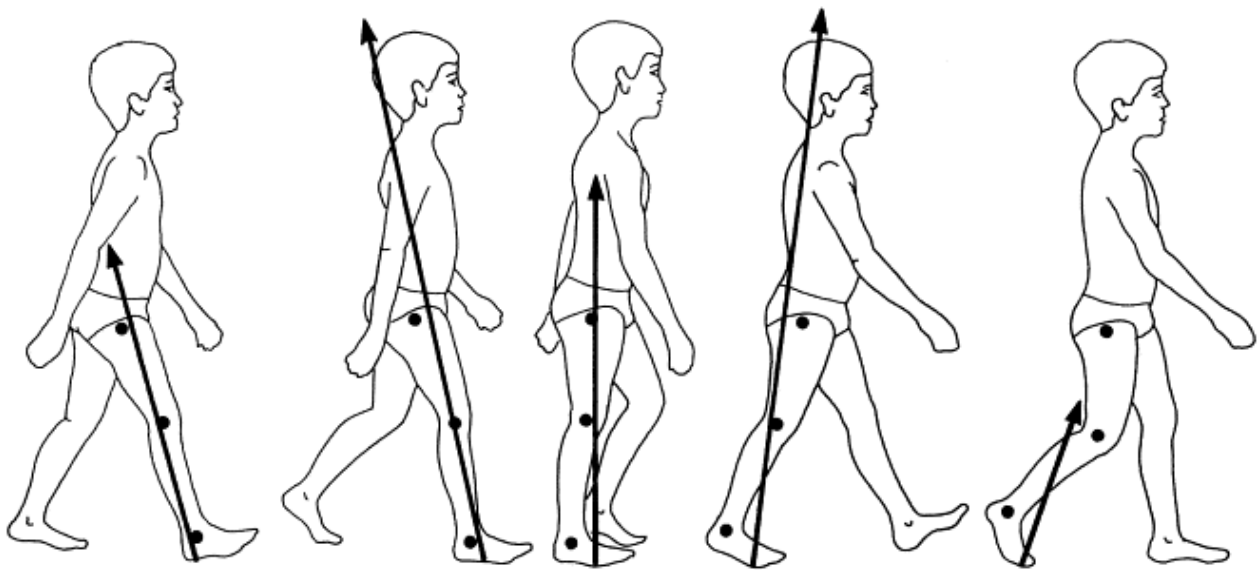


Figura 1.17. Vector de fuerza de reacción del suelo durante el caminado (tomado de [Vaughan, 2003]).

1.5. Elementos básicos del modelado de robots

Debido a la gran cantidad de **DOF** y múltiples extremidades presentes en un humanoide su modelado presenta diversas peculiaridades. Sin embargo, los métodos utilizados en el modelado de robots seriales son también aplicables en los humanoides.

Uno de los problemas que se presentan en el modelado de un humanoide es la colocación del marco de referencia, el cual además debe ser definido ya que dependiendo de su ubicación el modelado puede cambiar. Además, se debe considerar que el robot se mueve en el espacio, por lo que es muy fácil perder la referencia de éste en algún momento durante

su desplazamiento.

Una de las alternativas más usadas dentro de la literatura es tratar a un robot humanoide como un robot redundante y obtener su cinemática inversa mediante métodos iterativos.

A continuación, se presenta una breve síntesis de los conceptos conocidos y las fórmulas básicas del modelado de robots. Para mayores detalles sobre estos conceptos ver [Spong *et al.*, 2006, Craig, 2005, Sciavicco y Siciliano, 2012].

Transformación entre marcos La transformación entre marcos de referencia es de suma utilidad en el ámbito de la robótica, ya que asignando marcos de referencia rígidamente acoplados a cuerpos rígidos y encontrando la relación entre ellos, es posible determinar la posición y orientación de un cuerpo respecto a otro en el espacio tridimensional. Existen dos tipos de transformaciones principales entre marcos de referencia conocidos, traslación que determina la posición y rotación que establece la orientación. La combinación entre las dos transformaciones permitirá definir un cuerpo rígido completamente. A continuación, se describen a detalle las transformaciones de rotación y traslación.

Rotación Para representar la rotación de un cuerpo rígido respecto a otro, es necesario adherir fijamente un marco de referencia a cada uno y describir la relación geométrica entre ellos. Las matrices de rotación \mathbf{R} permiten representar la rotación entre dos cuerpos en el espacio por las proyecciones de sus ejes coordenados. Asumiendo que se desea expresar la rotación de un marco (O_1, x_1, y_1, z_1) sobre un marco (O_0, x_0, y_0, z_0) La matriz de rotación \mathbf{R} que los relaciona está conformada por:

$$\mathbf{R} = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} \quad (1.4)$$

Matrices de rotación básicas Las rotaciones compuestas son difíciles de representar, por lo que generalmente se descomponen en rotaciones básicas sobre los ejes del marco, mostradas en las ecuaciones (1.5), (1.6) y (1.7).

$$\mathbf{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\text{sen} \theta \\ 0 & \text{sen} \theta & \cos \theta \end{bmatrix} \quad (1.5)$$

$$\mathbf{R}_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \text{sen}\theta \\ 0 & 1 & 0 \\ -\text{sen}\theta & 0 & \cos \theta \end{bmatrix} \tag{1.6}$$

$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos \theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.7}$$

donde $\mathbf{R}_{x,\theta}$ representa una rotación con un ángulo θ sobre el eje x , $\mathbf{R}_{y,\theta}$ una rotación θ sobre el eje y y $\mathbf{R}_{z,\theta}$ una rotación θ sobre el eje z .

Rotaciones compuestas De manera general para representar totalmente la orientación existente entre dos cuerpos en el espacio se requiere de al menos una composición de tres rotaciones básicas no consecutivas sobre el mismo eje. Una de las principales rotaciones compuestas utilizadas en el ámbito de la robótica es Euler zyz , descrita a continuación.

Euler zyz Las rotaciones de Euler zyz se basa en una composición de rotaciones sobre un marco actual. Para obtener esta representación suponiendo que se tiene un marco fijo ($O_0x_0y_0z_0$) y se desea expresar la orientación del marco (O_1, x_1, y_1, z_1) primero se realiza una rotación de ϕ sobre el eje z , posteriormente una rotación θ sobre el eje actual y , y finalmente una rotación de ψ sobre el eje actual z . En la figura 1.18 se observan las transformaciones realizadas, donde (O_a, x_a, y_a, z_a) y (O_b, x_b, y_b, z_b) son los marcos actuales mencionados.

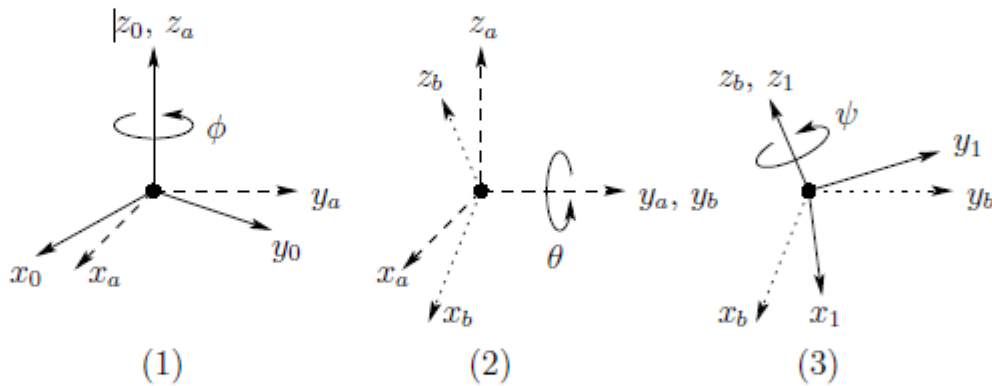


Figura 1.18. Representación de rotaciones de Euler zyz (tomado de [Spong *et al.*, 2006]).

En la ecuación (1.8) se observa la composición de matrices de rotación y (1.9) se muestra la matriz resultante

$$\mathbf{R}_{ZYZ} = \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.8)$$

$$\mathbf{R}_{ZYZ} = \begin{bmatrix} c\phi c\theta c\psi - s\phi s\psi & -c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta \\ s\phi c\theta c\psi + c\phi s\psi & -s\phi c\theta s\psi + c\phi c\psi & s\phi s\theta \\ -s\theta c\psi & s\theta s\psi & c\theta \end{bmatrix} \quad (1.9)$$

donde $s\phi$, $s\theta$ y $s\psi$ representan el $\text{sen}\phi$, $\text{sen}\theta$ y $\text{sen}\psi$, respectivamente y $c\phi$, $c\theta$ y $c\psi$ representan $\text{cos}\phi$, $\text{cos}\theta$ y $\text{cos}\psi$, respectivamente.

Traslación de cuerpos rígidos En robótica muchas veces es necesario expresar la posición de un cuerpo respecto a un marco de referencia dado. Asumiendo que se tiene un marco de referencia (O_0, x_0, y_0, z_0) y se desea expresar la relación traslacional que tiene con el marco (O_1, x_1, y_1, z_1) , ésta estará definida por un vector \mathbf{t}_0^1 de 3×1 el cual contiene las coordenadas del origen del marco (O_1, x_1, y_1, z_1) referenciado al marco coordenado (O_0, x_0, y_0, z_0) [Khatib, 2012]. En la figura 1.19 se observa la relación de traslación entre los marcos (O_0, x_0, y_0, z_0) y (O_1, x_1, y_1, z_1) .

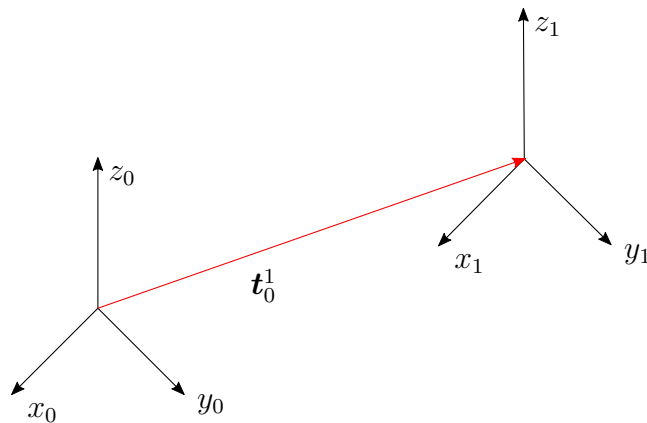


Figura 1.19. Traslación entre marcos de referencia.

Transformación de coordenadas Si se tiene un punto p y \mathbf{p}^1 es el vector de posición que lo asocia al marco (O_1, x_1, y_1, z_1) , y se busca expresar p en el marco (O_0, x_0, y_0, z_0) , es

necesario realizar transformaciones tanto de rotación como de traslación. La transformación necesaria está dada por

$$\mathbf{p}^0 = \mathbf{R}_0^1 \mathbf{p}^1 + \mathbf{t}_0^1 \quad (1.10)$$

donde \mathbf{p}^0 es el vector de posición que asocia p con marco (O_0, x_0, y_0, z_0) , \mathbf{R}_0^1 y \mathbf{t}_0^1 , es la matriz de rotación y el vector de traslación entre el marco (O_0, x_0, y_0, z_0) y el marco (O_1, x_1, y_1, z_1) respectivamente.

Matrices de transformación homogénea Las matrices de transformación homogénea permiten representar traslaciones y rotaciones conjuntas entre diversos marcos de referencia por medio de una sola matriz \mathbf{H} . Una matriz de transformación homogénea está compuesta por una matriz de rotación \mathbf{R} de tamaño 3×3 y un vector de traslación \mathbf{t} de 3×1 como se muestra en la ecuación (1.11).

$$\mathbf{H} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \quad (1.11)$$

Esta matriz permite expresar de forma matricial la ecuación (1.10) y hacer transformación de coordenadas por medio de multiplicaciones simples de matrices como se observa a continuación

$$\mathbf{P}^0 = \begin{pmatrix} \mathbf{R}_0^1 & \mathbf{t}_0^1 \\ 0 & 1 \end{pmatrix} \mathbf{P}^1 \quad (1.12)$$

$$\mathbf{P}^0 = \mathbf{H}_0^1 \mathbf{P}^1 \quad (1.13)$$

donde \mathbf{H}_0^1 es la matriz de transformación homogénea que asocia el marco (O_0, x_0, y_0, z_0) y el marco (O_1, x_1, y_1, z_1) , \mathbf{P}^0 y \mathbf{P}^1 son representaciones homogéneas de \mathbf{p}^0 y \mathbf{p}^1 y tienen la siguiente forma

$$\mathbf{P}^0 = \begin{bmatrix} \mathbf{p}^0 \\ 1 \end{bmatrix} \quad (1.14)$$

$$\mathbf{P}^1 = \begin{bmatrix} \mathbf{p}^1 \\ 1 \end{bmatrix} \quad (1.15)$$

1.5.1. Cinemática directa

El problema de Cinemática directa (CmD) de un robot serial consiste en determinar la posición y orientación de alguna parte del robot, generalmente la punta o efector, a partir de conocer una configuración articular \mathbf{q} dada. El vector \mathbf{q} contiene los valores de las juntas del robot, donde generalmente para juntas de revoluta se expresan en radianes y para juntas prismáticas en mm.

En robótica humanoide la CmD de un robot es obtenida multiplicando matrices de transformación homogénea o utilizando el método de Denavit-Hartenberg.

Convención Denavit-Hartenberg El método de Denavit-Hatemberg es un algoritmo que sirve para obtener el modelo cinemático directo de un robot serial. En este método, se logra representar la rotación y traslación entre dos marcos de referencia por medio de una representación mínima, es decir solo toma en cuenta 4 parámetros para crear una matriz de transformación homogénea que representa la transformación entre el marco asociado al punto de interés en el robot y un marco de referencia (por ejemplo, un marco fijo o inercial). Los parámetros Denavit-Hartenberg (D-H) se obtienen a partir de una rotación θ sobre el eje z , una traslación d sobre el eje z , una traslación a sobre el eje x y una rotación α sobre el eje x .

Asumiendo que se tienen un robot con n articulaciones, a éste se le asignarán $n+1$ marcos de referencia colocados en la base y en cada una de sus juntas, esto generará n matrices de transformación homogénea entre sus marcos. Utilizando los parámetros θ_i , d_i , a_i y α_i , donde i toma valores desde 1 a n se puede definir la transformación homogénea \mathbf{H}_{i-1}^i mostrada en la ecuación (1.17) que relaciona el marco $(O_{i-1}, x_{i-1}, y_{i-1}, z_{i-1})$ con (O_i, x_i, y_i, z_i) .

$$\mathbf{H}_{i-1}^i = \begin{pmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.16)$$

$$\mathbf{H}_{i-1}^i = \begin{pmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.17)$$

donde $c\theta_i$ es $\cos\theta_i$, $s\theta_i$ es $\sin\theta_i$, $c\alpha_i$ es $\cos\alpha_i$ y $s\alpha_i$ es $\sin\alpha_i$. La multiplicación consecutiva

de las matrices homogéneas obtenidas a partir de los parámetros D-H dan como resultado la cinemática directa del robot en cuestión. El procedimiento para enumerar los marcos de referencia y sus restricciones se puede consultar a detalle en [Spong *et al.*, 2006].

Jacobiano El jacobiano (\mathbf{J}) de un robot es una matriz de tamaño $6 \times n$ (donde n es el número de DOF) que permite relacionar las velocidades lineales ($\dot{\mathbf{x}}$) y angulares ($\boldsymbol{\omega}$) de un marco de referencia rígidamente adherido al efector final con las velocidades articulares $\dot{\mathbf{q}}$. Esta relación está dada por:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}\dot{\mathbf{q}} \quad (1.18)$$

con

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_w \end{bmatrix} \quad (1.19)$$

donde \mathbf{J}_v es el jacobiano de velocidades lineales y \mathbf{J}_w es el jacobiano de velocidades angulares.

El jacobiano tiene múltiples aplicaciones de gran importancia en la robótica como, su uso en el cálculo de la energía cinética y cinemática instantánea, la determinación de configuraciones singulares del robot, la transformación de fuerzas o torques del efector final a las juntas del robot, etc.

Existen diversas formas de obtener el jacobiano, siendo la más común, derivar parcialmente respecto a las variables articulares las ecuaciones de posición y orientación obtenidas a partir de la cinemática directa del robot. En este trabajo el jacobiano es calculado a partir de relaciones geométricas tomadas de [Spong *et al.*, 2006].

Para juntas de revoluta el cálculo de la columna i del jacobiano de velocidad lineal denotada como \mathbf{J}_{v_i} , se obtiene por medio de la ecuación (1.20) y la columna i del jacobiano de velocidad angular denotada como \mathbf{J}_{w_i} , por la ecuación (1.21).

$$\mathbf{J}_{v_i} = \mathbf{z}_{i-1} \times (\mathbf{O}_n - \mathbf{O}_{i-1}) \quad (1.20)$$

$$\mathbf{J}_{w_i} = \mathbf{z}_{i-1} \quad (1.21)$$

Donde \mathbf{z}_{i-1} representa el eje de rotación instantáneo, \mathbf{O}_n el origen asociado al marco de referencia asociado al efector final y \mathbf{O}_{i-1} el origen asociado a el eje de rotación instantáneo.

Para propósitos del presente trabajo es necesario obtener el jacobiano (\mathbf{J}_i) asociado al COM del eslabón i . Sustituyendo en la ecuación (1.20) el origen \mathbf{O}_n por un origen asociado al centro de masa \mathbf{O}_{cmi} del eslabón i se obtiene la columna l de \mathbf{J}_i , como se observa a continuación:

$$\mathbf{J}_l = \begin{cases} l < i & \mathbf{z}_{l-1} \times (\mathbf{O}_{cmi} - \mathbf{O}_l) \\ l \geq i & 0 \end{cases} \quad (1.22)$$

con $l = 1 \dots n$.

1.5.2. Cinemática inversa

El problema de cinemática inversa consiste en encontrar una configuración articular \mathbf{q} que satisfaga una posición y orientación dada para el efector final del robot. Existen diversos métodos propuestos para la obtención de dicha configuración, siendo los más utilizados la técnica de desacople cinemático y el método iterativo de Newton, descritos a continuación.

Desacople cinemático El desacople cinemático es un método principalmente utilizado para robots de 6 DOF (robots con muñeca esférica) donde es posible desacoplar la posición y la orientación del robot, es decir, se utilizan 3 DOF para calcular la posición y los 3 restantes para la determinación de la orientación del robot.

El proceso consiste en:

1. Proponer una posición y orientación para el efector final del robot (matriz de transformación Homogénea \mathbf{H}_0^6).
2. Determinar la posición del centro de la muñeca esférica (\mathbf{x}_s) del robot a partir de

$$\mathbf{x}_s = \mathbf{t}_0^6 - \mathbf{R}_0^6 * \begin{pmatrix} 0 \\ 0 \\ l_s \end{pmatrix} \quad (1.23)$$

donde \mathbf{R}_0^6 es la matriz de rotación que representa la orientación del efector final del robot respecto a la base del robot, \mathbf{t}_0^6 es el vector que contiene la posición del efector final respecto a la base del robot y l_s es el parámetro D-H d_6 .

3. Determinar las primeras tres variables articulares a partir de la geometría conocida del robot (las ecuaciones cambian dependiendo del modelo).
4. Calcular las matrices de transformación homogénea y obtener la matriz \mathbf{R}_0^3 .
5. Obtener la matriz \mathbf{R}_3^6 mediante $\mathbf{R}_0^{3T} * \mathbf{R}_0^6$.
6. Despejar las tres últimas variables asumiendo una configuración conocida de \mathbf{R}_3^6 (Euler zyz).

Una de las limitantes de este método es que solo está definido para robots de seis DOF con muñeca esférica, por lo que en el caso de un robot humanoide será necesario segmentar el robot por extremidades y aplicarlo secuencialmente.

Método iterativo El método iterativo para la cinemática inversa de un robot consiste en aplicar pequeños incrementos a las variables articulares para reducir el error existente entre la posición y orientación actual del efector y una deseada. los incrementos se calculan a partir de la ecuación

$$\Delta \mathbf{q} = \mathbf{J}^\dagger \Delta \mathbf{x} \quad (1.24)$$

donde $\Delta \mathbf{x} = \mathbf{x}^* - \mathbf{x}$, \mathbf{x}^* es la posición deseada, \mathbf{x} es la posición actual del efector, \dagger denota la pseudo inversa de Moore-Penrose, \mathbf{J}^\dagger es la pseudoinversa del jacobiano del robot y $\Delta \mathbf{q}$ es el incremento aplicado a las variables articulares. Este proceso se repite hasta que la magnitud de $\Delta \mathbf{x}$ sea lo suficientemente pequeña.

1.5.3. Robots redundantes

El concepto de redundancia en robótica tiene diversas definiciones y usos, donde muchas veces se considera como un robot redundante aquel que tenga más de seis grados de movilidad, sin embargo, esta definición aplica solo a aquellos robots que tengan una sola extremidad. En [Conkur y Buckingham, 1997] se analizan varias definiciones de redundancia y se formalizan las siguientes:

Redundancia cinemática Si las dimensiones del espacio articular son mayores a las del espacio del efector final entonces el dispositivo es **cinemáticamente redundante**.

Redundancia en tarea Si el espacio de tarea es completamente contenido dentro y de menores dimensiones que el espacio del efector final, se dice que el manipulador es **redundante en la tarea**.

En el caso de un robot humanoide dada la diversidad de tareas que puede ejecutar y que las dimensiones del espacio de tarea pueden variar estos conceptos son difíciles de aplicar, ya que por lo general se cuenta con múltiples extremidades y en consecuencia se tienen múltiples efectores. Una definición más adecuada para la redundancia en un humanoide se da en [Samson *et al.*, 1991] donde se dice que:

"Desde un punto de vista general, cualquier sistema robótico en donde la forma para ejecutar una tarea no es única se puede considerar redundante".

La característica principal de un robot redundante es que no existe solución única para su cinemática inversa. El hecho de contar con más **DOF** de los necesarios para ejecutar una tarea implica que desde un punto de vista matemático se tenga un sistema sobredeterminado de ecuaciones con soluciones infinitas.

Cinemática inversa de un robot redundante Debido a las infinitas soluciones de un robot redundante para alguna tarea determinada, no es posible obtener una solución analítica a su cinemática inversa y se opta por métodos iterativos y de optimización para obtener las mejores soluciones posibles. Dada la redundancia del sistema normalmente se busca solucionar una tarea primaria (por ejemplo, el posicionamiento) y con los **DOF** restantes se busca atender tareas secundarias, como las colisiones, minimización de energía, etc.

Una de las soluciones más utilizadas para la cinemática inversa en velocidad de un robot redundante [Buss, 2004a, D'Souza *et al.*, 2001, Liegeois, 1977] está dada por la ecuación (1.25)

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} - (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{z}} \quad (1.25)$$

donde $\dot{\mathbf{q}}$ es el vector de velocidades articulares, $\dot{\mathbf{x}}$ es el vector con la tarea principal, \mathbf{J} es el jacobiano que representa a la tarea primaria, $(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})$ es una proyección del espacio nulo de \mathbf{J} y $\dot{\mathbf{z}}$ representa el vector de velocidades cartesianas asociada a una tarea secundaria.

1.5.4. Cinemática directa del centro de masa

El problema de Cinemática directa del centro de masa (**CmDCOM**) consiste en encontrar la posición 3D del **COM** dado un vector de configuración articular. A diferencia de la **CmD**, la **CmDCOM** sí toma en cuenta la distribución de masas del conjunto de eslabones que componen el mecanismo articulado. Ésta permite determinar cómo cambia la posición y velocidad del **COM** a partir de las velocidades y posiciones articulares de cada eslabón en un mecanismo. La manipulación del **COM** en robótica humanoide, toma importancia ya que, como se mencionó anteriormente, desde un punto de vista estático la posición del **COM** determinará la estabilidad mecánica del robot en cuestión.

El **COM** \mathbf{x} de un robot con n eslabones se puede obtener a partir de la ecuación (1.26)

$$\mathbf{x} = \frac{\sum_{i=1}^n m_i \mathbf{x}_i}{M} \quad (1.26)$$

donde \mathbf{x}_i es el centro de masa del eslabón i , m_i su masa y M es la masa total del robot.

El cambio de posición respecto al tiempo del **COM** \mathbf{x} se puede calcular derivando la ecuación (1.26)

$$\dot{\mathbf{x}} = \frac{\sum_{i=1}^n m_i \dot{\mathbf{x}}_i}{M} \quad (1.27)$$

donde $\dot{\mathbf{x}}_i$ se calcula a partir del jacobiano \mathbf{J}_i de tamaño $3 \times n$ asociada a \mathbf{x}_i y la derivada del vector de variables articulares $\dot{\mathbf{q}}$, como se muestra en la ecuación (1.28)

$$\dot{\mathbf{x}}_i = \mathbf{J}_i \dot{\mathbf{q}} \quad (1.28)$$

sustituyendo la ecuación (1.28) en (1.27) se tiene:

$$\dot{\mathbf{x}} = \frac{\sum_{i=1}^n m_i \mathbf{J}_i}{M} \dot{\mathbf{q}} \quad (1.29)$$

Agrupando los términos de (1.29) se obtiene la expresión para representar la cinemática directa de velocidad del **COM** \mathbf{x}

$$\dot{\mathbf{x}} = \mathbf{J}_T \dot{\mathbf{q}} \quad (1.30)$$

donde \mathbf{J}_T es un jacobiano de dimensiones $3 \times n$ que relaciona las velocidades articulares del robot con las velocidades cartesianas del COM.

Dado que el valor de n puede variar dependiendo del robot a analizar, los jacobianos \mathbf{J}_i y \mathbf{J}_T pueden ser cuadrados, en cuyo caso existirá una solución analítica para la cinemática inversa del COM, o rectangulares donde se presentan dos casos: más renglones que columnas (no existe una solución para la cinemática inversa del COM), más columnas que renglones (robot redundante) donde existirán infinitas soluciones para la cinemática inversa del COM.

1.5.5. Cinemática inversa del centro de masa

El problema de CmICOM consiste en obtener la configuración articular \mathbf{q} dada una posición deseada del COM. A partir de (1.30) la velocidad angular se puede definir como

$$\dot{\mathbf{q}} = \mathbf{J}_T^\dagger \dot{\mathbf{x}} \quad (1.31)$$

donde \mathbf{J}_T^\dagger representa la pseudo-inversa del jacobiano \mathbf{J}_T .

Si tomamos en cuenta las relaciones denotada por (1.32) y (1.33) para incrementos infinitesimales

$$\dot{\mathbf{q}} \approx \frac{\Delta \mathbf{q}}{\Delta t} \quad (1.32)$$

$$\dot{\mathbf{x}} \approx \frac{\Delta \mathbf{x}}{\Delta t} \quad (1.33)$$

sustituyendo (1.32) y (1.33) en (1.31) y despejando Δt se obtiene (1.34)

$$\Delta \mathbf{q} = \mathbf{J}_T^\dagger \Delta \mathbf{x} \quad (1.34)$$

la cual permite aplicar incrementos a las variables articulares para reducir el valor de $\Delta \mathbf{x}$ el cual está definido como:

$$\Delta \mathbf{x} = \mathbf{x}^* - \mathbf{x} \quad (1.35)$$

donde \mathbf{x}^* es la posición deseada del COM y \mathbf{x} su posición actual.

En las figuras 1.20 y 1.21 se muestran ejemplos de cadenas cinemáticas seriales y las configuraciones obtenidas para una posición deseada del COM. Los eslabones están representados por las líneas azules, las juntas por los círculos y los centros de masa de cada eslabón por asteriscos rojos. En la figura 1.20 se observa una cadena cinemática serial de 4 DOFs con masas iguales para cada eslabón y donde se posiciona su COM en $x=5$, $y=4$ y $z=5$. Para la figura 1.21 se tiene una cadena cinemática serial de 8 DOFs y su COM se posiciona en $x=3$, $y=8$ y $z=0$.

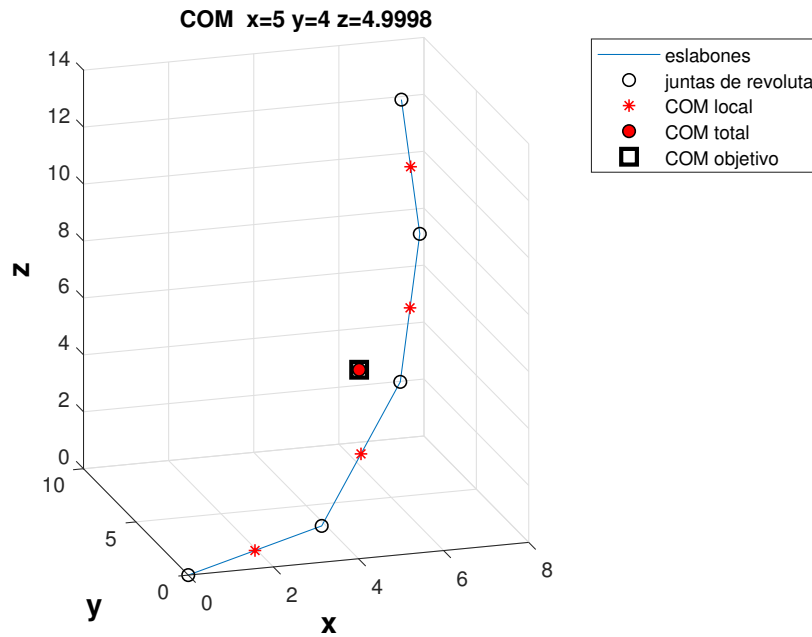


Figura 1.20. Configuración articular obtenida a partir de algoritmo de CmICOM para una cadena cinemática de 4 DOF.

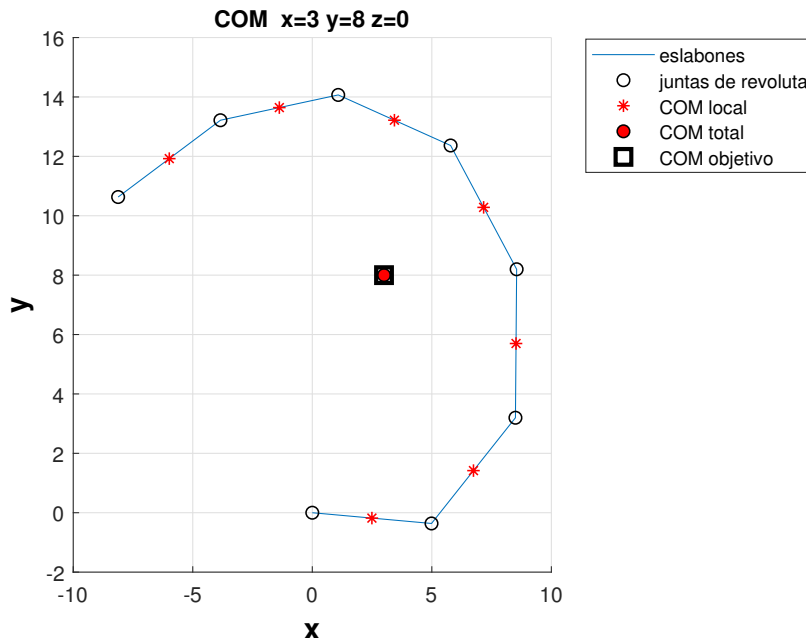


Figura 1.21. Configuración articular obtenida a partir de algoritmo de CmICOM para una cadena cinemática de 8 DOF.

1.6. Pseudo-inversa

Como se pudo observar anteriormente el empleo de la pseudo-inversa en los métodos iterativos para la obtención de la cinemática inversa de robots permite obtener soluciones aproximadas a diversas tareas asignadas al robot en cuestión. Sin embargo, la pseudo-inversa presenta varios inconvenientes durante su aplicación como su tendencia a llevar al robot a configuraciones singulares, su alto costo computacional y una convergencia lenta, por lo que es necesario aplicar diversas modificaciones para mejorar su rendimiento. La matriz pseudo-inversa o inversa de Moore-Penrose, es aquella que obtiene la mejor solución posible a la ecuación (1.36) en el sentido de los mínimos cuadrados [Buss, 2004b], es decir, que minimiza la relación $\| \mathbf{J} \Delta \mathbf{q} - \Delta \mathbf{x} \|$ o equivalentemente $\| \mathbf{J} \Delta \mathbf{q} - \Delta \mathbf{x} \|^2$. La pseudo-inversa tiene la particularidad de que está definida para cualquier matriz \mathbf{J} de $n \times m$, es decir matrices que no son cuadradas necesariamente o no tienen rango pleno.

$$\Delta \mathbf{x} = \mathbf{J} \Delta \mathbf{q} \quad (1.36)$$

Las ecuaciones (1.37) y (1.38) son dos de las representaciones posibles de la matriz pseudo-

inversa.

$$\mathbf{J}^\dagger = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \quad (1.37)$$

$$\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} \quad (1.38)$$

1.6.1. Pseudo-inversa ponderada

Una de las variaciones de la pseudo-inversa es la pseudo-inversa ponderada propuesta por [T. Komura, 1969]. Esta variación minimiza la expresión cuadrática mostrada en la ecuación (1.39), donde \mathbf{W} es una matriz de pesos diagonal positiva definida. Cada elemento de la diagonal de \mathbf{W} contiene un peso que pondera la contribución de la i -ésima componente de \mathbf{q} a la expresión cuadrática. Un peso mayor implica una mayor relevancia de la componente a la que se le aplica.

$$\mathbf{G} := \frac{1}{2} \Delta \mathbf{q}^T \mathbf{W} \Delta \mathbf{q} \quad (1.39)$$

La pseudo-inversa ponderada se define en la ecuación (1.40).

$$\mathbf{J}_w^\dagger = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1} \quad (1.40)$$

Sustituyendo la matriz \mathbf{J}^\dagger por \mathbf{J}_w^\dagger en la ecuación (1.24) se obtiene la ecuación (1.41)

$$\Delta \mathbf{q} = \mathbf{J}_w^\dagger \Delta \mathbf{x} \quad (1.41)$$

donde la matriz \mathbf{W} es usada para aumentar o disminuir los efectos de la variación de alguna de las variables. En [Komura *et al.*, 2003] para la obtención de cinemática inversa de robots por métodos iterativos se propone asignar un peso de .10 a las juntas con mayor movilidad y 10 a las juntas con mayor rigidez.

1.6.2. Pseudo-inversa amortiguada

El cálculo de la pseudo-inversa del jacobiano puede presentar problemas cuando el producto matricial $\mathbf{J} \mathbf{J}^T$ está cerca de la singularidad. Esto sucede frecuentemente en el ámbito de la robótica cuando el mecanismo se encuentra cerca de sus singularidades

geométricas. En [Wampller, 1986] se usa por primera vez la pseudo-inversa de Lavenberg-Marquardt para la cinemática inversa de mecanismos logrando sobrellevar el problema antes mencionado. La pseudo-inversa de Lavenberg-Marquardt propone una modificación a la ecuación (1.38), añadiendo una matriz identidad \mathbf{I} multiplicada por un factor escalar de amortiguamiento $\lambda > 0$ como se muestra en la ecuación (1.42).

$$\mathbf{J}^\dagger = (\mathbf{J}^T \mathbf{J} + \lambda^2 \mathbf{I})^{-1} \mathbf{J}^T \quad (1.42)$$

La adición de $\lambda^2 \mathbf{I}$ permite asegurar que la matriz resultante de $(\mathbf{J}^T \mathbf{J} + \lambda^2 \mathbf{I})$ no se convertirá en singular dentro del proceso iterativo. Existen diversos métodos para la selección del valor de λ , siendo el más simple asignar un valor constante, tomando en cuenta que un valor pequeño alcanzará más rápido la convergencia del error.

En [Wang *et al.*, 2013] durante un proceso de cinemática inversa de robots por métodos iterativos se propone reducir a la mitad el valor de λ si en la iteración actual se reduce el error y duplicar el valor si el error aumenta.

1.6.3. Pseudo-inversa ponderada-amortiguada

Al igual que la matriz pseudo-inversa, la pseudo-inversa ponderada tiene complicaciones en las cercanías de las singularidades geométricas del mecanismo. En este trabajo se propone utilizar el mismo procedimiento anteriormente mencionado introduciendo en la ecuación (1.40) una matriz identidad \mathbf{I} y un factor escalar de amortiguamiento $\lambda > 0$, como se muestra en la ecuación

$$\mathbf{J}_{w\lambda}^\dagger = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T + \lambda \mathbf{I})^{-1} \quad (1.43)$$

donde $\mathbf{J}_{w\lambda}^\dagger$ es la pseudo-inversa ponderada-amortiguada de \mathbf{J} . Esta definición combina las bondades de la pseudo-inversa ponderada y la pseudo-inversa amortiguada. En el cálculo de la Cinemática inversa (CmI) de robots la pseudo-inversa ponderada-amortiguada permitirá ejecutar el método iterativo sin obtener configuraciones singulares y además dar prioridad al movimiento de juntas específicas del robot.

1.7. Algoritmos límite de junta

Uno de los problemas persistentes en los métodos iterativos de cinemática inversa en robots, es la posibilidad de obtener configuraciones no realizables, principalmente por la existencia de límites físicos en el robot, es decir, por ejemplo aunque las juntas rotatorias teóricamente pueden moverse 360° , en la práctica su movimiento está limitado por el propio actuador o por colisión entre los dos eslabones que une. Existen diversas formas de atacar el problema, encontrándose entre los más comunes el método de descenso de gradiente, donde se propone una función de costos que permita alejar las juntas del robot de sus límites físicos. Esta función se proyecta a partir del espacio nulo del jacobiano asociado a la tarea primaria del robot como se muestra en la ecuación(1.44) la cual es una versión modificada de (1.25)

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} - (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \nabla G \quad (1.44)$$

donde ∇G es un vector de optimización que aleja las juntas de sus límites.

En [Liegeois, 1977] se propone la función de costos mostrada en la ecuación (1.45)

$$G(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \left(\frac{\theta_i - \theta_{ci}}{\theta_{ci} - \theta_{iM}} \right)^2 \quad (1.45)$$

$$\theta_{ci} = (\theta_{iM} + \theta_{im})/2 \quad (1.46)$$

donde n es el número total de juntas, θ_i es el valor actual de la junta i , θ_{im} es el valor mínimo de la junta i , θ_{iM} es su valor máximo y θ_{ci} es el valor medio de la junta i .

En [Jung et al., 1995] se propone una función similar a $G(\boldsymbol{\theta})$ nombrada como *Joint Range Availability* (JRA) y se puede observar en la ecuación (1.47)

$$JRA(\boldsymbol{\theta}) = \sum_{i=1}^n \left(\frac{\theta_i - \theta_{ci}}{\Delta\theta_i} \right)^2 \quad (1.47)$$

$$\Delta\theta_i = \theta_{iM} - \theta_{im} \quad (1.48)$$

donde $\Delta\theta_i$ es el máximo desplazamiento angular posible de la junta i .

En [Marey y Chaumette, 2010] se propone una función de costo G_s que permite activar o desactivar el efecto de la función dependiendo del valor actual de la junta i . Si el valor actual

de la junta i está dentro del rango operativo no existirá efecto de la función, en cambio si el valor actual de la junta i sobrepasa alguno de los límites, la función G_s aplicará una acción para regresarla a su rango operativo. La función G_s se presenta a continuación:

$$G_s(\boldsymbol{\theta}) = \frac{\beta}{2} \sum_{i=1}^n \frac{\Delta q_i}{\Delta \theta_i} \quad (1.49)$$

con

$$\Delta q_i = \begin{cases} \theta_i - \theta_{im} & \text{si } \theta_i < \theta_{im} \\ \theta_i - \theta_{iM} & \text{si } \theta_{iM} < \theta_i \\ 0 & \text{en otro caso} \end{cases} \quad (1.50)$$

donde Δq_i sirve para activar o desactivar la acción de la función G_s y β es un factor escalar que determina la amplitud del efecto de la función de optimización.

En [Zghal *et al.*, 1990] se propone la función de costos (1.51) la cual tiende a infinito tanto si el ángulo de la articulación se aproxima al límite inferior (valor mínimo permitido) como si se aproxima al límite superior (valor máximo permitido).

$$H(\theta) = \sum_{i=1}^n \frac{(\theta_{iM} - \theta_{im})^2}{4(\theta_{iM} - \theta_i)(\theta_i - \theta_{im})} \quad (1.51)$$

Como se mencionó anteriormente todas las metodologías hasta ahora mostradas para restringir el rango de movimiento de las juntas del robot, se basan en el descenso de gradiente y en proyectarse sobre el espacio nulo del jacobiano. Debido a que estas metodologías no cumplen con el rol de comportamiento primario del robot, no pueden asegurar que las juntas se mantengan siempre dentro del rango operativo. Además tienden a aumentar el tiempo de ejecución del algoritmo y generar cambios de configuración articular poco naturales.

1.7.1. Uso de ponderación para restringir límites en las juntas

Otra de las alternativas para mantener las juntas del robot dentro de sus límites operativos es el uso de la matriz de pesos \mathbf{W} . Una de las ventajas de esta alternativa es que el uso del espacio nulo del jacobiano puede ser utilizado para lograr objetivos adicionales. El proceso consiste en asignar pesos altos y restringir su movilidad a aquellas juntas tengan un movimiento en dirección a cualquiera de sus límites, mientras que se asignan pesos bajos

dándoles libertad de movimiento a las juntas que tengan un movimiento con dirección al rango medio de operación. Los pesos se determinan por la ecuación (1.52) propuesta en [Chan y Dubey, 1995]

$$w_i = \begin{cases} 1 + \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| & \text{si } \Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| \geq 0 \\ 1 & \text{si } \Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| < 0 \end{cases} \quad (1.52)$$

$\frac{\partial H(\theta)}{\partial \theta_i}$ y $\Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right|$ se definen como:

$$\frac{\partial H(\theta)}{\partial \theta_i} = \frac{(\theta_{iM} - \theta_{im})^2 (2\theta_i - \theta_{iM} - \theta_{im})}{4(\theta_{iM} - \theta_i)^2 (\theta_i - \theta_{im})^2} \quad (1.53)$$

$$\Delta \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| = \left| \frac{\partial H(\theta)}{\partial \theta_i} \right| - \left| \frac{\partial H(\theta)}{\partial \theta_i} \right|_{-1} \quad (1.54)$$

donde $\left| \frac{\partial H(\theta)}{\partial \theta_i} \right|$ es el valor absoluto de $\frac{\partial H(\theta)}{\partial \theta_i}$ en la iteración actual y $\left| \frac{\partial H(\theta)}{\partial \theta_i} \right|_{-1}$ es el valor absoluto de $\frac{\partial H(\theta)}{\partial \theta_i}$ en la iteración anterior. El comportamiento de $\frac{\partial H(\theta)}{\partial \theta_i}$ se puede observar en la figura 1.22.

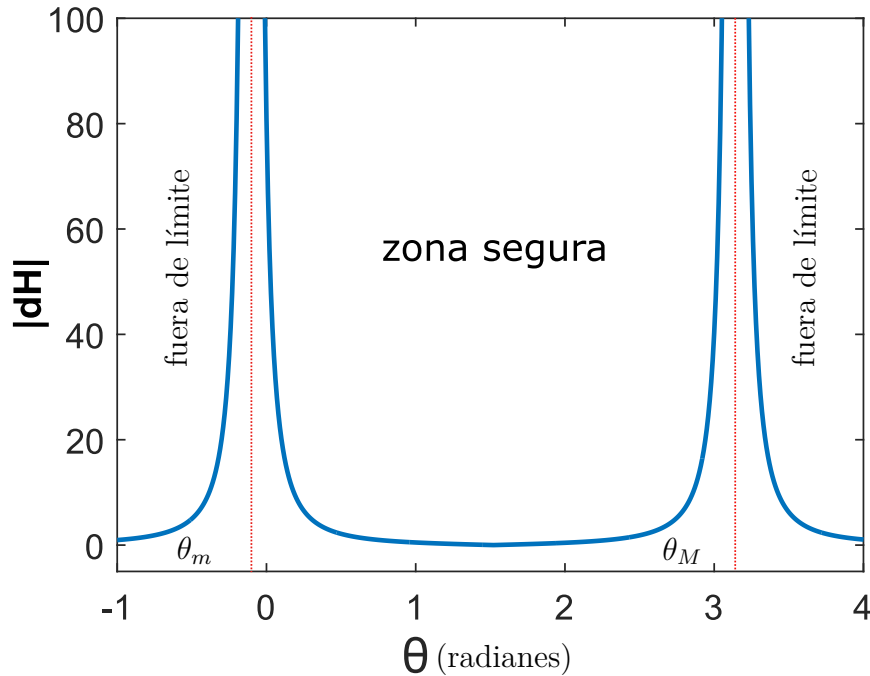


Figura 1.22. Comportamiento de función $|\partial H(\theta)/\partial \theta_i|$.

Cabe mencionar que esta metodología no tiene la capacidad de regresar a una junta fuera

de rango al intervalo seguro de movimiento.

1.8. Péndulo Invertido Lineal

Uno de los métodos más utilizados en la generación de trayectorias de marcha donde solo se considera el movimiento sobre el plano sagital, es el Péndulo Invertido Lineal (PIL). Esta metodología asume que un robot humanoide durante una trayectoria de marcha tendrá el comportamiento dinámico de un péndulo invertido. La principal ventaja de este método es que no es necesario calcular el modelo dinámico completo del humanoide, lo que simplifica en gran manera el cálculo del ZMP. Este método permite generar la trayectoria por donde el COM del robot debe moverse durante el caminado y a partir de ella generar las configuraciones articulares que cumplan dicho movimiento. El modelo del PIL asume que toda la masa del cuerpo se concentra en la ubicación del COM del robot y ésta se une al suelo mediante un eslabón con dos DOF como se muestra en la figura 1.23. Los DOF se representan por dos juntas, una revoluta asociada a un desplazamiento θ y un par τ y una junta prismática asociada a un desplazamiento traslacional r y una fuerza f .

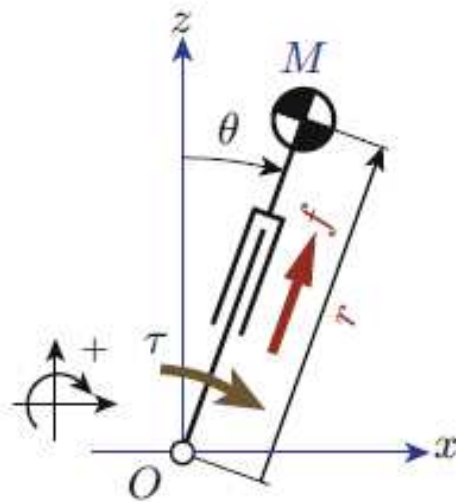


Figura 1.23. Péndulo Invertido Lineal (tomado de [Kajita *et al.*, 2014]).

Las ecuaciones de movimiento del péndulo son:

$$r^2\ddot{\theta} + 2r\dot{r}\dot{\theta} - gr\sin\theta = \tau/M \quad (1.55)$$

$$\ddot{r} - r\dot{\theta}^2 + g \cos \theta = f/M \quad (1.56)$$

donde $\dot{\theta}$ y $\ddot{\theta}$ representan la velocidad y aceleración angular del péndulo, \dot{r} y \ddot{r} la velocidad y aceleración radial, M es la masa total del robot y g la gravedad.

Debido a la limitada superficie de contacto que ofrece el pie de apoyo del robot contra la superficie, no se puede utilizar un par τ grande, ya que de lo contrario pueden propiciarse deslizamientos entre las superficies, siendo una de las principales limitantes del PIL. Para generar un movimiento del COM con solo componente horizontal, es decir manteniendo una altura fija, se prefiere usar la fuerza f , lo que propiciara que el eslabón se extienda para mantener la altura de COM. La fuerza f necesaria para mantener una altura constante z está dada por

$$f = \frac{Mg}{\cos \theta} \quad (1.57)$$

la cual se obtiene a partir del equilibrio de fuerzas verticales. La fuerza horizontal que se presenta en el COM está dada por la ecuación (1.58)

$$M\ddot{x} = f \sin \theta \quad (1.58)$$

donde \ddot{x} es la aceleración del COM sobre el eje horizontal. Sustituyendo (1.57) en (1.58) y despejando \ddot{x} se obtiene la dinámica horizontal del COM

$$\ddot{x} = \frac{g}{z}x \quad (1.59)$$

ya que z es constante la solución a la ecuación diferencial (1.59) está dada por:

$$x(t) = x(0) \cosh(t/Tc) + Tc\dot{x}(0) \sinh(t/Tc) \quad (1.60)$$

$$\dot{x}(t) = \frac{x(0)}{Tc} \sinh(t/Tc) + \dot{x}(0) \cosh(t/Tc) \quad (1.61)$$

donde Tc es una constante de tiempo y se define como por $Tc = \sqrt{z/g}$, $x(0)$ y $\dot{x}(0)$ son las condiciones iniciales de posición y velocidad.

1.8.1. Péndulo Invertido Lineal 3D

El PIL puede ser extendido a su versión de tres dimensiones el cual será de utilidad para generar la trayectoria del COM de robot humanoides con un desplazamiento en el espacio. En este caso se asume que el péndulo puede rotar libremente sobre el punto de soporte y extender su longitud mediante una junta prismática. A diferencia del péndulo invertido lineal se plantea un plano de restricción por el cual el COM debe moverse, como se observa en la figura 1.24.

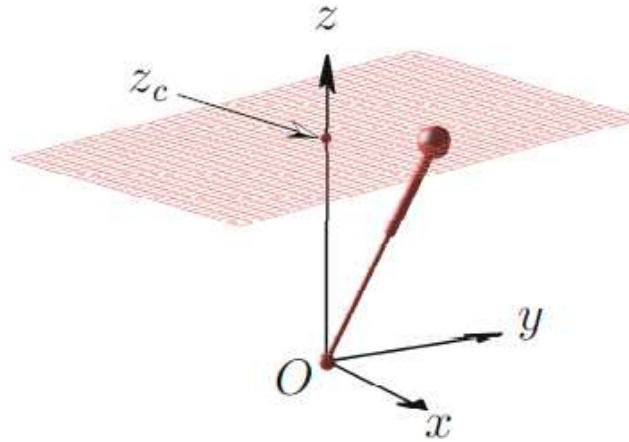


Figura 1.24. Péndulo Invertido Lineal 3D (tomado de [Kajita *et al.*, 2014]).

El plano de restricción se define como:

$$z = k_x x + k_y y + z_c \quad (1.62)$$

donde x y y son los componentes \hat{i} y \hat{j} del COM respectivamente, k_x y k_y son valores escalares constantes que determinan la inclinación del plano y z_c el punto de cruce del plano con el eje z . Realizando un proceso similar al usado en el desarrollo del péndulo invertido lineal se obtiene las ecuaciones (1.63) y (1.64) que representan la dinámica horizontal del COM para el péndulo invertido lineal en 3D.

$$\ddot{x} = \frac{g}{z_c} x \quad (1.63)$$

$$\ddot{y} = \frac{g}{z_c} y \quad (1.64)$$

Resolviendo las ecuaciones (1.63) y (1.64) se obtienen las ecuaciones de posición horizontal (1.65) y (1.67), y las ecuaciones de velocidad horizontal (1.66) y (1.68).

$$x(t) = x(0) \cosh(t/T_c) + T_c \dot{x}(0) \sinh(t/T_c) \quad (1.65)$$

$$\dot{x}(t) = \frac{x(0)}{T_c} \sinh(t/T_c) + \dot{x}(0) \cosh(t/T_c) \quad (1.66)$$

$$y(t) = y(0) \cosh(t/T_c) + T_c \dot{y}(0) \sinh(t/T_c) \quad (1.67)$$

$$\dot{y}(t) = \frac{y(0)}{T_c} \sinh(t/T_c) + \dot{y}(0) \cosh(t/T_c) \quad (1.68)$$

donde T_c es una constante de tiempo dada por $T_c = \sqrt{z_c/g}$, y $x(0)$ y $y(0)$ son las condiciones iniciales de posición, y $\dot{x}(0)$ y $\dot{y}(0)$ son las condiciones iniciales de velocidad. El desarrollo completo de las ecuaciones se puede consultar en [Kajita *et al.*, 2014].

En este capítulo se presenta el estado del arte de la robótica humanoide así como, una revisión de los conceptos básicos del modelado de robots y los criterios para la estabilidad mecánica de un humanoide. También se introduce la cinemática directa e inversa del COM y diversas definiciones de la pseudo-inversa las cuales son importantes para este trabajo y que se utilizarán en los siguientes capítulos. Finalmente se profundiza en diversas metodologías para mantener a las juntas del robot dentro de sus límites operativos y en el método del Péndulo Invertido Lineal para generar trayectorias del COM de un humanoide.

Modelado de robots humanoides

El modelado de robots humanoides presenta varias complicaciones que se deben tomar en cuenta. La primera de ellas es que en comparación con un robot industrial común, los humanoides cuentan con una cantidad mayor de **DOF**. Adicionalmente, los **DOF** de un humanoide se encuentran distribuidos en varias extremidades lo que complica aún más su modelado. Otro de los aspectos a considerar es que el humanoide no se encuentra fijo a una base y al desplazarse cambia constantemente su marco de referencia. Tanto el modelado cinemático directo como el inverso deben tener diversas consideraciones, e incluso existen varias formas de obtener los modelos.

En este capítulo se abordará el modelado cinemático de un robot humanoide haciendo hincapié en las diversas consideraciones que se deben tomar debido a las características antes mencionadas.

2.1. Modelado de robots humanoides

En este trabajo se considerará el modelado de robots humanoides que cuenten con cuatro extremidades: dos brazos y dos piernas, además se considerará que siempre hay al menos una planta de alguno de los pies apoyada fijamente en el suelo o superficie por donde se desplaza.

Para el modelado se considerará un humanoide que cuenta con seis **DOF** por cada pierna y tres **DOF** para cada brazo, obteniendo un total de 18 **DOF**. Este modelo permite representar de una forma cercana las articulaciones principales de un ser humano, además de que la mayoría de los robots actuales optan por una configuración similar por lo que es conveniente para este trabajo.

El modelo cinemático presentado se basa en el robot BIOLOID PREMIUM mostrado en figura 2.1 y su esquema representativo se muestra en la figura 2.2(a) donde el torso está

representado por color verde, la cabeza de color rojo, la ubicación de las juntas por los círculos azules y las líneas negras representan eslabones. Como se puede observar en la figura 2.2(a) existen dos eslabones compuestos por los tramos L_7 y L_8 , por simplicidad se tomarán en cuenta los tramos l que son la hipotenusa resultante de los segmentos L_7 y L_8 , y están denotados por líneas rojas punteadas en la figura.



Figura 2.1. Robot BIOLOID PREMIUM (vista frontal y lateral).

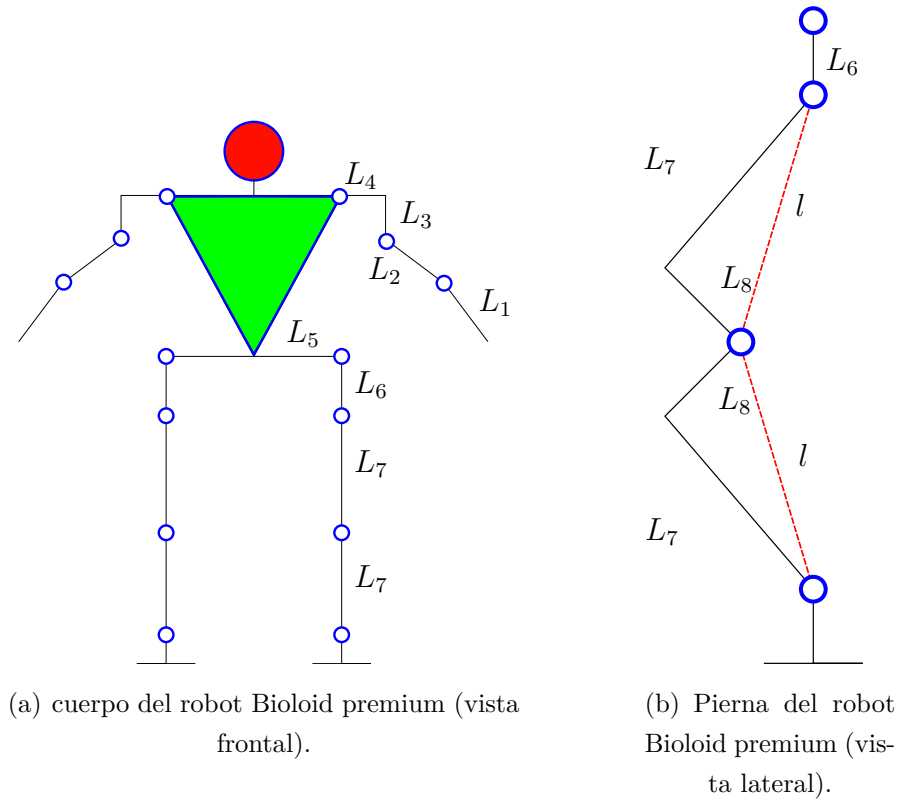


Figura 2.2. Modelo simplificado de robot Bioloid premium.

2.2. Cinemática directa de un robot humanoide

El método seleccionado para obtener la cinemática directa del robot será Denavit-Hartenberg, sin embargo, existen dos opciones a tomar en cuenta:

1. Colocar el marco de referencia global en la cadera.
2. Colocar el marco de referencia global en la planta del pie.

2.2.1. Cinemática directa respecto a la cadera

Con la opción 1 se tienen 4 cadenas cinemáticas independientes unidas en cadera (brazo izquierdo, brazo derecho, pierna izquierda y pierna derecha). Sin embargo, se requiere una transformación extra para posicionar la cadera respecto al suelo.

Cinemática directa de los brazos Para obtener la posición y orientación de las manos del robot humanoide referenciados al marco colocado en la cadera, primero se realiza una transformación homogénea del marco de la cadera a los marcos colocados en los hombros del robot. Estas transformaciones se presentan en la ecuación (2.1) para el brazo izquierdo y en la ecuación (2.2) para el brazo derecho

$$\mathbf{H}_c^{hi} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & ph \\ 1 & 0 & 0 & cp \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$

$$\mathbf{H}_c^{hd} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -ph \\ 1 & 0 & 0 & cp \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

donde \mathbf{H}_c^{hi} es la transformación homogénea entre el centro la cadera y el hombro izquierdo, \mathbf{H}_c^{hd} es la transformación homogénea entre el centro la cadera y el hombro derecho, cp es la distancia existente entre la cadera y el pecho del robot, ph la distancia entre el centro del pecho y el hombro del robot.

Una vez realizada la transformación de la cadera a los respectivos hombros, se pueden obtener los parámetros D-H de los brazos con la configuración mostrada en las figuras 2.3 para el brazo izquierdo y 2.4 para el brazo derecho. Los brazos considerados en este modelo solo cuentan con juntas de revoluta y están representadas por la letra j_i y sus variables asociadas por θ_i donde i es el número de la junta.

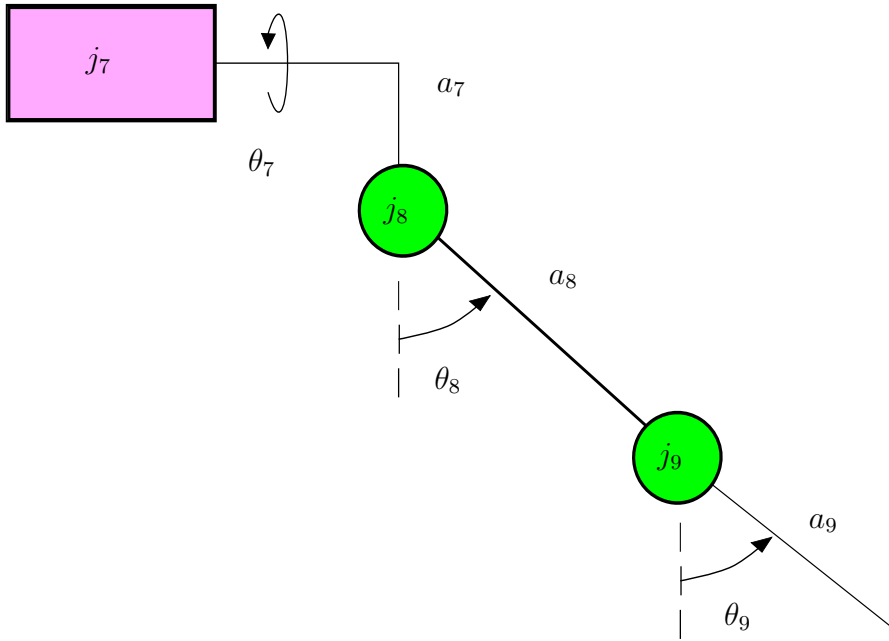


Figura 2.3. Representación de las juntas del brazo izquierdo del robot humanoide.

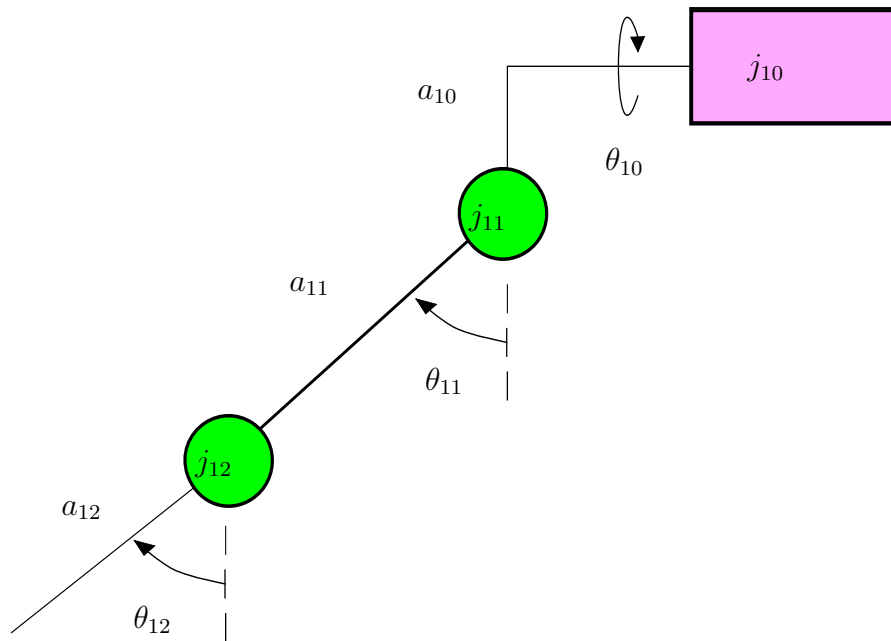


Figura 2.4. Representación de las juntas del brazo derecho del robot humanoide.

Los parámetros D-H para los brazos se muestran en la tabla 2.2.1.

Tabla 2.2.1. Parámetros D-H de los brazos.

Eslabón	a_i	α_i	d_i	θ_i	Eslabón	a_i	α_i	d_i	θ_i
7	a_7	90	0	θ_7	10	a_{10}	-90	0	θ_{10}
8	a_8	0	0	θ_8	11	a_{11}	0	0	θ_{11}
9	a_9	0	0	θ_9	12	a_{12}	0	0	θ_{12}

La transformación homogénea de las manos izquierda y derecha se obtiene a partir de las ecuaciones (2.3) y (2.4) respectivamente

$$\mathbf{H}_c^9 = \mathbf{H}_c^{hi} * \mathbf{H}_{hi}^7 * \mathbf{H}_7^8 * \mathbf{H}_8^9 \quad (2.3)$$

$$\mathbf{H}_c^{12} = \mathbf{H}_c^{hd} * \mathbf{H}_{hd}^{10} * \mathbf{H}_{10}^{11} * \mathbf{H}_{11}^{12} \quad (2.4)$$

donde \mathbf{H}_c^9 es la matriz de transformación homogénea que relaciona el marco de referencia del centro de la cadera con el marco de la mano izquierda, \mathbf{H}_c^{12} es la matriz de transformación homogénea que relaciona el marco de referencia del centro de la cadera con el marco de la mano derecha, \mathbf{H}_{hi}^7 es la matriz de transformación homogénea que relaciona el marco de referencia del hombro izquierdo con el marco de referencia de la junta 7 y \mathbf{H}_{hd}^{10} es la matriz de transformación homogénea que relaciona el marco de referencia del hombro derecho con el marco de referencia de la junta 10.

Cinemática directa de piernas Para obtener la CmD de las piernas se sigue un procedimiento similar al que se utilizó con los brazos. Primero se realiza una transformación homogénea del marco de la cadera al marco colocado en el centro de la junta esférica, la cual es representada por rojo, verde y azul en la figura 2.5.

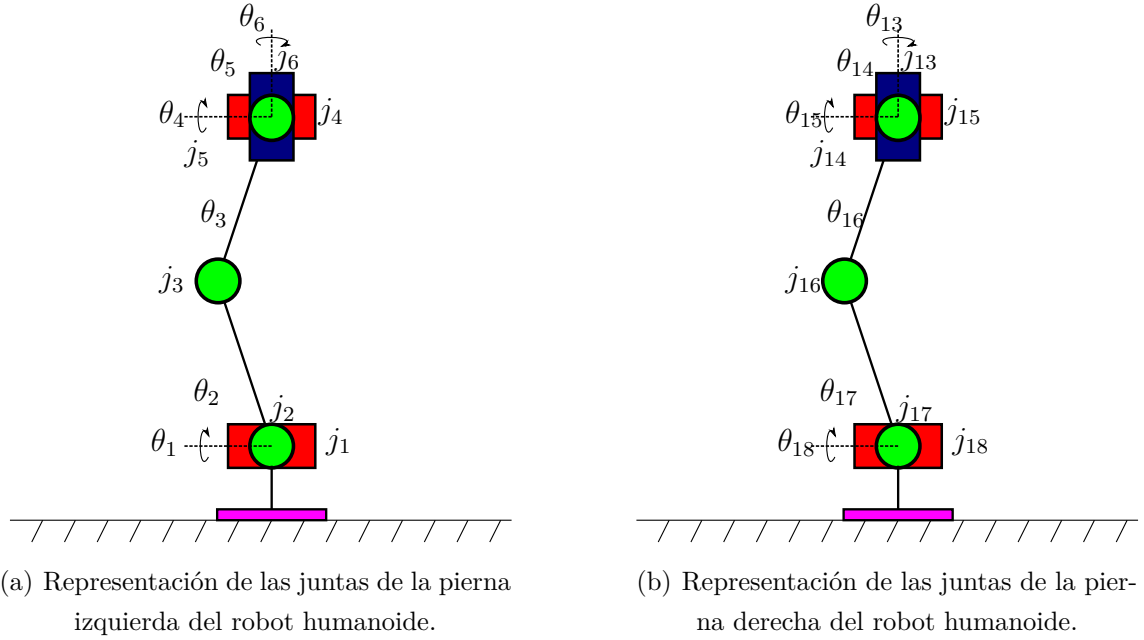


Figura 2.5. Representación de las juntas de las piernas.

Las transformaciones homogéneas que relacionan 1) el marco de la "cadera" en el de la junta esférica de la pierna izquierda está dada por la ecuación (2.5) y 2) el marco de la "cadera" en el de la junta esférica de la pierna derecha está dada por la ecuación (2.6).

$$H_c^{e1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & s_1 \\ 0 & 0 & 1 & s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

$$H_c^{e2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -s_1 \\ 0 & 0 & 1 & s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

donde s_1 es el componente en el eje y de la distancia entre la cadera y el centro de la junta esférica y s_2 el componente en el eje z . Una vez colocado el marco de referencia en la junta esférica se procedió a obtener los parámetros D-H desde la junta esférica hasta la planta del pie, los cuales se muestran en la tabla 2.2.2.

Tabla 2.2.2. Parámetros D-H de las piernas.

Eslabón	a_i	α_i	d_i	θ_i	Eslabón	a_i	α_i	d_i	θ_i
6	0	-90	0	θ_6	13	0	-90	0	θ_{13}
5	0	90	0	θ_5	14	0	90	0	θ_{14}
4	a_4	90	0	θ_4	15	a_{15}	90	0	θ_{15}
3	a_3	0	0	θ_3	16	a_{16}	0	0	θ_{16}
2	0	-90	0	θ_2	17	0	-90	0	θ_{17}
1	0	0	0	θ_1	18	0	0	0	θ_{18}

substituyendo los parámetros mostrados en la tabla 2.2.2 en (1.17) se obtienen las matrices de transformación entre eslabones. Las transformaciones existentes entre la cadera y la planta del pie se obtienen a partir de las siguientes ecuaciones

$$\mathbf{H}_c^1 = \mathbf{H}_c^{e1} * \mathbf{H}_{e1}^6 * \mathbf{H}_6^5 * \mathbf{H}_5^4 * \mathbf{H}_4^3 * \mathbf{H}_3^2 * \mathbf{H}_2^1 \quad (2.7)$$

$$\mathbf{H}_c^{18} = \mathbf{H}_c^{e2} * \mathbf{H}_{e2}^{13} * \mathbf{H}_{13}^{14} * \mathbf{H}_{14}^{15} * \mathbf{H}_{15}^{16} * \mathbf{H}_{16}^{17} * \mathbf{H}_{17}^{18} \quad (2.8)$$

donde \mathbf{H}_c^1 es la matriz de transformación homogénea entre el marco del centro de la cadera a el marco asociado a la junta 1, \mathbf{H}_c^{18} la matriz transformación homogénea entre el marco del centro de la cadera al marco asociado a la junta 18, \mathbf{H}_c^{e1} la matriz de transformación homogénea entre el marco del centro de la cadera y la junta esférica de la pierna izquierda, \mathbf{H}_{e1}^6 la matriz de transformación homogénea entre la junta esférica de la pierna izquierda y la junta 6, \mathbf{H}_c^{e2} la matriz transformación homogénea entre el marco del centro de la cadera y la junta esférica de la pierna derecha y \mathbf{H}_{e2}^6 la matriz de transformación homogénea entre la junta esférica de la pierna derecha y la junta 13.

2.2.2. Cinemática directa respecto a la planta del pie

En este caso se obtienen 3 cadenas cinemáticas conectadas en la cadera, es decir que las 3 cadenas comparten los eslabones pertenecientes a la pierna de apoyo. Las matrices de transformación que se obtiene dependen de más variables y por lo cual son más complejas. En contraste con calcular la cinemática directa respecto a la cadera, la Cinemática directa respecto a la planta del pie permite tener una referencia clara respecto al suelo, ya que uno de los pies se debe considerar en contacto y fijo sobre él.

Cinemática directa de la pierna La configuración de la pierna de apoyo vista desde la planta del pie es muy similar a la que se presenta en un robot manipulador de seis DOF, donde se cuentan con tres DOF para posicionar al efector final y tres para orientarlo. En este caso se obtendrá la **CmD** de la pierna de apoyo considerando como efector final el centro de la cadera, esto permitirá utilizar la **CmD** de los brazos y pierna previamente calculada.

Los últimos tres grados de libertad de cada pierna se pueden representar mediante una junta esférica. La junta esférica tiene la capacidad para orientar un cuerpo sobre tres ejes, por lo que, es ampliamente usada y descrita en la literatura.

Suponiendo que la pierna izquierda está apoyada en el piso, sus parámetros D-H se presentan en la tabla 2.2.3.

Tabla 2.2.3. Parámetros D-H de la pierna izquierda apoyada sobre el piso.

Eslabón	a_i	α_i	d_i	θ_i
1	0	-90	0	θ_1
2	a_2	0	0	θ_2
3	a_3	90	0	θ_3
4	0	-90	0	θ_4
5	0	90	0	θ_5
6	0	0	d_6	θ_6

Los parámetros D-H de la tabla 2.2.3 se sustituyen en la ecuación (1.17) y se obtienen las matrices de transformación homogénea entre las diversas juntas de la pierna.

La matriz de transformación entre el tobillo y la cadera de la pierna izquierda se establece utilizando la ecuación (2.9) y la transformación inversa está dada por (2.10)

$$H_0^6 = H_0^1 * H_1^2 * H_2^3 * H_3^4 * H_4^5 * H_5^6 \quad (2.9)$$

$$H_6^0 = H_6^5 * H_5^4 * H_4^3 * H_3^2 * H_2^1 * H_1^0 \quad (2.10)$$

además es necesario realizar una transformación extra para referenciar el tobillo a la planta del pie mediante la ecuación (2.11) y hacer una traslación hacia la cadera mostrada en la ecuación (2.12)

$$\mathbf{H}_p^0 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & t \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.11)$$

$$\mathbf{H}_6^c = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.12)$$

donde t representa la distancia que hay desde la planta del pie al tobillo. Multiplicando la (2.11) por (2.9) se obtiene la transformación entre la planta del pie de apoyo y la cadera \mathbf{H}_p^c

$$\mathbf{H}_p^c = \mathbf{H}_p^0 * \mathbf{H}_0^6 * \mathbf{H}_6^c \quad (2.13)$$

Cinemática directa de los brazos Una vez determinada la transformación entre la planta del pie de apoyo y la cadera, es posible obtener la transformación homogénea de las manos derecha e izquierda respecto a la planta del pie. Utilizando las matrices de transformación previamente mostradas para los brazos derecho e izquierdo (2.4) y (2.3), las transformaciones homogéneas que relaciona el marco del pie con el marco de las manos derecha e izquierda están dadas por las ecuaciones (2.14) y (2.15) respectivamente.

$$\mathbf{H}_p^{12} = \mathbf{H}_p^c * \mathbf{H}_c^{12} \quad (2.14)$$

$$\mathbf{H}_p^9 = \mathbf{H}_p^c * \mathbf{H}_c^9 \quad (2.15)$$

Cinemática directa de la pierna en vuelo De una forma similar a la **CmD** de los brazos, es posible obtener la **CmD** de la pierna que no se encuentra fija a el suelo. Multiplicando las ecuaciones (2.13) y (2.7) previamente mostradas es posible obtener la transformación homogénea entre la planta del pie de apoyo y el pie en vuelo como se observa en la ecuación (2.13).

$$\mathbf{H}_p^{18} = \mathbf{H}_p^c * \mathbf{H}_c^{18} \quad (2.16)$$

2.3. Cinemática inversa de un robot humanoide

Al igual que con la cinemática directa del robot humanoide contamos con varias opciones para obtener su cinemática inversa. siendo la más favorable, dividir el robot por extremidad, en cuyo caso se pueden obtener soluciones analíticas para cada extremidad.

Otra alternativa es usar un método iterativo que aproxime una configuración articular para alguna posición y orientación deseada de una o varias extremidades simultáneamente.

A continuación, se obtiene la cinemática inversa de una de las piernas y un brazo ya que debido a la simetría del robot .

2.3.1. Cinemática inversa de la pierna

En este caso la pierna está conformada por 6 **DOF** rotacionales, además tres de ellas se encuentran concentradas en un punto cumpliendo con la configuración de una junta esférica. Estas características, permiten utilizar el método de desacople cinemático mencionado en la sección 1.5.2.

Posición La posición de la cadera estará dada por las tres primeras articulaciones de la pierna a partir del suelo, por lo cual es necesario calcular la posición tridimensional del centro de la junta esférica con respecto al marco colocado en la planta del pie. En la ecuación 2.17 se calcula la posición de la junta esférica \mathbf{x}_s a partir de un vector de posición \mathbf{x}' que representa la posición deseada de la cadera y una matriz de rotación \mathbf{R}' que representa la orientación deseada del marco colocado en la cadera

$$\mathbf{x}_s = \mathbf{x}' - \mathbf{R}' * \begin{pmatrix} 0 \\ L_5 \\ L_6 \end{pmatrix} \quad (2.17)$$

donde L_5 y L_6 son los componentes y y z del vector que une el centro de la junta esférica con el centro de la cadera. Posteriormente mediante relaciones geométricas calcular los tres ángulos utilizando el modelo geométrico simplificado de la pierna mostrado en la figura 2.6, donde el centro de la junta esférica está marcado con un círculo rojo.

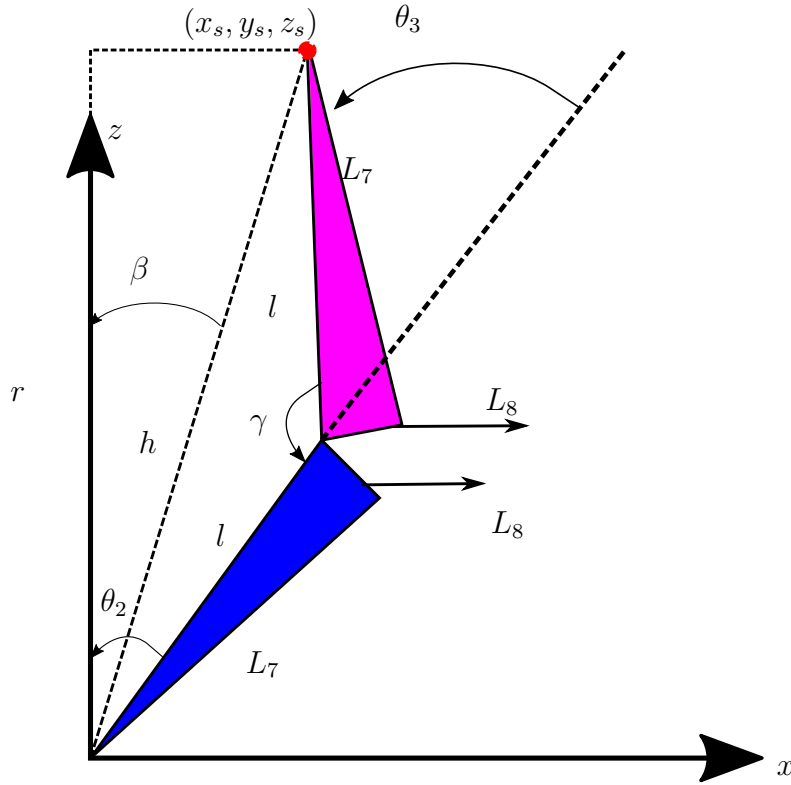


Figura 2.6. Esquema geométrico de una pierna del robot humanoide.

donde L_7 y L_8 son los componentes de los eslabones 2 y 3, x_s , y_s y z_s son los componentes \hat{i} , \hat{j} y \hat{k} de la posición de la junta esférica, l , r , h , γ y β son variables geométricas auxiliares que ayudan a obtener los ángulos de interés y se definen a continuación

$$l = \sqrt{L_7^2 + L_8^2} \quad (2.18)$$

$$r = \sqrt{z_s^2 + y_s^2} \quad (2.19)$$

$$h = \sqrt{r^2 + x_s^2} \quad (2.20)$$

$$\cos(\gamma) = \frac{h^2 - 2 * l^2}{-2 * l^2}; \quad (2.21)$$

$$\gamma = \arctan 2 \left(\sqrt{1 - \cos(\gamma)^2}, \cos(\gamma) \right) \quad (2.22)$$

$$\beta = -\arctan 2(x_s, r) \quad (2.23)$$

Las soluciones geométricas para θ_1 , θ_2 y θ_3 se muestran en la parte inferior.

$$\theta_1 = -\arctan 2(y_s, z_s) \quad (2.24)$$

$$\theta_2 = \beta - \frac{\pi - \gamma}{2} \quad (2.25)$$

$$\theta_3 = \pi - \gamma \quad (2.26)$$

Orientación El problema de orientación consiste en resolver los últimos tres variables articulares para obtener la orientación deseada.

Utilizando los valores de las variables articulares calculados en el paso anterior, se obtiene la matriz de rotación \mathbf{R}_0^3 la cual se introduce en la ecuación (2.27) para obtener la matriz \mathbf{R}_3^6 .

$$\mathbf{R}_3^6 = (\mathbf{R}_0^3)^T * \mathbf{R}' \quad (2.27)$$

La matriz \mathbf{R}_3^6 se puede considerar como una matriz de rotación de Euler zyz , de la cual se puede obtener los ángulos correspondientes ϕ , θ y ψ , que coinciden con las variables articulares faltantes θ_4 , θ_5 y θ_6 como se observa a continuación

$$\theta_4 = \phi = \arctan 2(\mathbf{R}_3^6(2, 3), \mathbf{R}_3^6(1, 3)); \quad (2.28)$$

$$\theta_5 = \theta = \arctan 2\left(\sqrt{1 - \mathbf{R}_3^6(3, 3)^2}, \mathbf{R}_3^6(3, 3)\right); \quad (2.29)$$

$$\theta_6 = \psi = \arctan 2(\mathbf{R}_3^6(3, 2), -\mathbf{R}_3^6(3, 1)); \quad (2.30)$$

las ecuaciones mostradas fueron tomadas de [Spong *et al.*, 2006].

2.3.2. Cinemática inversa del brazo

Los brazos solamente cuentan con 3 DOF, por lo que, solo es posible obtener la configuración articular para una posición \mathbf{x}' dada.

En la figura 2.7 se observa el esquema geométrico del brazo izquierdo del robot humanoide mostrando con un círculo rojo el punto de interés de la mano.

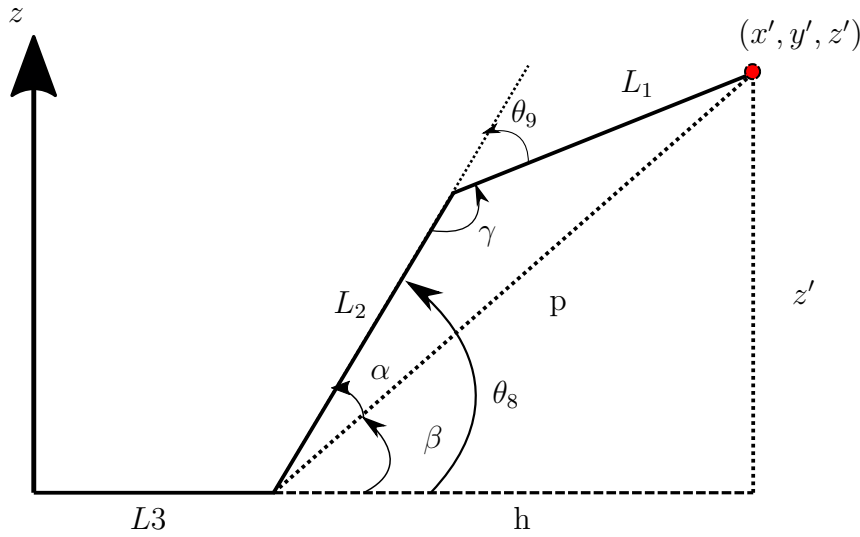


Figura 2.7. Esquema geométrico del brazo izquierdo del robot humanoide.

donde L_1 , L_2 y L_3 son las dimensiones de los eslabones del brazo, x' , y' y z' son los componentes de \mathbf{x}' , r , h , p , α , β y γ son variables auxiliares que se obtienen a partir de la figura 2.7 y se definen debajo

$$r = \sqrt{x'^2 + y'^2} \quad (2.31)$$

$$h = r - L_3 \quad (2.32)$$

$$\beta = \arctan 2(z, h) \quad (2.33)$$

$$p = \sqrt{h^2 + z^2} \quad (2.34)$$

$$\cos(\gamma) = \frac{p^2 - L_2^2 - L_1^2}{-2L_2L_1} \quad (2.35)$$

$$\gamma = \arctan 2 \left(\sqrt{1 - \cos(\gamma)^2}, \cos(\gamma) \right) \quad (2.36)$$

$$\cos(\alpha) = \frac{L_1^2 - L_2^2 - p^2}{-2L_2p} \quad (2.37)$$

$$\alpha = \arctan 2 \left(\sqrt{1 - \cos(\alpha)^2}, \cos(\alpha) \right) \quad (2.38)$$

utilizando las definiciones anteriores es posible obtener las variables articulares del brazo como sigue:

$$\theta_7 = \arctan 2(y, x) \quad (2.39)$$

$$\theta_8 = \beta + \alpha \quad (2.40)$$

$$\theta_9 = \pi - \gamma \quad (2.41)$$

2.3.3. Cinemática inversa del cuerpo completo

Se puede obtener una configuración articular \mathbf{q} que defina la postura del cuerpo del humanoide utilizando la cinemática inversa mostrada en la parte superior, sin embargo, se debe considerar al menos un pie fijo respecto al suelo, el cual será la referencia. La configuración articular se calculará a partir de un vector y una matriz dadas \mathbf{x}_p^c y \mathbf{R}_p^c que representa la posición y orientación deseadas del centro de la cadera respectivamente, un vector y una matriz \mathbf{x}_p^f y \mathbf{R}_p^f que representa la posición y orientación del pie en vuelo, y dos vectores \mathbf{x}_p^i y \mathbf{x}_p^d que simbolizan la posición de la mano izquierda y mano derecha respectivamente. Todas con respecto al marco de referencia del pie de apoyo. A continuación, se muestra el algoritmo para la cinemática inversa del robot completo con 18 DOF:

1. Pie de apoyo.
 - Se introducen el vector de posición \mathbf{x}_p^c y la matriz de rotación \mathbf{R}_p^c deseados para la cadera.

- Se obtiene la configuración articular \mathbf{q}_1^6 de la pierna que está en contacto con el suelo, donde \mathbf{q}_1^6 es un vector que contiene las variables articulares de la 1 a la 6. \mathbf{q}_1^6 se calcula por medio de la cinemática inversa para la pierna desarrollada anteriormente en la sección 2.3.1.

2. Pie flotante

- A partir de la configuración articular de la pierna de apoyo se obtiene la matriz de transformación homogénea \mathbf{H}_c^p que relaciona el marco de la cadera con el marco de la planta del pie.
- Multiplicando \mathbf{R}_0^f por una matriz de similaridad se obtiene \mathbf{R}_c^f que representa la orientación del pie en vuelo respecto al marco de la cadera.
- Se cambia de referencia \mathbf{x}_0^f al marco de la cadera obteniendo \mathbf{x}_c^f y se forma la matriz de transformación homogénea \mathbf{H}_c^{18} que relaciona el marco de la cadera con la planta del pie en vuelo. La matriz \mathbf{H}_c^{18} se muestra a continuación:

$$\mathbf{H}_c^{18} = \begin{bmatrix} \mathbf{R}_f & \mathbf{x}_f \\ 0 & 1 \end{bmatrix}. \quad (2.42)$$

- Se invierte la matriz \mathbf{H}_c^{18} , obteniendo \mathbf{H}_{18}^c .
- Se calcula la configuración articular de la pierna en vuelo \mathbf{q}_{13}^{18} , donde \mathbf{q}_{13}^{18} es un vector que contiene las variables articulares de la 13 a la 18. \mathbf{q}_{13}^{18} se obtiene por medio de la cinemática inversa de una pierna mostrado en la sección 2.3.1.

3. Brazo izquierdo.

- Se obtiene la matriz de transformación homogénea \mathbf{H}_p^h que relaciona la planta del pie con el hombro izquierdo y se invierte para conseguir \mathbf{H}_{hi}^p .
- Se cambia de referencia \mathbf{x}_p^i al marco que se encuentra en el hombro obteniendo \mathbf{x}_{hi}^i .
- Se obtienen las variables articulares \mathbf{q}_7^9 , donde \mathbf{q}_7^9 es un vector que contienen las variables articulares de la 7 a la 9. \mathbf{q}_7^9 se calcula por medio de la cinemática inversa del brazo, desarrollada anteriormente en la sección 2.3.2.

4. Brazo derecho.

- Se obtiene la matriz de transformación homogénea \mathbf{H}_p^{hd} que relaciona la planta del pie con el hombro derecho y se invierte para conseguir \mathbf{H}_{hd}^p .

- Se cambia de referencia \mathbf{x}_0^d al marco que se encuentra en el hombro obteniendo \mathbf{x}_{hd}^d .
- Se obtienen las variables articulares \mathbf{q}_{10}^{12} , donde \mathbf{q}_7^9 es un vector que contiene las variables articulares de la 7 a la 9. \mathbf{q}_7^9 se calcula por medio de la cinemática inversa del brazo, desarrollada anteriormente en la sección 2.3.2.

Es necesario decir que es complicado determinar en este punto si alguna de las posiciones que estamos introduciendo al algoritmo sea realizable, es decir alguna que se encuentren dentro del espacio del robot y de la extremidad, por lo que es muy propenso a obtener un error matemático durante su ejecución.

En la figura 2.9 se puede observar una pose obtenida a partir del algoritmo de cinemática inversa del cuerpo; las extremidades están representadas por líneas y el torso se representa como un triángulo verde, el frente del robot está orientado hacia la dirección x positiva.

2.4. Generación de trayectorias cinemáticas

La generación de trayectorias cinemáticas de marcha, está basada en la hipótesis de que el robot humanoide se mueve lo suficientemente lento para no generar fuerzas inerciales. El robot estará estable (en equilibrio) mientras esa la proyección de su COM sobre el piso permanezca dentro del polígono de apoyo. El objetivo de la generación de trayectorias cinemáticas es encontrar movimientos del robot que permitan generar una marcha de baja velocidad mientras se mantiene la estabilidad o equilibrio mecánico del mismo. Estos movimientos estarán definidos por las poses (configuraciones) inicial y final del robot para cada segmento de movimiento.

El procedimiento utilizado para la generación de trayectorias se describe a continuación.

1. Determinar los puntos tridimensionales donde la cadera y el pie en movimiento del robot deben posicionarse.
 2. Por medio de *splines* cúbicas segmentarias se interpolan puntos intermedios que permitan realizar una trayectoria suave en el movimiento del robot.
 3. Usando cinemática directa e inversa del cuerpo completo, se obtienen las configuraciones articulares que cumplen las posiciones indicadas previamente.
-

4. Se calcula la trayectoria que siguió el COM y se verifica la estabilidad del robot durante el trayecto.

Los pasos anteriores se pueden visualizar gráficamente en la figura 2.8.

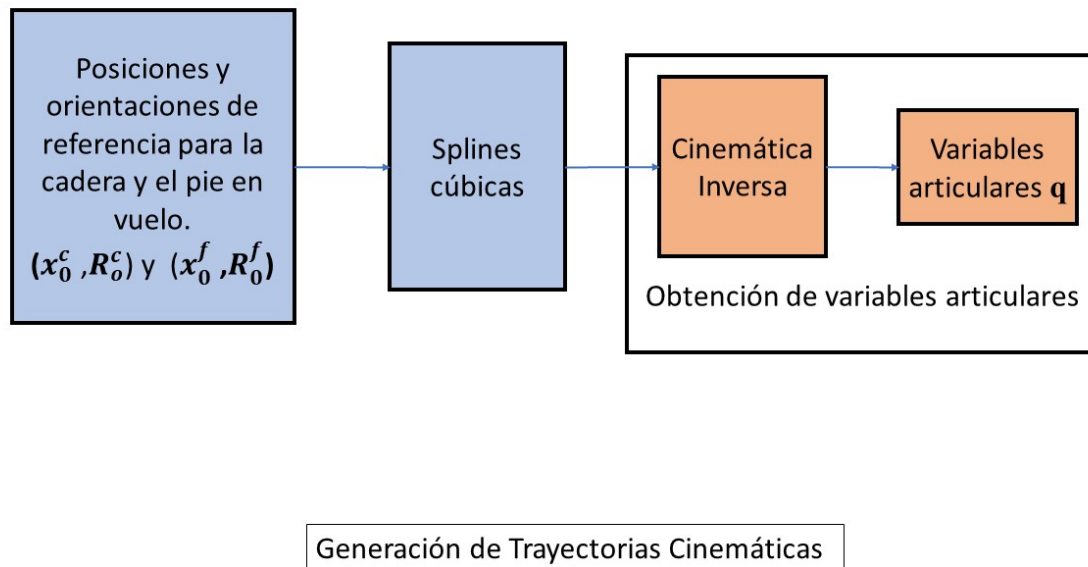
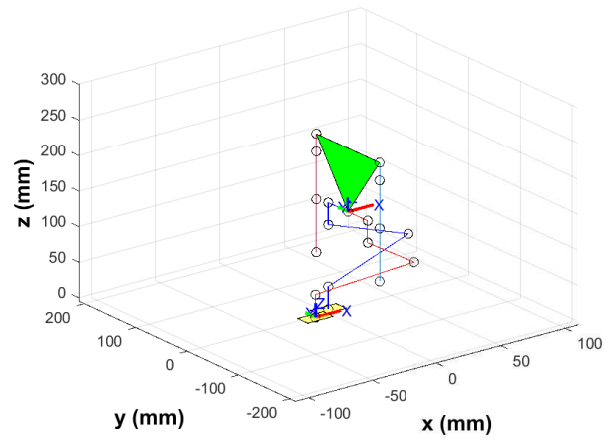


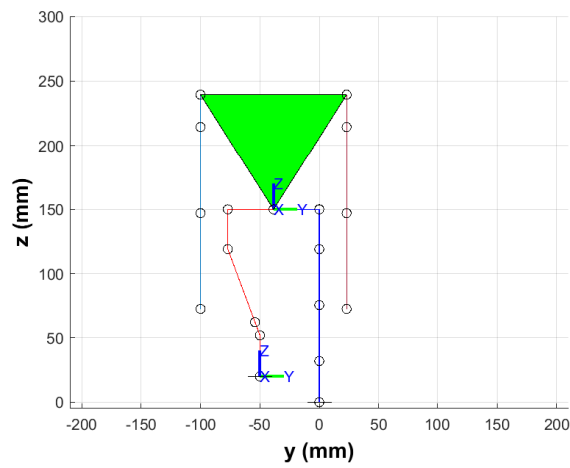
Figura 2.8. Algoritmo utilizado para la generación de trayectorias cinemáticas.

En la figura 2.10 se muestra una serie de poses representando el caminado de un humanoide, a partir del algoritmo descrito arriba. En la pose 1 se observa la posición inicial del robot donde la planta del pie izquierdo se encuentra fija al suelo y fungirá como referencia del robot, durante las poses 2 y 3 el robot se inclina hacia la pierna izquierda para mantener el COM dentro del polígono de apoyo, en las poses siguientes el robot mueve el pie para ejecutar un paso finalizándolo en la pose 10.

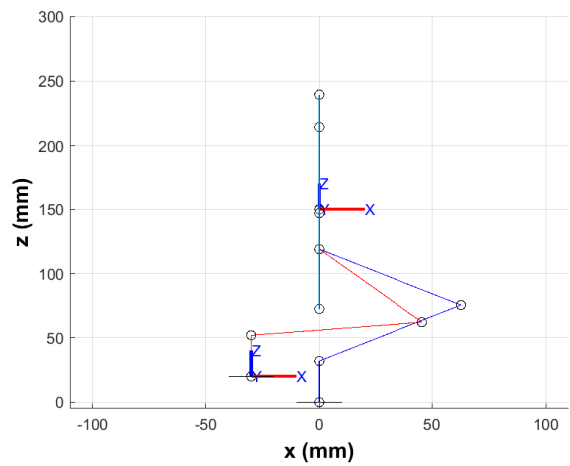
Debido a que este algoritmo no controla directamente el posicionamiento del COM es propenso a generar trayectorias de marcha inestables, por lo que es necesario integrar este aspecto. En los capítulos siguientes se desarrollan algoritmos que generan trayectorias cinemáticas que aseguran estabilidad.



(a) Vista 3D.



(b) Vista frontal.



(c) vista lateral.

Figura 2.9. Pose obtenida a partir del algoritmo de cinemática inversa para el robot completo.

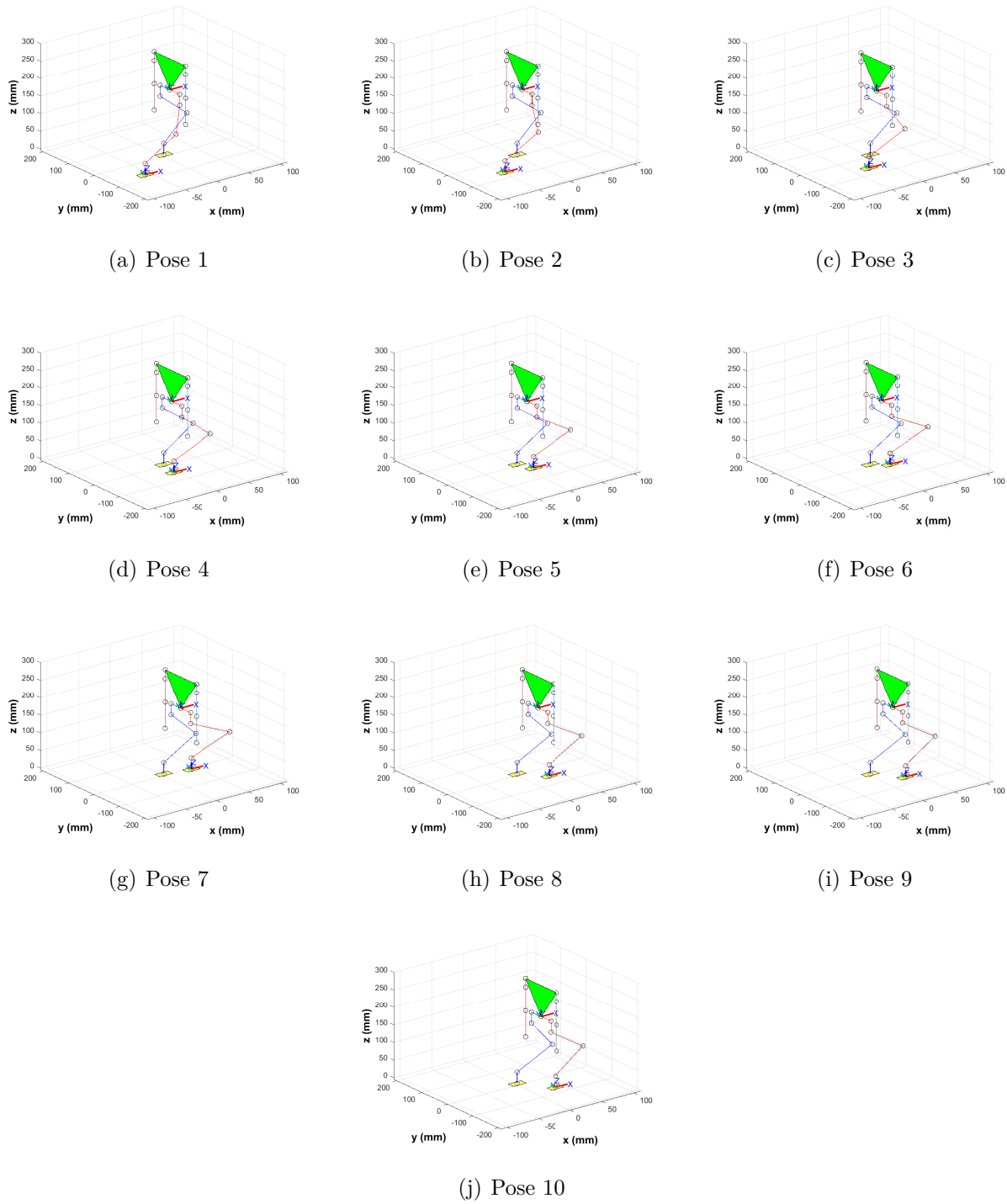


Figura 2.10. Trayectoria cinemática generada para un paso.

2.5. Conclusión del capítulo

En este capítulo se desarrolló el modelado cinemático directo e inverso de un robot humanoide. El robot humanoide considerado para este desarrollo cuenta con 18 **DOF**. Este robot permite representar la mayoría de las articulaciones de un cuerpo humano y además es una configuración ampliamente adoptada dentro de la robótica humanoide.

Se presentan dos formas de abordar la cinemática directa del humanoide, siendo la más adecuada para el trabajo la cinemática directa con marco de referencia en el pie de apoyo. Para futuros capítulos los jacobianos obtenidos a partir de este desarrollo serán de mayor utilidad para la generación de trayectorias de marcha estables.

Se obtuvieron las expresiones para la cinemática inversa de una pierna y un brazo, y se creó el algoritmo para aplicarlas al cuerpo del humanoide. Finalmente, se usó el algoritmo de cinemática inversa del cuerpo completo para generar una trayectoria cinemática de caminado aunque sin garantía de estabilidad o equilibrio del robot.

Control del centro de masa para robots humanoides

A pesar de que cinemática inversa para la obtención de una configuración articular que coloque al robot humanoide en una pose estable es funcional y se use comúnmente, se pueden presentar errores al posicionar el COM. El problema del algoritmo de Cinemática inversa del cuerpo completo es que no se controla el COM explícitamente. Sin embargo, se puede usar el algoritmo para generar teóricamente poses estables. Para usar el algoritmo de Cinemática inversa del cuerpo completo, hay que asociar una pose (posición y orientación) de la cadera con una posición del COM deseada. Posteriormente se usaría el algoritmo para determinar la pose del robot. Uno inconvenientes viene de que la solución de la CMI no es exacta y entonces ese error de posicionamiento trasladado al COM (lo cual ya no se calcula) puede generar inestabilidad fácilmente. Además, suelen existir errores relacionados a la asociación del COM con el movimiento de la cadera, particularmente en robots en cuyo torso es relativamente ligero. Dentro de este capítulo se presentan diversos algoritmos que permitan diseñar trayectorias cinemáticas estables de marcha, en términos directamente del COM del robot sobre el polígono de apoyo.

3.1. Generación de poses estables

Como se menciona en la sección 1.4.1, la estabilidad estática del robot se logra posicionando el COM en todo momento sobre el polígono de apoyo. Posicionar el COM no es una tarea sencilla y no existe una forma matemática cerrada para lograrlo. En la sección 1.5.5 se describe el uso de métodos iterativos para atacar este problema de posicionamiento, la cual es la opción más obvia. En el caso específico de un humanoide de n DOF, posicionar el COM presenta ciertas dificultades, como el hecho de tener una gran cantidad de eslabones, cada uno de los cuales contribuye con una fracción de la masa total del robot. Además, los

eslabones están distribuidos en diversas cadenas cinemáticas lo que dificulta la obtención de los jacobianos asociados a los COM individuales de cada eslabón. A continuación, se desarrollan diversas metodologías en busca de facilitar el posicionamiento del COM.

3.1.1. Obtención de distribución de masas objetivo

Un robot humanoide puede concebirse como cinco cadenas cinemáticas (dos piernas, dos brazos y el torso) interconectadas. Cada cadena cinemática tiene asociada una masa que llamaremos "principal" m_k ubicada en la posición \mathbf{x}_k , donde $k = 1, \dots, 5$ indica el número de la cadena. El robot tiene una masa total M dada por la suma de las masas de cada cadena. La relación entre el COM del robot (\mathbf{x}) y el de cada cadena cinemática (\mathbf{x}_k) está dada por:

$$\mathbf{x} = \frac{\sum_{k=1}^5 m_k \mathbf{x}_k}{M}. \quad (3.1)$$

La variación de \mathbf{x} respecto al tiempo se define por:

$$\dot{\mathbf{x}} = \frac{\sum_{k=1}^5 m_k \dot{\mathbf{x}}_k}{M} \quad (3.2)$$

aproximando las derivadas por incrementos se obtiene

$$\Delta \mathbf{x} = \frac{\sum_{k=1}^5 m_k \Delta \mathbf{x}_k}{M} \quad (3.3)$$

que puede ser expresado en forma matricial de la siguiente manera

$$\Delta \mathbf{x} = \mathbf{J}_c \Delta \mathbf{x}_K \quad (3.4)$$

donde $\Delta \mathbf{x}_K$ tiene la forma

$$\Delta \mathbf{x}_K = \begin{bmatrix} \Delta \mathbf{x}_1 \\ \vdots \\ \Delta \mathbf{x}_5 \end{bmatrix} \quad (3.5)$$

y \mathbf{J}_c se compone como:

$$\mathbf{J}_c = \frac{1}{M} \begin{pmatrix} \mathbf{n}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{n}_5 \end{pmatrix} \quad (3.6)$$

con

$$\mathbf{n}_i = [m_i \quad m_i \quad m_i] \quad (3.7)$$

la cual relaciona los incrementos en las posiciones de las masas m_k de forma cartesiana y el cambio en la posición de \mathbf{x} .

El problema que se plantea es cómo hay que distribuir las masas m_k para lograr \mathbf{x}^* , la posición deseada de \mathbf{x} .

Enseguida se desarrolla algoritmo de cinemática inversa del COM para obtener una distribución de masas que asegure una configuración estable para el robot. Para lo anterior primeramente, se deben determinar las ubicaciones objetivo \mathbf{x}_k^* de las masas principales dada la ubicación deseada del COM \mathbf{x}^* .

1. Se obtienen \mathbf{x} , \mathbf{x}_k y \mathbf{J}_c a partir de la configuración articular actual \mathbf{q} y las características físicas del robot humanoide.
2. se calcula el error existente entre \mathbf{x}^* y \mathbf{x} como sigue:

$$\Delta \mathbf{x} = \mathbf{x}^* - \mathbf{x} \quad (3.8)$$

3. Se utiliza el algoritmo de cinemática inversa del COM mostrado en la ecuación (3.9) para obtener la distribución de las masas m_k que satisfaga \mathbf{x}^* .

$$\Delta \mathbf{x}_K = \mathbf{J}_c^\dagger \Delta \mathbf{x} \quad (3.9)$$

4. Se aplican los incrementos a \mathbf{x}_k :

$$\hat{\mathbf{x}}_k = \mathbf{x}_k + \Delta \mathbf{x}_k \quad (3.10)$$

donde $\hat{\mathbf{x}}_k$ es el valor actualizado de \mathbf{x}_k .

5. Se recalcula \mathbf{x} con $\hat{\mathbf{x}}_k$ usando la ecuación (3.1).

6. Se calcula $\Delta \mathbf{x}$:

- o Si $\Delta \mathbf{x}$ es suficientemente pequeño

$$\mathbf{x}_k^* = \mathbf{x}_k \quad (3.11)$$

donde \mathbf{x}_k^* son las posiciones de las masas principales m_k que cumplen con \mathbf{x}^* . El proceso se finaliza.

- o Si $\Delta \mathbf{x}$ no es suficientemente pequeño se regresa al paso número 3.

Cabe recalcar que este algoritmo no toma en cuenta ninguna limitación física que presente el robot por lo que se puede obtener una distribución de masas que sea imposible de lograr en la práctica debido por ejemplo a colisiones entre las propias extremidades del robot o si se exceden los límites de movimiento de las juntas físicas. En la figura 3.1 se puede observar la distribución de masas principales m_k dentro de un cuerpo humanoide, las cuales están marcadas en amarillo y rojo, correspondientes a una postura arbitraria.

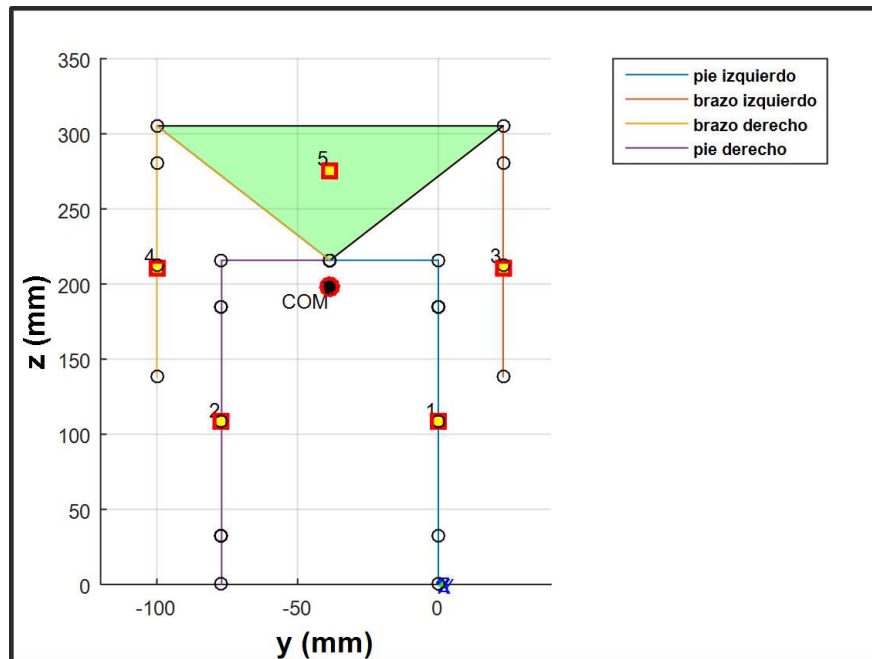


Figura 3.1. Distribución de masas principales m_k .

3.2. Obtención de configuración articular

Una vez que se tiene la ubicación deseada del COM del robot y de las masas principales, el siguiente paso es determinar la configuración articular del robot para lograrlas. Para

la obtención de la configuración articular del robot que satisfaga \mathbf{x}^* , \mathbf{q}^* , se plantean dos soluciones posibles: la primera, dividir el robot por extremidades y obtener la configuración articular de manera independiente para cada cadena cinemática o la segunda, manipular todas las juntas (y por tanto cadenas cinemáticas) del robot de manera simultánea.

3.2.1. División por Extremidades (DE)

Utilizando la distribución de las masas m_k que satisfacen \mathbf{x}^* , \mathbf{x}_k^* , obtenida en el algoritmo anterior (sección 3.1.1), es posible crear una metodología que permita obtener la configuración articular \mathbf{q}^* del humanoide. La metodología División por Extremidades (DE) consiste en reducir el error entre las posiciones actuales \mathbf{x}_k de las masas m_k , y su objetivo \mathbf{x}_k^* , a partir de aplicar incrementos en las variables articulares. Los incrementos en las variables articulares se obtienen dividiendo el cuerpo del humanoide en cuatro ya que la masa correspondiente al torso se moverá a partir de las variables articulares de la pierna de apoyo. Con este propósito, es conveniente dividir el vector de variables articulares \mathbf{q} como sigue:

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_a \\ \mathbf{q}_b \\ \mathbf{q}_c \\ \mathbf{q}_d \end{pmatrix} \quad (3.12)$$

donde en \mathbf{q}_a se almacenan las variables articulares pertenecientes a la pierna de apoyo, en \mathbf{q}_b las variables articulares del brazo derecho, en \mathbf{q}_c las variables articulares del brazo izquierdo y en \mathbf{q}_d las variables articulares de la pierna en vuelo. Es importante mencionar que alguna de las plantas de los pies se mantendrá fija sobre la superficie del suelo durante la duración del algoritmo, la cual, servirá como referencia. A continuación, se presenta el algoritmo que permitirá encontrar \mathbf{q}^* .

Algoritmo de División de Extremidades (DE)

1. Se realizan los pasos mostrados anteriormente en la sección 3.1.1 para obtener el vector objetivo \mathbf{x}_k^* .
2. Se obtiene la configuración articular de la pierna de apoyo \mathbf{q}_a , ya que ésta permitirá ubicar las masas asociadas a la pierna y el torso. Para encontrar la configuración

articular \mathbf{q}_a se utiliza el algoritmo de **CmICOM** mostrado en la sección 1.5.5, realizando varias iteraciones de la ecuación 3.13

$$\Delta \mathbf{q}_a = \alpha \mathbf{J}_a^\dagger \Delta \mathbf{x}_a \quad (3.13)$$

donde $\Delta \mathbf{x}_a$ es un vector aumentado de dimensiones 6×1 que contienen el error en la posición de las masas m_1 y m_5 que corresponden a la pierna de apoyo y el torso respectivamente, y está definido en la ecuación (3.14). \mathbf{J}_a que es un jacobiano aumentado que contiene los jacobianos que asocian los errores en posición de las masas $\Delta \mathbf{x}_1$ y $\Delta \mathbf{x}_5$ con el vector \mathbf{q}_a los cuales se obtiene a partir de la ecuación (1.30). \mathbf{J}_a se define en la ecuación (3.15).

$$\Delta \mathbf{x}_a = \begin{pmatrix} \Delta \mathbf{x}_1 \\ \Delta \mathbf{x}_5 \end{pmatrix} \quad (3.14)$$

$$\mathbf{J}_a = \begin{pmatrix} \mathbf{J}_1 \\ \mathbf{J}_5 \end{pmatrix} \quad (3.15)$$

3. Una vez obtenida la configuración articular de la pierna de apoyo \mathbf{q}_a , ésta se fija y se procede a encontrar la configuración articular de las extremidades restantes \mathbf{q}_b , \mathbf{q}_c y \mathbf{q}_d mediante el algoritmo de **CmICOM** aplicado independientemente a cada extremidad.
4. Se recalcula \mathbf{x} con la actualización de \mathbf{q} usando la ecuación (3.1).
5. Se calcula $\Delta \mathbf{x}$:

- Si $\Delta \mathbf{x}$ es suficientemente pequeño

$$\mathbf{q}^* = \mathbf{q} \quad (3.16)$$

donde \mathbf{q}^* representa la configuración articular que cumple con \mathbf{x}^* . El proceso se finaliza.

- Si $\Delta \mathbf{x}$ no es suficientemente pequeño se repite el algoritmo utilizando la configuración articular actualizada.

En la figura 3.2 se presenta el esquema utilizado en este algoritmo.

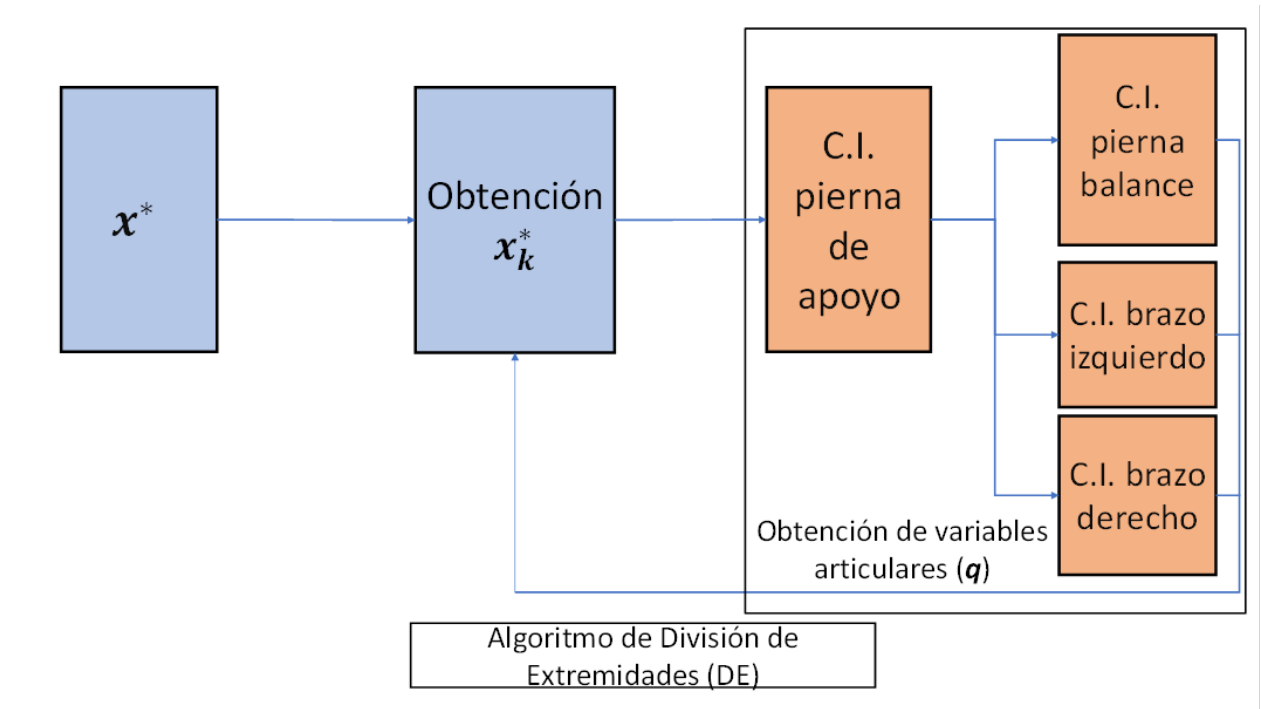


Figura 3.2. Algoritmo de División por Extremidades (DE).

3.2.2. Manipulación Simultánea de masas principales (MS)

Esta estrategia realiza un cálculo simultáneo de la configuración articular \mathbf{q} del robot para posicionar las masas principales m_k , siendo el equivalente a realizar cinco tareas de posicionamiento paralelas. El algoritmo Manipulación Simultanea (MS) se diferencia del algoritmo DE en los pasos 2 y 3, ya que, en lugar de hacer un cálculo independiente de la configuración articular de las extremidades, MS Posiciona todas las masas principales en un solo proceso de optimización. Para este algoritmo es necesario encontrar los cinco jacobianos \mathbf{J}_k que permitan manipular la posición de las 5 masas principales m_k . Sin embargo, en este caso se deben de tomar en cuenta todos los DOF involucrados en sus desplazamientos, iniciando en el pie que se encuentra apoyado en el suelo. El tamaño de las matrices \mathbf{J}_k será de 3×18 cada una y al acoplarlas se obtendrá un jacobiano aumentado \mathbf{J}_s de dimensiones 15×18 .

La configuración articular se obtiene realizando varias iteraciones de la ecuación (3.17)

$$\Delta \mathbf{q} = \alpha \mathbf{J}_s^\dagger \Delta \mathbf{x}_K \tag{3.17}$$

$$\mathbf{J}_s = \begin{pmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_5 \end{pmatrix} \quad (3.18)$$

donde \mathbf{J}_s es un jacobiano aumentado compuesto por los jacobianos que describen el cambio de los cinco vectores \mathbf{x}_k respecto a \mathbf{q} , $\Delta\mathbf{x}_K$ es un vector que contiene los errores entre \mathbf{x}_k y \mathbf{x}_k^* y $\Delta\mathbf{q}$ es el vector de incrementos para las variables articulares. En la figura 3.3 se puede observar el diagrama esquemático de la metodología.

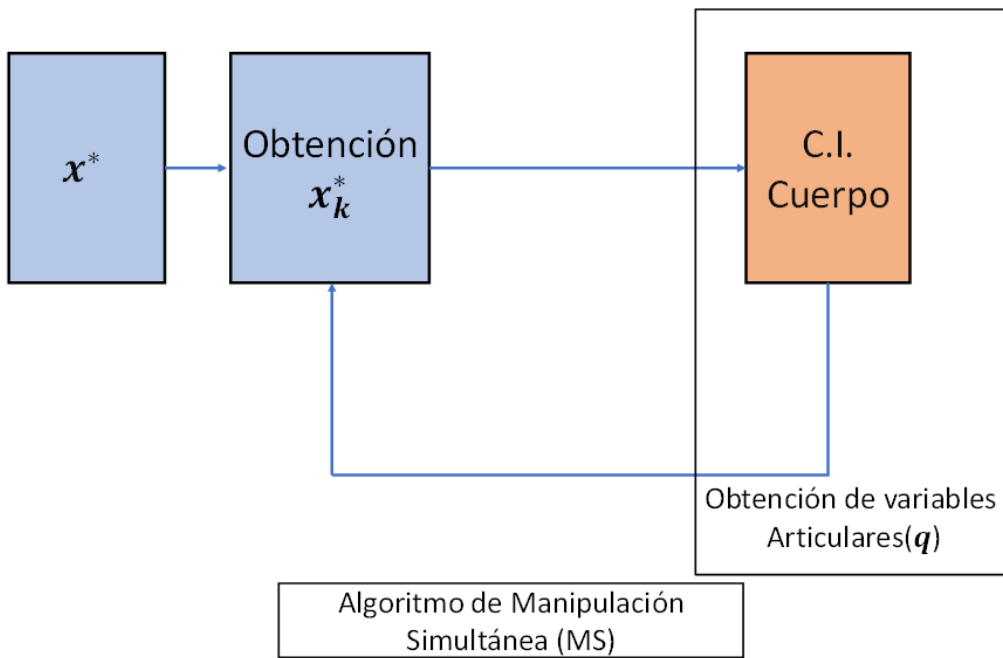


Figura 3.3. Algoritmo de Manipulación Simultánea.

3.2.3. Manipulación del Centro de Masa Total (MCMT)

Otra de las alternativas para la obtención de una configuración articular \mathbf{q} que satisfaga a \mathbf{x}^* , es utilizar el algoritmo de cinemática inversa del COM mencionado en la sección 1.5.5 aplicado directamente al COM \mathbf{x} . En este caso se obtiene un jacobiano \mathbf{J} de tamaño 3×18 que representa la variación de \mathbf{x} respecto a \mathbf{q} . Dado que en un humanoide los DOF suelen ser mucho mayores a 3, generalmente se observa un alto grado de redundancia en el algoritmo. La redundancia permitirá encontrar infinitas soluciones a \mathbf{x}^* , sin embargo, debido a la gran cantidad de DOF del robot, el algoritmo de cinemática inversa del COM se torna inestable.

Como alternativa para mejorar la convergencia, se utiliza una variante a la pseudo-inversa descrita en la sección 1.6.3, donde se añade una matriz de pesos \mathbf{W} para limitar el movimiento de las juntas que tienen un mayor efecto en el vector \mathbf{x} y en combinación con valores bajos en el factor de amortiguamiento λ .

Manipulación del Centro de Masa Total (MCMT)

1. Se calcula $\Delta\mathbf{x}$.
2. Se aplica obtiene los incrementos a las variables articulares por medio de la ecuación (1.34) mostrada en la sección 1.5.5.
3. Se actualiza el vector de variables articulares:

$$\hat{\mathbf{q}} = \mathbf{q} + \Delta\mathbf{q} \quad (3.19)$$

donde $\hat{\mathbf{q}}$ es el vector de variables articulares actualizado.

4. Se recalcula $\Delta\mathbf{x}$:
 - Si $\Delta\mathbf{x}$ es suficientemente pequeño

$$\mathbf{q}^* = \mathbf{q} \quad (3.20)$$

donde \mathbf{q}^* representa la configuración articular que cumple con \mathbf{x}^* . El proceso se finaliza.

- Si $\Delta\mathbf{x}$ no es suficientemente pequeño se repite el algoritmo desde el paso 2 utilizando la configuración articular actualizada.

En la figura 3.4 se puede observar el diagrama esquemático del algoritmo.

3.3. Pruebas de rendimiento

Se realizaron varias simulaciones utilizando un modelo simplificado del robot humanoide Bioloid Premium de 18 DOF, para verificar la precisión y tiempo de ejecución de las metodologías desarrolladas durante este capítulo. Se utilizó la variación de la pseudo inversa descrita en la sección 1.6.3, la cual permite añadir un valor de amortiguamiento λ en busca de mejorar la convergencia y una matriz de pesos \mathbf{W} para dar rigidez a ciertas juntas. Se

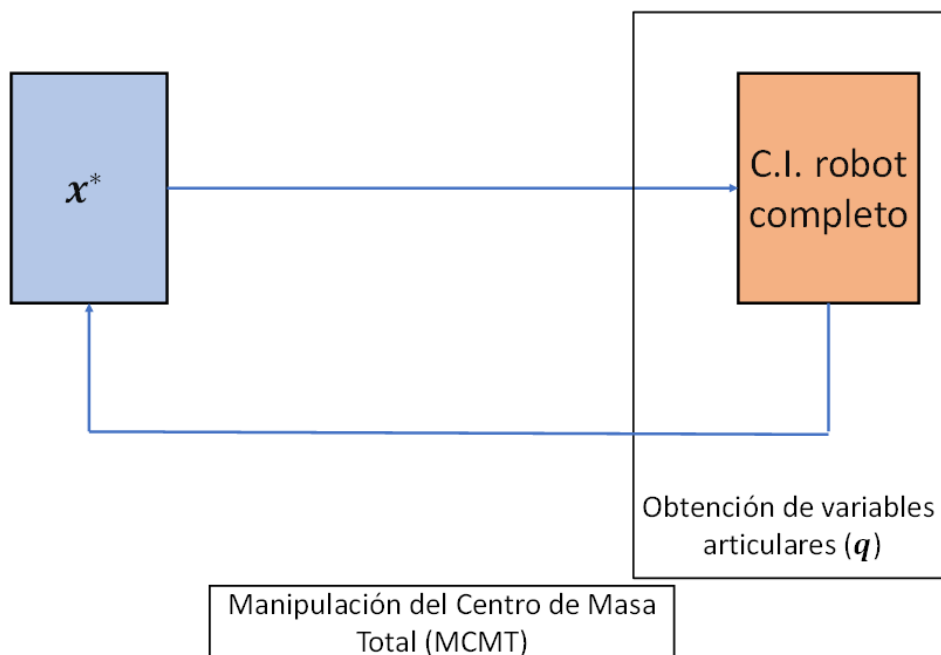


Figura 3.4. Algoritmo de Manipulación del Centro de Masa Total.

efectuaron seis simulaciones para cada método con diversos valores de las constantes α , la cual ayuda a dar estabilidad numérica al algoritmo, y el factor de amortiguamiento λ . Los valores utilizados de α y λ durante las simulaciones se muestran en la tabla 3.3.1.

Tabla 3.3.1. Valores de las constantes λ y α usados en las simulaciones de los algoritmos de posicionamiento del COM.

Simulación	Parámetros	
	λ	α
sim 1	0.1	0.1
sim 2	0.1	0.5
sim 3	0.1	1
sim 4	0.5	1
sim 5	0.5	0.5
sim 6	0.5	0.1

Además, se añadió una variación a los métodos DE y MS, llamados División por Extremidades con Amortiguamiento Variable (DEAV) y Manipulación Simultánea con Amortiguamiento Variable (MSAV) donde el valor de λ se modifica durante el algoritmo

para acelerar la convergencia. El valor de λ se reducirá a la mitad si $\Delta\mathbf{x}$ disminuye en la iteración, en cambio el valor de λ se duplicará si $\Delta\mathbf{x}$ aumenta, además de eliminar la iteración. La matriz de pesos \mathbf{W} permaneció constante en todas las simulaciones y se muestra en el anexo B. Los errores de posicionamiento obtenidos se muestran en la tabla 3.3.2 y los tiempos de ejecución en la tabla 3.3.3.

Tabla 3.3.2. Errores de posicionamiento(mm) de las simulaciones de los algoritmos de posicionamiento del COM.

Método	sim1	sim2	sim3	sim4	sim5	sim6
DE	0.0200	0.0153	0.0092	0.0166	0.0094	0.1510
DEAV	0.2134	0.0800	0.1333	0.0047	0.1679	0.2100
MS	0.1601	0.1508	0.0063	0.0069	0.0527	0.1203
MSAV	0.1039	0.1084	0.1110	0.1112	0.1120	0.1079
MCTM	0.0005	3.0664	4.4929	4.3493	2.6956	0.0005

Tabla 3.3.3. Tiempos de ejecución (ms) de las simulaciones de los algoritmos de posicionamiento del COM.

Método	sim1	sim2	sim3	sim4	sim5	sim6
DE	21.3	129.4	50.1	136.4	105.5	93.9
DEAV	86.3	40.3	73.9	73.9	83.1	51.5
MS	141.5	107.9	376.1	133.5	146.3	148.6
MSAV	105.8	109.8	90.7	89.3	115.8	143.8
MCTM	31.6	61.0	70.4	72.1	61.6	25.9

Como se puede observar en la figura 3.6 la simulación con un mayor tiempo de ejecución fue la realizada para MS con valores de $\lambda = 0.1$ y $\alpha = 1$. Además, MS también es el algoritmo con mayor tiempo promedio de ejecución con 175.6 ms. El algoritmo que presentó mayor error promedio es Manipulación del Centro de Masa Total (MCMT) con 2.4342 mm, sin embargo, este último también es el algoritmo con un tiempo de ejecución promedio más bajo. El algoritmo DE presentó el tiempo de ejecución más bajo usando los valores de $\lambda = 0.1$ y $\alpha = 0.1$.

Los mejores resultados considerando el tiempo de ejecución y el error obtenido son para la simulación 1 y 6 del algoritmo MCMT. En estas simulaciones el valor de $\alpha = 0.1$ coincide, por lo que se asume que MCMT tiene un mejor comportamiento con valores pequeños de α .

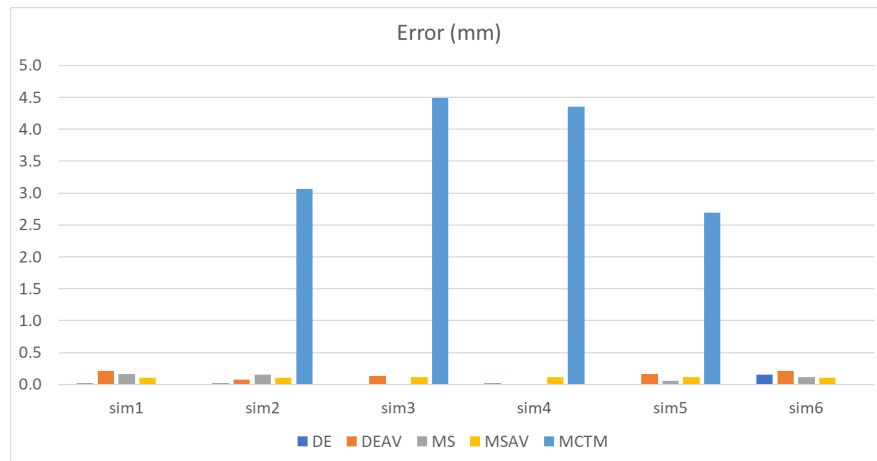


Figura 3.5. Errores en simulación de los algoritmos de posicionamiento del COM.

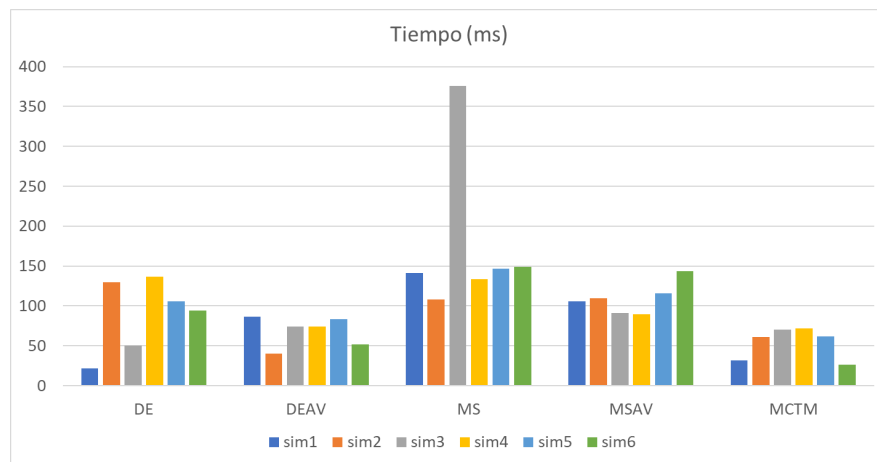


Figura 3.6. Tiempos de ejecución resultantes durante las simulaciones de los algoritmos de posicionamiento del COM.

Otro de los aspectos a considerar, son las poses obtenidas por los algoritmos. Debido a que no existe una limitación o tarea específica para las extremidades, éstas pueden adoptar posturas poco naturales cuando se compara con las de un ser humano. En las figuras 3.7, 3.8 y 3.9, se observan poses obtenidas de las simulaciones con una configuración articular deseada para los propósitos de este trabajo.

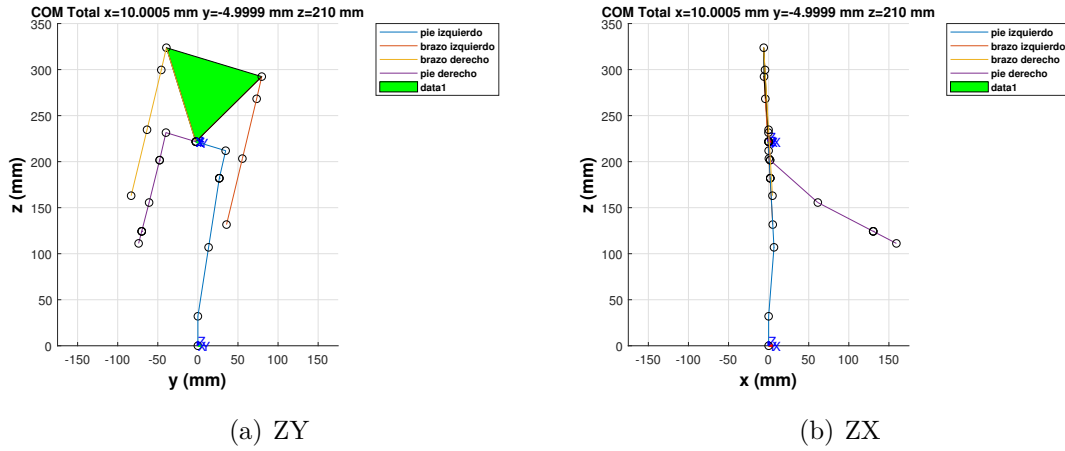


Figura 3.7. Poses obtenidas con el algoritmo MCMT y valores de $\lambda = 0.1$ y $\alpha = 0.1$.

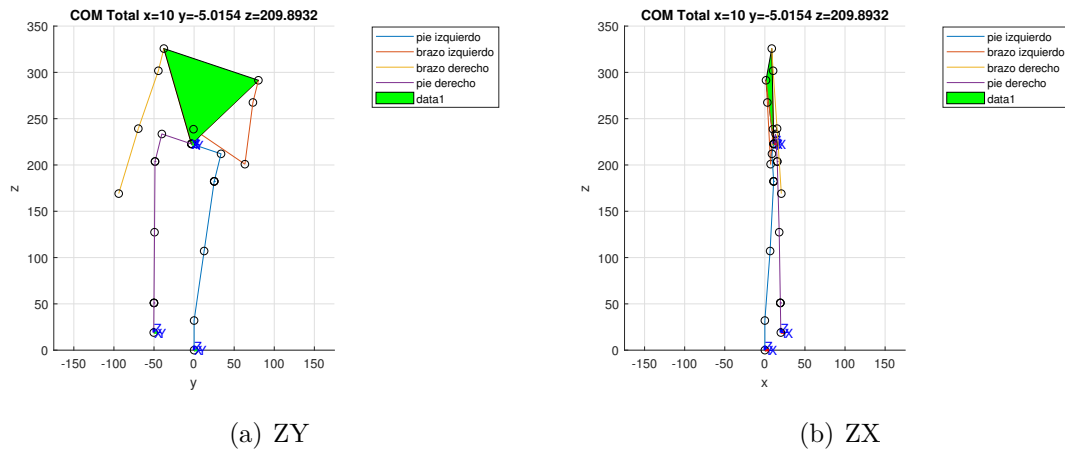


Figura 3.8. Poses obtenidas con el algoritmo MSAV y valores de $\lambda = 0.5$ y $\alpha = 0.1$.

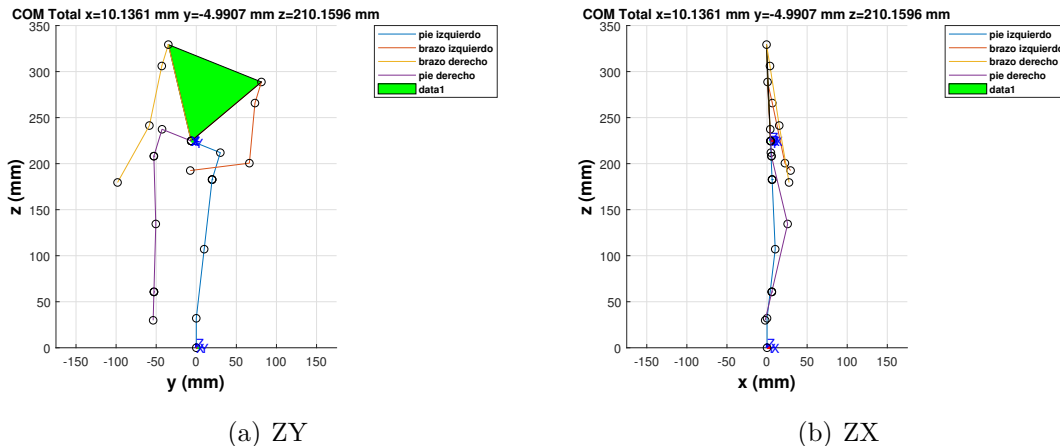


Figura 3.9. Poses obtenidas con el algoritmo DEAV y valores de $\lambda = 0.5$ y $\alpha = 0.1$.

3.4. Conclusiones del capítulo

Se propusieron y validaron en simulación tres algoritmos DE, MS y MCMT para determinar poses estáticas de un robot humanoide de 18 DOF y ubicar su COM en una posición deseada.

Los algoritmos DE y MS dividen el cuerpo en cinco cadenas cinemáticas (4 extremidades y el torso), donde dada una posición deseada del COM del humanoide, se determinan las posiciones deseadas de las masas principales asociadas a las cadenas cinemáticas consideradas. Posteriormente se calcula la configuración articular que cumpla con dicho posicionamiento. El algoritmo MCMT posiciona el COM del robot por medio del algoritmo de CmICOM introduciendo una un factor de amortiguamiento λ y una matriz de pesos \mathbf{W} en la pseudo-inversa del jacobiano que mejorar el rendimiento del algoritmo tanto en tiempo de ejecución como en precisión.

Se realizaron pruebas de rendimiento a los algoritmos para verificar el error de posicionamiento del COM, su tiempo de ejecución y las poses resultantes obtenidas a partir de los algoritmos. Los algoritmos propuestos tuvieron un buen desempeño en términos de posicionamiento del COM respecto al COM objetivo obteniendo en su mayoría errores en el orden de décimas y centésimas de mm. Los tiempos de convergencia de los algoritmos propuestos no son aptos para su implementación en línea en un robot humanoide, tomando en cuenta que además se añadirá una tarea complementaria, como el caminado en baja

velocidad. El algoritmo con mejores resultados tanto en posicionamiento y tiempo de ciclo es **MCMT** con valores de $\lambda = 0.1$ y $\alpha = 0.1$. La variación de los algoritmos **DEAV** y **MSAV** también presentaron un comportamiento aceptable superando a sus versiones originales **DE** y **MS** en cuanto a tiempo promedio de ejecución y las poses obtenidas.

Aplicación de control de centro de masa a tareas de caminado de un robot humanoide

Una de las tareas más complejas de un humanoide es caminar. La principal problemática, es mantener el equilibrio mientras el robot camina, por lo que se implementará el control del centro de masa presentado en el capítulo anterior en busca de obtener trayectorias de marcha estables y realizables.

Además, el robot debe ejecutar múltiples tareas mientras realiza la marcha como mover el pie y mantener el equilibrio, lo que aumenta la dificultad del movimiento.

4.1. Generación de trayectorias de marcha

La generación de trayectorias de marcha se enfoca en dos objetivos principales: el posicionamiento del **COM** y el posicionamiento del pie en vuelo. Las trayectorias para estos dos objetivos dependerán de las características del terreno donde se llevará a cabo la marcha y las capacidades físicas del robot en cuestión.

Existen diversas condiciones de terreno donde un humanoide puede desplazarse, como: superficie plana, superficie con pendiente constante, escaleras y terreno irregular. El **COM** debe de adaptar su desplazamiento dependiendo del terreno, por ejemplo, en una superficie plana bastará con que el **COM** se desplace a una altura constante, en cambio en una escalera, el **COM** se tendrá que desplazar con una pendiente.

Enseguida, se abordarán dos algoritmos para la obtención de las trayectorias del **COM** y el pie en desplazamiento.

4.1.1. Generación de trayectoria por *splines* cúbicas

Una de las opciones es el uso de *splines* cúbicas, para crear una trayectoria por donde el COM y el pie en vuelo se muevan. En este caso se busca que la trayectoria del COM pase en todo momento por el polígono de apoyo. Se asumirá que la superficie por donde se camina es plana y que el COM permanecerá en una misma altura durante todo el trayecto. A continuación, se presentan los pasos para generar la trayectoria:

Generación de trayectoria por *splines* cúbicas

1. Se determina la posición inicial, la posición final y la altura de movimiento del COM.
2. Se determina la posición inicial, la posición final y la máxima altura de movimiento del pie que ejecutará el desplazamiento.
3. Se discretiza el trayecto del COM y el pie en movimiento en al menos tres puntos.
4. Por medio de *splines* cúbicas se crean puntos intermedios para las trayectorias del COM y el pie en vuelo.
5. Se verifica que la trayectoria del COM permanezca siempre sobre el polígono de apoyo.

Una vez obtenidas las trayectorias del COM y el pie en vuelo se procederá a obtener la configuración articular del robot que cumpla con estos objetivos.

4.1.2. Generación de trayectoria por Péndulo Invertido Lineal 3D

La generación de trayectorias por Péndulo Invertido Lineal 3D descrito en la sección 1.8.1, permite determinar la trayectoria a seguir del COM basándonos en las ecuaciones dinámicas de un péndulo invertido. Al utilizar este modelo no es posible asegurar la estabilidad estática. Sin embargo, debido a la simplificación que se genera, es posible conseguir trayectorias que cumplan con los criterios de estabilidad dinámica. A continuación, se presenta el procedimiento para generar la trayectoria:

Generación de trayectoria por Péndulo Invertido Lineal 3D

1. Se determina la posición inicial $x(0)$ y $y(0)$ del péndulo, así como, la altura z_c y el tiempo de ciclo t_c .
-

2. Se calculan las velocidades iniciales $\dot{x}(0)$ y $\dot{y}(0)$ para cumplir t_c .
3. Se generan las trayectorias en posición a partir de las ecuaciones (1.65) y (1.67), así como, las trayectorias en velocidad con las ecuaciones (1.66) y (1.68).
4. Se determina la posición inicial, la posición final y la máxima altura de movimiento del pie que ejecutará el desplazamiento.
5. Se discretiza el trayecto del pie en movimiento en al menos tres puntos.
6. Por medio de *splines* cúbicas se crean puntos intermedios para la trayectoria del pie en vuelo.

De la misma forma que con la generación de trayectorias por *splines* cúbicas, la configuración articular que cumpla con los objetivos se calcula en un proceso posterior.

4.2. Definición de tareas

Para ejecutar la marcha es necesario ejecutar diversas tareas durante el movimiento. En este caso llevaremos a cabo una tarea de posicionamiento del COM, el posicionamiento del pie que se encuentra en vuelo y su orientación, y por último la orientación del torso

4.2.1. Posicionamiento del centro de masa

una tarea de posicionamiento se puede considerar como la regulación a cero de una función Delta de una variable, por ejemplo, en este caso la tarea de posicionamiento del COM \mathbf{x} respecto al marco de referencia colocado en la planta del pie fijo se define como:

$$\Delta \mathbf{x} = \mathbf{x}^* - \mathbf{x} \quad (4.1)$$

donde \mathbf{x}^* es la posición deseada del COM y $\Delta \mathbf{x}$ de dimensiones 3×1 es el error entre la posición deseada y su posición actual. El jacobiano asociado \mathbf{J} se obtiene a partir de la cinemática directa del centro de masa, mencionada en la sección 1.5.4.

4.2.2. Posicionamiento y orientación del pie en vuelo

La tarea de posicionamiento del pie en vuelo respecto a el marco de referencia colocado en la planta del pie fijo $\Delta \mathbf{x}_f$ de dimensiones 3×1 está dada por

$$\Delta \mathbf{x}_f = \mathbf{x}_f^* - \mathbf{x}_f \quad (4.2)$$

donde \mathbf{x}_f^* es la posición deseada del pie en vuelo y \mathbf{x}_f es su posición actual. El jacobiano asociado \mathbf{J}_f es obtenido a partir de la cinemática directa de la pierna suspendida respecto al marco de referencia colocado en la planta del pie fijo.

La tarea de orientación del pie en vuelo respecto al marco de referencia de la planta del pie fijo $\Delta \mathbf{x}_{of}$ con dimensiones 2×1 está dada por

$$\Delta \mathbf{x}_{of} = \begin{bmatrix} 1 - \hat{\mathbf{k}} \cdot \hat{\mathbf{k}}_f \\ 1 - \hat{\mathbf{i}} \cdot \hat{\mathbf{i}}_f \end{bmatrix} \quad (4.3)$$

donde $\hat{\mathbf{k}}$ e $\hat{\mathbf{i}}$ son los vectores unitarios asociados al marco colocado en la planta del pie fijo, y $\hat{\mathbf{k}}_f$ e $\hat{\mathbf{i}}_f$ los vectores unitarios asociados al marco de referencia del pie en vuelo. El jacobiano \mathbf{J}_{of} se obtiene derivando parcialmente respecto a las variables articulares los productos $\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}_f$ e $\hat{\mathbf{i}} \cdot \hat{\mathbf{i}}_f$.

4.2.3. Orientación del torso

La tarea de orientación del torso respecto a el marco de referencia colocado en la planta del pie fijo $\Delta \mathbf{x}_{ot}$ de dimensiones 1×1 está dada por

$$\Delta \mathbf{x}_{ot} = 1 - \hat{\mathbf{k}} \cdot \hat{\mathbf{k}}_t \quad (4.4)$$

donde $\hat{\mathbf{k}}$ es el vector unitario asociado al marco de referencia colocado en la planta del pie fijo, y $\hat{\mathbf{k}}_t$ es el vector unitario asociado al marco de referencia de la cadera. El Jacobiano \mathbf{J}_{ot} se obtiene derivando parcialmente respecto a las variables articulares el producto $\hat{\mathbf{k}} \cdot \hat{\mathbf{k}}_t$.

4.3. Marcha

Para la ejecución de la marcha, en general se propusieron dos metodologías con diversas variaciones:

1. Múltiples tareas como Vector Aumentado de Tareas (VAT). En esta estrategia se apilan los vectores de las tareas dentro de un vector y un jacobiano asociado.
2. Múltiples tareas como Control Jerárquico (CJ).

estas dos metodologías permiten controlar múltiples tareas o múltiples partes del robot de manera simultánea mediante el método iterativo de mínimos cuadrados. Enseguida se detallan las dos metodologías.

4.3.1. Múltiples tareas como vector aumentado de tareas

En este caso se construye un vector aumentado de tareas $\Delta \mathbf{x}_A$, que consiste en apilar dentro de un mismo vector los vectores de las tareas individuales, junto con un jacobiano asociado \mathbf{J}_A que se construye apilando los jacobianos de las tareas. Las ventajas de la metodología VAT, es que todas las tareas apiladas en el vector aumentado se controlan con una misma prioridad buscando el mejor resultado posible. En este caso el vector $\Delta \mathbf{x}_A$ de tamaño 9×1 y la matriz \mathbf{J}_A de tamaño 9×18 están definidas como:

$$\Delta \mathbf{x}_A = \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{x}_f \\ \Delta \mathbf{x}_{of} \\ \Delta \mathbf{x}_{ot} \end{bmatrix} \quad (4.5)$$

$$\mathbf{J}_A = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_f \\ \mathbf{J}_{of} \\ \mathbf{J}_{ot} \end{bmatrix} \quad (4.6)$$

ejecutando de forma iterativa la ecuación (4.7) se puede obtener una configuración articular \mathbf{q} que satisfaga las tareas.

$$\Delta \mathbf{q} = \mathbf{J}_A^{\dagger w} \Delta \mathbf{x}_A \quad (4.7)$$

Esto permitirá controlar la posición del centro de masa, la posición y orientación del pie y la orientación vertical del torso.

A continuación, se muestra una secuencia de poses obtenidas mediante esta metodología para una trayectoria del pie y COM en un robot de 18 DOF que cumple con los criterios de estabilidad estática. En la pose 1 se observa el comienzo del movimiento, durante la figura 10, 20 y 30 el robot carga su torso sobre la superficie del pie izquierdo, a partir de la pose 40 el robot inicia el desplazamiento del pie en vuelo en la dirección x positiva, final mente el la pose 100 termina el movimiento.

En las figuras 4.2 y 4.3 se puede observar las trayectorias de referencia para el COM y el pie en vuelo, y los resultados obtenidos. Como se puede observar en las figuras el desplazamiento del pie y del COM del robot cumplen el trayecto sin inconveniente.

Como se ve en las figuras 4.4 y 4.5, los errores obtenidos en el posicionamiento del pie y posicionamiento del COM respecto a sus objetivos están en el orden de las centésimas de milímetro, dejando claro la efectividad de la metodología. También se observan oscilaciones en las dos figuras en el intervalo de poses 1 a 20 las cuales corresponden a la sección de movimiento en donde los dos pies están en contacto con el suelo.

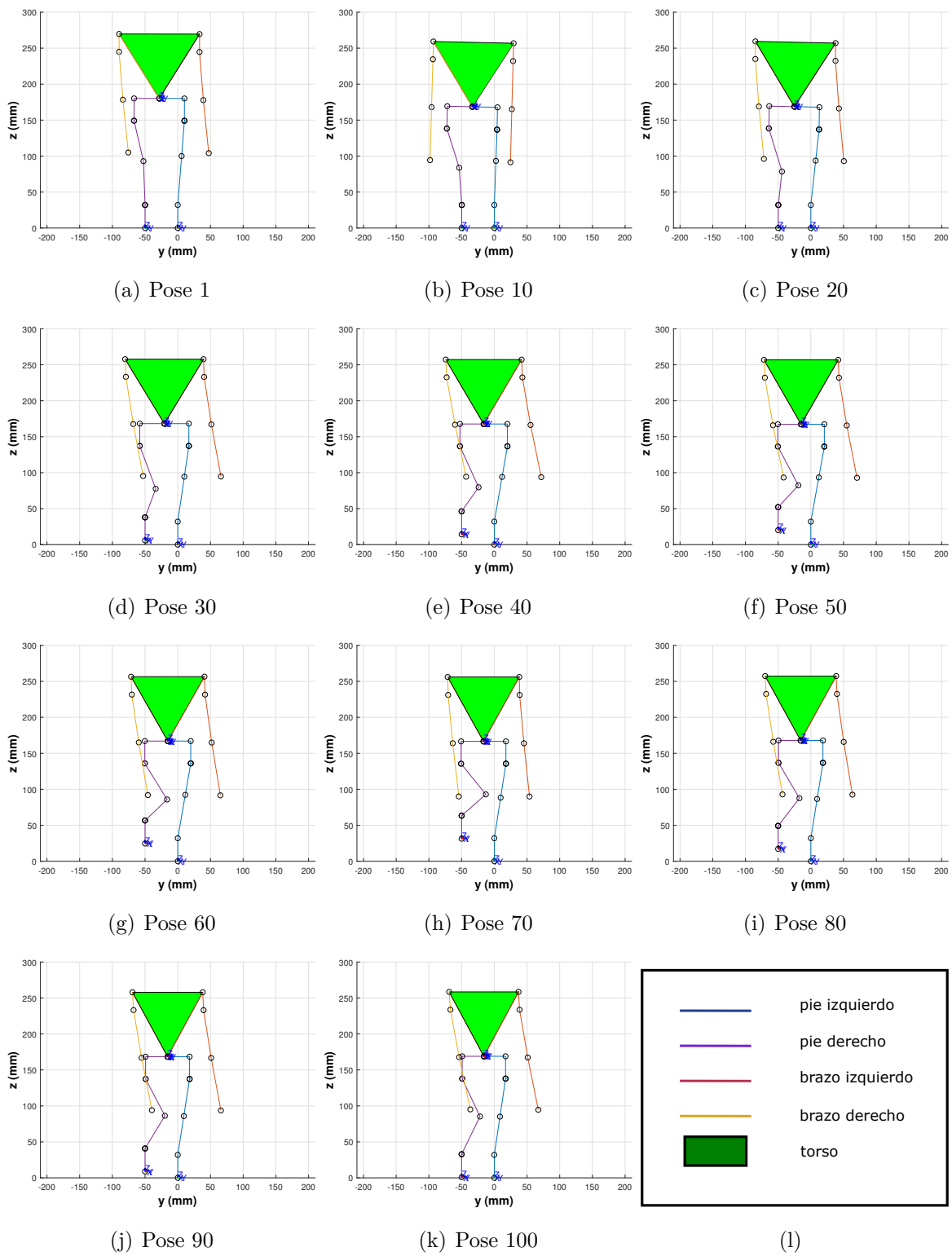


Figura 4.1. Secuencia de movimientos generados para un paso usando la estrategia de VAT.

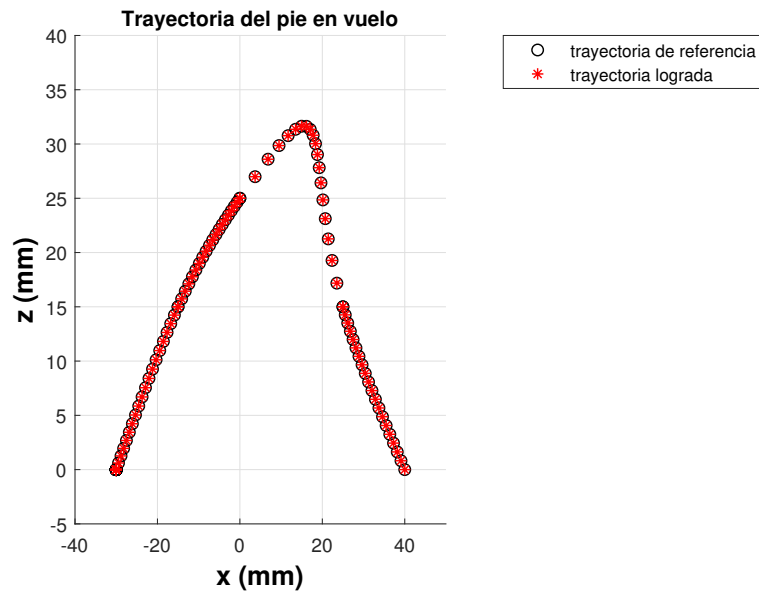


Figura 4.2. Trayectoria de pie en vuelo para VAT.

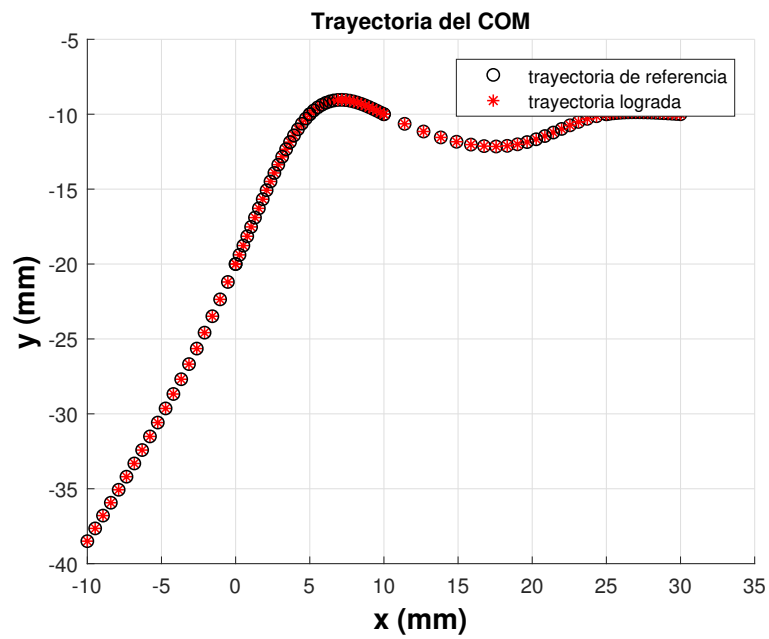


Figura 4.3. Trayectoria del COM para VAT.

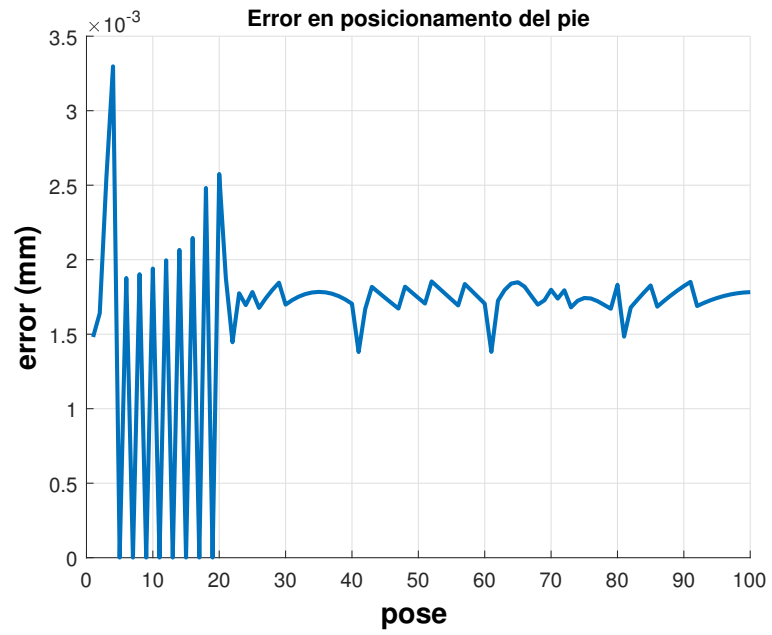


Figura 4.4. Error de posicionamiento de pie en vuelo para VAT.

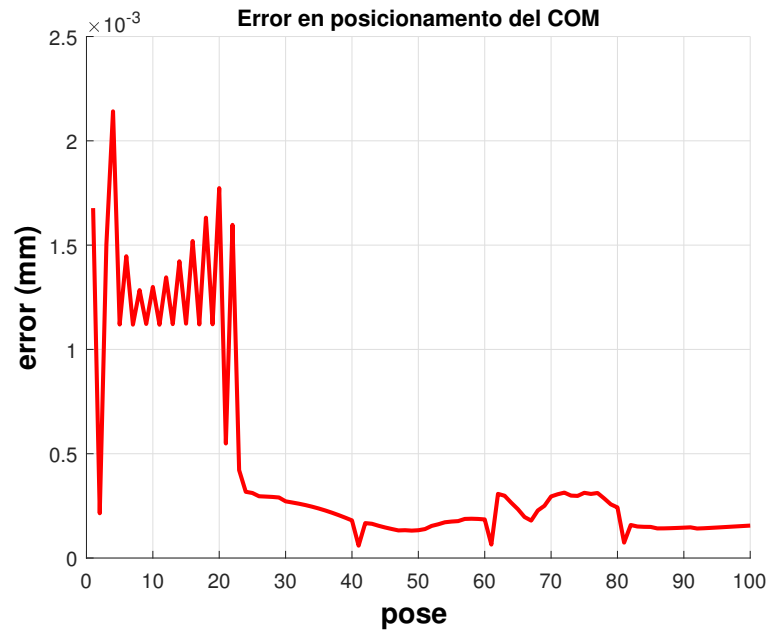


Figura 4.5. Error de posicionamiento del COM para VAT.

4.3.2. Múltiples tareas como control jerárquico

El segundo método propuesto para generar las trayectorias de marcha es **CJ**. Este esquema aplica solo para robots redundantes, ya que debido a su exceso de **DOF** es posible definir prioridades para las tareas a realizar. El **CJ** se asigna una tarea primaria con un jacobiano asociado, dada la redundancia del sistema el espacio nulo del jacobiano no es trivial y existirán vectores que se puedan proyectar sobre él que no afecten el comportamiento de la tarea primaria. Los vectores proyectados sobre el espacio nulos son considerados como una tarea de prioridad secundaria. La ecuación que rige la metodología **CJ** está dada por

$$\Delta \mathbf{q} = \mathbf{J}_A^{\dagger w} \Delta \mathbf{x}_A - (\mathbf{I} - \mathbf{J}_A^{\dagger w} \mathbf{J}_A) \Delta \mathbf{q}_s \quad (4.8)$$

donde $(\mathbf{I} - \mathbf{J}_A^{\dagger w} \mathbf{J}_A)$ es el espacio nulo de \mathbf{J}_A y $\Delta \mathbf{q}_s$ es un comportamiento secundario en las variables articulares. El **CJ** puede extenderse a más de 2 prioridades; en el caso de una tarea terciaria ésta se proyectará sobre el espacio nulo del jacobiano de la tarea secundaria.

4.3.2.1. Prioridad primaria a posicionamiento y orientación del pie en vuelo y orientación de cadera

En este caso el vector $\Delta \mathbf{x}_A$ con dimensiones 6×1 , estará conformado por los vectores de tarea $\Delta \mathbf{x}_f$, $\Delta \mathbf{x}_{of}$ y $\Delta \mathbf{x}_{ot}$ definidos en la sección 4.2. $\Delta \mathbf{x}_A$ se muestra en la ecuación (4.9) y su jacobiano \mathbf{J}_A con dimensiones 6×18 se muestra en la ecuación (4.10)

$$\Delta \mathbf{x}_A = \begin{bmatrix} \Delta \mathbf{x}_f \\ \Delta \mathbf{x}_{of} \\ \Delta \mathbf{x}_{ot} \end{bmatrix} \quad (4.9)$$

$$\mathbf{J}_A = \begin{bmatrix} \mathbf{J}_f \\ \mathbf{J}_{of} \\ \mathbf{J}_{ot} \end{bmatrix} \quad (4.10)$$

y $\Delta \mathbf{q}_s$ es el vector de variables articulares que posiciona el **COM** y se obtiene mediante la ecuación (4.11)

$$\Delta \mathbf{q}_s = \mathbf{J}^{\dagger} \Delta \mathbf{x} \quad (4.11)$$

Utilizando las mismas trayectorias usadas para probar el método de VAT, se evalúa el esquema de CJ con prioridad en el posicionamiento del pie descrito arriba. Las poses obtenidas durante la evaluación muestran debajo. En la pose 1 se observa el comienzo del movimiento, durante la figura 10, 20 y 30 el robot carga su torso sobre la superficie del pie izquierdo, a partir de la pose 40 el robot inicia el desplazamiento del pie en vuelo en la dirección x positiva, final mente la pose 100 termina el movimiento.

En las figuras 4.7 y 4.8 se pueden ver las trayectorias de referencia para el pie en vuelo y el COM, así como, los resultados obtenidos. En la figura 4.8 se puede percibir una menor coincidencia entre la referencia y el resultado, debido a que en ésta se representa la tarea con prioridad secundaria. En la figura 4.8 se percibe que la peor coincidencia durante la trayectoria es en el inicio, cuando los dos pies se mantienen fijos en el suelo.

Los errores obtenidos durante los desplazamientos del pie y el COM se pueden observar en las figuras 4.9 y 4.10. La tarea primaria (posicionamiento de pie en vuelo) obtuvo un error similar al presentado en el algoritmo de VAT. Sin embargo, el error presentado en el posicionamiento del COM crece drásticamente, obteniendo un máximo de 5.3 mm. También se percibe que los errores máximos se presentan en las primeras 20 poses, en las cuales los dos pies del robot están en contacto con el suelo, coincidiendo con el comportamiento observado en el algoritmo VAT.

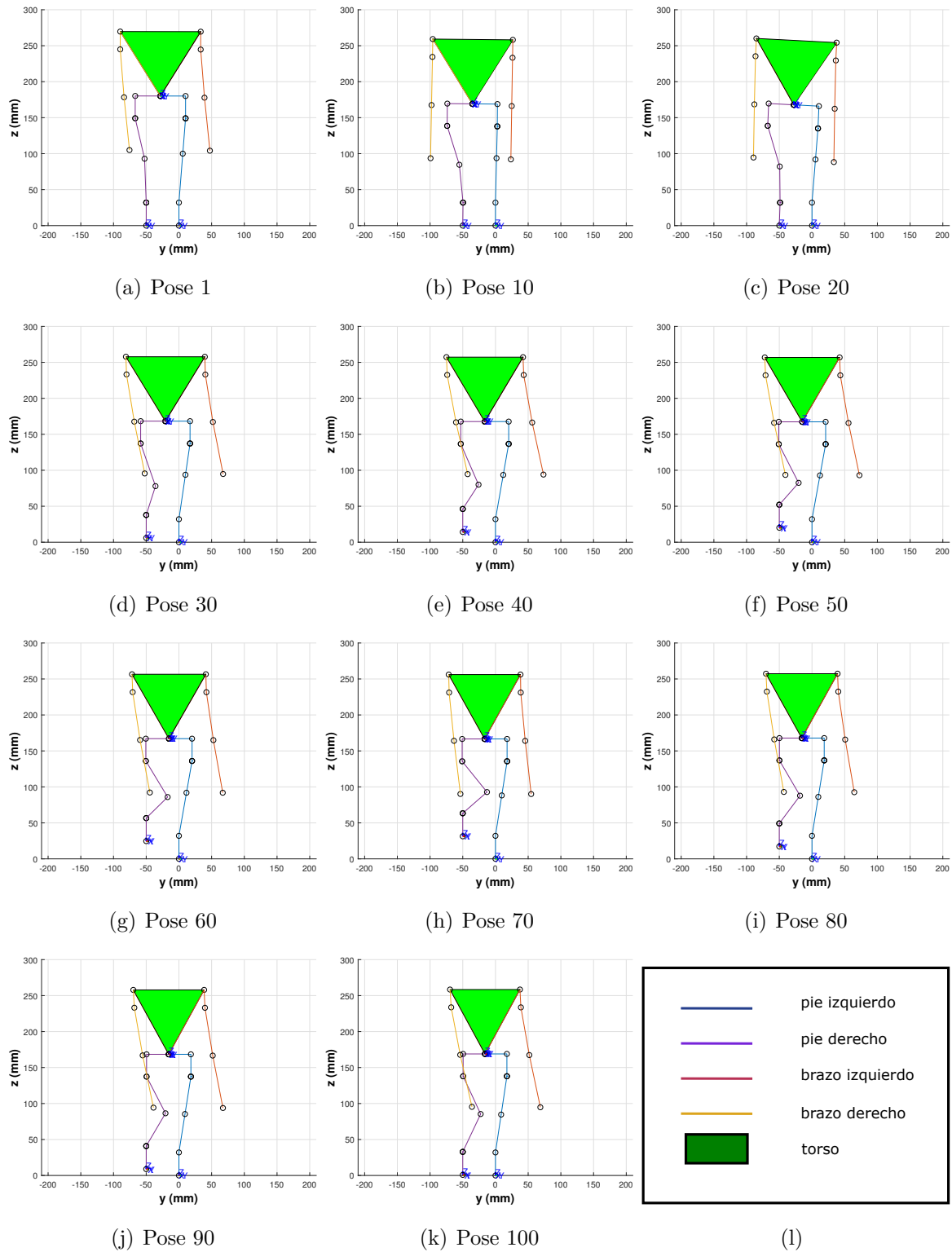


Figura 4.6. Secuencia de movimientos generados para un paso usando la estrategia de CJ en el posicionamiento del pie.

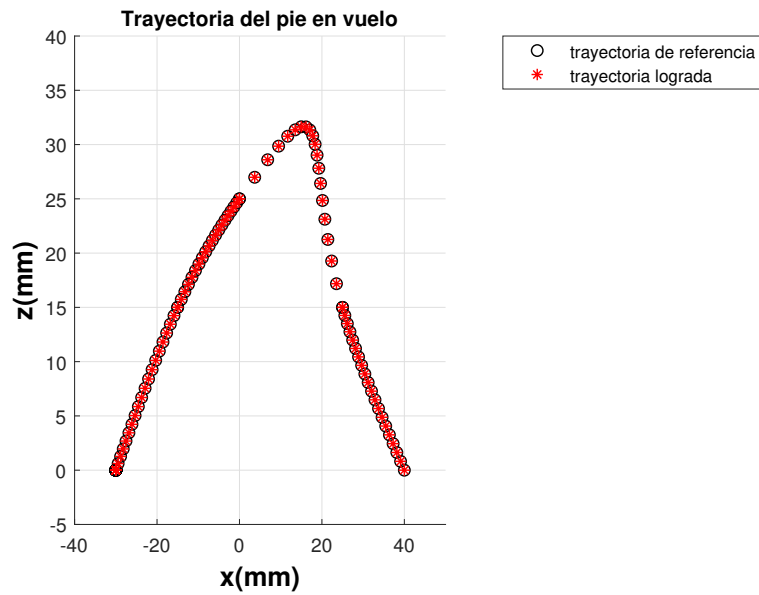


Figura 4.7. Trayectoria de pie en vuelo para CJ con prioridad en el posicionamiento del pie.

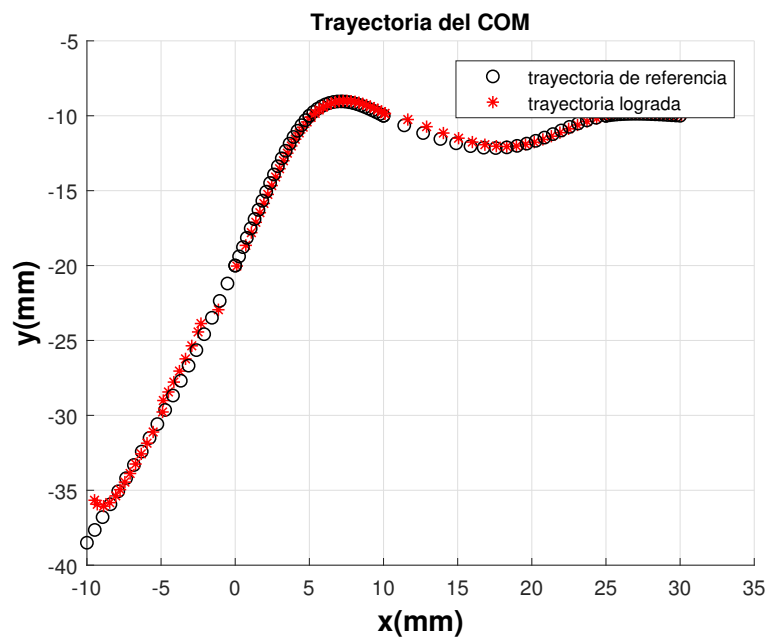


Figura 4.8. Trayectoria del COM para CJ con prioridad en el posicionamiento del pie.

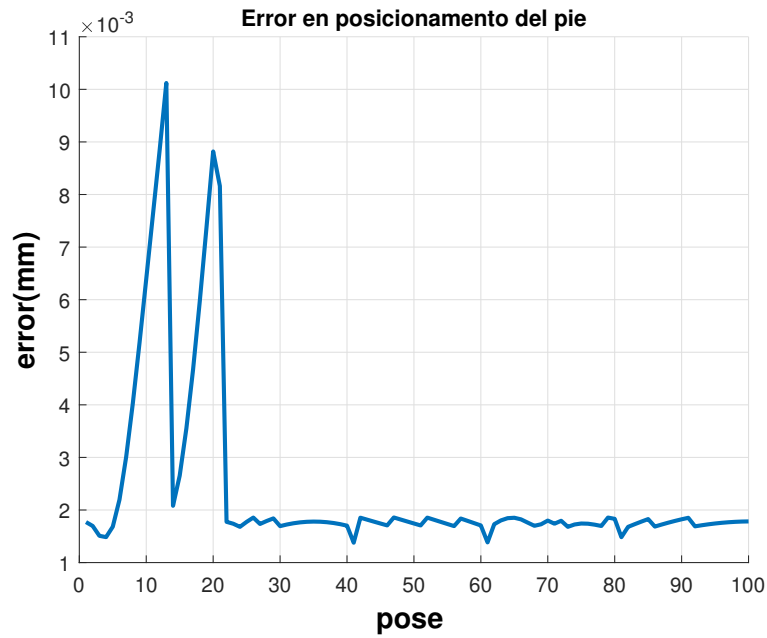


Figura 4.9. Error de posicionamiento de pie en vuelo para CJ con prioridad en el posicionamiento del pie.



Figura 4.10. Error de posicionamiento del COM para CJ con prioridad en el posicionamiento del pie.

4.3.3. Prioridad primaria a posicionamiento del centro de masa, la orientación del pie en vuelo y orientación de cadera

En este caso, intercambiamos la prioridad entre la posición del pie en vuelo y el COM obteniendo un vector de tarea $\Delta \mathbf{x}_A$ de dimensiones 6×1 y su respectivo jacobiano \mathbf{J}_A de dimensiones 6×18 como se muestre en las ecuaciones (4.12) y (4.13)

$$\Delta \mathbf{x}_A = \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{x}_{of} \\ \Delta \mathbf{x}_{ot} \end{bmatrix} \quad (4.12)$$

$$\mathbf{J}_A = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_{of} \\ \mathbf{J}_{ot} \end{bmatrix} \quad (4.13)$$

la configuración articular secundaria $\Delta \mathbf{q}_s$ se obtiene a partir de la ecuación (4.14).

$$\Delta \mathbf{q}_s = \mathbf{J}^\dagger \Delta \mathbf{x} \quad (4.14)$$

En la figura 4.11 se pueden observar poses del humanoide obtenidas a partir de esta variación del algoritmo CJ, utilizando las mismas trayectorias usadas previamente para la evaluación del método VAT. En la pose 1 se observa el comienzo del movimiento, durante la figura 10, 20 y 30 el robot carga su torso sobre la superficie del pie izquierdo, a partir de la pose 40 el robot inicia el desplazamiento del pie en vuelo en la dirección x positiva, finalmente en la pose 100 termina el movimiento.

En las figuras 4.12 y 4.13 se observa las trayectorias de referencia del pie en vuelo y COM, contra las trayectorias obtenidas. Como se observa en la figura 4.12 la trayectoria del pie obtenida tiene una menor coincidencia con la trayectoria de referencia ya que esta tarea es el comportamiento secundario. En la figura 4.13 se observa que la trayectoria obtenida para el COM coincide de forma muy cercana a la referencia.

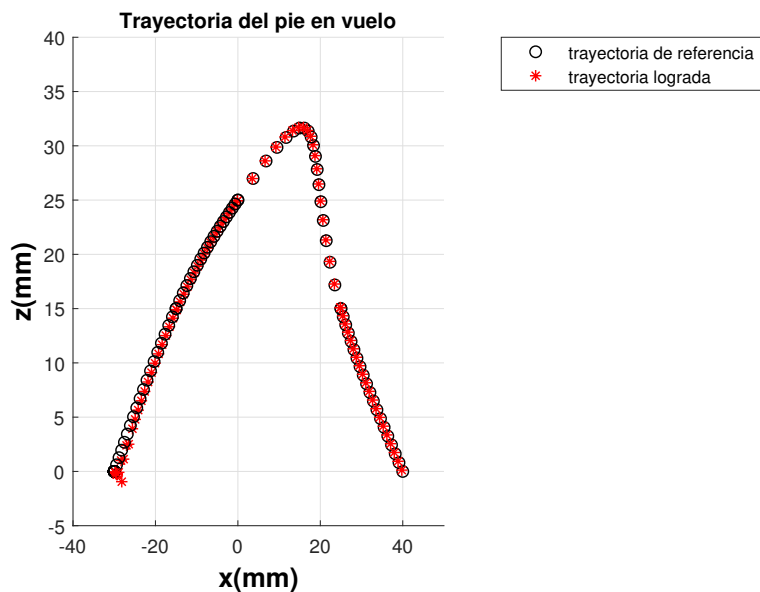


Figura 4.12. Trayectoria de pie en vuelo para CJ con prioridad en el posicionamiento del COM.

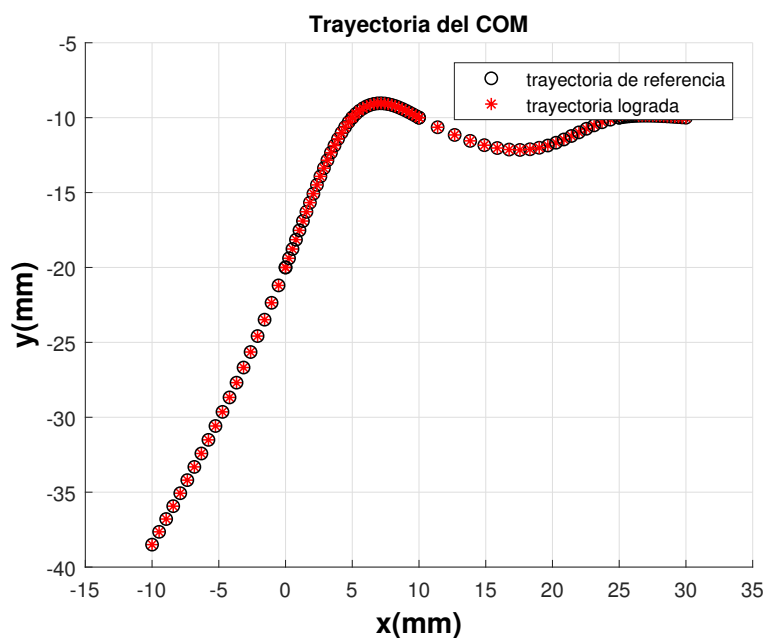


Figura 4.13. Trayectoria del COM para CJ con prioridad en el posicionamiento del COM.

Los errores obtenidos durante las trayectorias se presentan en las figuras 4.14 y 4.15. En este caso se obtiene el mejor comportamiento para el posicionamiento del COM, obteniendo un error máximo de 4.5×10^{-4} mm. Con respecto al posicionamiento del pie, se obtiene un error máximo de 2.3 mm el cual se presenta en el inicio del ciclo. También se percibe un

comportamiento oscilatorio entre las poses 20 y 30 en las cuales el pie en vuelo se despega del suelo.

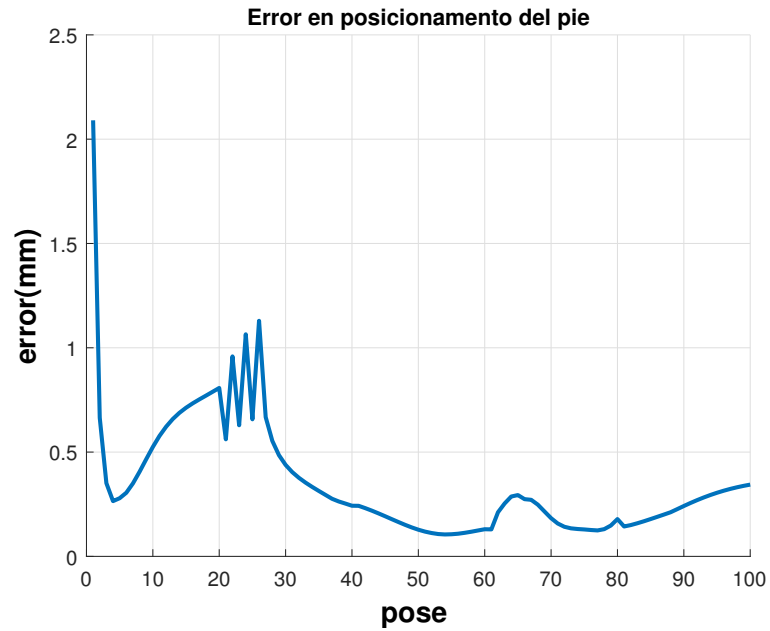


Figura 4.14. Error de posicionamiento de pie en vuelo para CJ con prioridad en el posicionamiento del COM.

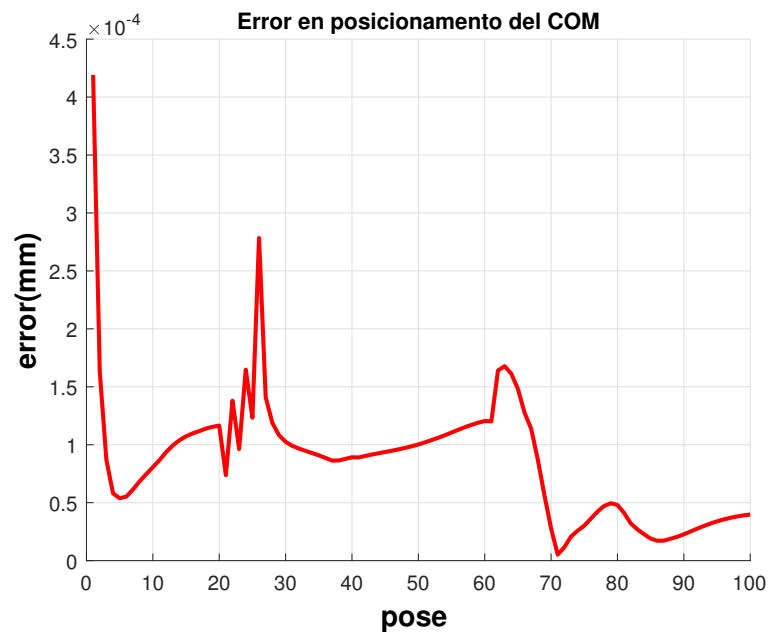


Figura 4.15. Error de posicionamiento del COM para CJ con prioridad en el posicionamiento del COM.

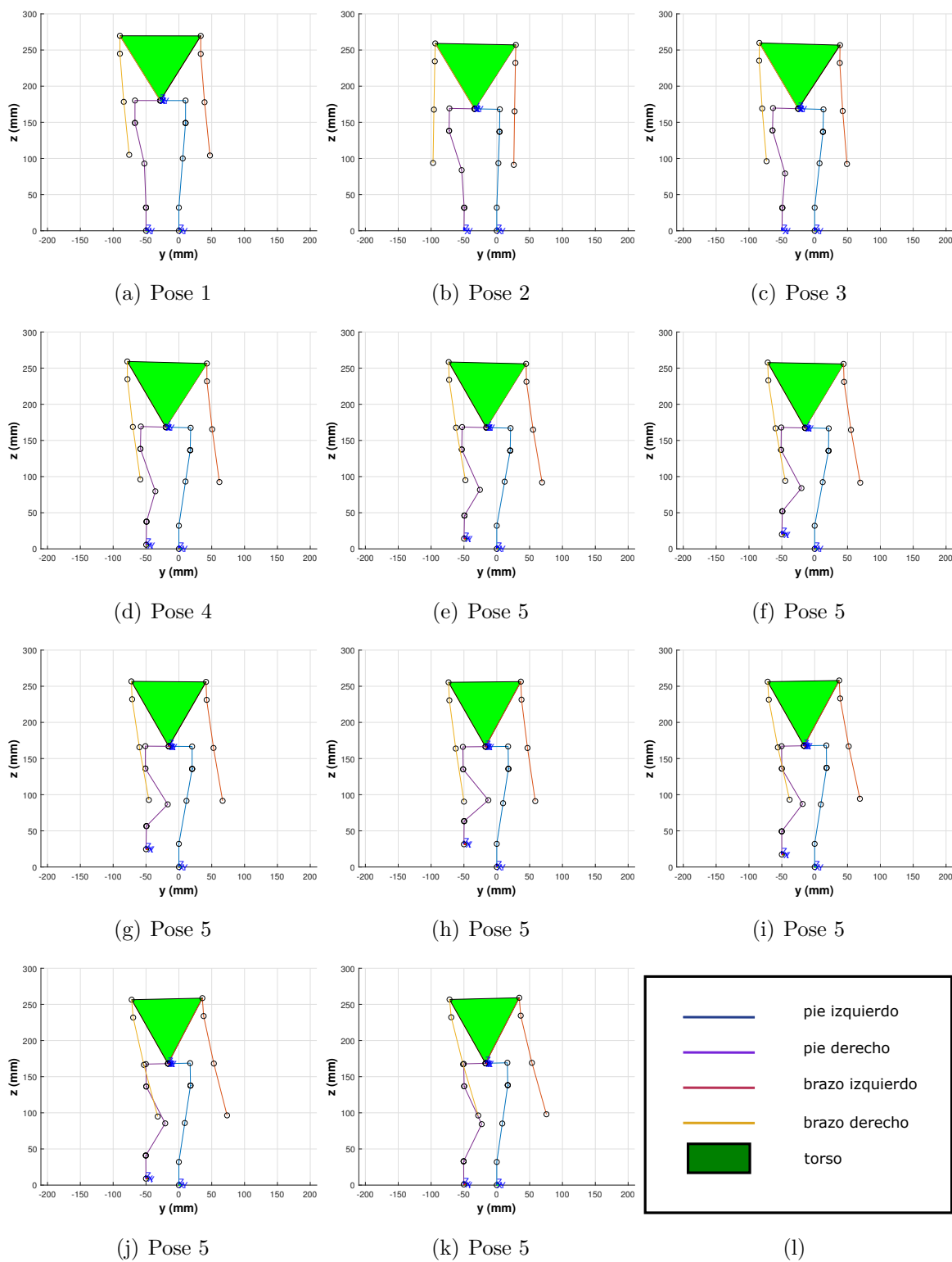


Figura 4.11. Secuencia de movimientos generados para un paso usando la estrategia de CJ con prioridad en el posicionamiento del COM.

4.4. Adición de límite de juntas a algoritmos

Como se menciona en la sección 1.7 uno de los problemas persistentes en los métodos de cinemática inversa de forma iterativa, es la obtención de poses no realizables debido a una configuración articular que sobrepasa alguno de los límites operativos de las juntas. En este trabajo se probaron diversos algoritmos que ayudan a mantener las juntas dentro de sus límites operativos durante una trayectoria de marcha, sin embargo, solo el uso de la matriz de pesos \mathbf{W} logró resultados satisfactorios. Como se describe en la sección 1.7.1 el uso de ponderación para restringir límites en las juntas permite asignar pesos a las juntas que se encuentran cerca de sus límites operativos, restringiendo sus movimientos fuera de ellos. La principal ventaja de la ponderación para restringir límites en las juntas es que no hace uso del espacio nulo del jacobiano, por lo que puede ser aplicado indistintamente para los métodos de VAT o CJ.

4.4.1. Múltiples tareas como vector aumentado y límites de junta

El método Vector Aumentado de Tareas con Límites (VATL) incorpora a la metodología VAT el uso de la matriz de pesos \mathbf{W} para restringir el movimiento de las juntas que se acercan a los límites de operación. Los pesos se calculan de acuerdo con el procedimiento mostrado en la sección 1.7.1. En este caso se utilizarán las definiciones del vector $\Delta \mathbf{x}_A$ y el jacobiano \mathbf{J}_A usadas en la sección 4.3.1 en la definición del algoritmo VAT.

Utilizando el algoritmo VATL y los límites mostrados en el anexo A se obtienen las secuencias de poses mostradas en la figura 4.16. En la pose 1 se observa el comienzo del movimiento, durante la figura 10, 20 y 30 el robot carga su torso sobre la superficie del pie izquierdo, a partir de la pose 40 el robot inicia el desplazamiento del pie en vuelo en la dirección x positiva, final mente en la pose 100 termina el movimiento.

Las trayectorias de referencia y los resultados obtenidos para el pie en vuelo y el COM, se muestran en las figuras 4.17 y 4.18. El comportamiento obtenido tanto del posicionamiento del pie como del posicionamiento del COM coincide con las trayectorias de referencia, sin presentar un desvío aparente en las en las gráficas.

A partir de la figura 4.19 se observa que el error en el posicionamiento de pie en vuelo no tiene una tendencia aparente en el inicio del movimiento como se presentó en VAT, también se observa un mayor error promedio en comparación directa con el algoritmo VAT. En 4.20 se puede observar que el error en el posicionamiento del COM tiene valores muy similares a los

presentados por el algoritmo VAT. Además, se puede ver que a diferencia del algoritmo VAT, en este caso no existen oscilaciones notables dentro de las primeras 20 poses. El desempeño del algoritmo es aceptable, incluso superando el rendimiento obtenido usando las estrategias de CJ.

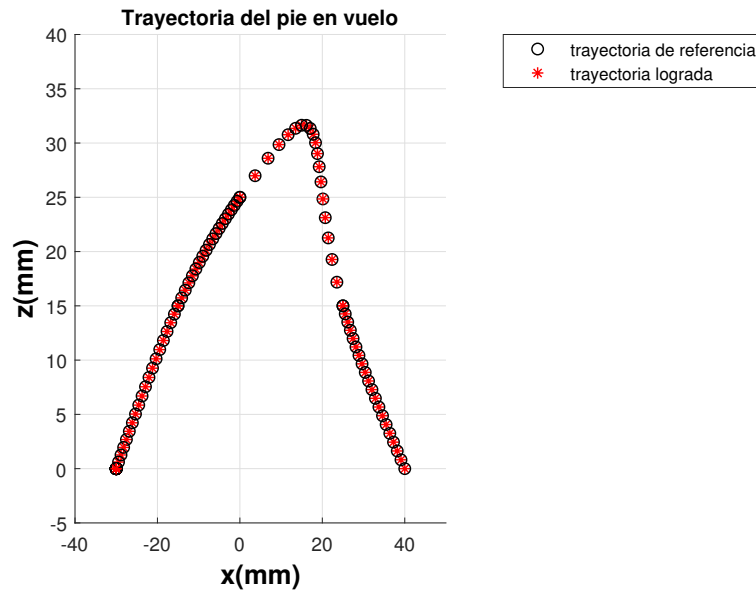


Figura 4.17. Trayectoria de pie en vuelo para VATL.

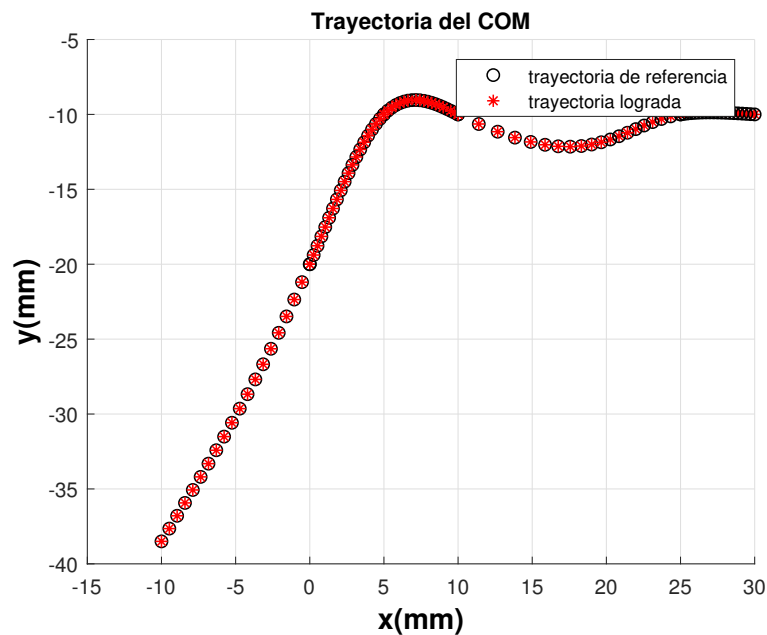


Figura 4.18. Trayectoria del COM para VATL.

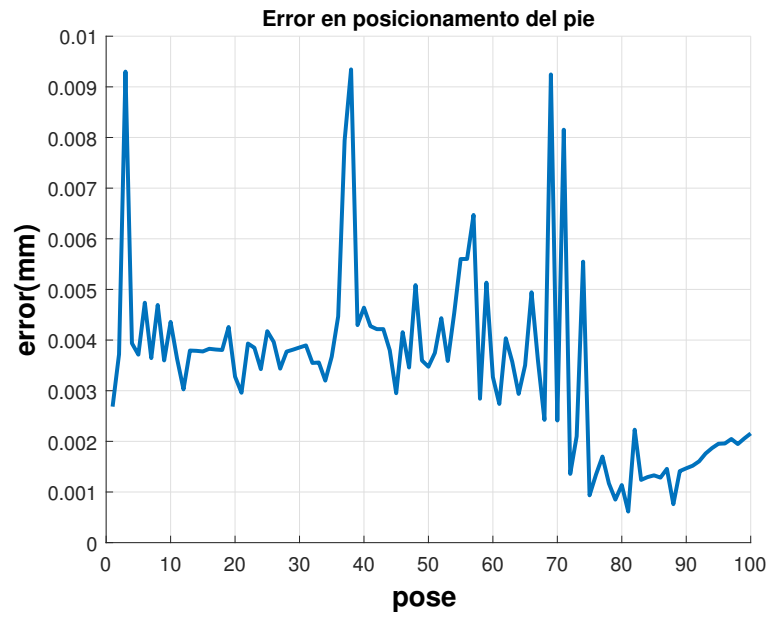


Figura 4.19. Error de posicionamiento de pie en vuelo para VATL.

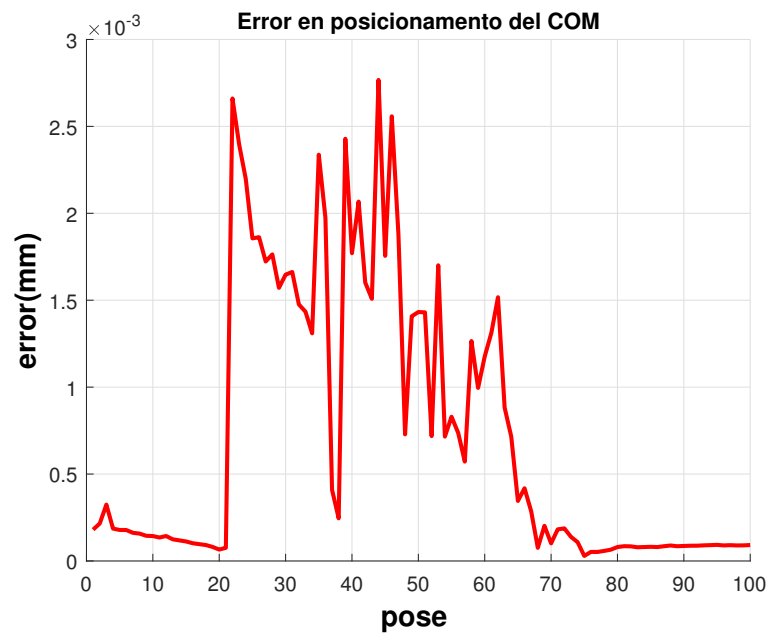


Figura 4.20. Error de posicionamiento del COM para VATL.

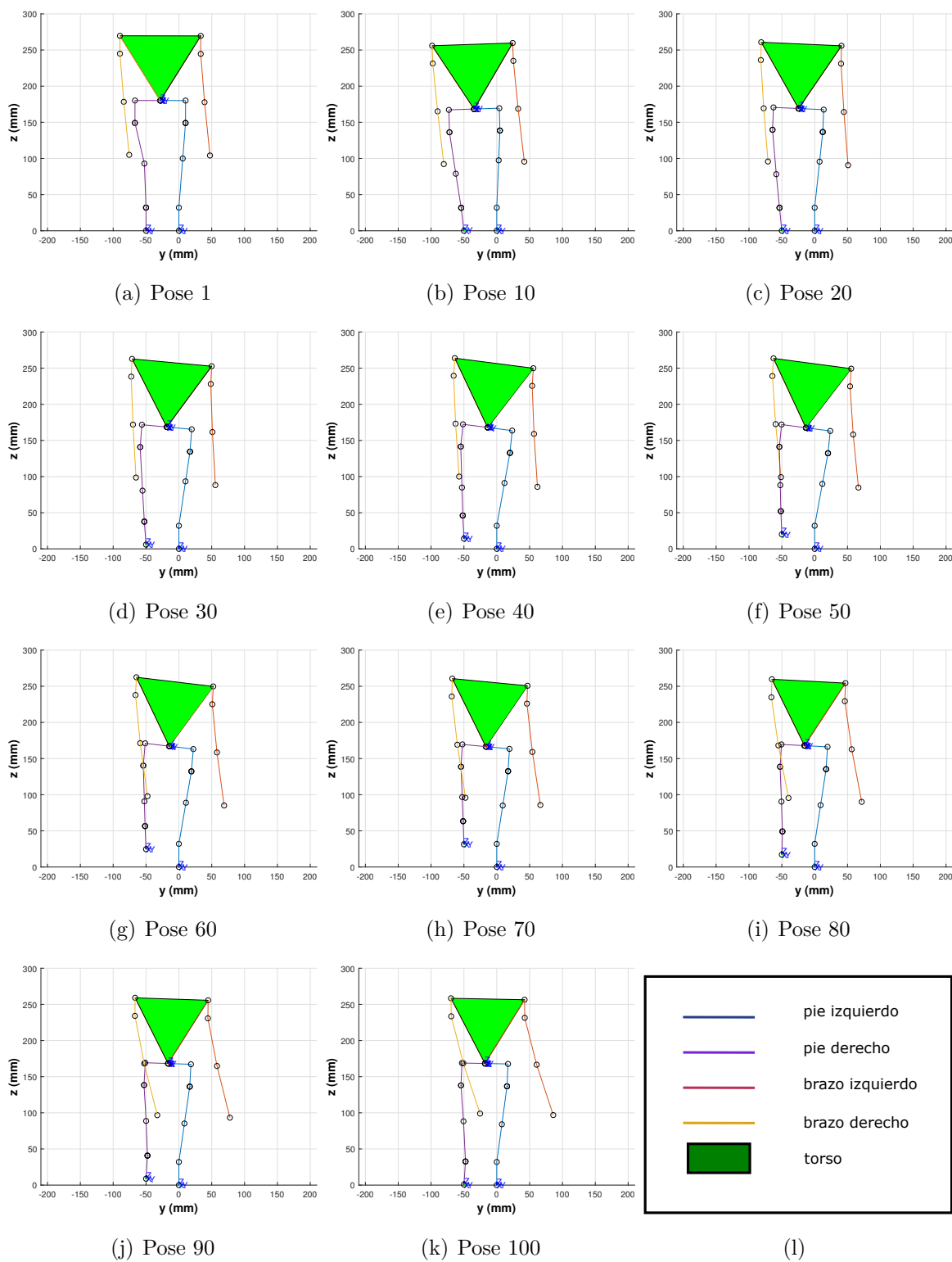


Figura 4.16. Secuencia de movimientos generados para un paso usando la estrategia de VATL.

4.4.2. Control jerárquico y límites de junta

Como se mencionó anteriormente el uso de la ponderación para restringir límites en las juntas se puede aplicar a las estrategias de CJ sin problema. Ya que al restringir las juntas a partir de la matriz de pesos \mathbf{W} se evita el uso del espacio nulo del jacobiano de la tarea primaria, éste queda libre para proyectar un comportamiento extra. A continuación, se muestran la incorporación de la ponderación para restringir límites en las juntas a los esquemas de CJ descritos en la sección 4.3.2.

4.4.2.1. Prioridad en el posicionamiento del pie de apoyo con límites

En esta versión de Control Jerárquico con Límites (CJL) se usan las definiciones de los vectores $\Delta \mathbf{x}_A$, $\Delta \mathbf{q}_s$ y \mathbf{J}_A usadas en la sección 4.3.2.1, donde se muestra el CJ con prioridad en el posicionamiento del pie en vuelo. Las ponderaciones se calculan a partir del procedimiento mostrado en la sección 1.7.1.

En la figura 4.21 se observan una secuencia de poses obtenidas a partir de la implementación de CJL con prioridad en el posicionamiento del pie en vuelo. En la pose 1 se observa el comienzo del movimiento, durante la figura 10, 20 y 30 el robot carga su torso sobre la superficie del pie izquierdo, a partir de la pose 40 el robot inicia el desplazamiento del pie en vuelo en la dirección x positiva, final mente en la pose 100 termina el movimiento.

En las figuras 4.22 se puede ver que el robot mantuvo la trayectoria del pie en vuelo sin algún error aparente, mientras que en la 4.23 se observa que la trayectoria seguida por el COM tuvo dificultades para seguir su trayecto presentando grandes desvíos durante toda la trayectoria.

El error del posicionamiento del pie se presenta en la figura 4.24, y se puede ver un comportamiento deseable, presentando los mayores errores dentro del intervalo de las primeras 20 poses. La magnitud de los errores se encuentra entre las 2 y 9 micras lo que es un error aceptable para la aplicación. En la figura 4.25 se presenta el error obtenido en la trayectoria del COM, donde se ve un crecimiento considerable comparado con la versión de la metodología de CJ sin incluir los límites. EL error máximo que presenta es de 10 mm, lo cual no es deseado ya que puede propiciar la inestabilidad del robot.

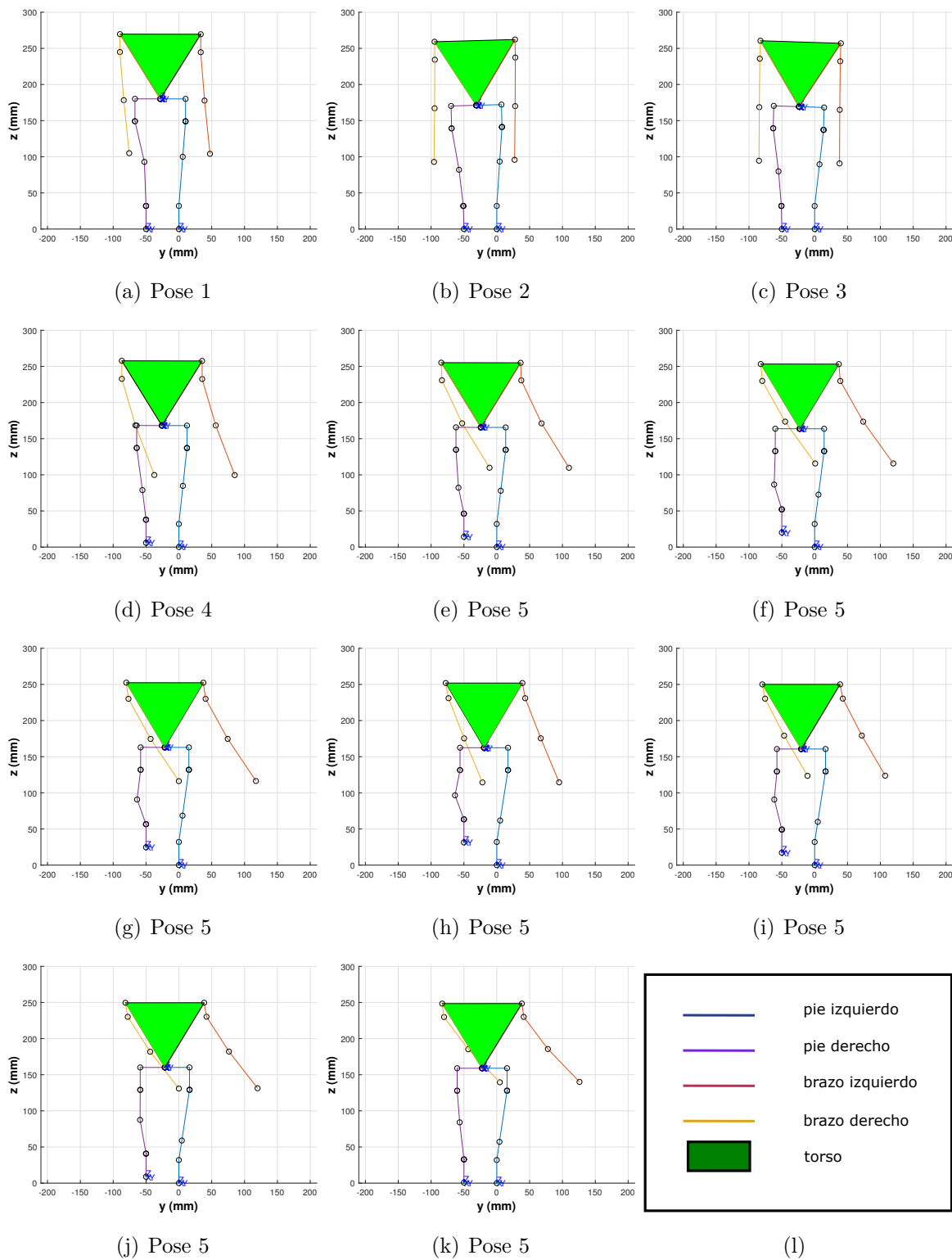


Figura 4.21. Secuencia de movimientos generados para un paso usando la estrategia de **CJL** con prioridad en el posicionamiento del pie.

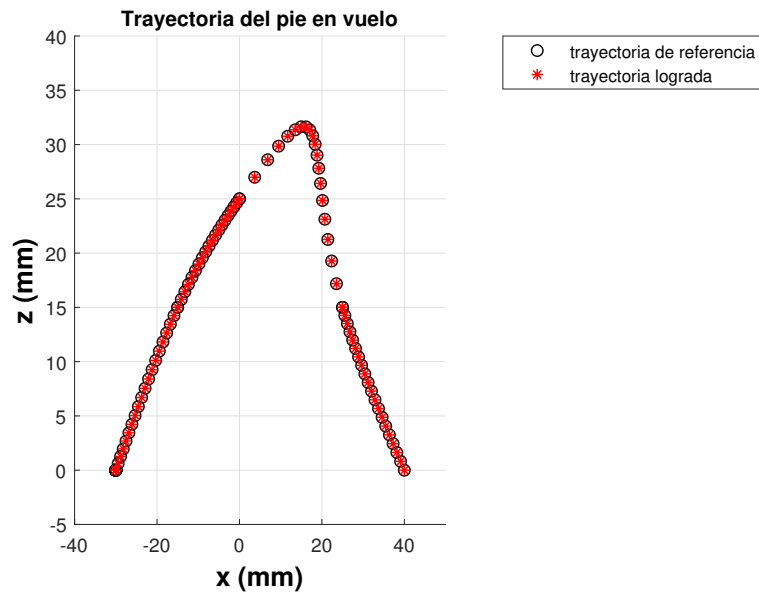


Figura 4.22. Trayectoria de pie en vuelo para CJL con prioridad en el posicionamiento del pie.

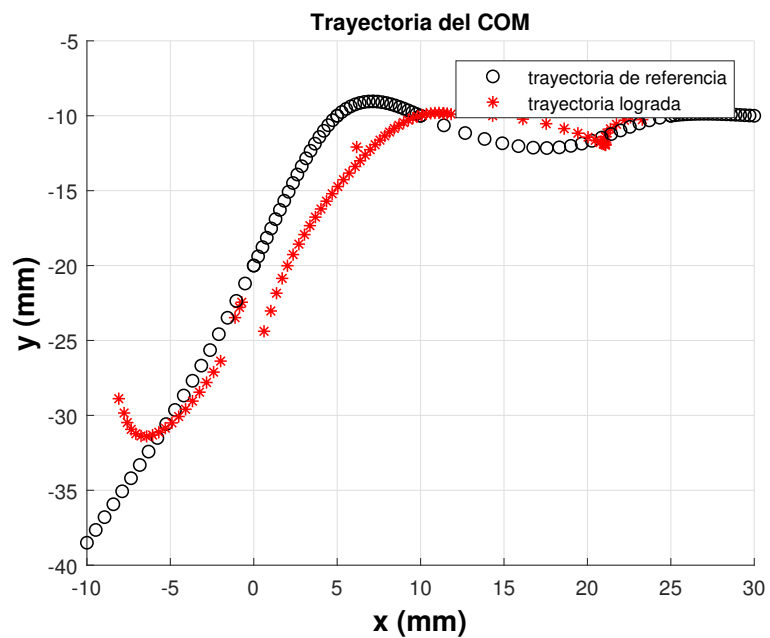


Figura 4.23. Trayectoria del COM para CJL con prioridad en el posicionamiento del pie.

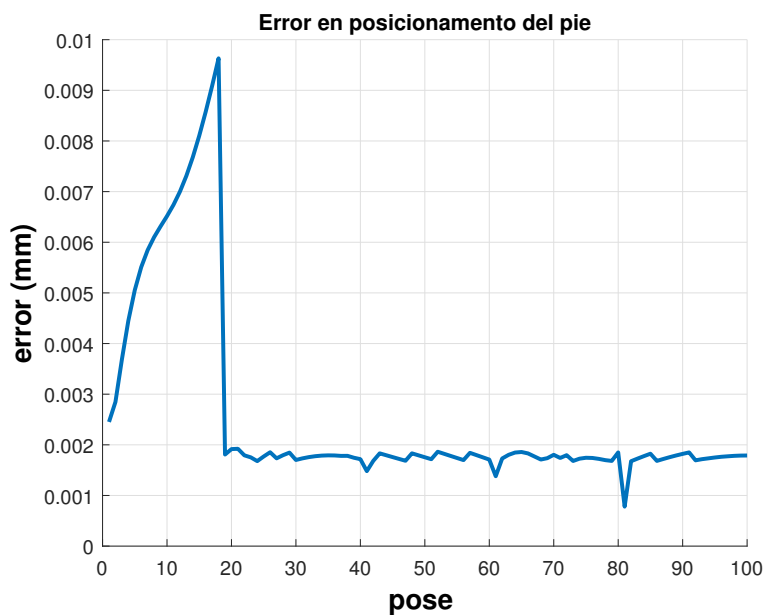


Figura 4.24. Error de posicionamiento de pie en vuelo para CJL con prioridad en el posicionamiento del pie.

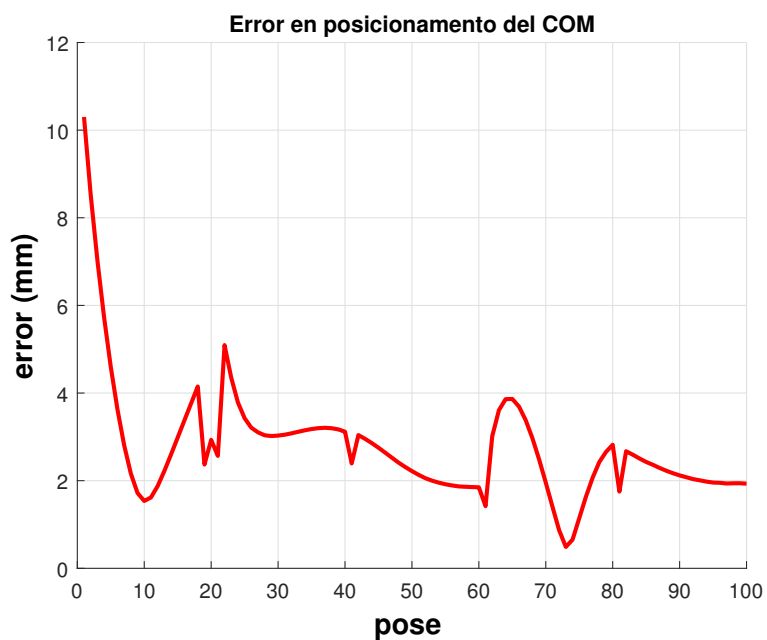


Figura 4.25. Error de posicionamiento del COM para CJL con prioridad en el posicionamiento del pie.

El comportamiento de este método no garantiza la estabilidad del robot debido al gran error que se presenta en el posicionamiento del COM.

4.4.2.2. Prioridad en el posicionamiento del COM con límites

Para este caso de CJL se usa la definición de los vectores $\Delta \mathbf{x}_A$, $\Delta \mathbf{q}_s$ y \mathbf{J}_A usadas en la sección 4.3.3 en donde la tarea primaria se considera el posicionamiento del COM. Los pesos se calculan a partir del procedimiento mostrado en la sección 1.7.1.

En la figura 4.26 se pueden ver una secuencia de poses obtenidas a partir del CJL con prioridad en el posicionamiento del COM. En la pose 1 se observa el comienzo del movimiento, durante la figura 10, 20 y 30 el robot carga su torso sobre la superficie del pie izquierdo, a partir de la pose 40 el robot inicia el desplazamiento del pie en vuelo en la dirección x positiva, final mente en la pose 100 termina el movimiento.

La trayectoria del pie en vuelo se puede ver en la figura 4.27, donde se percibe que a pesar de que se sigue el trayecto existen varias poses que tuvieron desvíos de consideración, principalmente en el inicio de la trayectoria. El trayecto del COM se puede observar en la figura 4.28. El COM siguió el trayecto sin ningún error aparente.

El error en posicionamiento del pie en vuelo se puede ver en la figura 4.29, donde el error máximo presentado durante el trayecto es de 2.5 mm, teniendo lugar en las primeras tres poses lo cuando los dos pies están en contacto con el suelo. Teniendo en cuenta el error mostrado en la figura 4.29 y la trayectoria de la figura 4.27 en el punto de error máximo en la trayectoria del pie en vuelo se presenta una interferencia con el suelo.

El error en el posicionamiento del COM se muestra en la figura 4.30 donde se obtiene resultados excelentes, teniendo un error máximo de 0.7 micras.

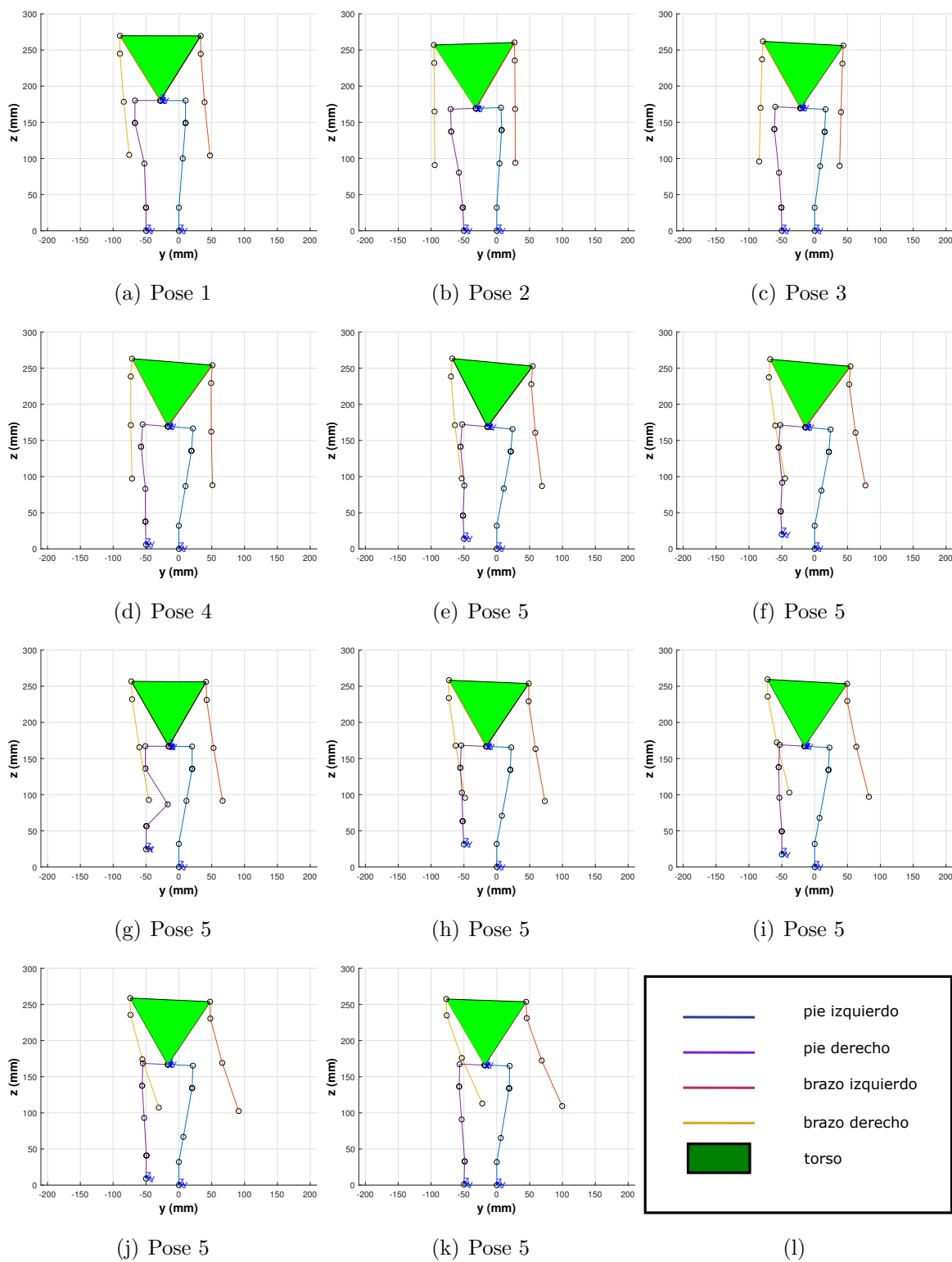


Figura 4.26. Secuencia de movimientos generados para un paso usando la estrategia de CJL con prioridad en el posicionamiento del COM.

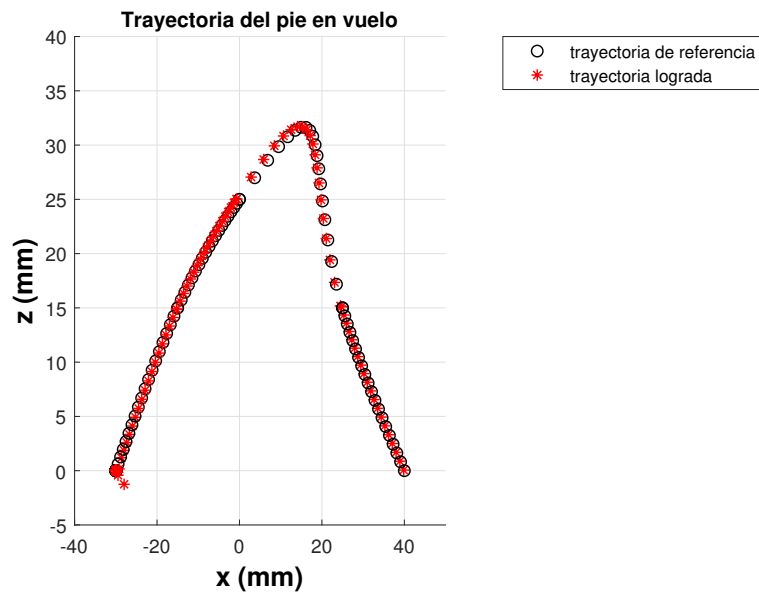


Figura 4.27. Trayectoria de pie en vuelo para CJL con prioridad en el posicionamiento del COM.

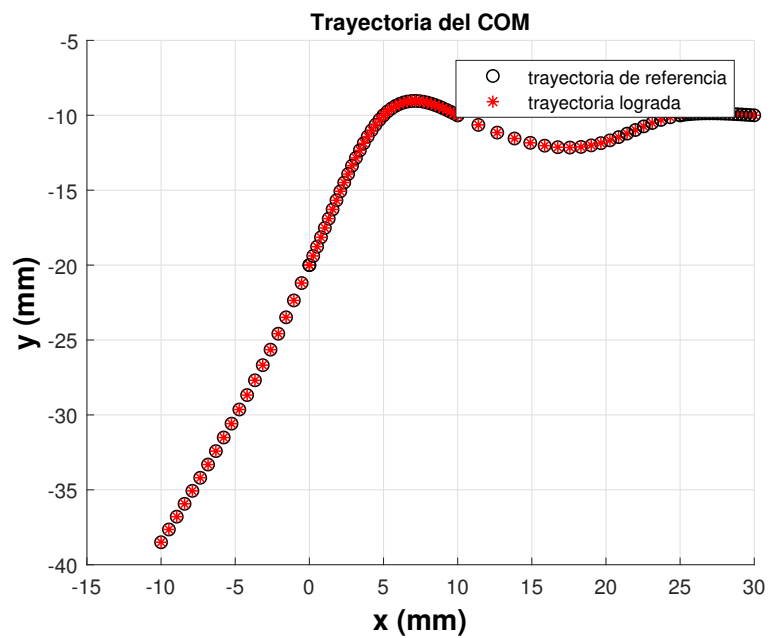


Figura 4.28. Trayectoria del COM para CJL con prioridad en el posicionamiento del COM.



Figura 4.29. Error de posicionamiento de pie en vuelo para CJL con prioridad en el posicionamiento del COM.

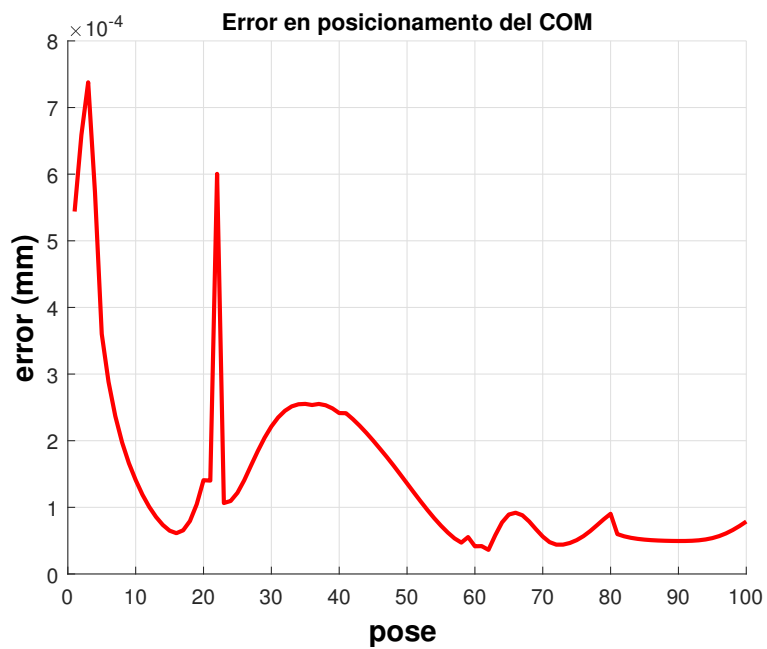


Figura 4.30. Error de posicionamiento del COM para CJL con prioridad en el posicionamiento del COM.

El algoritmo de **CJL** con prioridad en el posicionamiento de masa es una opción factible para realizar la tarea de marcha debido a la gran precisión que logra en el posicionamiento del **COM**, sin embargo, se deben tomar acciones respecto a las posibles interferencias del pie en vuelo con el suelo.

El uso de la matriz de peso **W** para restringir el movimiento de las juntas dentro de cierto rango se adapta a cualquiera de los esquemas de control descritos dentro del capítulo. Sin embargo, en su aplicación existe un incremento en el tiempo de ejecución de los algoritmos, por lo que se debe tener en consideración.

4.5. Tiempos de ejecución

Uno de los aspectos a considerar es el tiempo de ejecución de cada uno de los algoritmos ya que éstos se buscan implementar en tiempo real en un robot humanoide. En la figura 4.31 se muestra gráficamente los tiempos de ejecución por pose en ms. de cada uno de los algoritmos multitarea incluidos dentro de este capítulo, donde **CJ1** representa el algoritmo de **CJ** con prioridad en el posicionamiento del pie en vuelo y **CJ1L** su versión con límites en las juntas, **CJ2** representa el algoritmo de **CJ** con prioridad en el posicionamiento del **COM** y **CJ2L** su versión con límites en las juntas.

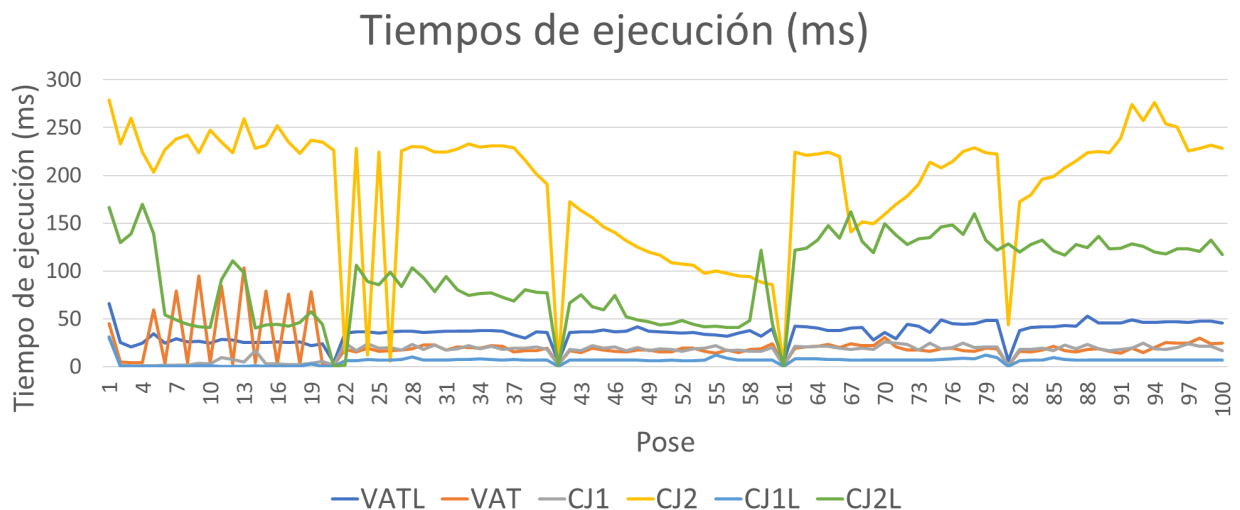


Figura 4.31. Tiempos de ejecución de los algoritmos multitarea (ms).

Como se puede observar en la gráfica el algoritmo más lento es el **CJ** con prioridad en posicionamiento de masa con un tiempo promedio de 189.6 ms por pose. El algoritmo más

rápido es **CJL** con prioridad en el posicionamiento del pie, sin embargo, éste es el que cuenta con un mayor error en el posicionamiento del **COM**. Las opciones más viables son **VAT** y **VATL** debido a su buen tiempo de ejecución promedio y el error obtenido en las trayectorias tanto del pie en vuelo como del **COM**.

4.6. Conclusiones del capítulo

A lo largo de este capítulo se presentaron diversas estrategias, tanto para crear trayectorias de marcha como para seguirlas con un robot humanoide. Se presentan dos estrategias para generar las trayectorias del pie y el **COM**. En la generación de trayectorias por *splines* cúbicas, se especifican manualmente los puntos por los que se desea que se localice el pie y el **COM**, por lo cual, es la estrategia más apropiada para lograr la estabilidad estática del robot. Con el uso del péndulo invertido 3D se pueden obtener trayectorias del **COM** que faciliten el cálculo de la estabilidad dinámica, ya que todo el cuerpo se considera como una masa concentrada, se reduce la complejidad de la obtención del **ZMP**.

Para la obtención de las configuraciones articulares que cumplan las trayectorias calculadas, se usan dos estrategias principales: **VAT** y **CJ**. El uso del método **VAT** permite ejecutar múltiples tareas con una prioridad igual, lo que da como resultado una precisión similar entre ellas. El método **CJ**, permite asignar diversas prioridades entre las tareas, lo que da como resultado una mayor precisión en la tarea primaria, mientras que, en la tarea secundaria el error aumenta considerablemente. Otro de los aspectos a tomar en cuenta es el tiempo de ejecución de los algoritmos, donde el método **CJ** con prioridad de posicionamiento del pie tiene un mejor desempeño con un tiempo promedio por pose de 16.12 ms., el método **VAT** 21.88 ms. y el **CJ** con prioridad de posicionamiento del **COM** 189.6 ms.

Finalmente se implementó el uso de la matriz de pesos para limitar el desplazamiento de las juntas. Los resultados obtenidos de su aplicación en conjunto con la estrategia de **VAT** son satisfactorios, tanto en el error de posicionamiento del pie en vuelo y el **COM** durante sus trayectos, y el tiempo de ejecución del algoritmo. En las estrategias de **CJL** se percibe un aumento considerable del error en las tareas secundarias, por lo cual no se puede asegurar un comportamiento confiable para la generación de trayectorias de marcha

Conclusiones

En este trabajo se desarrollaron varias estrategias para la generación y control de trayectorias cinemáticas del centro de masa (COM) de un robot humanoide en tareas tales como marcha, subir o bajar escaleras, levantar objetos, entre otras. Las estrategias propuestas permiten asegurar la estabilidad mecánica del humanoide durante el seguimiento de trayectorias en baja velocidad y se validaron en una aplicación de marcha.

Se obtuvo el modelo cinemático directo e inverso de un robot humanoide de 18 grados de libertad. En este modelado se considera un humanoide de 4 extremidades (2 brazos y 2 piernas).

Se desarrollaron múltiples estrategias de control para estabilizar trayectorias de marcha para diferentes arquitecturas de robots humanoides. Las estrategias de control se basan en controlar explícitamente el COM del robot. Al utilizar estas estrategias es posible generar y analizar trayectorias cinemáticas que cumplan los criterios de estabilidad mecánica.

Las estrategias de control propuestas se validaron mediante diversas simulaciones de un robot humanoide tanto para poses estáticas como para trayectorias de marcha.

Como resultado principal de este trabajo se desarrolló un algoritmo que incorpora una estrategia de control para posicionar el COM como tarea primara, lo cual que permite la ejecución de trayectorias estables de locomoción. Además, esta estrategia permite definir tareas adicionales tales como el control de la posición y orientación del pie en vuelo y del torso lo cual es muy relevante para que las posturas del robot durante la locomoción sean

más "naturales" o parecidas a la locomoción humana. Este algoritmo también incorpora restricciones en las juntas para evitar que salgan de sus límites operativos, lo que previene posibles auto colisiones o la obtención de poses no realizables por el robot. Este algoritmo incorpora diversos desarrollos descritos a continuación.

En busca de asegurar la estabilidad estática de un humanoide de n grados de libertad, se crearon 3 algoritmos para controlar y posicionar su COM. Los algoritmos se validaron en simulación, aplicados al modelo simplificado del robot Bioloid Premium de 18 DOF. Estos algoritmos se pueden extender y aplicar a cualquier robot con múltiples extremidades y arquitectura de árbol. Al controlar explícitamente el COM del robot, estos algoritmos permiten tener una garantía de que las trayectorias del robot son estables a bajas velocidades.

Con la finalidad de realizar una marcha estable, se implementaron dos formas de generar trayectorias del COM: a) por medio de *splines* cúbicas y b) por la metodología del Péndulo Invertido Lineal 3D. La generación de trayectorias por *splines* cúbicas permitió definir las posiciones en donde se debía colocar el COM para mantener el equilibrio estático del robot humanoide y obtener una transición suave entre ellas. El uso del Péndulo Invertido Lineal 3D permite calcular trayectorias de cinemáticas del COM basándonos en las ecuaciones dinámicas de un péndulo invertido. Además, esta metodología sirve como un inicio al estudio de la estabilidad dinámica de un humanoide mientras camina.

Para la ejecución de la marcha del robot humanoide, se optó por el uso de estrategias de control de múltiples tareas simultáneas, éstas permiten asignar diversos comportamientos al robot, como: el posicionamiento del COM, el posicionamiento del pie en vuelo, etc., y ejecutarlos de forma paralela. Las estrategias desarrolladas son: a) Vector Aumentado de Tareas que consiste en apilar las tareas dentro de un vector y junto con un jacobiano asociado se obtienen incrementos mediante el método iterativo, b) el Control Jerárquico que consiste en asignar prioridades a las tareas y aprovechando la redundancia del sistema obtener vectores asociados a las tareas secundarias que no afecten el comportamiento primario del robot. Se implementó de forma exitosa las estrategias de Vector Aumentado de Tareas y el Control Jerárquico. El uso de la estrategia de Vector Aumentado de Tarea obtuvo un excelente desempeño tanto en tiempo de ejecución, como en el error obtenido durante las tareas de posicionamiento. El Control Jerárquico obtuvo un buen desempeño, y aunque el error presentado en este algoritmo es mayor, es aceptable para su aplicación.

Se atacó el problema de tomar en consideración los límites operativos de las juntas durante el control de trayectorias. Para lo anterior, se propusieron dos estrategias: a) adición de vectores de optimización, los cuales alejan a las juntas de los límites añadiendo un

comportamiento secundario a la cinemática inversa, b) el uso de una matriz de pesos, la cual restringe el movimiento de las juntas al acercarse a alguno de los límites operativos. La matriz de pesos, logró obtener un resultado satisfactorio tanto en poses individuales como en trayectorias con un tiempos de ejecución y errores de posicionamiento aceptables. La adición de los vectores de optimización incrementa el tiempo de cómputo y aunque se logran corregir los problemas con los límites de juntas en una pose, éstos no aseguran mantener las configuraciones articulares en su rango operativo durante una trayectoria de marcha.

Las estrategias de control desarrolladas se basan en el modelo cinemático del robot, por lo que presenta una metodología para modelar robots humanoides, haciendo énfasis en las consideraciones que se deben tomar durante el proceso. La metodología desarrollada se utilizó para obtener el modelo cinemático de un robot Bioid Premium de 18 grados de libertad. En primera instancia se utilizó este modelo para generar trayectorias cinemáticas de marcha, sin embargo, éstas no aseguran la estabilidad del robot, por lo que no es un método confiable para lograr la tarea planteada de locomoción.

Trabajo a futuro

Enseguida se mencionan algunas líneas de desarrollo que a futuro permitirían avanzar en la aplicación de las técnicas desarrolladas en este trabajo.

- Validación experimental de los algoritmos presentados en este trabajo.
 - Extender los algoritmos de control del **COM** a diversas arquitecturas de árbol.
 - Extender los algoritmos a control en velocidad.
 - Análisis de la estabilidad dinámica de un humanoide por métodos de Péndulo Invertido Lineal.
-

Bibliografía

- [ABB, 2018] ABB (2018). Yumi® - creando un futuro automatizado juntos. <http://new.abb.com/products/robotics/es/robots-industriales/yumi>. [acceso, 20/01/2018].
- [Asimov, 2004] Asimov, I. (2004). *I, robot*, volumen 1. Spectra.
- [Borelli, 2012] Borelli, G. A. (2012). *On the movement of animals*. Springer Science & Business Media.
- [Buss, 2004a] Buss, S. R. (2004a). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16.
- [Buss, 2004b] Buss, S. R. (2004b). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16.
- [Čapek, 1925] Čapek, K. (1925). *RUR (Rossum's universal robots): a fantastic melodrama*. Doubleday, Page.
- [Chan y Dubey, 1995] Chan, T. F. y Dubey, R. V. (1995). A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292.

- [Conacyt, 2016] Conacyt, A. I. (2016). Muestran avances de robótica humanoide en México. <http://www.conacytprensa.mx/index.php/tecnologia/robotica/11091-muestran-avances-de-robotica-humanoid-en-mexico>. [acceso, 20/01/2018].
- [Conkur y Buckingham, 1997] Conkur, E. S. y Buckingham, R. (1997). Clarifying the definition of redundancy as used in robotics. *Robotica*, 15(5):583–586.
- [Craig, 2005] Craig, J. J. (2005). *Introduction to robotics: mechanics and control*, volumen 3. Pearson/Prentice Hall Upper Saddle River, NJ, USA:.
- [Da Vinci, 2012] Da Vinci, L. (2012). *The notebooks of Leonardo da Vinci*, volumen 1. Courier Corporation.
- [Dederick, 1868] Dederick, Z. (1868). Improvement in steam-oaebiage. US Patent 75,874.
- [D’Souza *et al.*, 2001] D’Souza, A., Vijayakumar, S., y Schaal, S. (2001). Learning inverse kinematics. En *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volumen 1, pp. 298–303 vol.1.
- [DYNAMICS, 2017a] DYNAMICS, B. (2017a). Atlas. <https://www.bostondynamics.com/robots>. [acceso 05/09/2017].
- [DYNAMICS, 2017b] DYNAMICS, B. (2017b). Boston dynamics. <https://www.bostondynamics.com/>. [acceso 14/11/2017].
- [Eaton, 2015] Eaton, M. (2015). *Humanoid Robots, Their Simulators, and the Reality Gap*, pp. 33–58. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Fitzpatrick *et al.*, 2016] Fitzpatrick, P., Harada, K., Kemp, C. C., Matsumoto, Y., Yokoi, K., y Yoshida, E. (2016). Humanoids. En *Springer Handbook of Robotics*, pp. 1789–1818. Springer.
- [Guizzo y Ackerman, 2015] Guizzo, E. y Ackerman, E. (2015). A compilation of robots falling down at the darpa robotics challenge. <https://spectrum.ieee.org/automaton/robotics/humanoids/darpa-robotics-challenge-robots-falling>. [acceso, 26/01/2018].
- [Hirose y Ogawa, 2007] Hirose, M. y Ogawa, K. (2007). Honda humanoid robots development. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 365(1850):11–19.

- [HONDA, 2017] HONDA (2017). Asimo the world's most advanced humanoid robot. <http://asimo.honda.com/asimo-specs/>. [acceso, 01/02/2017].
- [Huang *et al.*, 1999] Huang, Q., Kajita, S., Koyachi, N., Kaneko, K., Yokoi, K., Arai, H., Komoriya, K., y Tanie, K. (1999). A high stability, smooth walking pattern for a biped robot. En *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volumen 1, pp. 65–71. IEEE.
- [Humanoid Robotics Institute, 2008] Humanoid Robotics Institute, W. U. (2008). Wabot-waseda robot-. webpage. [acceso 23/11/2017].
- [Inoue *et al.*, 2000] Inoue, Hirochika and Tachi, Susumu and Tanie, Kazuo and Yokoi, Kazuhito and Hirai, Shigeoki and Hirukawa, Hirohisa and Hirai, Kazuo and Nakayama, Shigeto and Sawada, Kazuya and Nishiyama, Takashi and others (2000). Hrp: Humanoid robotics project of miti. *Journal of the Robotics Society of Japan*.
- [Jung *et al.*, 1995] Jung, E. S., Kee, D., y Chung, M. K. (1995). Upper body reach posture prediction for ergonomic evaluation models. *International Journal of Industrial Ergonomics*, 16(2):95–107.
- [Kajita *et al.*, 2014] Kajita, S., Hirukawa, H., Harada, K., y Yokoi, K. (2014). *Introduction to humanoid robotics*. Springer Tracts in Advanced Robotics, 101. Springer, Heidelberg.
- [Kajita *et al.*, 2003] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., y Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. En *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volumen 2, pp. 1620–1626. IEEE.
- [Kajita y Tani, 1996] Kajita, S. y Tani, K. (1996). Experimental study of biped dynamic walking. *IEEE Control Systems*, 16(1):13–19.
- [Kaneko *et al.*, 2008] Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., y Akachi, K. (2008). Humanoid robot hrp-3. En *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2471–2478.
- [Kaneko *et al.*, 2004] Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., e Isozumi, T. (2004). Humanoid robot hrp-2. En *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volumen 2, pp. 1083–1090 Vol.2.

- [Kaneko *et al.*, 2011] Kaneko, K., Kanehiro, F., Morisawa, M., Akachi, K., Miyamori, G., Hayashi, A., y Kanehira, N. (2011). Humanoid robot hrp-4 - humanoid robotics platform with lightweight and slim body. En *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4400–4407.
- [KAWADA INDUSTRIES, 2017] KAWADA INDUSTRIES, I. (2017). Humanoid robot hrp-2 "promet". [acceso 27/11/2017].
- [Kemp *et al.*, 2008] Kemp, C. C., Fitzpatrick, P., Hirukawa, H., Yokoi, K., Harada, K., y Matsumoto, Y. (2008). *Humanoids*, pp. 1307–1333. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Khatib, 2012] Khatib, O. (2012). *Advanced Robotic Manipulation*. Stanford University. Lecture Notes, CS327A, Spring 2012.
- [Komura *et al.*, 2003] Komura, T., Kuroda, A., Kudoh, S., Lan, T. C., y Shinagawa, Y. (2003). An inverse kinematics method for 3d figures with motion data. *Proceedings Computer Graphics International 2003, Computer Graphics International, 2003. Proceedings, Computer graphics*, p. 266.
- [Kormushev *et al.*, 2010] Kormushev, P., Calinon, S., Saegusa, R., y Metta, G. (2010). Learning the skill of archery by a humanoid robot icub. En *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp. 417–423.
- [Laboratory, 2015] Laboratory, T. (2015). Takanishi laboratory. <http://www.takanishi.mech.waseda.ac.jp>. [acceso 13/09/2017].
- [Liegeois, 1977] Liegeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE transactions on systems, man, and cybernetics*, 7(12):868–871.
- [Limer, 2015] Limer, E. (2015). The most hilarious robo-falls from the darpa robotics challenge. POPULAR MECHANICS, <http://www.popularmechanics.com/technology/robots/a15907/best-falls-from-darpa-robot-challenge/>. [acceso, 20/01/2018].
- [Marey y Chaumette, 2010] Marey, M. y Chaumette, F. (2010). New strategies for avoiding robot joint limits: Application to visual servoing using a large projection operator. En *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 6222–6227. IEEE.

- [Mechanics, 2010] Mechanics, P. (2010). Mediaeval mechanics. webpage. [acceso, 24/11/2017].
- [Meirovitch, 2010] Meirovitch, L. (2010). *Methods of analytical dynamics*. Courier Corporation.
- [Metta *et al.*, 2010] Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Rosander, K., Lopes, M., Santos-Victor, J., Bernardino, A., y Montesano, L. (2010). The icub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23(8):1125 – 1134. Social Cognition: From Babies to Robots.
- [Miura y Shimoyama, 1984] Miura, H. y Shimoyama, I. (1984). Dynamic walk of a biped. *The International Journal of Robotics Research*, 3(2):60–74.
- [Nelson *et al.*, 2012] Nelson, G., Saunders, A., Neville, N., Swilling, B., Bondaryk, J., Billings, D., Lee, C., Playter, R., y Raibert, M. (2012). Petman: A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, 30(4):372–377.
- [Ogura *et al.*, 2006] Ogura, Y., Aikawa, H., Shimomura, K., Morishima, A., Lim, H.-o., y Takanishi, A. (2006). Development of a new humanoid robot wabian-2. En *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 76–81. IEEE.
- [Park *et al.*, 2005] Park, I.-W., Kim, J.-Y., Lee, J., y Oh, J.-H. (2005). Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo). En *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pp. 321–326.
- [Pratt *et al.*, 1997] Pratt, J., Dilworth, P., y Pratt, G. (1997). Virtual model control of a bipedal walking robot. En *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volumen 1, pp. 193–198. IEEE.
- [Pratt y Pratt, 1998] Pratt, J. y Pratt, G. (1998). Intuitive control of a planar bipedal walking robot. En *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volumen 3, pp. 2014–2021. IEEE.
- [robot, 2017] robot, D. (2017). Hovis lite. <http://www.dstrobot.com/jsp/cms/view.jsp?code=100795>. [acceso, 07/12/2017].
- [Robotics, 2017a] Robotics, A. (2017a). Robot paralelo / 4 ejes / 3 ejes / de manipulación irb 360 series. [acceso, 04/12/2017].

- [Robotics, 2017b] Robotics, S. (2017b). Hovis lite. <http://www.sastrarobotics.com/products/hovis-lite/>. [acceso, 04/12/2017].
- [ROBOTRADERS, 2017] ROBOTRADERS (2017). New robots. <http://www.robotraders.co.uk/catalog/new-robots-0>. [acceso 04/12/2017].
- [Samson *et al.*, 1991] Samson, C., Espiau, B., y Borgne, M. L. (1991). *Robot control: the task function approach*. Oxford University Press.
- [Sciavicco y Siciliano, 2012] Sciavicco, L. y Siciliano, B. (2012). *Modelling and control of robot manipulators*. Springer Science & Business Media.
- [Shop, 2017] Shop, R. (2017). Bioloid premium robot kit. <http://www.robotshop.com/en/robotis-bioloid-premium-robot-kit.html>. [acceso, 04/12/2017].
- [Spong *et al.*, 2006] Spong, M. W., Hutchinson, S., y Vidyasagar, M. (2006). *Robot modeling and control*, volumen 3. Wiley New York.
- [Stasse *et al.*, 2008] Stasse, O., Escande, A., Mansard, N., Miossec, S., Evrard, P., y Kheddar, A. (2008). Real-time (self)-collision avoidance task on a hrp-2 humanoid robot. En *2008 IEEE International Conference on Robotics and Automation*, pp. 3200–3205.
- [STMicroelectronics, 2017] STMicroelectronics (2017). Stm32f103re. <http://www.st.com/en/microcontrollers/stm32f103re.html>. [acceso, 07/12/2017].
- [Sugano y Kato, 1987] Sugano, S. y Kato, I. (1987). Wabot-2: Autonomous robot with dexterous finger-arm–finger-arm coordination control in keyboard performance. En *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volumen 4, pp. 90–97.
- [T. Komura, 1969] T. Komura, Kuroda, A. K. S. T. C. L. S. Y. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems, Man-Machine Systems, IEEE Transactions on, IEEE Trans. Man-Machine Syst*, (2):47.
- [Vaughan, 2003] Vaughan, C. L. (2003). Theories of bipedal walking: an odyssey. *Journal of Biomechanics*, 36(4):513 – 523.
- [Velasco, 2011] Velasco, J. (2011). Petman, el inquietante robot que prueba la ropa de los soldados. <https://hipertextual.com/2011/10/petman-soldado-robot>. [acceso el 05/09/2017].

- [Vukobratović y Borovac, 2004] Vukobratović, M. y Borovac, B. (2004). Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173.
- [Vukobratovic y Juricic, 1969] Vukobratovic, M. y Juricic, D. (1969). Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*, (1):1–6.
- [Wampler, 1986] Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics, Systems, Man and Cybernetics, IEEE Transactions on, IEEE Trans. Syst., Man, Cybern.*, (1):93.
- [Wang *et al.*, 2013] Wang, F., Zhang, Z., y Ou, Y. (2013). Inverse kinematics for center of mass through iteration. *Proceedings of the 32nd Chinese Control Conference*, p. 5833.
- [Wang *et al.*, 2014] Wang, H., Zheng, Y. F., Jun, Y., y Oh, P. (2014). Drc-hubo walking on rough terrains. En *2014 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 1–6.
- [Wiedebach *et al.*, 2016] Wiedebach, G., Bertrand, S., Wu, T., Fiorio, L., McCrory, S., Griffin, R., Nori, F., y Pratt, J. (2016). Walking on partial footholds including line contacts with the humanoid robot atlas. En *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 1312–1319.
- [Yates *et al.*, 2011] Yates, D. R., Vaessen, C., y Roupret, M. (2011). From leonardo to da vinci: the history of robot-assisted surgery in urology. *BJU International*, 108(11):1708–1713.
- [Zghal *et al.*, 1990] Zghal, H., Dubey, R. V., y Euler, J. A. (1990). Efficient gradient projection optimization for manipulators with multiple degrees of redundancy. En *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 1006–1011 vol.2.

Anexos

Límites de junta

A continuación, se presentan los límites físicos en las juntas del robot humanoide Bioloid Premium. En las tablas siguientes se presentan los valores máximos y mínimos que puede tomar la junta j_i expresados en grados. Además se incluyen los valores que representan la pose de inicio del robot Bioloid Premium.

Tabla A.1. Ángulos límite de robot Bioloid Premium en los brazos.

Juntas	Inicio	Mínimo	Máximo	Juntas	Inicio	Mínimo	Máximo
j10	-81.15	-150	150	j7	80.86	-150	150
j11	-68.26	-80.26	84.67	j8	67.97	-84.67	80.6
j12	-14.65	-36.65	84.35	j9	-14.65	-84.35	36.6

Tabla A.2. Ángulos límite de robot Bioloid Premium en las piernas.

Juntas	Inicio	Mínimo	Máximo	Juntas	Inicio	Mínimo	Máximo
j13	-45.12	-67.09	24.02	j6	45.8	-24.8	67.09
j15	-1.46	-40.46	5.57	j4	1.17	-5.57	40.17
j14	-50.01	-130.1	25.9	j5	49.8	-25.2	130.6
j16	-79.69	-129.69	0	j3	79.39	0	129.44
j17	39.55	-25	100	j2	-39.84	-100.84	25.2
j18	-1.46	-50	50	j1	1.17	-50.68	50.1

Matriz de Pesos

A continuación, se presentan los pesos asignados a las juntas del robot humanoide durante las pruebas de rendimiento de los algoritmos **DE** presentado en la sección 3.2.1, **DEAV** presentado en la sección 3.3, **MS** presentado en la sección 3.2.2, **MSAV** presentado en la sección 3.3 y **MCMT** presentado en la sección 3.2.3.

Tabla B.1. Pesos asignados a las juntas del robot para las simulaciones.

Junta	Peso	Descripción
1	.1	articulación tobillo izquierdo 1
2	.1	articulación tobillo izquierdo 2
3	.1	articulación rodilla izquierda
4	10	articulación de cadera izquierda 1
5	10	articulación de cadera izquierda 2
6	10	articulación de cadera izquierda 3
7	.1	articulación hombro izquierdo 1
8	10	articulación hombro izquierdo 2
9	10	articulación codo izquierdo
10	.1	articulación hombro derecho 1
11	10	articulación hombro derecho 2
12	10	articulación codo derecho
13	10	articulación de cadera derecha 3
14	.1	articulación de cadera derecha 2
15	10	articulación de cadera derecha 1
16	.1	articulación rodilla derecha
17	10	articulación tobillo izquierdo 2
18	10	articulación tobillo izquierdo 1

