



# UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

Centro de Investigación y Estudios de Posgrado

## **“Seguimiento de Trayectorias de un Robot Bípedo usando Control Difuso”**

Tesis para obtener el grado de:

Maestro en Ingeniería Eléctrica,  
opción en Control Automático

Presenta:

Karla Lourdes Luna Gallegos

Asesores de tesis:

Dra. Elvia Ruth Palacios Hernández

Dr. Alejandro Rodríguez Ángeles (CINVESTAV-IPN)



*Dedicado a mi familia,  
con todo mi cariño, respeto y admiración.  
A mi hijo, por ser la motivación más grande para terminar este proyecto.*



# Agradecimientos

A mi familia, por el apoyo incondicional que me han brindado y por acompañarme en cada etapa de mi vida.

A mis asesores de tesis por transmitirme sus conocimientos y experiencia, por su ayuda y paciencia.

A mis profesores de posgrado quienes fueron formadores de habilidades y capacidades durante la maestría.

A mis sinodales por los consejos y observaciones.

Al Consejo Nacional de Ciencia y Tecnología por haberme becado durante mis estudios.

A mis compañeros y amigos, al igual que a todas aquellas personas que colaboraron directa o indirectamente en la elaboración de esta tesis.



# Resumen

Este trabajo aborda el problema de seguimiento de trayectorias en las articulaciones de un robot bípedo de 10 grados de libertad (gdl). Actualmente se resuelve implementando controladores PID independientes en cada una de las articulaciones del robot. En nuestro caso, se diseñan, simulan y se comparan con los PID, controladores PI+D difusos. Este tipo de controladores se diseñan utilizando las herramientas de la lógica difusa (funciones de pertenencia, reglas de inferencia, agregación, defuzzificación, etc.) sin tener que recurrir a tablas de datos, es decir, al final del diseño se obtienen expresiones matemáticas las cuales reemplazan directamente aquellas expresiones de los PID convencionales. Además, las ganancias de este tipo de controladores son variables, resultando en un mejor desempeño en la tarea de seguimiento.

Para este trabajo se consideran trayectorias deseadas obtenidas a partir del estudio de la marcha humana, las cuales toman en cuenta los principales criterios utilizados para la estabilidad de un robot bípedo: el punto de momento cero (ZMP) y el centro de masa (CoM). Si el ZMP se puede medir o calcular mientras el robot se desplaza, es posible realizar la acción de caminar de forma constante y generar un control que estabilice el robot bípedo utilizando esta variable.

Para verificar el comportamiento cinemático del robot bípedo por medio de simulación, se utiliza el software MATLAB® que a partir del desarrollo matemático muestra de manera gráfica los resultados, observando que el robot bípedo mantiene el equilibrio estático y dinámico durante el ciclo de marcha. Con estos resultados se puede comprobar que es posible realizar la locomoción de un robot bípedo con la implementación de controladores difusos ofreciendo un alto grado de similitud al caminar humano.





# Índice general

<b>Lista de tablas</b>	<b>v</b>
<b>Lista de figuras</b>	<b>ix</b>
<b>Introducción</b>	<b>1</b>
Motivación y Planteamiento del problema . . . . .	2
Definición del problema . . . . .	3
Objetivo de la tesis . . . . .	3
Objetivos específicos . . . . .	3
Contribución de la tesis . . . . .	4
Organización de la tesis . . . . .	4
<b>1. Estado del arte</b>	<b>5</b>
1.1. Estado del Arte de los robots bípedos . . . . .	5
1.2. Locomoción bípeda y algoritmos de control . . . . .	9
1.2.1. Punto de Momento Cero (ZMP) . . . . .	9
1.2.2. Locomoción pasiva y dinámica . . . . .	10
1.2.3. SIMBICON: Control Sencillo de Locomoción Bípeda . . . . .	12
1.3. Control Difuso en robots bípedos . . . . .	13
1.4. Marcha Humana . . . . .	14
1.5. Trayectorias deseadas . . . . .	17
1.6. Condiciones de equilibrio . . . . .	21
1.6.1. Centro de Masa (CoM) . . . . .	22
1.6.2. Punto de Momento Cero (ZMP) . . . . .	24
1.7. Validación de Trayectorias . . . . .	26
1.8. Conclusiones . . . . .	27
<b>2. Modelo matemático</b>	<b>29</b>
2.1. Cinemática del robot bípedo . . . . .	29

2.1.1.	Métodos para definir la cinemática . . . . .	30
2.1.2.	Notación Denavit-Hartenberg . . . . .	31
2.1.3.	Derivando la cinemática del robot . . . . .	34
2.1.4.	Cinemática Inversa . . . . .	39
2.1.5.	Modelo Cinemática Inversa de las piernas del robot . . . . .	41
2.2.	Jacobiano del sistema . . . . .	44
2.3.	Conclusiones . . . . .	47
<b>3.</b>	<b>Introducción al Control Difuso</b>	<b>49</b>
3.1.	Control Difuso . . . . .	49
3.2.	Diferentes dominios de aplicación . . . . .	50
3.3.	El concepto de la lógica difusa . . . . .	50
3.4.	Las bases del control difuso . . . . .	51
3.4.1.	Las funciones de pertenencia . . . . .	51
3.4.2.	Los operadores de la lógica difusa . . . . .	52
3.4.3.	Universo de discurso y clases de pertenencia . . . . .	54
3.5.	Esquema de un control difuso . . . . .	55
3.6.	Conclusiones . . . . .	56
<b>4.</b>	<b>Diseño de un controlador difuso</b>	<b>57</b>
4.1.	Diseño del Control . . . . .	57
4.2.	Ley de control . . . . .	58
4.3.	Conclusiones . . . . .	65
<b>5.</b>	<b>Simulación y análisis de resultados</b>	<b>67</b>
5.1.	Simulaciones . . . . .	67
5.2.	Conclusiones . . . . .	83
	<b>Conclusiones y trabajo futuro</b>	<b>85</b>
	<b>Bibliografía</b>	<b>87</b>
	<b>A. Ecuaciones</b>	<b>91</b>
	<b>B. Programas</b>	<b>103</b>

# Índice de tablas

1.1. Porcentajes del ciclo de la marcha [1] . . . . .	16
1.2. Tabla de puntos para interpolación de trayectorias humanas en cadera y rodilla .	20
1.3. Tabla de puntos para interpolación de trayectorias humanas en cadera y tobillo (plano frontal) . . . . .	21
1.4. Localización del CoM ( ${}^{i-1}\vec{c}_i$ ) respecto al eje $i - 1$ y masas ( $m_i$ ) de cada eslabón [1].	23
2.1. Parámetros Denavit-Hartenberg para la pierna derecha . . . . .	35
2.2. Parámetros Denavit-Hartenberg para la pierna izquierda . . . . .	35
5.1. Valores de $K_p, K_i, K_d$ y $L$ para el control difuso . . . . .	79



# Índice de figuras

1.1. Robots desarrollados en la universidad de Waseda[4]. . . . .	6
1.2. Robot de la serie E[4]. . . . .	8
1.3. Planos de análisis[8]. . . . .	15
1.4. Marcha Humana . . . . .	16
1.5. Movimiento Estático y Dinámico . . . . .	17
1.6. Secuencia de Marcha [1] . . . . .	17
1.7. CoM del Robot bípedo[1]. . . . .	22
1.8. ZMP y eslabones del robot como cuerpos rígidos[1]. . . . .	24
1.9. Planificación de la marcha . . . . .	27
2.1. Representación de juntas de un robot[12] . . . . .	31
2.2. Juntas de un robot para establecer parámetros D-H[12] . . . . .	32
2.3. Estructura del bípedo . . . . .	35
2.4. Estructura del bípedo 2 . . . . .	36
2.5. Soluciones al modelo de cinemática inversa para un brazo planar de 3 juntas . . . . .	40
3.1. Regiones de combinación de entrada del control difuso PI . . . . .	52
3.2. Función de pertenencia en forma de campana . . . . .	52
3.3. Función de pertenencia del ejemplo del operador No . . . . .	53
3.4. Función de pertenencia del ejemplo del operador Y . . . . .	53
3.5. Función de pertenencia del ejemplo del operador O . . . . .	54
3.6. Esquema a bloques del control difuso . . . . .	55
4.1. Esquema a bloques del controlador PI+D difuso que se implementará . . . . .	57
4.2. Sistema de control PID convencional en tiempo continuo . . . . .	59
4.3. Sistema de control PID difuso . . . . .	61
4.4. Regiones de combinación de entrada del control difuso PI . . . . .	61
4.5. Funciones de pertenencia de $e(nT)$ y $v(nT)$ . . . . .	62
4.6. Regiones de combinación de entrada del control difuso D . . . . .	64

4.7. Funciones de pertenencia para el control difuso D . . . . .	65
5.1. Trayectorias deseadas, pierna derecha. . . . .	67
5.2. Trayectorias deseadas, pierna izquierda. . . . .	68
5.3. Representación del movimiento que realiza la pierna derecha (fase a). . . . .	68
5.4. Representación del movimiento que realiza la pierna izquierda (fase a). . . . .	69
5.5. Seguimiento de trayectorias, pierna derecha (fase a). . . . .	69
5.6. Seguimiento de trayectorias, pierna izquierda (fase a). . . . .	70
5.7. Movimiento en 2D de la pierna derecha (fase a). . . . .	70
5.8. Movimiento en 2D de la pierna izquierda (fase a). . . . .	71
5.9. Representación del movimiento que realiza la pierna derecha (fase b) . . . . .	71
5.10. Representación del movimiento que realiza la pierna izquierda (fase b) . . . . .	72
5.11. Seguimiento de trayectorias, pierna derecha (fase b). . . . .	72
5.12. Seguimiento de trayectorias, pierna izquierda (fase b) . . . . .	73
5.13. Movimiento en 2D de la pierna derecha (fase b). . . . .	73
5.14. Movimiento en 2D de la pierna izquierda (fase b). . . . .	74
5.15. Representación del movimiento que realiza la pierna derecha (fase c-d). . . . .	74
5.16. Representación del movimiento que realiza la pierna izquierda (fase c-d). . . . .	75
5.17. Seguimiento de trayectorias, pierna derecha (fase c-d). . . . .	75
5.18. Seguimiento de trayectorias, pierna izquierda (fase c-d). . . . .	76
5.19. Movimiento en 2D de la pierna derecha (fase c-d). . . . .	76
5.20. Movimiento en 2D de la pierna izquierda (fase c-d). . . . .	77
5.21. Secuencia de pasos de la locomoción de la pierna derecha de la estructura del robot. . . . .	77
5.22. Esquema del lazo abierto . . . . .	78
5.23. Seguimiento de trayectorias lazo abierto . . . . .	78
5.24. Esquema a bloques del controlador PI+D difuso lazo cerrado . . . . .	79
5.25. Comparación con la señal de referencia el PID convencional y el PI+D difuso . .	80
5.26. Acercamiento de la señal de referencia, el PID convencional y el PI+D difuso . .	80
5.27. Comparación del error que presentan los dos controladores, pierna derecha . . .	81
5.28. Criterio integral de error cuadrático ISE . . . . .	82
5.29. Criterio integral del tiempo multiplicado por el valor absoluto del error . . . . .	82
A.1. Función de pertenencia, región 1 . . . . .	91
B.1. Sistema lazo abierto . . . . .	113
B.2. PID convencional . . . . .	113

B.3. control PID convencional [1] . . . . .	114
B.4. PI+D difuso . . . . .	114
B.5. PI+D difuso . . . . .	114





# Introducción

La robótica es la rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots. Actualmente, el concepto de robótica ha evolucionado hacia los sistemas móviles autónomos, que son aquellos capaces de desenvolverse por sí mismos en entornos desconocidos y parcialmente cambiantes sin necesidad de supervisión.

Un robot móvil se define como un dispositivo formado por componentes físicos y computacionales, divididos en cuatro subsistemas: locomoción, percepción, razonamiento y comunicación. Los robots móviles terrestres son aquellos capaces de desplazarse en superficies, una plataforma robótica móvil es un robot para experimentación e investigación en tiempo real, dentro de esta área se encuentra el estudio de los robots con patas[2].

Los robots con patas presentan como ventaja la posibilidad de desplazarse en terrenos rugosos teniendo una velocidad mayor y siendo capaz de pasar sobre un obstáculo o evadirlo sin afectar el objetivo de control, además de causar menor daño al terreno pues tiene un contacto discreto, también poseen intrínsecamente una suspensión activa adaptando la longitud de sus patas a las irregularidades del terreno. Como desventaja, necesitan un equilibrio dinámico y dependen de los grados de libertad. Ambas complicaciones dependen del número de patas, mientras más patas es menor el problema de equilibrio. La complejidad del mecanismo es proporcional al número de patas y a las articulaciones del robot. La electrónica se torna compleja mientras más grados de libertad y más actuadores sean implementados al sistema, el control debe ser más preciso y más elaborado. Los robots con patas reciben diferentes nombres dependiendo del número de extremidades que lo conforman: 1 pata, monopodos; 2 patas, bípedos; 4 patas, cuadrúpedos; 6 patas, hexápodos.

La investigación en robots humanoides se ha desarrollado en generación de procesos de caminado, generación de movimientos, cooperación del cuerpo completo, etc. El diseño de robots bípedos ha sido largamente influenciado por un bípedo más sofisticado y versátil, el hombre. Por lo tanto, muchos de los modelos ó máquinas desarrollados tienen un gran parecido con los

humanos, por esta razón, para tener suficiente versatilidad, es esencial en robots humanoides bípedos desplazarse entre humanos incluyendo en las escaleras[22].

Los robots humanoides básicamente han sido desarrollados a la conveniencia de los humanos. Estos robots asistirán a la gente en emergencias, entre otras actividades. Hasta ahora, los robots humanoides como ASIMO, WABIAN, HRP y VOHNE fueron exitosamente desarrollados y tienen diferentes desempeños.

En general, un sistema de locomoción bípeda consiste en varios miembros que están interconectados con eslabones, en esencia, un robot bípedo no es más que un robot manipulador con una base móvil y desprendible. Casi cualquier modelo o máquina puede ser caracterizada en dos miembros inferiores a través de un miembro central que intenta emular.

El modo de caminar de los robots bípedos es una aproximación lo más real posible al caminar humano, para poder obtener las trayectorias que regularán el movimiento del robot se realiza un estudio de la marcha humana. El problema es complejo ya que para el estudio de trayectorias, estas se dividen en trayectorias en el plano sagital y trayectorias en el plano frontal. El robot bípedo de 10gdl que se considera en este trabajo de tesis está conformado por: 1 articulación en el tobillo, 1 articulación en la rodilla y 1 articulación en la cadera para el plano sagital; 1 articulación en el tobillo y 1 articulación en la cadera en el plano frontal, para cada extremidad.

## Motivación y Planteamiento del problema

Uno de los principales objetivos en el estudio de robots bípedos es la capacidad de poder desplazarse en todo tipo de terrenos y entornos manteniendo el equilibrio a lo largo del trayecto. A pesar de que la estabilidad de un robot bípedo caminando ha sido estudiada enormemente, esta no ha sido resuelta completamente[23].

El robot bípedo que camina es un sistema multi-etapas, el cual tiene una dinámica muy compleja y muchos factores no lineales, estas son las razones por las cuales es muy difícil mover un robot bípedo de forma estable.

La arquitectura del robot bípedo que se considera en este trabajo consta de 11 eslabones. Cada pierna del robot consta de 5gdl<sup>1</sup> de los cuales 3 brindan movimiento de revoluta en el plano sagital emulando cadera, rodilla y tobillo. Dos grados de libertad ofrecen un movimiento angular sobre el plano frontal (cadera y tobillo). El robot tiene un espacio de trabajo similar en escala al espacio de trabajo de un humano debido a que se emulan los principales movimientos de la biomecánica del humano.

Los 11 eslabones conforman un robot bípedo de 10gdl. La posición de cada articulación se

---

<sup>1</sup>grados de libertad

obtiene mediante potenciómetros de 5K Ohms en un rango de  $0^\circ$  a  $190^\circ$ . En cada articulación se emplea un motor de CD con reductor de velocidad. Para que el robot camine, es decir, para que exista una locomoción bípeda, se requiere que las articulaciones del robot sigan una trayectoria. Como son 10 articulaciones se tienen 10 trayectorias deseadas. Para esto se emplea un control inteligente como el presentado en [1] donde para cada articulación la entrada de control depende solamente del valor de posición correspondiente a su eje; en este trabajo se implementaron 10 PID para el seguimiento de trayectorias.

## Definición del problema

Una vez que la arquitectura del robot bípedo fue brevemente descrita, el problema de control puede ser formulado. En este sentido se hace notar que el mayor interés es explotar las propiedades de los controladores difusos, los cuales no requieren de modelo para su diseño para proponer una solución al problema establecido. Entonces el problema puede ser definido como:

*Considerar un robot bípedo de 10 gdl y bajo la hipótesis que:*

*A.1 Cada articulación acepta una ley de control independiente de las posiciones del resto.*

*A.2 Existe una retroalimentación de posición de cada articulación.*

*Bajo estas condiciones, el problema es diseñar 10 leyes de control  $u_i$ ,  $i = 1, \dots, 10$  tal que el  $\lim_{t \rightarrow \infty} (\theta_i - \theta_{di}) = 0$  donde  $\theta_i$  son la posición articular de cada junta y  $\theta_{di}$  son las posiciones articulares deseadas de cada junta, es decir, las trayectorias obtenidas a través del análisis del paso humano.*

## Objetivo de la tesis

El objetivo de esta tesis es diseñar y simular un sistema de control difuso para resolver el problema de seguimiento de trayectorias deseadas en las articulaciones de un robot bípedo de 10 gdl. La implementación de una estrategia de control difuso que consta de 10 controladores PI+D difusos en simulaciones numéricas.

## Objetivos específicos

- Obtener el modelo cinemático del robot bípedo.
- Obtener la cinemática inversa del robot bípedo.

- Analizar la marcha humana para la comprensión del problema.
- Simular en Matlab del modelo matemático del bípedo.
- Diseñar un controlador PI+D difuso para el seguimiento de trayectorias.
- Validar y verificar del seguimiento de trayectorias mediante resultados numéricos y simulaciones.
- Realizar una comparación entre un PID convencional y un PI+D difuso.

## Contribución de la tesis

Los objetivos contemplados fueron cubiertos logrando las siguientes contribuciones:

- Desarrollo matemático del modelo cinemático del robot para la correcta simulación de ambas piernas del robot.
- Implementación de trayectorias de locomoción humana sobre el plano sagital y el plano frontal en cada articulación del robot durante el ciclo de marcha.
- Diseño de 10 controladores PI+D difusos que regulan el seguimiento de trayectorias.
- Validación y comprobación del modelo por medio de simulaciones en un ambiente de realidad virtual empleando MATLAB.

## Organización de la tesis

Esta tesis está organizada de la siguiente manera:

En el capítulo 1 se presenta el estado del arte del seguimiento de trayectorias en robots bípedos, así como una breve descripción de los elementos necesarios para la comprensión de este tipo de robots. En el capítulo 2 se desarrolla el modelado matemático del robot bípedo, se describen la cinemática directa e inversa, así como el Jacobiano. En el capítulo 3, se presenta un panorama general del control difuso y en el capítulo 4 se desarrolla el diseño de los controladores PI+D difusos para el problema establecido en esta tesis. En el capítulo 5 se presentan los resultados numéricos de la implementación de este tipo de controladores al robot bípedo. Finalmente se establecen las conclusiones del trabajo de tesis.

# Capítulo 1

## Seguimiento de Trayectorias en robots bípedos: Estado del arte

### 1.1. Estado del Arte de los robots bípedos

En general, la historia de la robótica puede ser clasificada en cinco generaciones. Las dos primeras se centraron en la gestión de tareas repetitivas con autonomía limitada. La tercera generación impulsó la visión artificial. La cuarta incluye movilidad avanzada en exteriores e interiores y la quinta entraría en el dominio de la inteligencia artificial en lo cual se está trabajando actualmente.

Los robots bípedos, desde el punto de vista semántico de la palabra bípedo, se pueden definir como aquellos en los que el movimiento del robot depende del movimiento de dos extremidades mecánicas a modo de piernas. Estos robots surgen del deseo de los humanos de construir máquinas que se asemejen a ellos y les sirvan de ayuda en sus labores cotidianas adaptándose lo más posible al entorno que se ha creado. Desde un punto de vista histórico, los robots bípedos se engloban en dos grupos: los robots pasivos y los robots activos.

Los robots pasivos fueron cronológicamente los primeros. Estos sistemas mecánicos tenían como característica fundamental que podían caminar sin necesidad de control ni de ningún tipo de actuadores o motores, es decir, que su movimiento era generado únicamente por la acción de la fuerza de la gravedad. La principal ventaja de estos robots pasivos es que no requerían de energía externa para caminar, pero su desventaja radicaba principalmente en que su direccionalidad está limitada y era prácticamente imposible que estos robots subieran pendientes.

Los robots activos son aquellos que sí requieren la actuación de fuerzas externas, es decir,

motores que posibiliten su movimiento. Estos robots activos han ido evolucionando desde su diseño más simple, con un único actuador, hasta robots mucho más complejos que cuentan con 30 motores, con sus respectivos sistemas de control, controlados por autómatas. [3]

Una consecuencia directa del aumento de complejidad de estos robots es que requieren para su funcionamiento un mayor número de componentes mecánicas, eléctricos y electrónicos. Los humanoides es una rama de la robótica muy activa, son muchos los proyectos e investigaciones que se están desarrollando actualmente a nivel mundial. Por tal motivo, se verán los principales proyectos que han tenido mayor trascendencia mundial hasta la fecha.

En 1967 se diseñó un mecanismo a partir del estudio de las piernas humanas, el robot WAM-1 que asentaría las bases para posteriores diseños que emplean un sistema de locomoción bípedo.

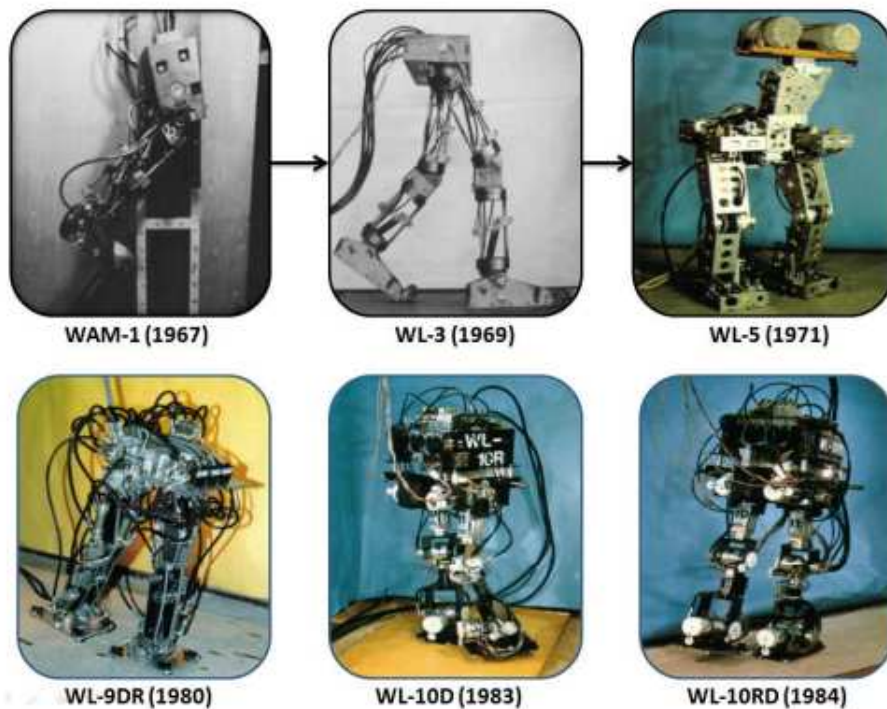


Figura 1.1: Robots desarrollados en la universidad de Waseda[4].

El robot WAM-1 fue el primero de la serie de robots que culminará con el desarrollo del robot WL-12RDIII (1986). Esta serie de robots ha servido de base para el desarrollo de otros sistemas y ha supuesto una valiosa fuente de conocimiento en el campo de los robots bípedos. El robot WAM-1 tuvo su sucesor en el WL-3, que era capaz de elevarse y sentarse por medio de actuadores electrohidráulicos. Otra mejora en este robot se encontraba en el sistema de control, introduciendo el concepto de maestro-esclavo.

En 1968, aparece el primer sistema bípedo teleoperado, el mecanismo de Witt, desarrollado por la Universidad de Oxford con objeto de obtener un sistema capaz de ayudar a las personas discapacitadas. El sistema contaba con dos piernas telescópicas, pero su movimiento se encontraba muy restringido, sin poder llevar a cabo un movimiento con cambios de plano.

En 1969, Vukobratovic junto con el equipo de desarrollo del instituto Mikhail Pupin de Belgrado enuncia el concepto de Punto de Momento Cero (Zero Moment Point, ZMP). Es aquí, donde tiene lugar uno de los avances fundamentales en el campo de la investigación, que permitió el posterior desarrollo de mecanismos y robots mucho más complejos. El empleo del ZMP se hace fundamental a la hora de diseñar el control dinámico de cualquier robot bípedo, y es un elemento básico a la hora de planificar la estrategia de diseño, tanto de la máquina como de la secuencia de movimiento.

En 1971 aparece el siguiente robot de esta serie, el WL-5, al que se le añadió un nuevo grado de libertad en la cadera que le permitía desplazar su centro de gravedad en el plano horizontal, consiguiendo así un movimiento mucho más estable. Este robot podía cambiar de dirección mientras caminaba, gracias al sistema de control implementado por medio de una minicomputadora. El robot constaba de un total de 11gdl y fue la base de posteriores diseños de la serie WL, así como del robot WABOT-1.

La serie de robots bípedos continuó a lo largo de los años ochenta, introduciéndose sucesivas mejoras en cada uno de los modelos desarrollados. En 1980 aparece el modelo WL-9DR, que fue el primer sistema que imitaba de forma casi perfecta la cadencia del paso humano. Empleaba una computadora de 16-bits para llevar a cabo la acción de control. El robot tenía una cadencia de 10 seg/paso y el pie tenía cuatro puntos de apoyo.

En los años 1982 y 1984 aparecen dos nuevas mejoras en esta serie de robots, los modelos WL-10R y WL-10RD. En estos robots se emplearon nuevos materiales: fibra de carbono y plásticos (más ligeros que disminuyeron el peso del robot facilitando su control). Se añadieron grados de libertad de forma que estos robots fueran capaces de andar hacia adelante, hacia atrás y lateralmente. El menor peso de los robots permitía una mayor velocidad de paso, de aproximadamente 4.4 seg/paso. El robot WL-10RD presentaba un control dinámico del movimiento, para llevar a cabo este control se añadieron al robot sensores en tobillos y caderas, pudiendo conocer así las fuerzas en estas articulaciones en todo momento.

La serie WL tuvo su culminación en el año 1986 con los modelos WL-12 y WL-12RDIII. Este último modelo podía desenvolverse en un entorno humano, es decir, era capaz de subir y bajar escalones como también podía andar en un entorno con múltiples obstáculos. El movimiento era mucho más parecido al del hombre que en el caso de otros robots. Estos bípedos presentaban un método de control muy elaborado, de tipo adaptativo, que dotaba al sistema de mayor estabilidad. Empleando los conocimientos adquiridos durante el desarrollo de las series WL y

WAM (serie que seguía los principios similares a los de la serie WL, pero que luego se centró en las extremidades superiores), la Universidad de Waseda desarrolla el robot WABOT-1 (1973), siendo el primer robot bípedo con forma humanoide y autónomo, construido en el mundo.

A lo largo de los años ochenta, la empresa japonesa Honda, empezó a desarrollar una serie de sistemas bípedos a partir de los cuales surgieron en los noventa diseños de humanoides más complejos. Estos primeros sistemas fueron la serie E, construyéndose siete modelos, desde el E0 (1986), hasta el E6 (1993). El modelo E0 era un robot de dos piernas que imitaba el movimiento humano, desplazando una pierna a continuación de otra, sin embargo la velocidad de paso era muy baja, así como la estabilidad del diseño. Los siguientes modelos E1, E2, E3, pretendían solucionar estos problemas. Fueron desarrollados entre los años 1987 y 1991, mejorándose sustancialmente la velocidad de paso y la estabilidad, de forma sucesiva con cada uno de los modelos.

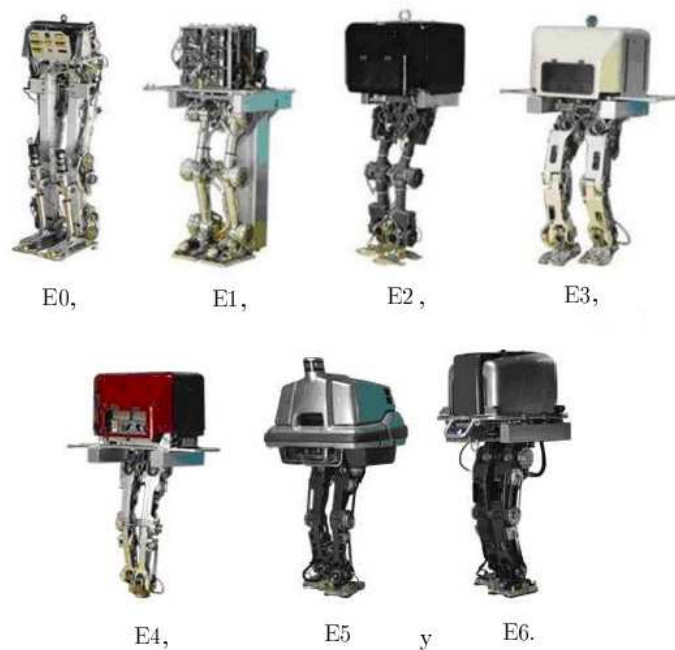


Figura 1.2: Robot de la serie E[4].

La serie culmina con los modelos E4, E5 y E6, desarrollados entre los años 1991 y 1993. Con estos tres modelos, Honda consolida su tecnología de estabilización del paso, basada en tres técnicas de control básicas:

1. Control de la reacción del suelo.
2. Control de la posición del ZMP.



3. Planificación del posicionamiento de los pies en el paso.

Una vez desarrollado el modelo E6, el siguiente objetivo de Honda fue unir el sistema de locomoción desarrollado a un tronco, con objeto de diseñar un robot bípedo capaz de interactuar con el ser humano. Así, en el año 2000, Honda desarrolla el que es su último modelo de robot humanoide, el ASIMO. [4]

## 1.2. Locomoción bípeda y algoritmos de control

En esta sección se revisará el estado del arte de la locomoción de robots bípedos. Con base en el trabajo de investigación realizado en este campo, se hará un análisis descriptivo de los aspectos más importantes acerca del estudio de diversos algoritmos de control representativos.

### 1.2.1. Punto de Momento Cero (ZMP)

El punto de momento cero (ZMP), por sus siglas en inglés Zero-Moment Point, es una técnica que se utiliza en el control de robots bípedos introducida por primera vez hace más de 35 años, y es actualmente una de las más extendidas. Establece un criterio de estabilidad dinámica para el robot que permite generar patrones de locomoción. Este concepto fue aplicado con éxito por primera vez en 1984 en Waseda University, Laboratorio de Ichiro Kato, en el robot WL-10RD, el primero equilibrado dinámicamente, y desde entonces en otros múltiples robots.

El ZMP se puede definir como el punto  $p_{ZMP}$  en el suelo tal que el momento neto de las fuerzas externas no tiene componente sobre los ejes horizontales. Cuando  $p_{ZMP}$  existe dentro del polígono de soporte, el contacto entre el suelo y el pie es estable. Cuanto más cercano esté  $p$  al centro de la superficie de soporte, más robustez se conseguirá. Cuando el ZMP está fuera del polígono de soporte, el robot se inclina rotando sobre alguno de los bordes de dicho polígono. El criterio de que el ZMP exista dentro del polígono de soporte es condición necesaria y suficiente para garantizar la estabilidad dinámica del robot.

De forma intuitiva, la condición del ZMP asegura que el movimiento del cuerpo será tal que el pie estará plano en el suelo y por tanto no caerá. ASIMO (Advanced Step in Innovative Mobility), creado por la empresa Honda, es uno de los ejemplos más representativos de los robots caminadores bípedos basados en el algoritmo del ZMP. Las ventajas de aplicación del concepto del ZMP y los algoritmos derivados podrían ser resumidas como sigue:

- En sí mismo, el ZMP constituye una metodología sistemática para la generación de trayectorias para la locomoción.

- Se asegura que las trayectorias de locomoción sean dinámicamente estables.
- Constituye una variable de control en línea (on-line). Aunque las trayectorias se planeen generalmente fuera de línea (off-line), lógicamente existen perturbaciones. Se puede establecer un control articular tal que en caso de que el ZMP se desvíe de su posición calculada, regrese a ésta.
- En caso que la perturbación no se consiga compensar y el ZMP salga del polígono de soporte, el cálculo del grado de separación del ZMP permite tener una medida de la perturbación. Esta se puede corregir entonces modificando la trayectoria de tal manera que el nuevo paso se sitúe de tal forma que el nuevo polígono de soporte incluya el ZMP, volviendo a recuperar la estabilidad.

Los siguientes puntos se plantean como inconvenientes al ZMP, siempre recordando que los mejores robots que caminan usan este método:

- La resolución del algoritmo es de alta complejidad. Con el crecimiento de capacidad de cálculo de los ordenadores no supondrá un problema en el futuro.
- El uso de cinemática inversa también conlleva problemas. Debido a que se necesita gran precisión para tener una trayectoria estable, los errores numéricos pueden tener implicaciones muy perjudiciales de desequilibrio.
- El uso de cinemática y dinámica inversas puede tener problemas por singularidades, ya que cerca de ellas se producen errores numéricos muy importantes. Los más comunes son divisiones por cero que conllevan explosiones numéricas, y por tanto comportamientos bruscos del robot.
- Dependiendo de la configuración (articulación en su límite, soporte sobre un pie, sobre dos, sobre un borde de un pie...) cambian los grados de libertad del sistema, y por tanto deben usarse matrices de dinámica diferentes, esto eleva la complejidad. Otra opción sería evitar configuraciones cercanas a singularidades, pero esto limitaría la movilidad “natural” del robot.

### 1.2.2. Locomoción pasiva y dinámica

#### Locomoción Pasiva

El concepto de locomoción pasiva, es decir, sin actuadores, fue introducido por primera vez en 1988 por Tad McGeer, quien estudió y construyó un caminador pasivo. La virtud de

este mecanismo es que no necesita energía externa ni ningún tipo de control para caminar por una pendiente, el movimiento viene “propulsado” por la energía gravitacional de la bajada. El sistema actúa como dos péndulos acoplados. La pierna soporte es un péndulo invertido, y la pierna oscilante es un péndulo normal unido a la pierna soporte. Con la distribución de masas y longitudes adecuada, se consigue un ciclo límite estable, una trayectoria nominal que se repite. Inicialmente el mecanismo no tenía rodillas, pero una mejora posterior implementó esta articulación manteniendo la estabilidad. El movimiento lateral se inhibe al contar con cuatro piernas.

McGeer hizo un estudio exhaustivo de la mecánica y dinámica del caminador. Demostró las condiciones necesarias para asegurar la estabilidad del ciclo límite. Cabe destacar una diferencia sustancial respecto al algoritmo del ZMP: el caminador pasivo describe trayectorias inestables (por definición, se comporta como un péndulo invertido inestable), manteniendo el ciclo límite estable.

También, es de gran interés el hecho que los movimientos de este simple mecanismo, que no tienen actuación ni control, tengan una gran semejanza al de los humanos. Esto sugiere la idea de que los humanos aprovechamos las dinámicas propias de nuestro cuerpo para optimizar el consumo energético de la locomoción.

Este descubrimiento abrió una nueva puerta de investigación: robots que aprovechen los mismos conceptos de la locomoción pasiva aplicándolos para conseguir locomoción sobre llanos y subidas, lógicamente añadiendo elementos actuadores y de control. Esto se conoce como locomoción dinámica.

Algunas características de los robots basados en el caminador pasivo son:

- La forma de la “suela” de los pies está muy estudiada. Es una forma curvada que permite que el movimiento de avance sea suave. Gran parte del movimiento talón–punta del pie tan “humano” que tienen es gracias a esta característica. El robot del instituto de tecnología de Massachusetts (MIT) también tiene forma curvada en el plano lateral, facilitando su oscilación y asegurando estabilidad intrínseca en este plano.
- Presentan elementos pasivos, por ejemplo muelles. Estos elementos con las características adecuadas permiten almacenar energía en una fase de la locomoción para liberarla en otra fase. De esta forma se consigue un ahorro energético muy importante, además de un mejor comportamiento especialmente en el impacto del pie en el suelo. También pueden incluir amortiguadores mecánicos que proporcionan estabilidad en el momento del impacto con el suelo.

El resultado de estos estudios de robots bípedos basados en caminadores pasivos se puede resumir en los siguientes puntos:

- Controladores simples.
- No usan actuadores de alta potencia.
- No usan controladores de alta frecuencia, que no serían comparables al comportamiento humano.
- Tienen eficiencias y movimientos parecidos a los humanos.

## Locomoción Dinámica

Steve Collins y sus compañeros hicieron interesantes estudios de robots humanoides que usan los conceptos de la locomoción pasiva. Desarrollaron tres robots diferentes, sustituyendo la “propulsión” gravitacional por actuadores simples.

El bípedo “Cornell” está basado en el mecanismo pasivo, y propulsado por motores eléctricos y muelles que ayudan al movimiento de los tobillos cuando se levantan los pies. Tiene cinco grados de libertad (dos tobillos, dos rodillas y la cadera), los brazos están unidos mecánicamente a la pierna opuesta. El bípedo “Delft” tiene una morfología similar, pero con actuación neumática. Y el último fue desarrollado en el MIT (Massachusetts Institute of Technology). Tiene la peculiaridad de estar controlado mediante un aprendizaje con refuerzo (reinforcement learning), que converge automáticamente a la estrategia de control óptima.

El bípedo “Cornell” está específicamente diseñado para minimizar la energía de propulsión. Mediante una política en los actuadores adecuada, se consigue que siempre aporten trabajo positivo y no se usen como disipadores. Este hecho está en contraposición con el problema que presentan muchos otros robots: en el momento en el que el pie impacta en el suelo se disipa energía debido a que el robot está frenando, afectando así al rendimiento, ya que luego hay que recuperarla. Aunque los robots del MIT y “Delft” no fueron diseñados específicamente para usar poca energía ambos tienen las ventajas inherentes de los caminadores pasivos en lo que respecta a bajo consumo.

### 1.2.3. SIMBICON: Control Sencillo de Locomoción Bípeda

SIMBICON (Simple Biped Locomotion Control) es un algoritmo de control para robots bípedos que usa técnicas de retroalimentación orientadas a conseguir el equilibrio a partir del estudio de las dinámicas involucradas. Permite una gran cantidad de estilos de locomoción en tiempo real (hacia delante, hacia atrás, de lado, girando, corriendo, saltando, gateando...).

Este algoritmo funciona bien tanto en 2D como en 3D, y con parámetros adecuados se puede conseguir un estilo de caminar muy parecido al humano y muy estable frente a perturbaciones

externas o cambios de terreno. Cabe destacar que sólo se ha probado con simulaciones físicas (con parámetros reales), pero no en robots bípedos.

El algoritmo SIMBICON proporciona una serie de ventajas: gran facilidad para crear estilos de locomoción, resulta muy estable frente a perturbaciones, y su simplicidad y facilidad de programación. Puede ser una opción muy interesante para el desarrollo de la locomoción de un robot humanoide.[5]

### 1.3. Control Difuso en robots bípedos

Numerosas instituciones educativas han construido su propio robot humanoide, en el cual su desarrollo implica la aplicación de diversas herramientas de la Ingeniería Mecatrónica como el diseño de controladores, inteligencia artificial, cálculo de las ecuaciones matemáticas, etc. En nuestro país este campo no ha sido desarrollado lo suficiente, sin embargo, es interesante trabajar con estos modelos en el desarrollo de controladores que permitan probar las diferentes estrategias de control desde el punto de vista cinemático y dinámico. Técnicas como el control difuso proveen una poderosa herramienta para resolver problemas con ambientes impredecibles. Los controladores difusos pueden desempeñarse mejor que un controlador PID lineal puesto que estos no poseen características no lineales.

En la Universidad Nacional de Ingeniería de Lima-Perú se diseñó un robot bípedo de 6gdl, el cual fue modelado dinámicamente para lograr controlar la trayectoria de cada una de sus piernas. Este control está basado en algoritmos de lógica difusa, los cuales están aplicados directamente a los motores DC ubicados en cada articulación del robot. El problema que el control PID presenta está en la sintonización de cada parámetro. El diseño mencionado cuenta con 18 parámetros para ser sintonizados; se optó por algoritmos de lógica difusa para resolver este inconveniente, utilizando Mamdani, además de las reglas de pertenencia. Se sustituyó entonces el PID inicial por un PD difuso.[6]

En 2010, Lin, Chang, Lyu, Wang y Li [7] aplicaron la lógica difusa en el ajuste de la estabilización de un robot bípedo, logrando desplazamientos en un plano horizontal y en el ascenso de escalones, suprimiendo la inestabilidad inherente al sistema y reduciendo la duración de oscilaciones transitorias.

En la Universidad de Guadalajara, se implementó el control PD Difuso para controlar el balance de los robots humanoides, debido a que no es necesario un modelado matemático de la planta por ser técnicas libres de modelo. Además, usando lógica difusa, es factible aprovechar el conocimiento de un humano experto sobre el proceso que se desea controlar. [8]

En Korea, se realizó una tesis donde el objetivo principal fue mantener el balance del robot al caminar. Una de las soluciones para resolver este problema es implementar un algoritmo de

control inteligente y robusto basado en el concepto de ZMP. En el artículo [9] la postura del robot bípedo al caminar es controlada por un algoritmo difuso basado en sensores de fuerza.

En Portugal se realizó una investigación donde la solución propuesta para el control de un robot bípedo estuvo basada en una red neuronal difusa de 1er orden (Takagi-Sugeno). Esta red usa la posición del momento cero y su variación como entrada y la corrección longitudinal del cuerpo del robot como salida. La estructura del robot que se presenta constituye un péndulo invertido. El control neuronal difuso permite excelentes resultados al controlar sistemas complejos, el control del robot bípedo se logra con datos de simulación, manipulación del sistema real y con el conocimiento de un balance bípedo.[10]

El artículo [11] se enfoca en realizar un control total del movimiento de un robot bípedo de manera on-line basado en un control de retroalimentación sensorial. El robot bípedo mantendrá el balance durante la locomoción en presencia de perturbaciones externas. El algoritmo utilizado para este trabajo se basó en un sistema neuronal difuso, donde las reglas de conexión son representadas en forma de si-entonces. Mientras más reglas existan la representación de los datos procesados será mejor, pero la complejidad del sistema se incrementará.

## 1.4. Marcha Humana

La marcha es un proceso mecánico de locomoción del ser humano con el que el cuerpo se desliza de un lugar a otro siendo este sostenido por ambas piernas. La locomoción humana normal se ha descrito como una serie de movimientos alternantes, rítmicos, de las extremidades y del tronco que determinan un desplazamiento hacia delante del centro de gravedad. A pesar de existir diferencias en la forma de caminar entre un individuo y otro, estas diferencias son mínimas y caen dentro de ciertos límites. [24]

El caminar humano suele adaptarse a su entorno según el terreno sobre el cual se desliza, superficies suaves, rugosas, duras, resbalosas, ásperas, subidas o bajadas, etc. También existen otros factores que afectan el movimiento de la marcha tales como el calzado, carga, práctica deportiva, condiciones físicas del individuo, entre otros. Puesto que la marcha en su totalidad es una actividad muy compleja, se toma como modelo simplificado el andar de un sujeto normal en una superficie plana.

Tradicionalmente el análisis de la marcha humana se estudia en tres planos de referencia primarios: el plano transversal, sagital y frontal o coronal.

El ciclo de la marcha comienza cuando el pie contacta con el suelo y termina con el siguiente contacto con el suelo del mismo pie. La longitud del paso completo es la distancia lineal entre los sucesivos puntos de contacto del talón del mismo pie y la longitud del paso es la distancia

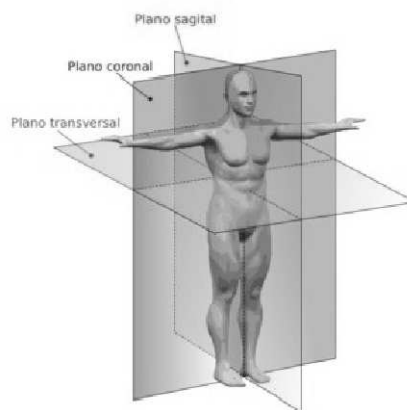


Figura 1.3: Planos de análisis[8]

lineal en el plano de progresión entre los puntos de contacto de un pie y el otro pie.[25]

Cada ciclo se compone de dos pasos durante una marcha normal; se pueden distinguir dos periodos o fases: la fase de apoyo en el que se tiene doble soporte representa aproximadamente un 20 % y la fase de balanceo se refiere al periodo cuando sólo una pierna está en contacto con el suelo y representa alrededor del 80 % del ciclo. La ausencia de un período de doble apoyo distingue el correr del andar.

La Tabla 1.1 muestra los porcentajes correspondientes a cada fase durante el ciclo de la marcha, donde las fases de soporte se definen de la siguiente manera:

- FSD: Intervalo de tiempo donde ambos pies están en contacto con el suelo. Fase de apoyo 20 % (Fase de soporte doble).
- FSS: Intervalo de tiempo en el cual una pierna se balancea y la otra permanece en el suelo. Fase de balanceo 80 % (Fase de soporte simple)

La caminata inicia con ambos pies extendidos y sobre el suelo en donde el equilibrio no presenta dificultad, al levantar uno de los pies la estabilidad debe mantenerse en un solo pie, debido a esto, dentro del ciclo de la marcha se debe poner principal atención a esta fase de balanceo. El medio ciclo termina cuando los dos pies vuelven a tocar el suelo, el siguiente medio ciclo es igual al anterior, solamente que hay un intercambio de piernas.

El desglose del ciclo de marcha no se puede aplicar en los primeros pasos, ya que las secuencias motrices no son las mismas. El partir de un estado de reposo presenta diferentes posiciones en los eslabones.

Periodo	Porcentaje
Primera FSD	0 % - 10 %
Primera FSS	10 % - 50 %
Segunda FSD	50 % - 60 %
Segunda FSs	60 % - 100 %

Tabla 1.1: Porcentajes del ciclo de la marcha [1]

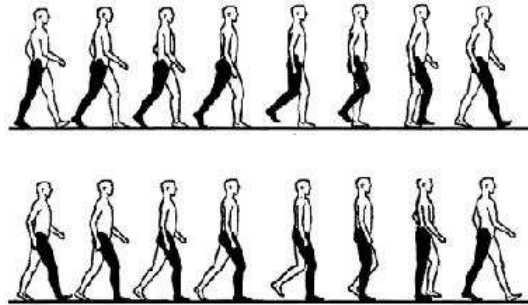


Figura 1.4: Marcha Humana

En la locomoción bípeda, una de las tareas más importantes es la conservación del equilibrio o balanceo del robot. Para esto se hace el estudio de los ciclos de la marcha desde dos enfoques, el estático y el dinámico.

El balanceo estático se basa en el concepto del centro de masa (CoM), el CoM del robot es el punto donde se concentra la masa total de la estructura física. La condición de balanceo estático se reduce a la proyección del CoM dentro del área formada por la planta de los pies. Esto implica que si el robot se detiene en cualquier momento, se encontrará en un estado de equilibrio. La marcha estática obtenida a partir de la definición anterior, presenta desventajas ya que realiza movimientos lentos y poco ágiles, en general este tipo de marcha requiere de pies grandes y un mayor consumo de energía por parte de los actuadores.

El balanceo dinámico engloba los efectos dinámicos y las fuerzas externas que actúan sobre el robot durante su desplazamiento. El concepto del punto de momento cero (ZMP) provee la condición necesaria y suficiente de equilibrio dinámico durante la locomoción bípeda, el ZMP se define como el punto dentro del área de soporte para el cual se mantiene que la sumatoria de los momentos que experimenta el robot son iguales a cero. Se realiza una marcha dinámica si el ZMP se encuentra dentro del área de soporte durante las fases del ciclo de la marcha. A diferencia de la marcha estática, durante una marcha dinámica la proyección del CoM puede encontrarse fuera del área de soporte en periodos cortos de tiempo. [26]



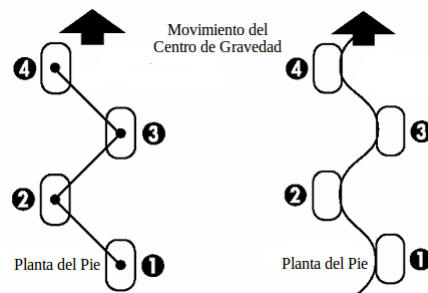


Figura 1.5: Movimiento Estático y Dinámico

En general la marcha dinámica ofrece mejor movilidad pues permite realizar movimientos más complejos y naturales, sin embargo, puede presentar comportamientos inestables.

## 1.5. Trayectorias deseadas

Para que el desplazamiento del robot sea autónomo, se considera una secuencia de marcha en el robot. Esta secuencia permitirá la locomoción en simulación del robot bípedo. La Figura 1.6 muestra la secuencia considerada.

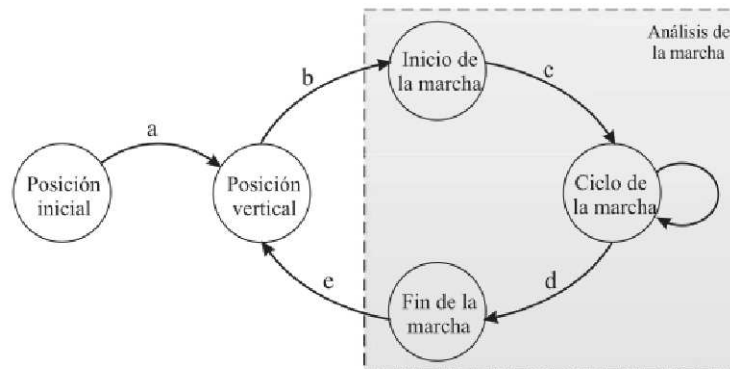


Figura 1.6: Secuencia de Marcha [1]

La secuencia está conformada por 5 etapas definidas como:

- Posición inicial.- etapa donde el robot adopta una postura en cuclillas, soportado por ambos pies, validadas por CoM.
- Posición vertical.- postura erguida del robot, donde todas las variables articulares son igual a cero ( $\theta = 0$ ).

- Inicio de marcha.- posición del robot donde la proyección del CoM sobre el piso reside en la planta del pie derecho.
- Ciclo de la marcha.- etapa donde se ejecutan las trayectorias del ciclo de la marcha, validadas por el ZMP.
- Fin de marcha.- posición donde la proyección del CoM sobre el piso reside en el polígono formado por ambos pies del robot.

Cabe destacar, que durante todas las etapas excepto “ciclo de la marcha” el robot bípedo permanece estático, mientras que en este último, el robot se encuentra en movimiento desarrollando el ciclo de la marcha.

La transición de una etapa a otra, se realiza mediante la implementación de trayectorias. Para que un robot bípedo siga unas trayectorias pre-definidas, la generación de estas trayectorias juega un papel muy importante. Son las trayectorias las que determinan la forma de la marcha, la velocidad del robot y su estabilidad. Para obtener estas trayectorias existen dos principales métodos: medir variables biométricas y generar las trayectorias de forma teórica. Este último método se considerará en el presente trabajo de tesis debido a que las variables biomédicas rara vez son determinísticas y éstas varían enormemente de una persona a otra, se optó entonces por tomar resultados de investigaciones ya realizadas en este estudio.

Para la generación de estas trayectorias se desea que la velocidad inicie y termine en cero, por tal motivo, se utilizan funciones cúbicas que tienen como perfil de velocidad una parábola y su perfil de aceleración es lineal[35]. Entonces, se definen los polinomios de la forma:

$$\theta(k) = a_0 + a_1k + a_2k^2 + a_3k^3 \quad (1.1)$$

cumpliendo con las siguientes restricciones:

$$\begin{aligned} \theta(k_0) &= \theta_0, \\ \theta(k_f) &= \theta_f, \\ \dot{\theta}(k_0) &= 0, \\ \dot{\theta}(k_f) &= 0, \end{aligned} \quad (1.2)$$

donde

$\theta_0$  indica la posición inicial de la articulación

$\theta_f$  la posición final ó deseada.

$k$  representa el ciclo de marcha total, ( $k_0 = 0\%$ ,  $k_f = 100\%$ )

Resolviendo lo anterior para los coeficientes del polinomio  $a_i$  se tiene:

$$\begin{aligned}
a_0 &= \theta_0, \\
a_1 &= 0, \\
a_2 &= \frac{3}{k_f^2}(\theta_f - \theta_0), \\
a_3 &= \frac{2}{k_f^3}(\theta_f - \theta_0).
\end{aligned} \tag{1.3}$$

De acuerdo a estas ecuaciones los polinomios que definen las trayectorias de cadera, rodilla y tobillo durante la transición del estado “**posición inicial - posición vertical**” son:

$$\begin{aligned}
\theta(k)_{cadera} &= -75 + 22.5 \times 10^{-3}k^2 - 15 \times 10^{-5}k^3, \\
\theta(k)_{rodilla} &= 100 - 3 \times 10^{-2}k^2 + 2 \times 10^{-4}k^3, \\
\theta(k)_{tobillo} &= -25 + 7.5 \times 10^{-3}k^2 - 5 \times 10^{-5}k^3,
\end{aligned} \tag{1.4}$$

para esta transición las articulaciones que prestan movimiento en el plano frontal (cadera y tobillo) se mantienen igual a cero, es decir:

$$\theta(k)_{cadera-frontal} = \theta(k)_{tobillo-frontal}. \tag{1.5}$$

los polinomios para la transición “**posición vertical - inicio de marcha**” de la pierna derecha son:

$$\begin{aligned}
\theta(k)_{cadera} &= (0.503 \times 10^{-3}k^2 - 0.435 \times 10^{-5}k^3) * 37.5, \\
\theta(k)_{rodilla} &= (1.229 \times 10^{-3}k^2 - 1.064 \times 10^{-5}k^3) * 3, \\
\theta(k)_{tobillo} &= (0.465 \times 10^{-3}k^2 - 0.402 \times 10^{-5}k^3) * -30, \\
\theta(k)_{cadera-frontal} &= (0.4194x10^{-3}k^2 - 0.363 \times 10^{-5}k^3) * 1.5 \\
\theta(k)_{tobillo-frontal} &= (0.719x10^{-3}k^2 - 0.622 \times 10^{-5}k^3) * -1.3
\end{aligned} \tag{1.6}$$

así mismo para la pierna izquierda se emplean:

$$\begin{aligned}
\theta(k)_{cadera} &= (0.27 \times 10^{-3}k^2 - 0.234 \times 10^{-5}k^3) * 78, \\
\theta(k)_{rodilla} &= (0.0268 \times 10^{-3}k^2 - 0.0232 \times 10^{-5}k^3) * 150, \\
\theta(k)_{tobillo} &= (-0.0672 \times 10^{-3}k^2 + 0.0582 \times 10^{-5}k^3) * 290, \\
\theta(k)_{cadera-frontal} &= (-0.0941 \times 10^{-3}k^2 + 0.0814 \times 10^{-5}k^3) * -13.32 \\
\theta(k)_{tobillo-frontal} &= (0.653 \times 10^{-3}k^2 - 0.566 \times 10^{-5}k^3) * -2.92
\end{aligned} \tag{1.7}$$

los polinomios que definen los valores de las articulaciones de la transición “**inicio de marcha - ciclo de la marcha - fin de marcha**” se obtienen por medio de la instrucción “polyfit” dentro del ambiente de MATLAB, esto es:

$$p(x) = \text{polyfit}(\vec{x}, \vec{y}, n), \quad (1.8)$$

donde  $\vec{x}$  representa el vector que contiene los valores  $x_i$ ,  $\vec{y}$  es el vector que contiene los valores  $y_i$  y  $n$  representa el grado del polinomio. Esta instrucción devuelve los coeficientes del polinomio  $p(x)$  de grado  $n$  de la forma:

$$p(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1} \quad (1.9)$$

La Tabla 1.2 muestra el conjunto de puntos recuperados de las trayectorias humanas empleados en el cálculo de los polinomios que definirán las trayectorias deseadas de cadera y rodilla sobre el plano sagital. En el caso particular del tobillo se desea que siempre se mantenga

Rodilla		Cadera	
$x_i$ [%ciclo]	$y_i$ [°]	$x_i$ [%ciclo]	$y_i$ [°]
0	5.65	0	25.5
3	6.5	5.8	29.75
10	13.3	15.9	23.375
12.7	20	27.35	0
20	15	40	-13.75
25	11	50	-19.375
30	7.22	66.3	0
39	4.44	78.9	22.5
50	12.22	83.15	25
60	38.88	87.35	26.25
72	59	97.35	22.5
80	50.5	100	23.4
89	13.88	-	-
96.5	4	-	-
100	4.44	-	-

Tabla 1.2: Tabla de puntos para interpolación de trayectorias humanas en cadera y rodilla

horizontal al piso, por lo tanto, la ecuación que soluciona la restricción anterior es:

$$p(k)_{\text{tobillo}} = -(p_{\text{cadera}} + p_{\text{rodilla}}) \quad (1.10)$$

Estas son las trayectorias deseadas para el plano sagital durante el ciclo de la marcha.

Para el plano frontal se realiza de la misma manera que el método anterior tomando ahora los datos de la Tabla 1.3. Para la definición de la trayectoria deseada del tobillo se emplea

Cadera - frontal	
$x_i$ [%ciclo]	$y_i$ [°]
0	1
15	6.5
37	9
50	4
70	0
72	-11
80	1.7
89	-1.5
100	1.5

Tabla 1.3: Tabla de puntos para interpolación de trayectorias humanas en cadera y tobillo (plano frontal)

el polinomio obtenido en la ecuación multiplicado por un factor de escalamiento  $f_e = -1.5$  quedando como:

$$p(k)_{\text{tobillo-frontal}} = (f_e)p(k)_{\text{cadera-frontal}} = (-1,5)p(k)_{\text{cadera-frontal}} \quad (1.11)$$

El factor de escalamiento  $f_e$  permite balancear al robot hacia un costado, lo que da posibilidad de levantar la planta del pie contraria al pie de soporte. El valor del  $f_e = -1.5$  se define por medio del proceso iterativo de planificación de marcha.

## 1.6. Condiciones de equilibrio

Planteado por Debra J, Rose 2005 [28], el equilibrio se define como el proceso por el cual se controla el centro de masa (CoM) del cuerpo respecto a la base de sustentación, sea estática o dinámica. Con frecuencia se considera que estar de pie y erguidos en el espacio constituye una tarea del equilibrio estático y que inclinarse en el espacio o caminar son tareas del equilibrio dinámico.

El CoM es el punto donde se concentra la masa total de la estructura física, el punto de momento cero (ZMP) es el punto en el piso donde la suma de todos los momentos de las fuerzas

activas es igual a cero. El CoM ofrece una metodología adecuada para conocer la condición de equilibrio estático y el ZMP para el equilibrio dinámico.

### 1.6.1. Centro de Masa (CoM)

Para obtener el CoM del robot bípedo se considera la localización individual de los centros de masa de cada eslabón. La masa total del robot, asumiendo que el robot tiene  $N=11$  eslabones, se obtiene mediante la siguiente ecuación:

$${}^0\vec{c}_i = \sum_{i=1}^N m_i {}^0\vec{c}_i / M$$

donde  $m_i$  es la masa del eslabón  $i$ ,  ${}^0\vec{c}_i$  el centro de masa del eslabón  $i$  respecto al sistema 0,  $M$  es la masa equivalente del robot y el vector  ${}^0\vec{c}$  es el CoM resultante de toda la estructura.

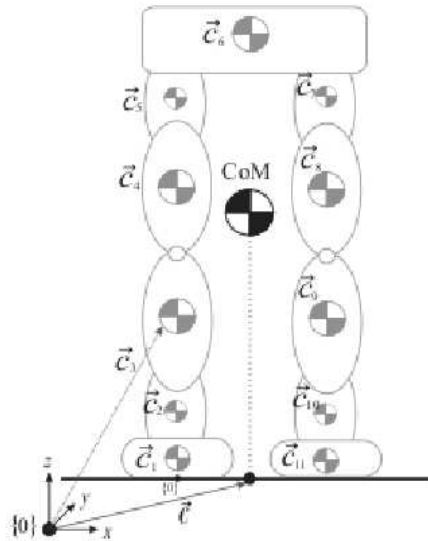


Figura 1.7: CoM del Robot bípedo[1].

Para la estabilidad se requiere solo de la proyección del CoM en el piso, por esta razón se desprecia el componente en  $z$ . Durante el proceso de locomoción del robot es fundamental conocer la ubicación del CoM en cada instante de tiempo.

Para conocer la localización del CoM del eslabón  $i$  respecto al sistema de referencia 0 se concatenan las transformaciones obtenidas del modelo cinemático con la transformación del CoM del eslabón  $i$  ( ${}^{i-1}T_{ci}$ ), es decir:

$${}^0T_{ci} = {}^0T_1 T_2 \dots {}^{i-2}T_{i-1} {}^{i-1}T_{ci}, \quad (1.12)$$

de [1] obtenemos la Tabla 2.3 que muestra la ubicación el CoM en cada eslabón.

i	Descripción	${}^{i-1}c_{ix}$	${}^{i-1}c_{iy}$	${}^{i-1}c_{iz}$
1	Pie derecho	16.68	8.25	-0.20
2	Unión tobillo-tibia-peroné derecho	35.95	-4.84	1.06
3	Tibia-peroné derecho	53.64	0.00	-7.27
4	Femur derecho	59.27	0.00	3.48
5	Unión femur-cadera derecha	32.50	-3.55	-3.07
6	Cadera	4.25	-62.52	-1.52
7	Unión cadera-femur izquierdo	32.50	-3.07	3.55
8	Femur izquierdo	31.39	0.00	3.48
9	Tibia-peroné izquierdo	51.34	0.00	-7.39
10	Unión tibia-tobillo izquierdo	38.50	-1.06	-4.86
11	Pie izquierdo	5.56	8.25	-0.28

Tabla 1.4: Localización del CoM ( ${}^{i-1}\vec{c}_i$ ) respecto al eje  $i - 1$  y masas ( $m_i$ ) de cada eslabón [1].

donde:

$${}^{i-1}T_{c_i} = \begin{bmatrix} 1 & 0 & 0 & i - 1_{ix}^c \\ 0 & 1 & 0 & i - 1_{iy}^c \\ 0 & 0 & 1 & i - 1_{iz}^c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.13)$$

Con los elementos (1,4), (2,4) y (3,4) de la ecuación 1.13, formamos el vector  ${}^0\vec{c}_{i\text{cinematico}} = [{}^0\vec{c}_{ix\text{cinematico}} \quad -{}^0\vec{c}_{iy\text{cinematico}} \quad {}^0\vec{c}_{iz\text{cinematico}}]$

El sistema de referencia  $\{0\}$  en la Figura 1.7 presenta una rotación respecto al sistema tomado para la obtención de la cinemática directa, debido a esto el  ${}^0\vec{c}_i$  se localiza como:

$${}^0\vec{c}_i = \begin{bmatrix} {}^0c_{iz\text{cinematico}} \\ -{}^0c_{iy\text{cinematico}} \\ {}^0c_{ix\text{cinematico}} \end{bmatrix} \quad (1.14)$$

Entonces el eslabón  $i=2$  está dado por:

$${}^0\vec{c}_2 = \begin{bmatrix} c_2z \\ -(s_1c_2x + c_1c_2y) \\ c_1c_2x - s_1c_2y + l_1 \end{bmatrix} \quad (1.15)$$

Siguiendo la misma metodología se obtienen las ecuaciones  ${}^0c_i$  para  $i > 2$  hasta  $i = 11$  [1].

### 1.6.2. Punto de Momento Cero (ZMP)

El ZMP es el punto en el piso donde la suma de todos los momentos de las fuerzas activas es igual a cero, dicho de otra forma, es el punto dentro del área de soporte para el cual se mantiene:

$$\begin{aligned}\sum M_x &= 0, \\ \sum M_y &= 0,\end{aligned}\tag{1.16}$$

al punto  $p = [p_x, p_y, p_z]^T$  que satisface la ecuación se denomina punto de momento cero [15]. La suma de momentos en el eje  $z$  es diferente de cero ( $\sum M_z \neq 0$ )[16]. Para determinar la localización del ZMP durante la locomoción del robot se emplea la teoría de la dinámica de cuerpos rígidos para describir el comportamiento de los eslabones [17]. Por lo tanto, se considera a cada eslabón como un cuerpo rígido  $\mathcal{S}$  donde la distancia entre cualquier par de puntos  $Q_i$  y  $Q_j$  se mantiene constante durante el desplazamiento, esto es

$$d_{ij} = |x_i - x_j| = \text{constante}, \quad \forall Q_i, Q_j \in \mathcal{S}\tag{1.17}$$

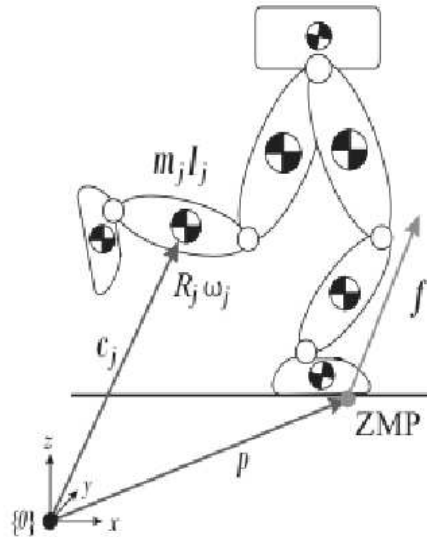


Figura 1.8: ZMP y eslabones del robot como cuerpos rígidos[1].

En el estudio del ZMP las dimensiones y formas geométricas de los eslabones que conforman el robot se toman en cuenta, por lo que es necesario contar con la información mecánica y cinemática de cada eslabón.



El cálculo del ZMP se desarrolla analizando la dinámica del robot en tres dimensiones Figura 2.6, considerando la masa  $m_i$  e inercia  $I_i$  de cada eslabón y la información concerniente a la cinemática del robot (posición del CoM, orientación, velocidad de los eslabones, etc.). Por lo tanto, el momento lineal total  $\mathcal{P} = [\mathcal{P}_x \mathcal{P}_y \mathcal{P}_z]^T$  se obtiene por medio de:

$$\mathcal{P} = \sum_{i=1}^N m_i \dot{c}_i \quad (1.18)$$

donde  $\dot{c}_i$  es la velocidad del CoM del  $i$ -ésimo eslabón. Similarmente, el momento angular total  $\mathcal{L} = [\mathcal{L}_x \mathcal{L}_y \mathcal{L}_z]^T$  se calcula por medio de la siguiente ecuación como:

$$\mathcal{L} = \sum_{i=1}^N [c_i x (m_i \dot{c}_i) + R_i I_i R_i^T w_i], \quad (1.19)$$

donde  $R_i$ ,  $I_i$  y  $w_i$  es la matriz de rotación de  $3 \times 3$ , el tensor de inercia y la velocidad angular del  $i$ -ésimo eslabón, respectivamente.

Aplicando una fuerza externa  $f$  y un momento  $\tau$ , la razón de cambio del momento angular y lineal es descrito por la ley de Newton y Euler como:

$$\begin{aligned} f &= \dot{\mathcal{P}} - M\vec{g}, \\ \tau &= \dot{\mathcal{L}} - \vec{c}x M\vec{g}, \end{aligned} \quad (1.20)$$

donde  $\vec{g} = [0, 0, -g]^T$  es el vector de aceleración debido a la gravedad. Por otro lado, suponiendo que una fuerza externa está actuando en el ZMP localizado en  $p$ ,

$$\tau = px f + \tau_{ZMP}, \quad (1.21)$$

donde  $\tau_{ZMP} = [\tau_{ZMP_x}, \tau_{ZMP_y}, \tau_{ZMP_z}]^T$  es el momento en el ZMP. Sustituyendo 1.20 en 1.21 se obtiene que:

$$\tau_{ZMP} = \dot{\mathcal{L}} - \vec{c}x M\vec{g} + (\dot{\mathcal{P}} - M\vec{g})xp, \quad (1.22)$$

reduciendo y representando la primera y segunda fila de 1.22 se tiene:

$$\begin{aligned} \tau_{ZMP_x} &= \dot{\mathcal{L}}_x + Mgc_y + \dot{\mathcal{P}}_y p_z - (\dot{\mathcal{P}}_z + Mg)p_y, \\ \tau_{ZMP_y} &= \dot{\mathcal{L}}_y + Mgc_x + \dot{\mathcal{P}}_x p_z - (\dot{\mathcal{P}}_z + Mg)p_x, \end{aligned} \quad (1.23)$$

el ZMP representado respecto al origen  $\{0\}$  ( $x, y, z$ ) puede ser localizado por medio de:

$$\begin{aligned} p_x &= \frac{Mgc_x + p_z \dot{\mathcal{P}}_x - \dot{\mathcal{L}}_y}{Mg + \dot{\mathcal{P}}_z}, \\ p_y &= \frac{Mgc_y + p_z \dot{\mathcal{P}}_y - \dot{\mathcal{L}}_x}{Mg + \dot{\mathcal{P}}_z}, \end{aligned} \quad (1.24)$$

en donde,  $g$  es la aceleración gravitacional,  $c_x$ ,  $c_y$  son las coordenadas del centro de masa equivalente,  $M$  es la masa total de los  $N$  eslabones rígidos [16][18][19]. Finalmente el componente  $z$  del ZMP se considera igual a cero ( $p_z = 0$ ) debido a que el origen  $\{0\}$  se encuentra fijo al suelo, de aquí que las ecuaciones 1.24 se reducen a

$$\begin{aligned} p_x &= \frac{Mgc_x - \dot{\mathcal{L}}_y}{Mg + \dot{\mathcal{P}}_z}, \\ p_y &= \frac{Mgc_y - \dot{\mathcal{L}}_x}{Mg + \dot{\mathcal{P}}_z}, \end{aligned} \quad (1.25)$$

Cabe destacar, que si el robot experimenta movimientos lentos o se encuentra en una condición estacionaria, el cálculo del ZMP se reduce a la proyección del CoM ( $\vec{c}$ ) sobre el piso, es decir, las velocidades tienden a cero y se reducen las ecuaciones 1.25 a:

$$\begin{aligned} p_x &= c_x, \\ p_y &= c_y, \end{aligned} \quad (1.26)$$

## 1.7. Validación de Trayectorias

Las trayectorias que se implementarán en el modelo del robot bípedo, deben ser validadas por los conceptos CoM o ZMP, a fin de establecer la condición de equilibrio (estático y dinámico) durante el desplazamiento.

Para la transición de “posición inicial - posición vertical” la condición de equilibrio se analiza por medio del CoM. Por otra parte, para las transiciones de “inicio de marcha - ciclo de marcha - fin de marcha” se verifica la condición de balanceo del robot por medio del ZMP.

La planificación de la marcha realizada en [1], se resume en la Figura 1.9 donde al final del proceso se determinan las trayectorias deseadas  $\theta_d(k)$  en función de los ángulos de articulación.

Primeramente, se consideraron las trayectorias de estudios médicos tanto para el plano frontal como para el sagital[36][37][38][39][40]. Estas trayectorias se introdujeron a un modelo en 3D del software SolidWorks para verificar su equilibrio durante la marcha, después de esto se realizó el cálculo del modelo matemático y posteriormente se validó el ZMP.

Una vez que se satisfacen las condiciones de equilibrio, se obtienen la trayectorias deseadas. Las trayectorias obtenidas son las presentadas en la sección 1.5

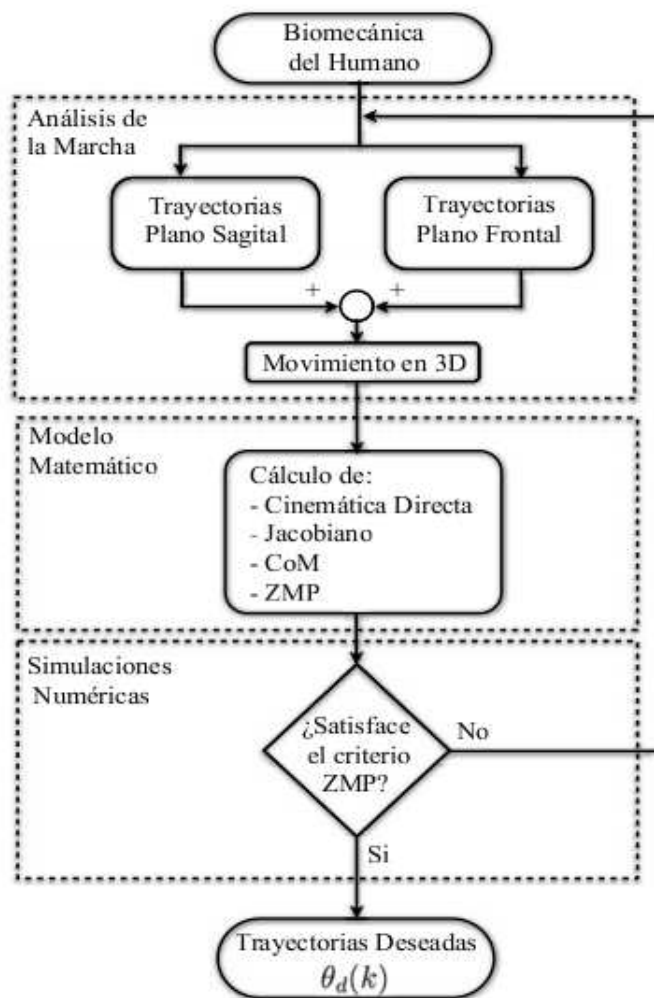


Figura 1.9: Planificación de la marcha

## 1.8. Conclusiones

Durante el presente capítulo, se empezó clasificando de manera general los distintos tipos de robots, llegando así a los robots bípedos. Se realizó una revisión a los principales proyectos involucrados en esta área de la robótica, mostrando la evolución que han tenido los humanoides de mayor trascendencia tecnológica actualmente construidos.

Después de este análisis, pasamos al estudio del estado del arte en lo que respecta a la locomoción bípeda de robots humanoides. Los resultados más exitosos se han conseguido con el algoritmo ZMP, un procedimiento sistemático que permite asegurar la estabilidad de las locomociones. Este algoritmo es siempre una referencia para el desarrollo de otros.

También existen otras múltiples aproximaciones, como Locomoción dinámica. Esta última

está bastante desarrollada, y permite a través de un diseño conjunto de la estructura física y el control reducir el consumo energético de la locomoción. Una aproximación inspirada en este planteamiento puede proporcionar propiedades muy interesantes al robot.

Aparte de los algoritmos usados, existen muchos otros, como por ejemplo el uso de modelos virtuales, aprendizaje con redes neuronales, aprendizaje por refuerzo, técnicas de optimización, etc.

En el área de la robótica, el interés en el control difuso ha ido en aumento. Se han realizado gran cantidad de investigaciones donde se demuestra la eficiencia en la implementación de este algoritmo en control de robots. En base a que la plataforma experimental proporciona 6gdl sobre el plano sagital y 4 para el plano frontal, la planificación de marcha se analiza desde un enfoque tridimensional. De esta forma, la coordinación del movimiento en el plano frontal con el plano sagital genera un desplazamiento en el espacio.

El enfoque adoptado para la definición de las trayectorias durante el ciclo del paso es tomado del estudio de la biomecánica del humano lo que implica un comportamiento similar al humano en articulaciones de cadera, rodilla y tobillo para el plano sagital y sobre las articulaciones de tobillo y cadera en el plano frontal. El objetivo final de este capítulo es dar una idea general del problema a resolver.

# Capítulo 2

## Modelo matemático de un robot bípedo de 10 gdl

El modelo matemático se emplea en el análisis de la marcha, la cinemática directa se obtiene con el método de Denavit-Hartenberg a fin de conocer la posición y orientación del robot. Se obtiene también la cinemática inversa y la matriz Jacobiana para definir la relación entre las velocidades articulares y las del espacio cartesiano. Finalmente con el análisis dinámico del robot se estudia la localización del centro de masa y del punto de momento cero.

### 2.1. Cinemática del robot bípedo

La cinemática es una descripción analítica del movimiento espacial del robot respecto a un sistema de coordenadas de referencia. El modelo cinemático directo es importante para conocer la posición, orientación, velocidad y aceleración de la planta del pie y cadera. La cinemática inversa es la técnica que permite determinar el movimiento de una cadena de articulaciones para lograr que un actuador final se ubique en una posición concreta[27].

En esta sección, se desarrolla la cinemática del robot. La derivación de la cinemática se compone de 4 secciones. Primero, una breve descripción de la cinemática y el método utilizados para describir la cinemática se presentan. Esto incluye como la posición y orientación de los eslabones del robot pueden representarse en el espacio. Se utiliza la notación y convención de Denavit-Hartenberg para describir los eslabones. En la segunda sección, los métodos son aplicados al bípedo para describir su cinemática. En la tercera sección, se determina la cinemática inversa del bípedo. En la cuarta sección, se obtienen el jacobiano del robot.

### 2.1.1. Breve descripción de los métodos para definir la cinemática

En la cinemática, el movimiento es tratado sin considerar las fuerzas que lo causan. Las variables estudiadas en la cinemática son posición, velocidad y aceleración. El robot es considerado como una cadena de cuerpos llamados “eslabones” conectados por juntas, donde una junta en caso del robot es 1 grado de libertad (1gdl) de junta de revoluta. Estas cadenas son conocidas como manipuladores. Para describir cada eslabón del robot en el espacio, es necesaria la representación de su posición y orientación.

#### Posición y orientación en el espacio

Dado un sistema de coordenadas de referencia  $\{A\}$ , cualquier eslabón puede ser localizado por un vector de posición de  $3 \times 1$ . Para obtener la orientación de un eslabón, un sistema de coordenadas  $\{B\}$  se incluye al eslabón. La orientación de  $\{B\}$  relativa al marco de referencia 1 es una transformación lineal llamada matriz de rotación o matriz coseno directo (DCM) Direct Cosine Matrix por sus siglas en ingles.

La matriz de rotación que transforma del marco de referencia  $\{B\}$  al  $\{A\}$  se escribe como  ${}^A_B\mathbf{R}$  [12]

$${}^A_B\mathbf{R} : A \leftarrow B; \quad {}^A p = {}^A_B\mathbf{R} {}^B p \quad (2.1)$$

donde  ${}^B p$  es el vector eslabón  $p$  visto desde el marco de referencia  $\{B\}$  y  ${}^A p$  es el vector eslabón  $p$  visto desde el marco de referencia  $\{A\}$ .

Si los tres ejes principales de  $\{B\}$  son descritos por un conjunto de vectores ortonormales  $\widehat{b}_1, \widehat{b}_2$  y  $\widehat{b}_3$ , entonces la matriz de rotación  ${}^A_B\mathbf{R}$  puede expresarse como:

$${}^A_B\mathbf{R} = [{}^A\widehat{b}_1 \quad {}^A\widehat{b}_2 \quad {}^A\widehat{b}_3] \quad (2.2)$$

Note que  ${}^A_B\mathbf{R} = {}^B_A\mathbf{R}^+ = {}^B_A\mathbf{R}^{-1}$

Cuando se utiliza la matriz de rotación (2.1), el origen del marco de referencia  $\{A\}$  y  $\{B\}$  coincide mostrando la Figura 2.1(a)

donde el marco de referencia  $\{A\}$  y  $\{B\}$  son unidos al mismo eslabón. Para unir un marco  $\{B\}$  a otro eslabón que no sea  $\{A\}$ , se requieren dos piezas de información para definirlo con respecto al marco  $\{A\}$ . En la Figura 2.1(a) el origen de  $\{A\}$  y  $\{B\}$  coinciden ya que están definidos relativos uno con otro por la matriz de rotación. En la Figura 2.1(b) la matriz de rotación y un vector de desplazamiento se requieren para definir  $\{B\}$ . El marco  $\{B\}$  esta definido relativo a

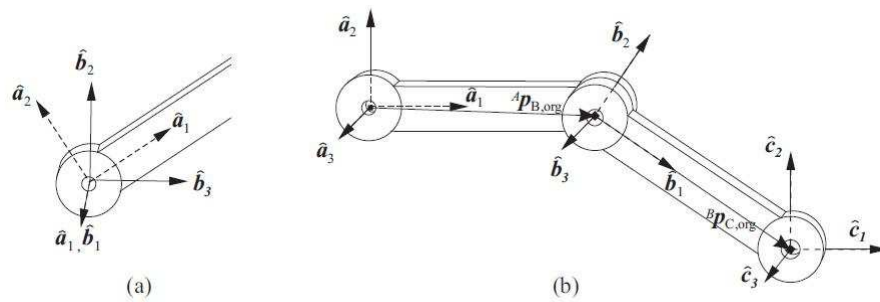


Figura 2.1: Representación de juntas de un robot[12]

{A}, mientras que el marco {C} está definido relativo a {B}. Estas son las matrices de rotación  ${}^A_B\mathbb{R}$  y el vector  ${}^A p_{B,origen}$  apuntando al origen de {B} desde el marco {A}. [12]

$${}^A p = {}^A_B \mathbb{R} {}^B p + {}^A p_{B,origen} \quad (2.3)$$

la Figura 2.1(b) muestra como esta transformación puede ser aplicada a varios marcos de referencia. La ecuación 2.3 puede ser re-escrita en forma matricial como:

$$\begin{bmatrix} {}^A p \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_B \mathbb{R} & {}^A p_{B,origen} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B p \\ 1 \end{bmatrix} \quad (2.4)$$

la matriz de  $4 \times 4$  (2.4) es llamada una transformación homogénea. Esta matriz de transformación  $T$  incluye toda la información necesaria acerca de la posición y orientación de un marco de referencia con respecto a otro marco. La ecuación 2.4 puede escribirse también como:

$${}^A p = {}^A_B T {}^B p \quad (2.5)$$

donde  $p$  es un vector de  $4 \times 1$ .

### 2.1.2. Notación Denavit-Hartenberg

En esta sección, se presenta la notación Denavit-Hartenberg para describir una cadena de cuerpo de un robot, esta notación establece que la cinemática de un robot puede ser representada por 4 parámetros para cada eslabón. Dos parámetros describen al eslabón y los otros dos describen su conexión con el eslabón siguiente [12]. El eslabón se define como un cuerpo rígido, el cual define la relación de los ejes de la junta de un robot. Un eje de la junta  $i$  es un vector dirección en el espacio y dice cual eslabón  $i$  rota relativo al eslabón  $i - 1$ . El lugar relativo de

dos ejes de junta pueden ser especificados por 2 parámetros, la distancia entre ellos y el ángulo o giro entre ellos.

La distancia  $a_{i-1}$  entre los dos ejes de junta  $i-1$  y  $i$  están definidos como la distancia de la línea mutua perpendicular a ambos ejes de junta, el ángulo entre sus proyecciones es el ángulo de giro de la unión como se ilustra en la Figura 2.2. La cantidad  $a_{i-1}$  también se conoce como la longitud del eslabón.

El segundo parámetro es el ángulo de giro  $\alpha_{i-1}$ , si ambos ejes son proyectados en un plano normal es la línea mutua perpendicular a ambos ejes de junta, el ángulo entre sus proyecciones es el ángulo de giro de la unión como se muestra en la Figura 2.2. Eslabones vecinos tienen un eje de junta común (Figura 2.2). Para describir la conexión, la notación D-H acomoda dos parámetros, el desplazamiento de la unión  $d_i$  y el ángulo de junta  $\theta_i$ .

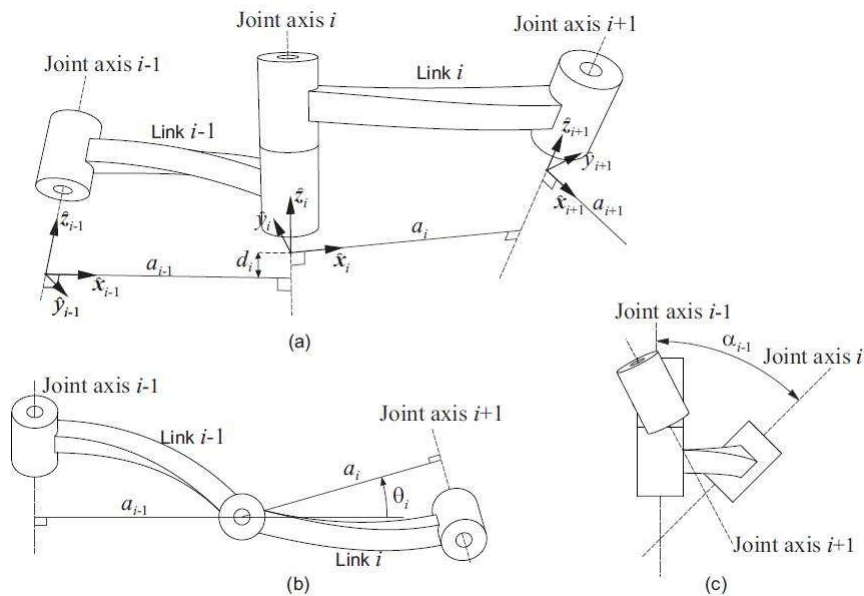


Figura 2.2: Juntas de un robot para establecer parámetros D-H[12]

El desplazamiento de la unión  $d_i$  es la distancia entre las dos líneas de longitud de eslabones  $a_{i-1}$  y  $a_i$  a lo largo del eje de junta  $i$ . El ángulo de junta  $\theta_i$  es el ángulo entre  $a_{i-1}$  y  $a_i$  medido en el eje de junta. Las dos identidades se muestran en la Figura 2.2, el desplazamiento  $d_i$  es cambiante y  $\theta_i$  es variante en el tiempo. Utilizando la notación D-H, la siguiente convención puede ser utilizada para fijar dos marcos de referencia a los eslabones.

En [12] una convención se presenta para fijar marcos de referencia a los eslabones de los robots o manipuladores en una cadena sencilla de eslabones fijos a alguna base fija.

1. El tercer eje  $\hat{z}_i$  del marco de referencia  $\{i\}$  coincide con el eje de junta  $i$ .



2. El origen del marco de referencia  $i$  se localiza donde la longitud de la unión  $a_i$  intersecta el eje de junta  $i$ .
3. El primer eje  $\hat{x}_i$  apunta a lo largo de la longitud de la unión  $a_i$  en la dirección de la junta  $i$  a  $i + 1$ .
4. El segundo eje  $\hat{y}_i$  esta formado por la regla de la mano derecha para completar un sistema de coordenadas.

El ángulo  $\alpha_i$  es medido en el sentido de la mano derecha alrededor de  $\hat{x}_i$ . Una cadena de eslabones con marcos fijos se presentan en la Figura 2.2. Notar que este marco de unión no es único, una de dos posibles direcciones en  $\hat{z}$  pueden elegirse. El primer eslabón de una cadena es la base del robot, llamado eslabón 0. El marco de referencia es llamado el marco  $\{0\}$  y es estacionario. Así, todas las otras posiciones de los eslabones pueden ser descritas en términos de este marco. El sistema de coordenadas es fijo para que coincida con el marco  $\{1\}$  cuando el ángulo de junta  $\theta_1$  es cero. Como resultado los parámetros de eslabones  $d_0, a_0$  y  $\alpha_0$  son siempre cero.

Utilizando el procedimiento de fijar marcos descritos anteriormente, los parámetros de unión y la variable de junta pueden ser encontradas como:

- $a_i$  = la distancia desde  $\hat{z}_i$  a  $\hat{z}_{i+1}$  medida a lo largo de  $\hat{x}_i$ .
- $\alpha_i$  = el ángulo entre  $\hat{z}_i$  a  $\hat{z}_{i+1}$  medida a lo largo de  $\hat{x}_i$ .
- $d_i$  = la distancia desde  $\hat{x}_{i-1}$  a  $\hat{x}_i$  medida a lo largo de  $\hat{z}_i$ .
- $\theta_i$  = los ángulos entre  $\hat{x}_i$  a  $\hat{x}_{i+1}$  alrededor de  $\hat{z}_i$ .

La matriz de transformación desde el marco  $\{i\}$  a  $\{i - 1\}$  se forma a partir de los cuatro parámetros de los eslabones,  $a_i, \alpha_i, d_i$  y  $\theta_i$  es

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & -\cos \alpha_{i-1} & -\cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Con esta breve descripción para asignar los marcos de referencia y derivando la matriz de transformación entre ellos, estos métodos son aplicados al robot para establecer el modelo cinemático del robot.

### 2.1.3. Derivando la cinemática del robot

La tarea de derivar la cinemática del robot se lleva a cabo en 3 pasos. El primer paso es fijar los marcos de referencia a la base del robot. El segundo paso es identificar los cuatro parámetros del eslabón para cada eslabón y el tercero es encontrar las matrices de transformación describiendo sus relaciones.

#### Fijando marcos de referencia al robot

En [12], se presenta una convención solamente para fijar marcos a una sola cadena de eslabones con una base fija, pero el bípedo no tiene base fija y consiste en múltiples cadenas. Sin embargo, en la fase de soporte simple, el bípedo puede verse como una cadena comenzando del pie de soporte separándose en otra cadena, la pierna de no soporte. Más aún, las cadenas del bípedo no tienen una base global fija, pero tienen una base fija durante el periodo de SSP. En [13] sugiere una solución de modelar el bípedo en 2 cadenas.

- Cadena 1, consiste en la pierna de soporte comenzando desde el pie y terminando en la cadera.
- Cadena 2, consiste en la pierna en el aire (no soporte), comenzando desde la junta de la cadera y terminando en el pie.

La cadena 2 esta unida a la cadena 1 en la junta de cadera usando matrices de transformación.

El marco de unión del bípedo se ve en la Figura 2.3(a). En la figura, el pie derecho es el pie de soporte. La pierna derecha se abrevia  $P_d$  y Pierna izquierda será  $P_i$ . El marco de unión que se ve en la Figura 2.3(b) presenta la dirección positiva de rotación de los ángulos de junta diferentes.

En la Figura 2.3(c) una vista del bípedo se presenta, mostrando las longitudes de los eslabones y los desplazamientos de junta y el vector localizado es la base de la pierna izquierda (no soporte), relativa al marco de la cadera  $P_{d5}$ . El marco-cero de la pierna izquierda  $P_{i0}$  se muestra en la Figura 2.4(a), estas están unidas a la cadera del bípedo. El conjunto de  $\{P_{i1}\}$  coincide con  $\{P_{i0}\}$  pero su orientación está dada por una rotación alrededor del eje z con los ángulos de junta  $\theta_{P_{i1}}$ . Así, cuando estos ángulos están en el marco cero  $\{P_{i1}\}$  coincide con el marco-cero. El marco base  $\{P_{d0}\}$  se ve en la Figura 2.4(b).

#### Identificando los parámetros del eslabón

Dados los marcos en la Figura 2.3(a) los parámetros del eslabón son listados en las Tablas 2.1 y 2.2.

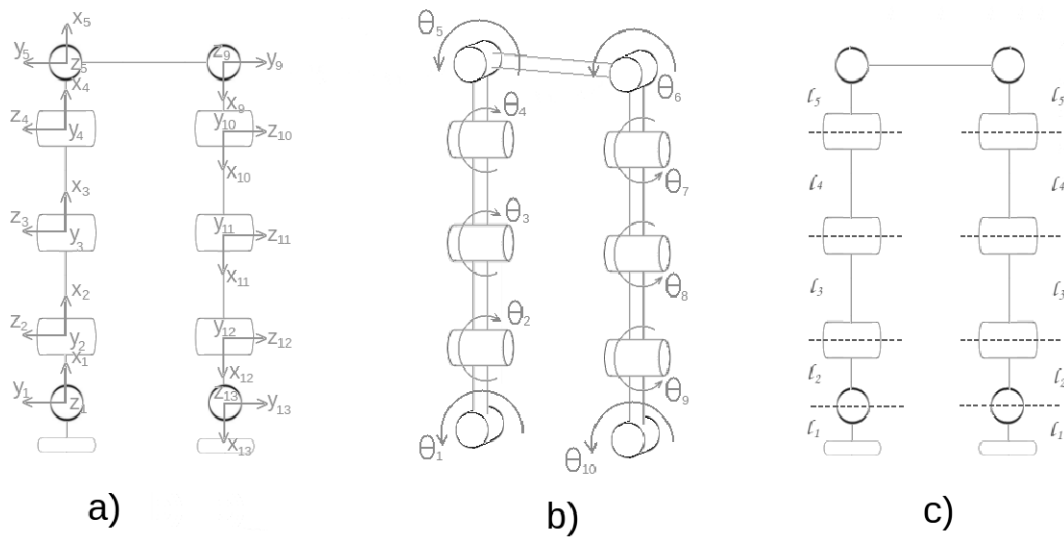


Figura 2.3: Estructura del bípedo

$i$	$a_{Pd,i-1}$	$\alpha_{Pd,i-1}$	$d_{Pd,i}$	$\theta_{Pd,i}$
1	$l_1-1.65\text{cm}$	$0^\circ$	0	$\theta_{Pd,1}^\circ$
2	$l_2-6.2\text{cm}$	$-90^\circ$	0	$\theta_{Pd,2}^\circ$
3	$l_3-10.5\text{cm}$	$0^\circ$	0	$\theta_{Pd,3}^\circ$
4	$l_4-8.3\text{cm}$	$0^\circ$	0	$\theta_{Pd,4}^\circ$
5	$l_5-6.4\text{cm}$	$90^\circ$	0	$\theta_{Pd,5}^\circ$
6	$l_6-2.8\text{cm}$	$90^\circ$	$l_7-6.45\text{cm}$	$0^\circ$

Tabla 2.1: Parámetros Denavit-Hartenberg para la pierna derecha

$i$	$a_{Pi,i-1}$	$\alpha_{Pi,i-1}$	$d_{Pi,i}$	$\theta_{Pi,i}$
1	0	$0^\circ$	$l_7-6.45\text{cm}$	$180^\circ$
2	$l_6-2.8\text{cm}$	$90^\circ$	0	$\theta_{Pi,5}^\circ$
3	$l_5-6.4\text{cm}$	$-90^\circ$	0	$\theta_{Pi,4}^\circ$
4	$l_4-8.3\text{cm}$	$0^\circ$	0	$\theta_{Pi,3}^\circ$
5	$l_3-10.5\text{cm}$	$0^\circ$	0	$\theta_{Pi,2}^\circ$
6	$l_2-6.2\text{cm}$	$90^\circ$	0	$\theta_{Pi,1}^\circ$
7	$l_1-1.65\text{cm}$	$0^\circ$	0	$0^\circ$

Tabla 2.2: Parámetros Denavit-Hartenberg para la pierna izquierda

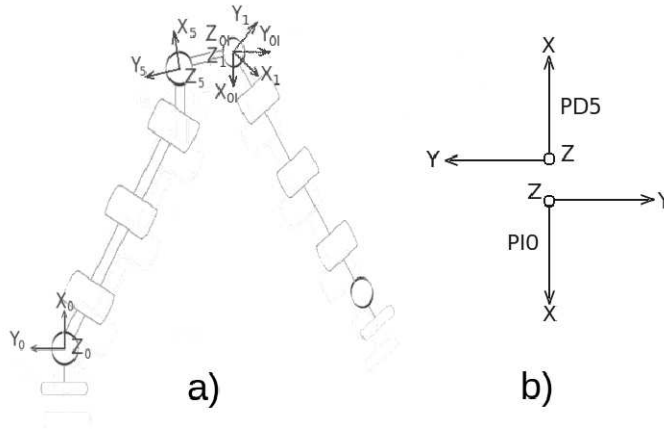


Figura 2.4: Estructura del bípido 2

El vector  ${}^{Pd5}P_{Pi0,origen}$  localiza el origen del marco  $\{P_{i0}\}$  relativa a  $\{P_{d0}\}$  tiene las siguientes coordenadas.

$${}^{Pd5}P_{Pi0,origen} = \begin{bmatrix} 0 \\ -6.45 \\ 0 \end{bmatrix} \quad (2.7)$$

El marco de orientación  $\{P_{i0}\}$  relativa a  $\{P_{d5}\}$  se obtiene derivando las matrices de rotación correspondientes. Esta matriz resulta de la ecuación 2.2 y por inspección de la Figura 2.4

$${}^{Pd5}_{Pi0,origen} \mathbf{R} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

con este vector, esta matriz de rotación y los parámetros de D-H, las matrices de transformación se pueden obtener.

### Derivación de las matrices de transformación

La matriz de transformación  ${}^{Pd0}_{Pd1} \Pi$  desde el marco  $\{P_{d1}\}$  a  $\{P_{d0}\}$  se encuentra sustituyendo los parámetros de la Tabla 2.1 en la ecuación 2.6.

$${}^{Pd0}_{Pd1} \Pi = \begin{bmatrix} \cos \theta_1 & -\text{sen } \theta_1 & 0 & l_1 \\ \text{sen } \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$\begin{matrix} P_{d1} \\ P_{d2} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_2 & -\text{sen } \theta_2 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ -\text{sen } \theta_2 & -\cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$$\begin{matrix} P_{d2} \\ P_{d3} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_3 & -\text{sen } \theta_3 & 0 & l_3 \\ \text{sen } \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

$$\begin{matrix} P_{d3} \\ P_{d4} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_4 & -\text{sen } \theta_4 & 0 & l_4 \\ \text{sen } \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

$$\begin{matrix} P_{d4} \\ P_{d5} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_5 & -\text{sen } \theta_5 & 0 & l_5 \\ 0 & 0 & -1 & 0 \\ \text{sen } \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

$$\begin{matrix} P_{d5} \\ P_{d6} \end{matrix} \Pi = \begin{bmatrix} 1 & 0 & 0 & l_6 \\ 0 & 0 & -1 & l_7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

El marco  $\{P_{i0}\}$  está definido por el marco traslación  $\{P_{d5}\}$  con el vector  ${}^{P_{d5}}P_{P_{i0},origen}$  y rotándolo por la matriz  ${}^{P_{d5}}_{P_{i0}}\mathbb{R}$  dada en 2.8. Estos dos datos se sustituyen en la ecuación 2.6 resulta la matriz de transformación  ${}^{P_{d5}}_{P_{i0}}\Pi$ . El resto de las matrices de transformación para la pierna izquierda se encuentran por la sustitución de los parámetros D-H de la pierna izquierda (Tabla 2.2) en la ecuación 2.6.

$$\begin{matrix} P_{d5} \\ P_{i0} \end{matrix} \Pi = \begin{bmatrix} {}^{P_{d5}}_{P_{i0}}\mathbb{R} & {}^{P_{d5}}P_{P_{i0},origen} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2.15)$$

usa el la

$$\begin{matrix} P_{i0} \\ P_{i1} \end{matrix} \Pi = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -l_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

$$\begin{matrix} P_{i1} \\ P_{i2} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_5 & -\text{sen } \theta_5 & 0 & l_6 \\ 0 & 0 & -1 & 0 \\ \text{sen } \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

$$\begin{matrix} P_{i2} \\ P_{i3} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_4 & -\text{sen } \theta_4 & 0 & l_5 \\ 0 & 0 & 1 & 0 \\ -\text{sen } \theta_4 & -\cos \theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

$$\begin{matrix} P_{i3} \\ P_{i4} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_3 & -\text{sen } \theta_3 & 0 & l_4 \\ \text{sen } \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

$$\begin{matrix} P_{i4} \\ P_{i5} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_2 & -\text{sen } \theta_2 & 0 & l_3 \\ \text{sen } \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.20)$$

$$\begin{matrix} P_{i5} \\ P_{i6} \end{matrix} \Pi = \begin{bmatrix} \cos \theta_1 & -\text{sen } \theta_1 & 0 & l_2 \\ 0 & 0 & -1 & 0 \\ \text{sen } \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.21)$$

$$\begin{matrix} P_{i6} \\ P_{i7} \end{matrix} \Pi = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.22)$$

Con estas matrices es posible determinar la posición y orientación de todos los marcos relativos con algún otro. Por ejemplo, la matriz de transformación que proporciona la posición de un punto en la cadera  $\{P_{d5}\}$  en el marco de referencia  $\{P_{d0}\}$  puede determinarse como:

$${}^{Pd0}_{Pd5}\Pi = {}^{Pd0}_{Pd1}T \quad {}^{Pd1}_{Pd2}T \quad {}^{Pd2}_{Pd3}T \quad {}^{Pd3}_{Pd4}T \quad {}^{Pd4}_{Pd5}T \quad {}^{Pd5}_{Pd6}T \quad {}^{Pd6}_{Pd7}T \quad (2.23)$$

Entonces, la matriz de transformación de la planta del pie a la cadera, es la ecuación 2.24 que describe la posición y la orientación cartesiana del eje  $\{6\}$  respecto al  $\{0\}$  de la pierna derecha.<sup>1</sup>

$${}^{pie}T_{cadera} = \begin{bmatrix} c_1c_{234}c_5 - s_1s_5 & c_1s_{234} & c_1c_{234}s_5 + s_1c_5 & P_x \\ s_1c_{234}c_5 + c_1s_5 & s_1s_{234} & s_1c_{234}s_5 - c_1c_5 & P_y \\ s_{234}c_5 & -c_{234} & s_{234}s_5 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

donde:

$$\begin{aligned} P_x &= c_1c_{234}(c_5l_6 + s_5l_7 + l_5) - s_1(s_5l_6 - c_5l_7) + c_1(c_{234}l_4 + c_2l_3 + l_2) + l_1, \\ P_y &= s_1c_{234}(c_5l_6 + s_5l_7 + l_5) + c_1(s_5l_6 - c_5l_7) + s_1(c_{234}l_4 + c_2l_3 + l_2), \\ P_z &= s_{234}(c_5l_6 + s_5l_7 + l_5) + s_{234}l_4 + s_2l_3. \end{aligned} \quad (2.25)$$

Y la matriz de transformación que representa la cinemática directa de la cadera  $\{0\}$  respecto a la plana del pie izquierdo  $\{7\}$  se ve en la ecuación 2.26

$${}^{cadera}T_{pie} = \begin{bmatrix} -c_1c_{234}c_5 + s_1s_5 & s_1c_{234}c_5 + c_1s_5 & c_5s_{234} & P_x \\ -c_1s_{234} & s_1s_{234} & c_{234} & P_y \\ -c_1c_{234}s_5 - s_1c_5 & s_1c_{234}s_5 - c_5c_1 & s_5s_{234} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

donde:

$$\begin{aligned} P_x &= -c_5c_{432}(c_1l_1 + l_2) - c_5(c_{43}l_3 + c_4l_4 + l_5) + s_5s_1l_1 - l_6, \\ P_y &= -s_{432}(c_1l_1 + l_2) - s_{43}l_3 - s_4l_4, \\ P_z &= -s_5c_{432}(c_1l_1 + l_2) - s_5(c_{43}l_3 + c_4l_4 + l_5) - c_5s_1l_1 - l_7, \end{aligned} \quad (2.27)$$

### 2.1.4. Cinemática Inversa

Para completar la información del modelo cinemático y lograr un movimiento similar al del humano, es necesario un mapeo desde la primer posición de una extremidad a los ángulos de las uniones en esa extremidad. El modelo de la cinemática directa puede expresarse como:

$$\mathbb{P} = f(x, y, z)(\theta) \quad (2.28)$$

---

<sup>1</sup> $c_1 = \cos \theta_1$ ,  $s_1 = \sin \theta_1$ ,  $c_{234} = \cos(\theta_2 + \theta_3 + \theta_4)$

donde  $f$  es una función que aplica el modelo de la cinemática directa a los ángulos de las uniones y  $P \in \mathbb{R}^{3 \times 5}$  es la posición de los eslabones. Cuando el modelo de la cinemática inversa tiene que determinarse se calcula la inversa de la función  $f(x, y, z)(\theta)$ . La función inversa puede expresarse como:

$$\theta = f^{-1}(x, y, z)(\Pi) \quad (2.29)$$

donde  $\Pi$  es el subconjunto de los vectores de posición  $p \in \mathbb{R}^3$ . Esta función calcula  $\theta$  basados en la posición deseada en alguna unión clave  $\Pi$ .

El cálculo del modelo de cinemática inversa puede parecer simple y directo, pero podemos obtener múltiples soluciones y/o ninguna, razón por la cual este cálculo es más complejo. Para que una solución exista, el punto objetivo debe estar en el espacio de trabajo del manipulador donde el espacio de trabajo es el volumen del espacio que el efector final del manipulador puede alcanzar. Otro problema puede surgir al existir múltiples soluciones. Un ejemplo es un brazo planar con 3 juntas de revoluta las cuales tienen un espacio de trabajo grande en el plano. La Figura 2.5 muestra este ejemplo, donde se muestran 6 soluciones en un plano. El número de soluciones aumenta rápidamente si el brazo puede moverse en 3 dimensiones.

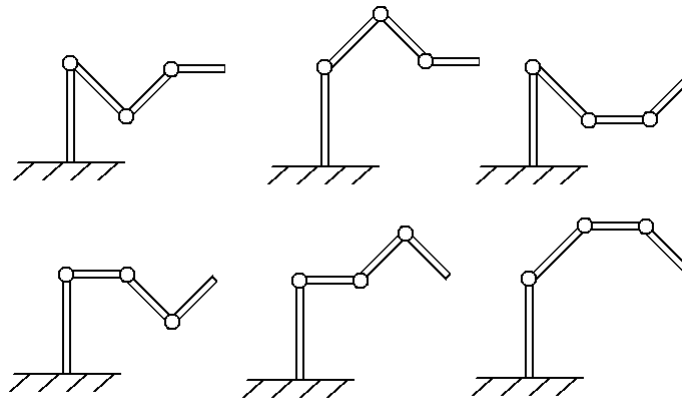


Figura 2.5: Soluciones al modelo de cinemática inversa para un brazo planar de 3 juntas

La definición de restricciones a las juntas puede reducir el número de soluciones a las ecuaciones de la cinemática inversa. Un ejemplo de restricción sería que la junta de la rodilla solo se le permite flexionarse hacia atrás en el plano sagital.



### 2.1.5. Modelo Cinemática Inversa de las piernas del robot

#### Pierna Derecha

Para la determinación de la cinemática inversa de la pierna derecha es necesario calcular la siguiente matriz  ${}_{Pd6}^{Pd0}\Pi$  la cual se retoma a continuación

$${}^0T_6 = \begin{bmatrix} c_1 c_{234} c_5 - s_1 s_5 & c_1 s_{234} & c_1 c_{234} s_5 + s_1 c_5 & P_x \\ s_1 c_{234} c_5 + c_1 s_5 & s_1 s_{234} & s_1 c_{234} s_5 - c_1 c_5 & P_y \\ s_{234} c_5 & -c_{234} & s_{234} s_5 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.30)$$

donde

$$\begin{aligned} P_x &= c_1 c_{234} (c_5 l_6 + s_5 l_7 + l_5) - s_1 (s_5 l_6 - c_5 l_7) + c_1 (c_{23} l_4 + c_2 l_3 + l_2) + l_1, \\ P_y &= s_1 c_{234} (c_5 l_6 + s_5 l_7 + l_5) + c_1 (s_5 l_6 - c_5 l_7) + s_1 (c_{23} l_4 + c_2 l_3 + l_2), \\ P_z &= s_{234} (c_5 l_6 + s_5 l_7 + l_5) + s_{23} l_4 + s_2 l_3. \end{aligned} \quad (2.31)$$

Considerar los componentes de la matriz 2.30  ${}^0T_6(1, 2) = c_1 s_{234}$  y  ${}^0T_6(2, 2) = s_1 s_{234}$ , resolviendo para  $s_{234}$  obtenemos

$$s_{234} = \frac{{}^0T_6(1, 2)}{c_1} \quad \text{y} \quad s_{234} = \frac{{}^0T_6(2, 2)}{s_1} \quad (2.32)$$

igualando las dos expresiones y juntando términos se obtiene

$$\frac{s_1}{c_1} = \frac{{}^0T_6(2, 2)}{{}^0T_6(1, 2)} = \tan \theta_1 \quad (2.33)$$

Resolviendo para  $\theta_1$  se obtiene

$$\theta_1 = \tan^{-1} \frac{{}^0T_6(2, 2)}{{}^0T_6(1, 2)} \quad (2.34)$$

De la misma manera se obtiene  $\theta_5$  de las componentes de la matriz 2.30,  ${}^0T_6(1, 3) = c_5 s_{234}$  y  ${}^0T_6(3, 3) = s_5 s_{234}$

$$\theta_5 = \tan^{-1} \frac{{}^0T_6(3, 3)}{{}^0T_6(1, 3)} \quad (2.35)$$

Ahora, se re-escribe la primer ecuación de 2.31 de la siguiente manera

$$P_x = \alpha c_1 - \beta s_1 + \gamma c_1 + c_1 l_2 + l_1 \quad (2.36)$$

donde  $\alpha = c_{234} (c_5 l_6 + s_5 l_7 + l_5)$ ,  $\beta = (s_5 l_6 - c_5 l_7)$  y  $\gamma = (c_{23} l_4 + c_2 l_3)$ . Se resuelve para  $\gamma$  obteniendo

$$\gamma = \frac{P_x - \alpha c_1 - \beta s_1 - c_1 l_2 - l_1}{c_1} = c_{23} l_4 + c_2 l_3. \quad (2.37)$$

Por otro lado de  $P_z$  de la matriz 2.30 se tiene

$$P_z = \alpha \frac{s_{234}}{c_{234}} + \Delta = {}^2T_4\alpha + \Delta \quad (2.38)$$

donde  $\Delta = s_{23}l_4 + s_2l_3$ , resolviendo para  $\Delta$

$$\Delta = P_z^2 T_4 \alpha = s_{23}l_4 + s_2l_3 \quad (2.39)$$

también se conoce  $\theta_{2-4}$  de  $\cos^{-1}({}^0T_6(3, 2))$  y sumando el cuadrado de 2.37 y 2.39 se tiene

$$\gamma^2 + \Delta^2 = l_4^2(c_{23}^2 + s_{23}^2) + l_3^2(c_2^2 + s_2^2) + 2l_3l_4(c_2c_{23} + s_2s_{23}) = l_4^2 + l_3^2 + 2l_3l_4c_3 \quad (2.40)$$

resolviendo para  $c_3$  tenemos

$$c_3 = \frac{\gamma^2 + \Delta^2 - l_4^2 - l_3^2}{2l_3l_4} \quad (2.41)$$

y para obtener  $\theta_3$  se tiene

$$\theta_3 = \cos^{-1}\left(\frac{\gamma^2 + \Delta^2 - l_4^2 - l_3^2}{2l_3l_4}\right). \quad (2.42)$$

Ahora, con las identidades  $c_{23} = c_2c_3 - s_2s_3$  y  $s_{23} = s_2c_3 + c_2s_3$  se expanden las ecuaciones 2.37 y 2.39 obteniendo

$$\gamma = c_2(c_3l_4 + l_3) - s_2(s_3l_4) \quad (2.43)$$

$$\Delta = c_2(s_3l_4) + s_2(c_3l_4 + l_3) \quad (2.44)$$

resolviendo para  $c_2$  y  $s_2$  por ecuaciones simultaneas y con  $\tan(\theta_2) = \frac{s_2}{c_2}$  se tiene

$$\theta_2 = \tan^{-1}\left(\frac{\Delta(c_3l_4 + l_3) - \gamma(s_3l_4)}{\gamma(c_3l_4 + l_3) + \Delta(s_3l_4)}\right). \quad (2.45)$$

Finalmente para obtener  $\theta_4 = \theta_{234} - \theta_2 - \theta_3$ .

## Pierna Izquierda

Para la determinación de la cinemática inversa de la pierna izquierda se utiliza el mismo procedimiento que se utilizó para la pierna derecha. Entonces la matriz  ${}_{Pd6}^{d0}\Pi$  para la pierna izquierda es

$${}^0T_7 = \begin{bmatrix} -c_1c_{234}c_5 + s_1s_5 & s_1c_{234}c_5 + c_1s_5 & c_5s_{234} & P_x \\ -c_1s_{234} & s_1s_{234} & -c_{234} & P_y \\ -s_5c_{234}c_1 - c_5s_1 & s_5c_{234}s_1 - c_5c_1 & s_{234}s_5 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.46)$$

donde

$$\begin{aligned} P_x &= -c_5 c_{432}(c_1 l_1 + l_2) + s_5 s_1 l_1 - c_5 c_{43} l_3 - c_5 c_4 l_4 - c_5 l_5 + l_6, \\ P_y &= s_{432}(c_1 l_1 + l_2) - s_{43} l_3 - s_4 l_4, \\ P_z &= -s_5 c_{432}(c_1 l_1 + l_2) + -c_5 s_1 l_1 - s_5 c_{43} l_3 - s_5 c_4 l_4 - s_5 l_5 - l_7, \end{aligned} \quad (2.47)$$

Considerar los componentes de la matriz 2.46  ${}^0T_6(2, 1) = -c_1 s_{432}$  y  ${}^0T_6(2, 2) = s_1 s_{432}$ , resolviendo para  $s_{432}$  obtenemos

$$s_{432} = \frac{{}^0T_6(2, 1)}{c_1} \quad \text{y} \quad s_{432} = \frac{{}^0T_6(2, 2)}{s_1} \quad (2.48)$$

igualando las dos expresiones y juntando términos se obtiene

$$\frac{s_1}{c_1} = \frac{{}^0T_6(2, 2)}{{}^0T_6(2, 1)} = \tan \theta_1 \quad (2.49)$$

Resolviendo para  $\theta_1$  se obtiene

$$\theta_1 = \tan^{-1} \frac{{}^0T_6(2, 2)}{{}^0T_6(2, 1)} \quad (2.50)$$

De la misma manera se obtiene  $\theta_5$  de las componentes de la matriz 2.46,  ${}^0T_6(1, 3) = c_5 s_{432}$  y  ${}^0T_6(3, 3) = s_5 s_{432}$

$$\theta_5 = \tan^{-1} \frac{{}^0T_6(3, 3)}{{}^0T_6(1, 3)} \quad (2.51)$$

Ahora, se re-escibe la primera ecuación de 2.47 de la siguiente manera

$$P_x = -\alpha_1 c_5 - \beta_1 c_5 - c_5 l_5 - l_6 + s_5 s_1 l_1 \quad (2.52)$$

donde  $\alpha_1 = c_{432}(c_1 l_1 + l_2)$ ,  $\beta_1 = c_{43} l_3 + c_4 l_4$ . Se resuelve para  $\beta_1$  obteniendo

$$\beta_1 = -\frac{P_x - c_5(\alpha_1 + l_5) + l_6 - s_5 s_1 l_1}{c_5}. \quad (2.53)$$

Por otro lado de  $P_y$  de la matriz 2.46 se tiene

$$P_y = -(T_{432} \alpha_1 + \Delta_1) \quad (2.54)$$

donde  $\Delta_1 = s_{43} l_3 + s_4 l_4$ , resolviendo para  $\Delta_1$

$$\Delta_1 = -(P_y + T_{432} \alpha_1) \quad (2.55)$$

sumando el cuadrado de 2.53 y 2.55 se tiene

$$\beta_1^2 + \Delta_1^2 = l_3^2(c_{43}^2 + s_{43}^2) + l_4^2(c_4^2 + s_4^2) + 2l_3 l_4(c_4 c_{43} + s_4 s_{43}) = l_4^2 + l_3^2 + 2l_3 l_4 c_3 \quad (2.56)$$

resolviendo para  $c_3$  tenemos

$$c_3 = \frac{\beta_1^2 + \Delta_1^2 - l_3^2 - l_4^2}{2l_3l_4} \quad (2.57)$$

y para obtener  $\theta_3$  se tiene

$$\theta_3 = \cos^{-1}\left(\frac{\beta_1^2 + \Delta_1^2 - l_4^2 - l_3^2}{2l_3l_4}\right). \quad (2.58)$$

Ahora, con las identidades  $c_{43} = c_4c_3 - s_4s_3$  y  $s_{43} = s_4c_3 + s_3c_4$  se expanden las ecuaciones 2.53 y 2.55 obteniendo

$$\beta_1 = (c_4c_3 - s_4s_3)l_3 - c_4l_4 \quad (2.59)$$

$$\Delta_1 = (s_4c_3 + s_3c_4)l_3 + s_4l_4 \quad (2.60)$$

resolviendo para  $c_4$  y  $s_4$  por ecuaciones simultaneas y con  $\tan(\theta_4) = \frac{s_4}{c_4}$  se tiene

$$\theta_4 = \tan^{-1}\left(\frac{\Delta_1(c_3l_3 + l_4) - \beta_1(s_3l_3)}{\beta_1(c_3l_3 + l_4) + \Delta_1(s_3l_3)}\right). \quad (2.61)$$

Finalmente para obtener  $\theta_2 = \theta_{432} - \theta_4 - \theta_3$ .

## 2.2. Jacobiano del sistema

Matemáticamente, las ecuaciones de cinemática directa definen una función entre el espacio de posiciones y orientaciones cartesianas, y el espacio de posiciones de las juntas. Las relaciones de velocidad se determinan entonces por medio del Jacobiano de esta función. Este Jacobiano es una función matricial que puede asociarse con la versión vectorial de la derivada ordinaria de una función. El Jacobiano o matriz Jacobiana, es una de las cantidades más importantes en el análisis y control del movimiento del robot. Es pertinente con virtualmente todos los aspectos de manipulación robótica[14]:

- Planificación y ejecución de trayectorias suaves.
- Determinación de configuraciones singulares.
- Ejecución de movimiento coordinado tipo antropomórfico.
- Derivación de ecuaciones de movimiento dinámicas.
- Determinación del índice de manipulabilidad de un robot.
- Transformación de fuerzas y torques ejercidos en el efector final a los torques requeridos en las juntas del manipulador.

Además, el Jacobiano es fundamental para la obtención del punto momento cero (ZMP).

La relación que existe entre las velocidades en el espacio cartesiano ( $\dot{\mathbb{X}}$ ) y las velocidades en el espacio articular ( $\dot{\Theta}$ ) a fin de encontrar la razón de cambio del último eslabón es:

$$(\dot{\mathbb{X}}) = J(\Theta)\dot{\Theta} \quad (2.62)$$

donde  $J(\Theta)$  es la matriz Jacobiana cuyos elementos son funciones lineales de las variables articulares ( $\theta$ ). Las ecuaciones 2.25 y 2.27 de la sección anterior, se observa que están en función de las variables articulares:

$$\begin{aligned} p_x &= f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ p_y &= f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \end{aligned} \quad (2.63)$$

$$p_z = f_3(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \quad (2.64)$$

y pueden ser representadas en notación vectorial como:

$$\mathbb{X} = F(\Theta) \quad (2.65)$$

derivando ambos lados de la ecuación con respecto al tiempo y empleando la regla de la cadena se obtiene:

$$\begin{aligned} \dot{\mathbb{X}} &= \frac{\partial F(\Theta)}{\partial \Theta} \dot{\Theta} \\ \dot{\mathbb{X}} &= J(\Theta)\dot{\Theta} \end{aligned} \quad (2.66)$$

El Jacobiano puede ser representado en notación matricial como:

$$J = \begin{bmatrix} \dot{j}_{11} & \cdots & \dot{j}_{1n} \\ \vdots & \ddots & \vdots \\ \dot{j}_{m1} & \cdots & \dot{j}_{mn} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(\Theta)}{\partial \theta_1} & \cdots & \frac{\partial f_1(\Theta)}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\Theta)}{\partial \theta_1} & \cdots & \frac{\partial f_m(\Theta)}{\partial \theta_n} \end{bmatrix} \quad (2.67)$$

donde  $m = 3$  que representa el número de grados de libertad cartesianos y  $n = 5$  es el número de variables articulares del robot. De acuerdo a lo anterior, la relación que existe entre las velocidades en el espacio articular y las velocidades en el espacio cartesiano para el modelo cinemático  ${}^{pie}T_{cadera}$  de la ecuación 2.24 es:

$${}^{pie}J_{cadera} = \begin{bmatrix} \dot{j}_{11} & \dot{j}_{12} & \dot{j}_{13} & \dot{j}_{14} & \dot{j}_{15} \\ \dot{j}_{21} & \dot{j}_{22} & \dot{j}_{23} & \dot{j}_{24} & \dot{j}_{25} \\ \dot{j}_{31} & \dot{j}_{32} & \dot{j}_{33} & \dot{j}_{34} & \dot{j}_{35} \end{bmatrix} \quad (2.68)$$

donde:

$$\begin{aligned}
\dot{j}_{11} &= (c_5 l_7 - s_5 l_6) c_1 + (-c_{234} s_5 l_7 - c_{234} l_5 - c_{234} c_5 l_6 - c_{23} l_4 - c_2 l_3 - l_2) s_1, \\
\dot{j}_{12} &= (-s_{234} c_5 l_6 - s_{234} s_5 l_7 - s_{234} l_5 - s_{23} l_4 - s_2 l_3) c_1, \\
\dot{j}_{13} &= (-s_{234} c_5 l_6 - s_{234} s_5 l_7 - s_{234} l_5 - s_{23} l_4) c_1, \\
\dot{j}_{14} &= -c_1 s_{234} c_5 l_6 - c_1 s_{234} s_5 l_7 - c_1 s_{234} l_5, \\
\dot{j}_{15} &= -c_1 c_{234} s_5 l_6 + c_1 c_{234} c_5 l_7 - s_1 c_5 l_6 - s_1 s_5 l_7, \\
\dot{j}_{21} &= (c_{234} c_5 l_6 + c_{234} s_5 l_7 + c_{234} l_5 + l_2 + c_{23} l_4 + c_2 l_3) c_1 + (c_5 l_7 - s_5 l_6) s_1, \\
\dot{j}_{22} &= (-s_{234} c_5 l_6 - s_{234} s_5 l_7 - s_{234} l_5 - s_{23} l_4 - s_2 l_3) s_1, \\
\dot{j}_{23} &= (-s_{234} c_5 l_6 - s_{234} s_5 l_7 - s_{234} l_5 - s_{23} l_4) s_1, \\
\dot{j}_{24} &= -s_1 s_{234} c_5 l_6 - s_1 s_{234} s_5 l_7 - s_1 s_{234} l_5, \\
\dot{j}_{25} &= -s_1 c_{234} s_5 l_6 + s_1 c_{234} c_5 l_7 + c_1 c_5 l_6 + c_1 s_5 l_7, \\
\dot{j}_{31} &= 0, \\
\dot{j}_{32} &= c_{234} c_5 l_6 + c_{234} s_5 l_7 + c_{234} l_5 + c_{23} l_4 + c_2 l_3, \\
\dot{j}_{33} &= c_{234} c_5 l_6 + c_{234} s_5 l_7 + c_{234} l_5 + c_{23} l_4, \\
\dot{j}_{34} &= c_{234} c_5 l_6 + c_{234} s_5 l_7 + c_{234} l_5, \\
\dot{j}_{35} &= -s_{234} s_5 l_6 + s_{234} c_5 l_7,
\end{aligned} \tag{2.69}$$

considerando el modelo cinemático directo  ${}^{cadera}T_{pie}$  de la ecuación 2.26 y derivando parcialmente con respecto a las variables articulares, el Jacobiano esta dado por:

$${}^{cadera}J_{pie} = \begin{bmatrix} \dot{j}_{11} & \dot{j}_{12} & \dot{j}_{13} & \dot{j}_{14} & \dot{j}_{15} \\ \dot{j}_{21} & \dot{j}_{22} & \dot{j}_{23} & \dot{j}_{24} & \dot{j}_{25} \\ \dot{j}_{31} & \dot{j}_{32} & \dot{j}_{33} & \dot{j}_{34} & \dot{j}_{35} \end{bmatrix} \tag{2.70}$$

donde:

$$\begin{aligned}
\dot{j}_{11} &= c_5 c_{234} s_1 l_1 + s_5 c_1 l_1, \\
\dot{j}_{12} &= c_5 s_{432} c_1 l_1 + c_5 s_{234} l_2, \\
\dot{j}_{13} &= c_5 s_{432} c_1 l_1 + c_5 s_{234} l_2 + c_5 s_{43} l_3, \\
\dot{j}_{14} &= c_5 s_{432} c_1 l_1 + c_5 s_{234} l_2 + c_5 s_{43} l_3 + c_5 s_4 l_4, \\
\dot{j}_{15} &= s_5 c_{432} c_1 l_1 + s_5 c_{234} l_2 + c_5 s_1 l_1 + s_5 c_{43} l_3 + s_5 c_4 l_4 + s_5 l_5, \\
\dot{j}_{21} &= s_{234} s_1 l_1, \\
\dot{j}_{22} &= -c_{234} c_1 l_1 - c_{234} l_2, \\
\dot{j}_{23} &= -c_{234} c_1 l_1 - c_{234} l_2 - c_{43} l_3, \\
\dot{j}_{24} &= -c_{234} c_1 l_1 - c_{234} l_2 - c_{43} l_3 - c_4 l_4, \\
\dot{j}_{25} &= 0, \\
\dot{j}_{31} &= s_5 c_{432} s_1 l_1 - c_5 c_1 l_1, \\
\dot{j}_{32} &= s_5 s_{432} c_1 l_1 + s_5 s_{234} l_2, \\
\dot{j}_{33} &= s_5 s_{432} c_1 l_1 + s_5 s_{234} l_2 + s_5 s_{43} l_3, \\
\dot{j}_{34} &= s_5 s_{432} c_1 l_1 + s_5 s_{234} l_2 + s_5 s_{43} l_3 + s_5 s_4 l_4, \\
\dot{j}_{35} &= -c_5 c_{432} c_1 l_1 - c_5 c_{234} l_2 + s_5 s_1 l_1 - c_5 c_{43} l_3 - c_5 c_4 l_4 - c_5 l_5,
\end{aligned} \tag{2.71}$$

Estas matrices establecen el mapeo entre las velocidades en el espacio articular y las velocidades del espacio cartesiano del robot bípodo. [1]

## 2.3. Conclusiones

En este capítulo se derivó el modelo matemático de un robot bípodo de 10 gdl, el cual consta de un modelo de la cinemática directa, un modelo de la cinemática inversa y el Jacobiano. La cinemática directa se determinó a través de los parámetros de D-H, los cuales proporcionan la posición de cualquier articulación del robot. La cinemática inversa es capaz de calcular todos los ángulos de las juntas de cualquier posición alcanzable de cualquier eslabón.





# Capítulo 3

## Introducción al Control Difuso

### 3.1. Control Difuso

El control de sistemas utilizando técnicas de razonamiento aproximado permite una mayor flexibilidad que el control clásico. Un control difuso transmite el raciocinio humano hacia un sistema y así es posible prescindir de la rigidez matemática, este control se basa en la lógica difusa.

En la lógica difusa todo es cuestión de grado, es decir, va más allá de una lógica normal que consiste en cero y uno. Este tipo de lógica es de gran utilidad para la navegación de robots, pues permite al robot reconocer su entorno con cierta gradualidad y poder moverse adecuándose a él.

Así el control difuso utiliza expresiones lingüísticas para formular las reglas que controlarán un sistema. La lógica difusa sugiere un cierto grado de pertenencia para los datos, esto permite al controlador difuso tomar diferentes grados de acción en un sistema, el cual lo realiza a través de la inferencia. La inferencia es el proceso mediante el cual se genera un mapeo para asignar a una entrada una salida utilizando lógica difusa. El proceso de inferencia provee las bases para la toma de decisiones del sistema. Este proceso involucra la utilización de funciones de pertenencia y las reglas generadas en la base de conocimiento. Existen diferentes métodos de inferencia, los más comunes son los de Mamdani y Takagi Sugeno-Kang. [29]

El método de Mamdani utiliza reglas tipo Si-Entonces. Una regla de la base de reglas o base de conocimiento tiene dos partes, el antecedente y la conclusión. En un sistema difuso tipo Mamdani tanto el antecedente como el consecuente de las reglas están dados por expresiones lingüísticas. Las reglas de la base de conocimiento de un sistema Takagi-Sugeno son diferentes a las de los sistemas Mamdani pues el consecuente de estas reglas ya no es una etiqueta lingüística

sino que es una función de la entrada que tenga el sistema en un momento dado. En los sistemas difusos tipo Takagi-Sugeno, los valores que arrojan los consecuentes de las diferentes reglas que se han activado en un momento determinado ya son valores numéricos por lo que no se necesita una etapa de defuzzificación.

Después de la inferencia, se tendrá una conclusión difusa, una variable lingüística cuyos valores han sido asignados por grados de pertenencia, sin embargo, usualmente necesitamos un escalar que corresponda a estos grados de pertenencia, a este proceso se le llama defuzzificación.

## 3.2. Diferentes dominios de aplicación

Las primeras aplicaciones del control difuso han sido desarrolladas en Europa por Mamdani, Van Lauta Lemke, Willayes, etc. en los años 1975, este tipo de control ha sido muy utilizado en Japón para finales de los años 80s y comienzo de los años 90s, por ejemplo, sobre el metro Sendai (julio 1987), en el diseño de una lavadora Aisaigo Day Fuzzy de Matsushita (febrero 1990). Debido al perfil de la evolución de la tecnología que permite abordar los problemas en tiempo real considerando dinámicas rápidas, la industria japonesa ha desarrollado sus productos al público en general con base en la lógica difusa.

- Aparatos electrodomésticos (lavadoras, aspiradoras, hornos,...)
- Sistemas de audio-vídeo (aparatos fotográficos autofocus, cámaras estabilizadoras de imágenes, copiadoras,...)
- Sistemas en automóviles (BVA, ABS, suspensión, aire acondicionado,...)
- Sistemas móviles autónomos
- Sistemas de transporte (tren, metro, elevadores,...)
- Sistemas de decisión, diagnóstico y reconocimiento.
- Sistemas de control de producción, transformación, tratamiento del producto, etc..

## 3.3. El concepto de la lógica difusa

Este concepto permite considerar las clases de objetos donde las fronteras no son determinadas claramente por la introducción de una función característica (función de pertenencia de los objetos a la clase) que toma valores entre 0 y 1, contrariamente a los conjuntos "booleanos", donde la función característica no toma más que 2 valores posibles: el 0 y 1.

Estos conjuntos difusos tienen la gran ventaja de constituir una representación matemática de etiquetas lingüísticas utilizadas en la expresión del conocimiento cualitativo. Entonces, estos conjuntos aparecen como un medio de realizar una interfaz entre la información numérica (cuantitativa) y la información simbólica (lingüística).

## 3.4. Las bases del control difuso

A este momento ya se mencionó el principio del control difuso y de la lógica difusa. Esta sección nos permitirá formalizar matemáticamente estos conceptos, tales como fuzzificar que es pasar de una variable real a una variable difusa, así como construir un esquema de control por la definición de las reglas de inferencia.

### 3.4.1. Las funciones de pertenencia

En un conjunto booleano podemos decir “ $x$  pertenece a un conjunto  $A$ ”, ya sea 1 si pertenece y 0 si no. En el caso de la lógica difusa este coeficiente de apreciación de ‘el aire tiene una temperatura igual a  $30^{\circ}\text{C}$  es caliente’ vale 0.6 que significa que esta temperatura corresponde a ‘más caliente’. Podemos para toda temperatura, es decir, para toda  $x$ , definir este coeficiente directamente a  $x$ . Esta propiedad se presenta fácilmente por una función de pertenencia  $\mu_A(x)$  que tiene valores dentro del intervalo  $[0, 1]$ , la notación significa ‘el coeficiente de pertenencia de  $x$  al conjunto caracterizado por  $A$ ’,  $x$  se refiere a la variable lingüística y el índice  $A$  designa al conjunto.

De la misma forma, una variable  $y$  pertenecerá a un conjunto  $B$  con una función de pertenencia  $\mu_B(y)$ , por ejemplo, ‘el viento está fuerte’. Se puede asociar  $x$  y  $y$  en una misma frase, por ejemplo, el conjunto  $C$ , ‘el aire está caliente y el viento es fuerte’. La variable  $z$  definida por ‘aire caliente y viento fuerte’ corresponde a la intersección de ‘aire caliente’ y de ‘viento fuerte’. El conjunto  $C$  corresponde a la intersección de los conjuntos  $A$  y  $B$ . El valor de  $\mu_C(z)$  se deduce de los valores de  $\mu_A(x)$  y  $\mu_B(y)$ . Existen diversas soluciones para traducir el problema matemáticamente. Un hecho con cierta certidumbre tendrá una función de pertenencia igual a 1 para el punto de funcionamiento considerado. Un hecho incierto tendrá una función de pertenencia inferior o igual a 1.

Estas funciones de pertenencia pueden ser de diferentes formas, las formas trapezoidales y triangulares son las más utilizadas. Estas son formas simples compuestas por trozos de rectas[30]. En la figura 3.1 se muestran unas funciones de pertenencia de forma triangular donde están completamente definidas por los valores de  $a, b, c$ . En el caso de las formas trapezoidales están caracterizadas completamente por cuatro valores:  $a, b, c, d$ .

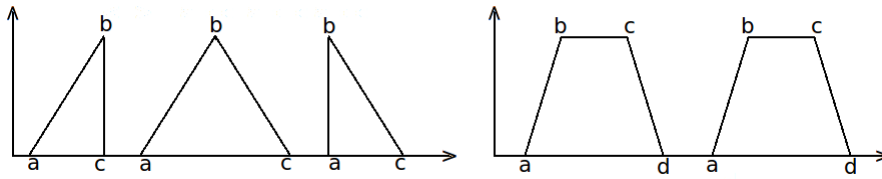


Figura 3.1: Regiones de combinación de entrada del control difuso PI

Existen otras formas de funciones de pertenencia llamadas: formas de campana. La primera de ellas esta expresada por

$$\mu(x) = \frac{1}{1 + \left(\frac{x-x_0}{a}\right)^2}; \quad -\infty < x < +\infty \quad (3.1)$$

y mostrada en la figura 3.2. En este caso  $x_0$  determina la posición de la punta de la curva  $\mu = 1$ , el parámetro  $\beta$  impone la anchura del dominio. Otra posibilidad de forma de función de

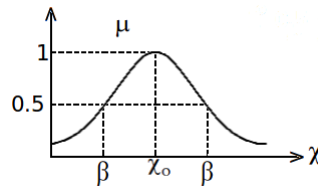


Figura 3.2: Función de pertenencia en forma de campana

pertenencia se obtiene a partir de las funciones trigonométricas, por ejemplo

$$\mu(x) = \frac{1}{2} \left( 1 + \cos\left(\frac{\pi(x-x_0)}{2a}\right) \right); \quad -2a \leq x \leq x_0 + 2a. \quad (3.2)$$

### 3.4.2. Los operadores de la lógica difusa

Es importante saber componer entre los diversos predicados y sus funciones de pertenencia como en el ejemplo ‘el aire está frío y el viento está fuerte’ o en ‘si el aire está frío o si el viento está fuerte’ debemos cerrar la puerta. Aparecen dos tipos de composición  $y$  y  $o$  a los cuales hay que añadir la negación.

Notemos que  $x$  y  $y$  son las variables lingüísticas caracterizando la temperatura del aire y la fuerza del viento y  $\mu_A(x), \mu_B(y), \mu_Y(z), \mu_O(z)$  y  $\mu_C(z)$  con  $z = \{x, y\}$  son las funciones de pertenencia asociadas a las propiedades de ‘el aire está frío’, ‘el viento está fuerte’, ‘el aire está frío y el viento está fuerte’, ‘el aire está frío o el viento está fuerte’ y ‘el aire no está frío’, respectivamente.

### Operador No

La propiedad ‘el aire no está frío’ puede ser caracterizada de una forma evidente por la función de pertenencia

$$\mu_C(z) = 1 - \mu_A(x) \quad (3.3)$$

La figura 3.3 pone en evidencia esta relación.[32]

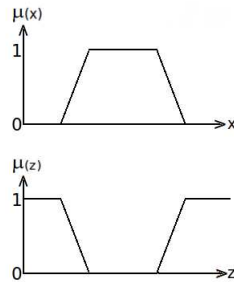


Figura 3.3: Función de pertenencia del ejemplo del operador No

### Operador Y

La solución más simple y la más utilizada para caracterizar la satisfacción simultánea de dos propiedades es el operador Y

$$\mu_Y(z) = \mu_{A \text{ y } B}(z) = \min\{\mu_A(x), \mu_B(y)\} \quad (3.4)$$

el cual es el operador mínimo o también conocido como operador intersección. Esta operación se muestra en la figura 3.4. [32]

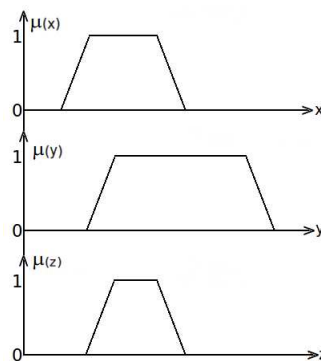


Figura 3.4: Función de pertenencia del ejemplo del operador Y

### Operador O

La realización del operador en la lógica difusa se hace en general por la formación del máximo aplicado a las funciones de pertenencia  $\mu_A(x)$  y  $\mu_B(y)$  de dos conjuntos  $A$  y  $B$ , es decir,

$$\mu_O(z) = \mu_{A \circ B}(z) = \text{máx}\{\mu_A(x), \mu_B(y)\} \quad (3.5)$$

en la figura 3.5 se muestra esta operación.[32]

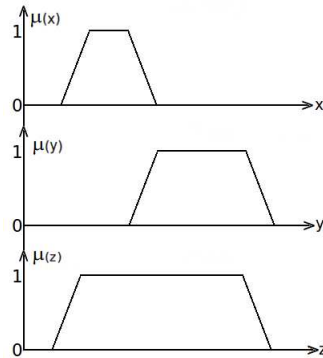


Figura 3.5: Función de pertenencia del ejemplo del operador O

### 3.4.3. Universo de discurso y clases de pertenencia

El usuario generalmente describe la variable ‘temperatura’ por un cierto número de palabras, por ejemplo ‘caliente’, ‘frío’, ‘tibio’, o ‘muy caliente’, ‘calientito’, ‘muy frío’. Para cada uno de estos predicados podemos definir una función de pertenencia. [31]

El universo de discurso será considerado como el dominio de funcionamiento del proceso. Los problemas que pueden surgir son:

- ¿cuantos predicados son necesarios para el control y como elegirlos?
- ¿es necesario elegir cada predicado?

En general, se puede dar un número de reglas de control importante, pero el interés del control difuso es que solo un número pequeño de reglas es necesario. El número de predicados depende esencialmente de la manera que el experto puede describir el proceso y de la precisión deseada. Por ejemplo, en el control difuso clásico 5 predicados son un buen compromiso (grande, muy grande, mediano, pequeño, muy pequeño), algunas veces tres predicados son suficientes (grande, mediano, pequeño) o (positivo, negativo, cercano al cero). Los primeros tres sirven para la regulación, mientras que los segundos sirven para el seguimiento.

El control difuso permite de asegurar una transición armoniosa entre los dos modos de funcionamiento. El universo de discurso de una variable cubrirá el conjunto de valores tomados por esta variable.

### 3.5. Esquema de un control difuso

En la implementación de un control difuso aparecen tres grandes módulos. El primer módulo trata las entradas del sistema. Primeramente se define un universo de discurso, una partición de este universo en clases para cada entrada y las funciones de pertenencia para cada una de las entradas, por ejemplo: presión: grande, pequeña, flujo: débil, nulo, fuerte. Esta primera etapa, llamada fuzzificación, consiste en atribuir al valor real de cada entrada, en el instante  $t$ , su función de pertenencia a cada una de las clases ya definidas, es decir, transforma una entrada real a un subconjunto difuso.

El segundo módulo consiste en la aplicación de reglas tipo 'si la diferencia de temperatura es grande, entonces hay que disminuir el flujo de combustible'. Estas reglas permiten pasar de un grado de pertenencia de una cantidad a grados de pertenencia de un control. Este módulo está constituido por una base de reglas y de un motor de inferencia que permite el cálculo.

El tercer y último módulo describe la etapa de defuzzificación que es la transformación inversa del primer módulo. Este módulo permite pasar de un grado de pertenencia del control a la determinación del valor numérico de este control.

El esquema de control se presenta en la figura 3.6, [29]

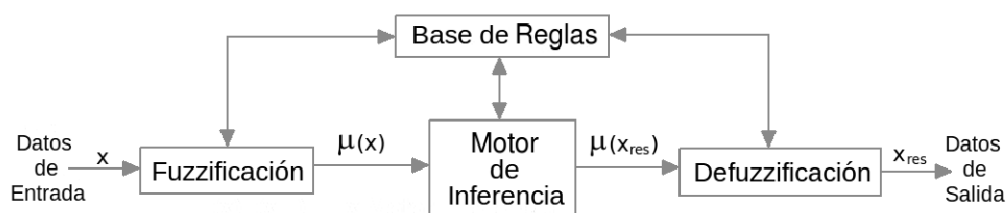


Figura 3.6: Esquema a bloques del control difuso

donde  $x$  representa el vector de entradas,  $x_{res}$  es el vector de control,  $\mu(x)$  y  $\mu(y)$  son las funciones de pertenencia correspondientes.

### 3.6. Conclusiones

Es este capítulo se describió brevemente los conceptos básicos de la lógica difusa, y del control difuso. Se expusieron los conceptos del universo del discurso, funciones de pertenencia, conjuntos difusos, predicados y el esquema básico de un control difuso.



# Capítulo 4

## Diseño de un controlador difuso para el problema de seguimiento en un robot bípedo

### 4.1. Diseño del Control

Para simplificar el diseño de control para el seguimiento de trayectorias en las articulaciones de un robot bípedo de 10 gdl se emplea o se diseña un control independiente para cada articulación, donde la entrada de control de la articulación depende solo del valor de la posición correspondiente a su eje. Entonces el esquema de control propuesto para resolver el problema establecido se presenta en la Figura 5.24.

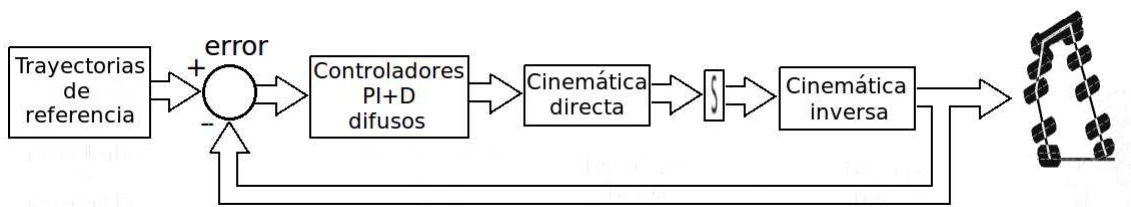


Figura 4.1: Esquema a bloques del controlador PI+D difuso que se implementará

## 4.2. Ley de control

Los controladores PID convencionales son quizá los más conocidos y extensamente usados en la industria moderna. Los controladores PID son simples, fiables y efectivos [20]. El uso de control PID de ganancia fija, resulta ser una buena alternativa para controlar sistemas dinámicos; ya que proporcionan tiempos de respuesta rápidos, sin embargo entre mayor es la precisión requerida en el sistema el ajuste de este tipo de control es más difícil ya que son bastante sensibles a las señales de ruido y en ocasiones introducen oscilaciones cuando se presentan retardos en el sistema. El controlador PID Difuso es generalmente superior a uno convencional, particularmente para sistemas de orden superior, con tiempo de retardo y no lineales donde el control PID convencional se dificulta respecto a cálculos. Para sistemas lineales de bajo orden, ambos controladores trabajan bien, entonces, en casos simples el control convencional PID es recomendado debido a que su estructura es más simple. [33] [34]

La implementación de controladores PID en hardware basados en lógica difusa es motivada por su capacidad de implementar un comportamiento de control altamente flexible. Con esto se logra que nuestros sistemas puedan ajustarse a condiciones cambiantes que son muchas veces imposibles de predecir, tales como los cambios ambientales o las condiciones de desgaste en sus componentes físicos, por citar algunos ejemplos. El problema de utilizar controladores PID Difusos es que sus métodos de diseño son ligeramente más avanzados y sus fórmulas resultantes son un tanto más complicadas pues contienen ganancias variables de control cuando en el control PID convencional las ganancias de control son constantes. Aunque el control PID Difuso es diseñado por matemática difusa, sus fórmulas finales de control son de controladores convencionales. De esta forma estas pueden ser usadas directamente para remplazar las fórmulas convencionales. Además sus ganancias variables de control son convertidas en expresiones que tienen la capacidad de adaptarse a los efectos de retardo, no linealidad e incertidumbre del sistema dado.

El algoritmo de control PID actúa sobre la señal de error, sin embargo en el momento de un cambio brusco de la referencia, la derivada del error se hace infinita, esta actuación puede resultar nociva para la planta. La solución a este inconveniente resulta en mover el modo diferencial al lazo de retroalimentación, puesto que la señal de salida del sistema  $y(t)$  es más suave que la señal de error. De esta forma se evita que aparezcan señales de control muy bruscas o excesivamente elevadas cuando la consigna varía bruscamente.

Para el diseño de un controlador difuso PI+D que se utilizará en este caso, se considera primeramente el controlador PI+D convencional mostrado en la Figura 4.2.

donde  $r$  es la referencia,  $e$  es el error,  $u_{PI}$  es el controlador PI,  $u_D$  es el controlador D, y  $y$  es la salida.

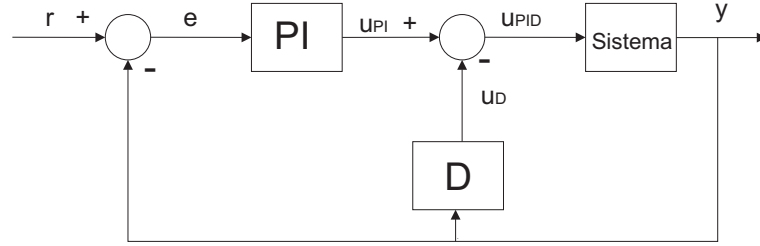


Figura 4.2: Sistema de control PID convencional en tiempo continuo

Los controladores PI y D serán diseñados de forma individual para después combinarlos. Primeramente, para el diseño de un controlador difuso PI, la función de transferencia de un controlador PI clásico es

$$u_{PI}(s) = \left( K_p^c + \frac{K_i^c}{s} \right) E(s) \quad (4.1)$$

donde  $K_p^c$  y  $K_i^c$  son las ganancias proporcional e integral respectivamente y  $E(s)$  es la señal del error de seguimiento. Transformando esta ecuación a su versión discreta aplicando la transformación bilineal (4.2) se obtiene

$$s = \frac{2z-1}{Tz+1} \quad (4.2)$$

$$u_{PI}(z) = \left( K_p^c - \frac{K_i^c T}{2} + \frac{K_i^c T}{1-z^{-1}} \right) E(z) \quad (4.3)$$

definiendo ahora

$$K_p = K_p^c - \frac{K_i^c T}{2} \quad y \quad K_i = K_i^c T \quad (4.4)$$

y tomando la transformada z inversa se tiene

$$u_{PI}(nT) - u_{PI}(nT - T) = K_p(e(nT) - e(nT - T)) + K_i T e(nT) \quad (4.5)$$

dividiendo por  $T$

$$\Delta u_{PI} = K_p v(nT) + K_i e(nT) \quad (4.6)$$

donde

$$\Delta u_{PI} = \frac{u_{PI}(nT) - u_{PI}(nT - T)}{T} \quad (4.7)$$

$$\Delta e(nT) = \frac{(e(nT) - e(nT - T))}{T} = v(nT) \quad (4.8)$$

de (4.7) se tiene

$$u_{PI}(nT) = u_{PI}(nT - T) + T \Delta u_{PI} \quad (4.9)$$

reemplazando  $T$  por la ganancia de control difusa  $K_{u,PI}$  resulta en

$$u_{PI}(nT) = u_{PI}(nT - T) + K_{u,PI} \Delta u_{PI} \quad (4.10)$$

Para el controlador difuso D, la función de transferencia de la entrada  $y$  y  $u_D$  como salida es

$$U_D = sK_d^c Y(s) \quad (4.11)$$

donde  $K_d^c$  es la ganancia de control y  $Y(s)$  es la señal de salida. Utilizando la transformación bilineal (4.2), obtenemos la siguiente función de transferencia

$$U_D = K_d^c \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} Y(z) \quad (4.12)$$

y realizando el mismo procedimiento que para el PI se obtiene

$$\Delta u_D = K_d \Delta y(nT) \quad (4.13)$$

donde  $\Delta u_D(nT) = \frac{u_D(nt) + u_D(nt-T)}{T}$  es la salida de control difuso D,  $\Delta y(nT) = \frac{y(nT) - y(nT-T)}{T}$  es la tasa de cambio de la salida  $y$  y  $K_d = \frac{2K_d^c}{T}$ .

El único dato relevante para el desempeño de la salida que nos proporciona la ecuación 4.13 es  $\Delta y(nT)$ . En base a esta única señal, es imposible llegar a una ley de control difuso útil. Por lo tanto, se busca otra señal que proporcione información acerca de la salida, la elección más lógica es la señal negativa del error. Entonces, la señal de salida  $y$  estará por encima de la referencia  $r$ .

Ahora, se modifica la ecuación (4.13) añadiendo en la parte derecha la señal  $Ky_d(nT)$ , donde  $y_d(nT) = y(nT) - r(nT) = -e(nT)$  obteniendo

$$\Delta u_D = K_d \Delta y(nT) + Ky_d(nT) \quad (4.14)$$

con  $K = 1$ , por otro lado se sabe que

$$u_D(nt) = -u_D(nt - T) + K_{u,D} \Delta u_D(nT) \quad (4.15)$$

donde  $K_{u,D}$  es la ganancia de control D difuso.

Finalmente, para el controlador PI+D difuso se suman la ley de control PI difuso (4.10) y la ley de control D difuso (4.15) para obtener

$$\begin{aligned} u_{PID}(nT) &= u_{PI}(nT) - u_D(nT) \\ &= u_{PI}(nT - T) + K_{u,PI} \Delta u_{PI}(nT) \\ &\quad + u_D(nT - T) - K_{u,D} \Delta u_D(nT) \end{aligned} \quad (4.16)$$

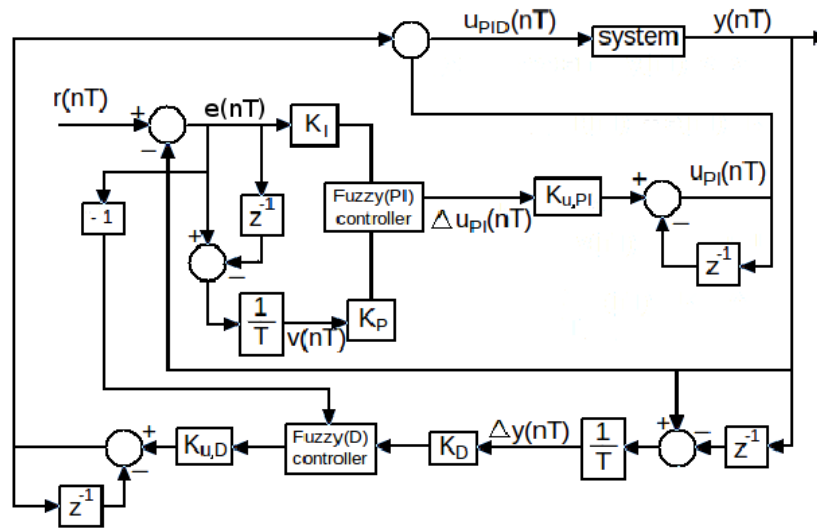


Figura 4.3: Sistema de control PID difuso

que es la ley de control difusa PI+D implementada en la Figura 4.3. [20]

En resumen, en el sistema de control difuso PI+D solo es necesario ajustar las ganancias  $K_i, K_p, K_{u,PI}, K_D$  y  $K_{u,D}$ , donde para simplificar más el diseño se considera  $K_{u,PI} = K_{u,D} = T$  y las acciones de control  $\Delta u_{PI}(nT)$  y  $\Delta u_D(nT)$  que se obtienen a continuación.

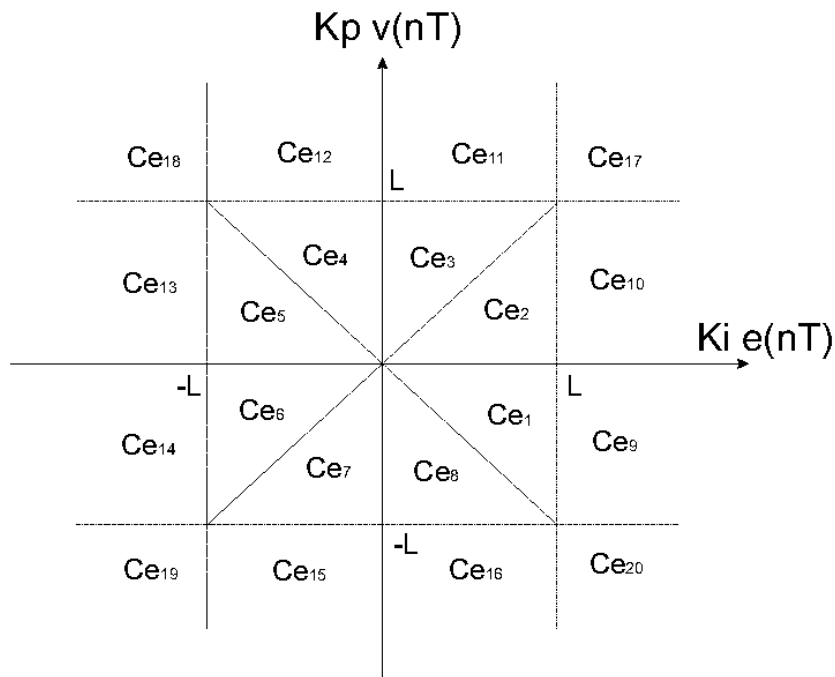


Figura 4.4: Regiones de combinación de entrada del control difuso PI

En el control PI Difuso primero se introduce un parámetro constante  $L > 0$ , y se descompone el plano limitado por  $L$  en 20 regiones que representan las combinaciones de entrada como la que se muestra en la Figura 4.4, donde  $K_i e(nT)$  y  $K_p v(nT)$  son llamadas señal de error y rango del cambio de la señal de error, respectivamente. Las funciones de pertenencia para las 2 entradas (error y rango) se muestran en la Figura 4.5 y corresponden a las ecuaciones 4.17.

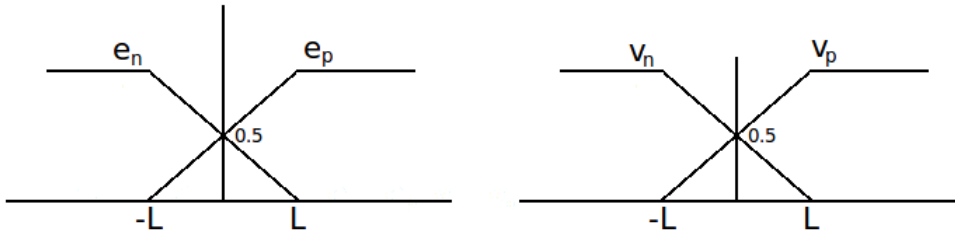


Figura 4.5: Funciones de pertenencia de  $e(nT)$  y  $v(nT)$

$$\begin{aligned}
 e_n &= \frac{L - k_i e(nt)}{2L} \\
 e_p &= \frac{L + k_i e(nt)}{2L} \\
 v_n &= \frac{L - k_p v(nt)}{2L} \\
 v_p &= \frac{L + k_p v(nt)}{2L}
 \end{aligned} \tag{4.17}$$

Entonces de acuerdo a la localización de las entradas ( $K_i e(nT)$ ,  $K_p v(nT)$ ) al bloque de control difuso, el incremento de la salida de control es procesada por las siguientes reglas difusas:

- $R_1$  : SI  $e = e_n$  Y  $v = v_n$  ENTONCES  $Salida_{PI} = o_n$ .
- $R_2$  : SI  $e = e_n$  Y  $v = v_p$  ENTONCES  $Salida_{PI} = o_z$ .
- $R_3$  : SI  $e = e_p$  Y  $v = v_n$  ENTONCES  $Salida_{PI} = o_z$ .
- $R_4$  : SI  $e = e_p$  Y  $v = v_p$  ENTONCES  $Salida_{PI} = o_p$ .

donde  $e = r - y$  es el error y  $v = \dot{e} = 0 - \dot{y} = -\dot{y}$  es la tasa de cambio del error.  $e_n, e_p, v_n, v_p$  son las funciones de pertenencia de las entradas: error negativo, error positivo, etc. y  $o_n, o_z, o_p$  output negative (salida negativa), output zero (salida cero) y output positive (salida positiva), respectivamente. El operador ‘Y’ es el mínimo y la fórmula de defuzzificación es el promedio ponderado.

Si se considera la región  $C_{e1}$  donde las reglas se activarán de la siguiente forma

$$R_1 : e_n < 0.5 \text{ y } v_n > 0.5 = \min\{e_n, v_n\} = e_n \rightarrow \text{Salida}_{PI} = o_n$$

$$R_2 : e_n < 0.5 \text{ y } v_p < 0.5 = \min\{e_n, v_p\} = e_n \rightarrow \text{Salida}_{PI} = o_z$$

$$R_3 : e_p > 0.5 \text{ y } v_n > 0.5 = \min\{e_p, v_n\} = v_n \rightarrow \text{Salida}_{PI} = o_z$$

$$R_4 : e_p > 0.5 \text{ y } v_p < 0.5 = \min\{e_p, v_p\} = v_p \rightarrow \text{Salida}_{PI} = o_p$$

donde la salida es

$$\Delta u(nT) = \frac{e_n \times o_n + e_n \times o_z + v_n \times o_z + v_p \times o_p}{e_n + e_n + v_n + v_p}, \quad (4.18)$$

sustituyendo las funciones de pertenencia correspondientes y  $o_n = -L$ ,  $o_z = 0$  y  $o_p = L$  se obtiene

$$\Delta u(nT) = \frac{L(K_i e(nT) + K_p v(nT))}{2(2L - K_i |e(nT)|)} \quad (4.19)$$

El desarrollo de  $\Delta u(nT)$  en todas las regiones se presenta en el Apéndice A. Con las ecuaciones obtenidas de cada región se forma la ley de control difuso PI mostradas en la ecuación siguiente.

$$\begin{aligned} \Delta u_{PI}(nT) &= \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L - K_i |e(nT)|)}, & C_{e1}, C_{e2} \\ &= \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L - K_p |v(nT)|)}, & C_{e3}, C_{e4} \\ &= \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L + K_i |e(nT)|)}, & C_{e5}, C_{e6} \\ &= \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L + K_p |v(nT)|)}, & C_{e7}, C_{e8} \\ &= \frac{1}{2}[L + K_p v(nT)], & C_{e9}, C_{e10} \\ &= \frac{1}{2}[L + K_i e(nT)], & C_{e11}, C_{e12} \\ &= \frac{1}{2}[-L + K_p v(nT)], & C_{e13}, C_{e14} \\ &= \frac{1}{2}[-L + K_i e(nT)], & C_{e15}, C_{e16} \\ &= 0, & C_{e18}, C_{e20} \\ &= L, & C_{e17} \\ &= -L, & C_{e19} \end{aligned} \quad (4.20)$$

Para la acción de control  $\Delta u_D(nT)$  se realiza el mismo procedimiento que para el PI difuso, solo que ahora las entradas al controlador difuso D son  $y_d(nT)$  y  $\Delta y(nT)$  y las combinaciones de entrada  $(K y_d(nT), K_d \Delta y(nT))$  para este controlador están mostradas en la figura (4.6). Con las funciones de pertenencia (4.7) se obtienen las siguientes reglas

$R_5$  : SI  $y_d = y_{dp}$  Y  $\Delta y = \Delta y_p$  ENTONCES  $Salida_D = o_z$ .

$R_6$  : SI  $y_d = y_{dp}$  Y  $\Delta y = \Delta y_n$  ENTONCES  $Salida_D = o_p$ .

$R_7$  : SI  $y_d = y_{dn}$  Y  $\Delta y = \Delta y_p$  ENTONCES  $Salida_D = o_n$ .

$R_8$  : SI  $y_d = y_{dn}$  Y  $\Delta y = \Delta y_n$  ENTONCES  $Salida_D = o_z$ .

la acción de control difusa D es

$$\begin{aligned}
 \Delta u_D(nT) &= \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L - K|y_d(nT)|)}, & C_{e1}, C_{e2}, C_{e3}, C_{e8} \\
 &= \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L + K|y_d(nT)|)}, & C_{e4}, C_{e5}, C_{e6}, C_{e7} \\
 &= \frac{1}{2}[L - K_d\Delta y(nT)], & C_{e9}, C_{e10} \\
 &= \frac{1}{2}[-L + Ky_d(nT)], & C_{e11}, C_{e12} \\
 &= \frac{1}{2}[-L - K_d\Delta y(nT)], & C_{e13}, C_{e14} \\
 &= \frac{1}{2}[L + Ky_d(nT)], & C_{e15}, C_{e16} \\
 &= 0, & C_{e17}, C_{e19} \\
 &= L, & C_{e18} \\
 &= -L, & C_{e20}
 \end{aligned} \tag{4.21}$$

El desarrollo de  $\Delta u_D(nT)$  se presenta en el Apéndice A.

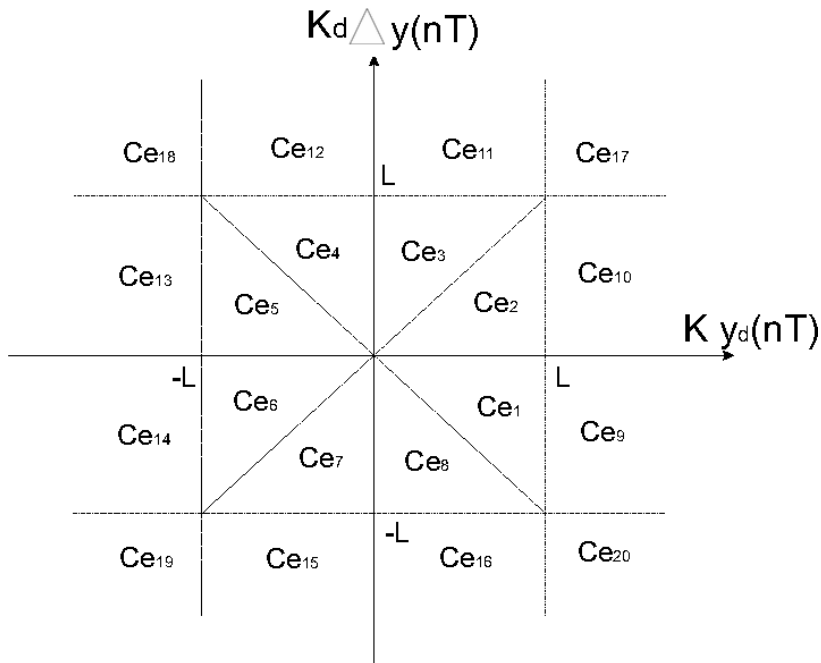


Figura 4.6: Regiones de combinación de entrada del control difuso D



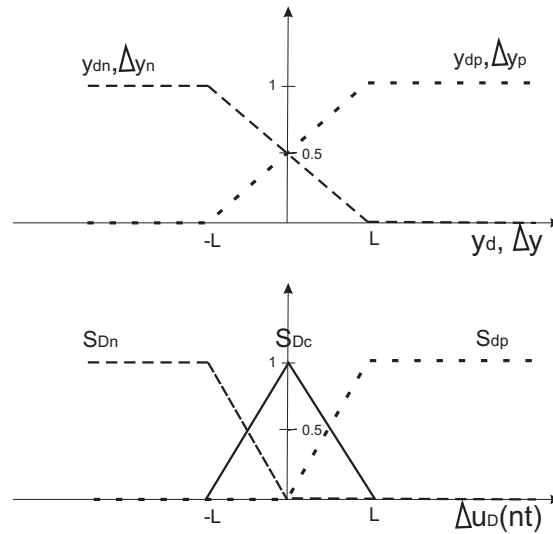


Figura 4.7: Funciones de pertenencia para el control difuso D

### 4.3. Conclusiones

En este capítulo se presenta el diseño de los controladores difusos PI+D, aunque el desarrollo de los controladores parece complejo, el resultado son formulas fácilmente intercambiables para efecto de la implementación, donde solo es necesario ajustar los parámetros  $K_p, K_i, K_d$  y  $L$ .



# Capítulo 5

## Simulación y análisis de resultados

### 5.1. Simulaciones

Primero se realizó el movimiento del robot directo, es decir, se introdujeron las trayectorias de referencia directamente al modelo del robot para ver su comportamiento sin control y sin retroalimentación, estos fueron los resultados. Las gráficas de la Figura 5.1 muestran las trayectorias de referencia dadas para la pierna derecha.

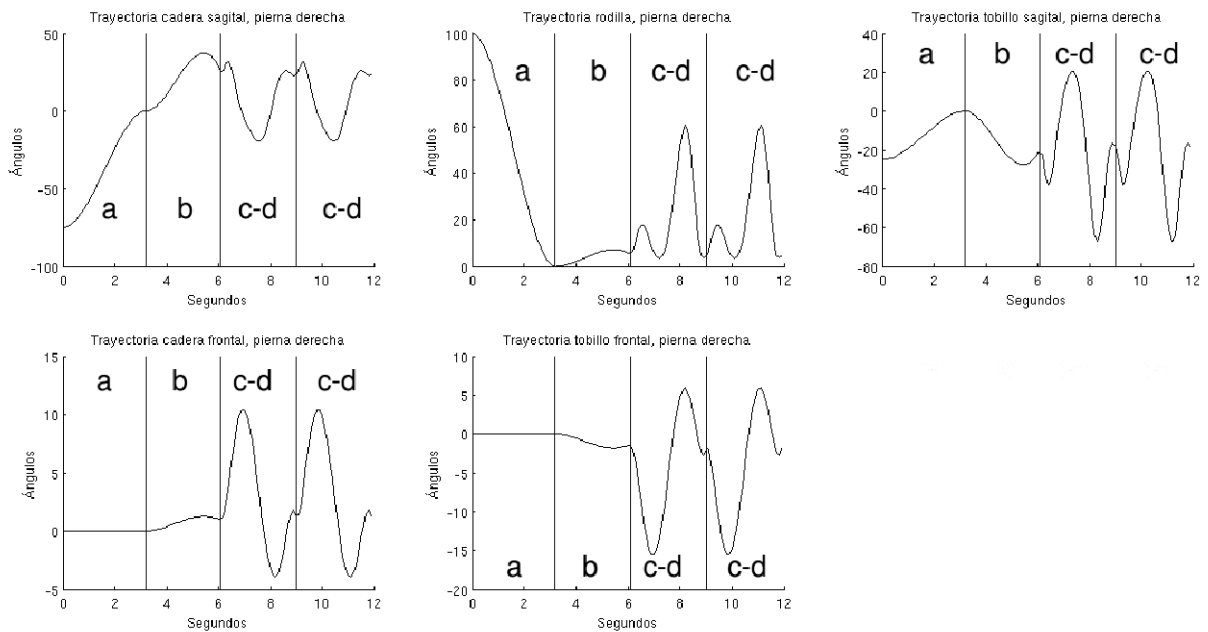


Figura 5.1: Trayectorias deseadas, pierna derecha.

Y en las gráficas de la Figura 5.2 se observan las trayectorias de referencia para la pierna izquierda.

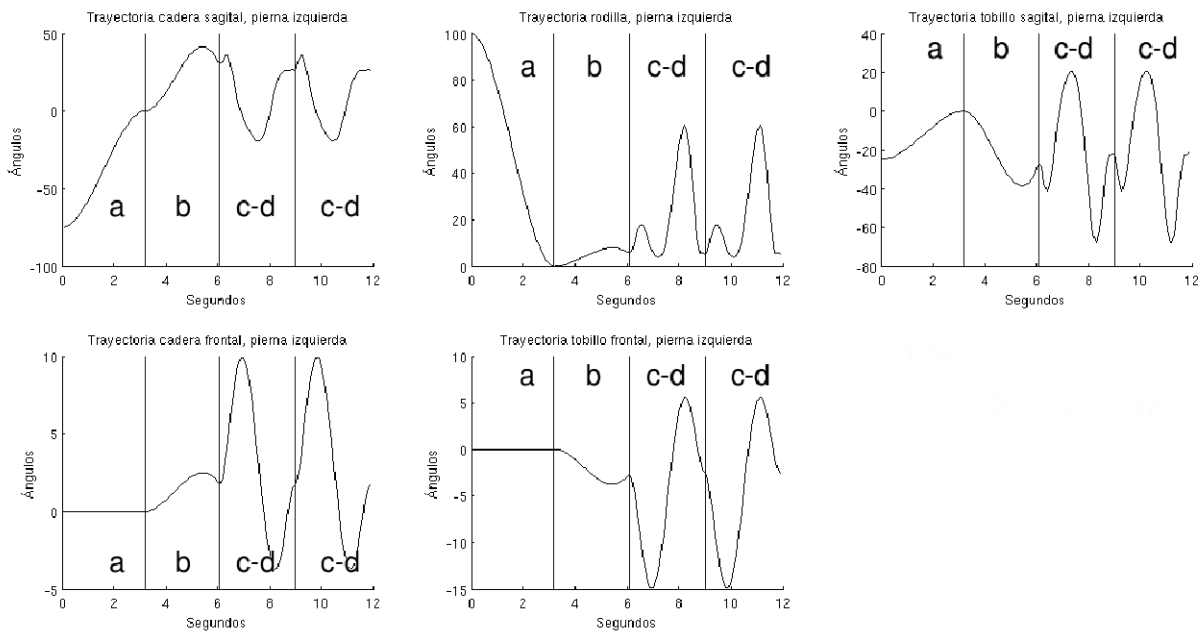


Figura 5.2: Trayectorias deseadas, pierna izquierda.

Para tener una mejor apreciación, se revisará el modelo fase por fase. En las Figuras 5.3 y 5.4 se ve el movimiento que realiza en la transición de “posición inicial - posición vertical” para ambas piernas, respectivamente.

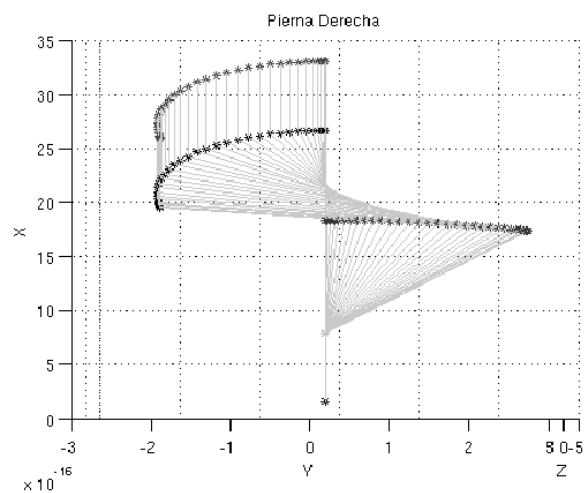


Figura 5.3: Representación del movimiento que realiza la pierna derecha (fase a).

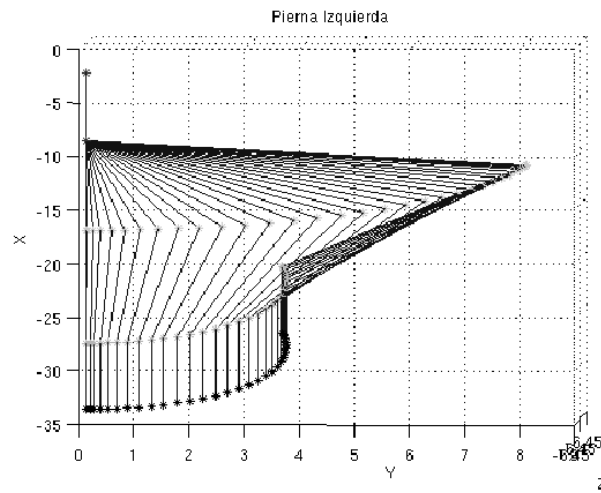


Figura 5.4: Representación del movimiento que realiza la pierna izquierda (fase a).

En las Figuras 5.5 y 5.6 se ven las trayectorias correspondientes a esta fase para cada junta del robot.

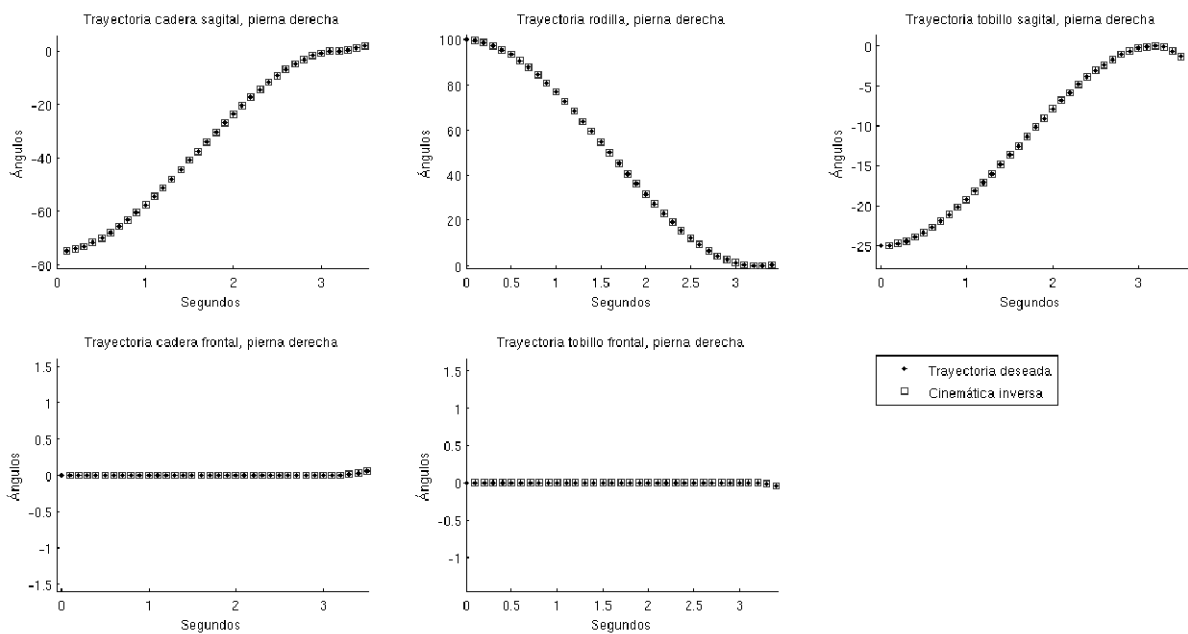


Figura 5.5: Seguimiento de trayectorias, pierna derecha (fase a).

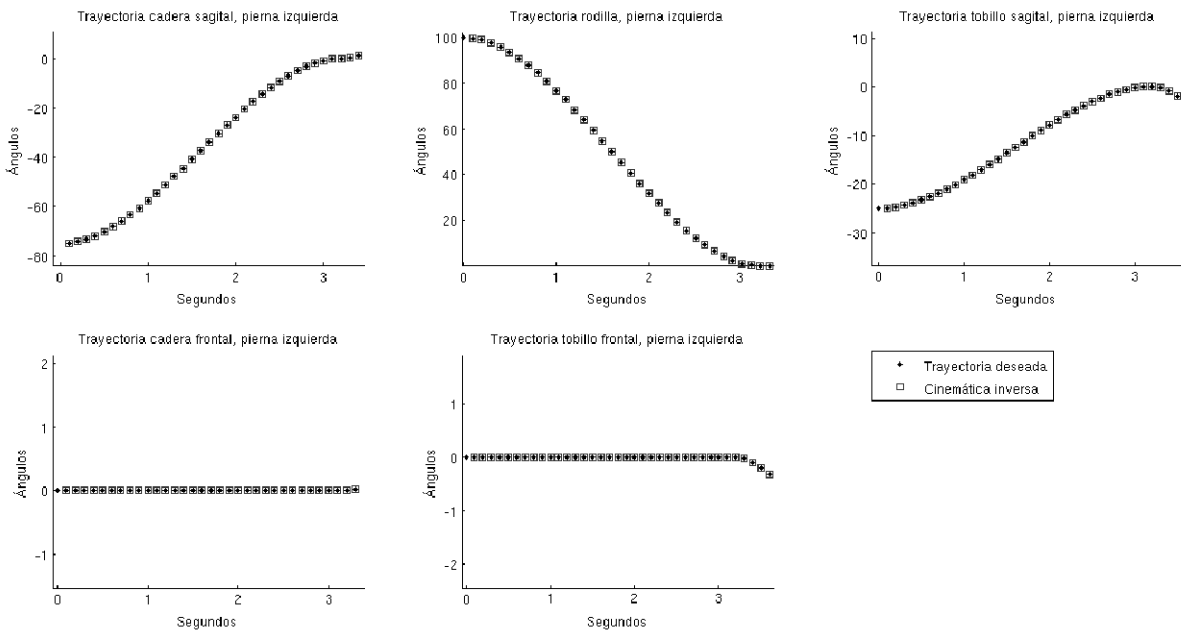


Figura 5.6: Seguimiento de trayectorias, pierna izquierda (fase a).

Se realizaron gráficas en 2 dimensiones para ver con más claridad el movimiento efectuado, en la Figura 5.7 se observa el movimiento realizado por la pierna derecha y en la Figura 5.8 el de la pierna izquierda.

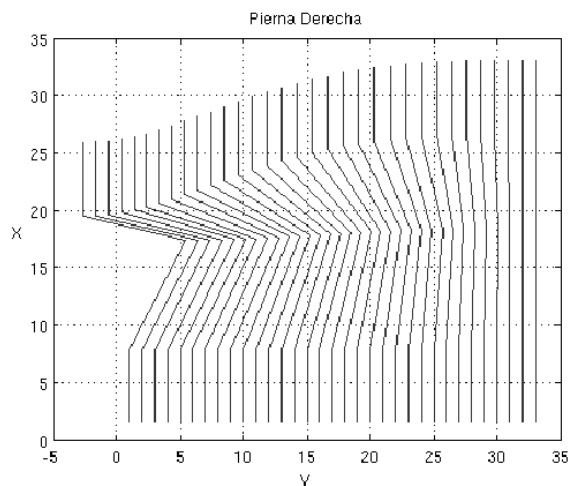


Figura 5.7: Movimiento en 2D de la pierna derecha (fase a).

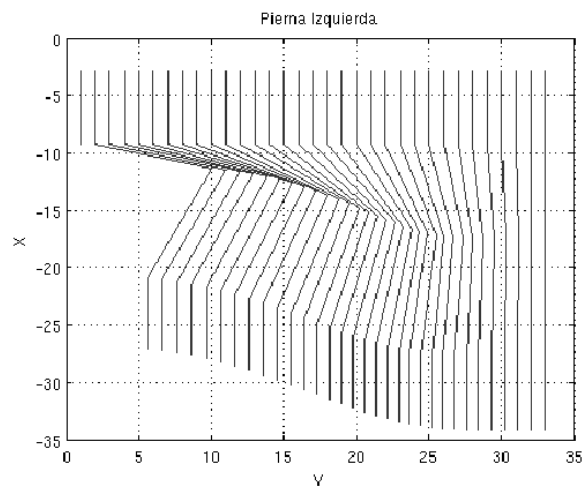


Figura 5.8: Movimiento en 2D de la pierna izquierda (fase a).

Para la secuencia b) que es donde se posiciona el robot para comenzar con el primer paso se tienen las siguientes capturas, la Figura 5.9 presenta el movimiento del robot para la pierna derecha y la Figura 5.10 para la pierna izquierda.

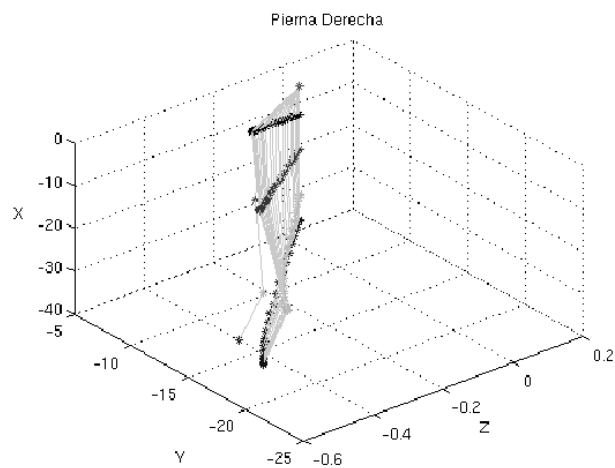


Figura 5.9: Representación del movimiento que realiza la pierna derecha (fase b)

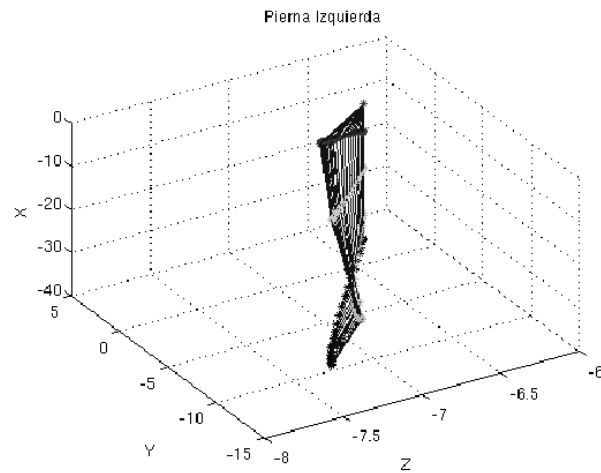


Figura 5.10: Representación del movimiento que realiza la pierna izquierda (fase b)

La Figura 5.11 muestra las trayectorias que sigue el robot en esta fase para la pierna derecha y la Figura 5.12 para la pierna izquierda.

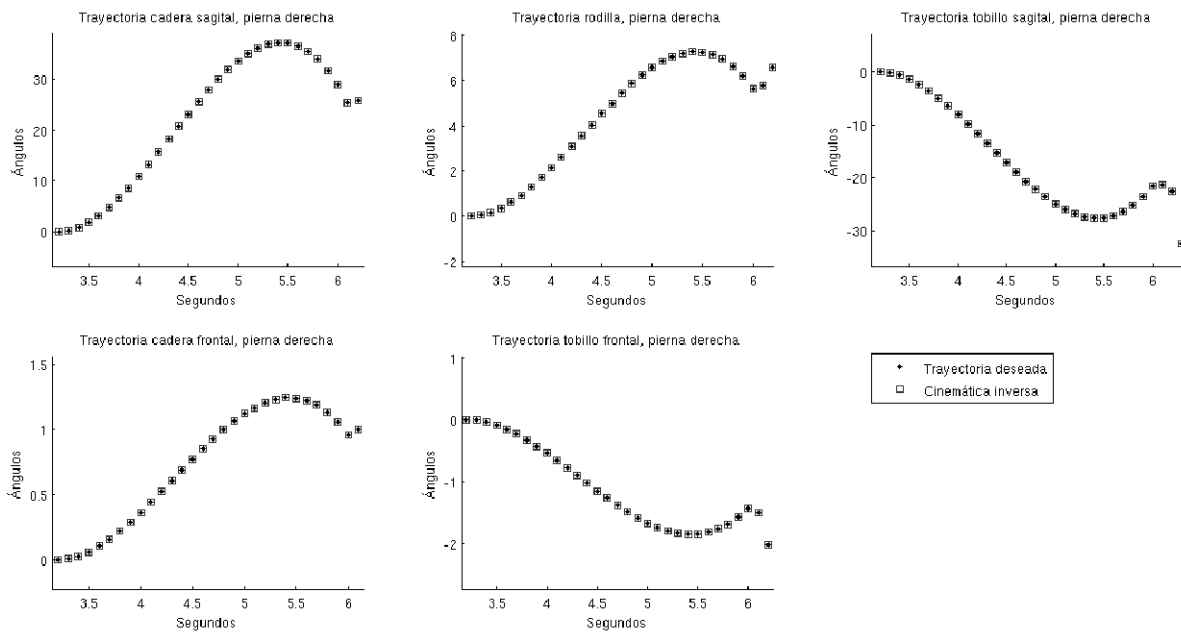


Figura 5.11: Seguimiento de trayectorias, pierna derecha (fase b).



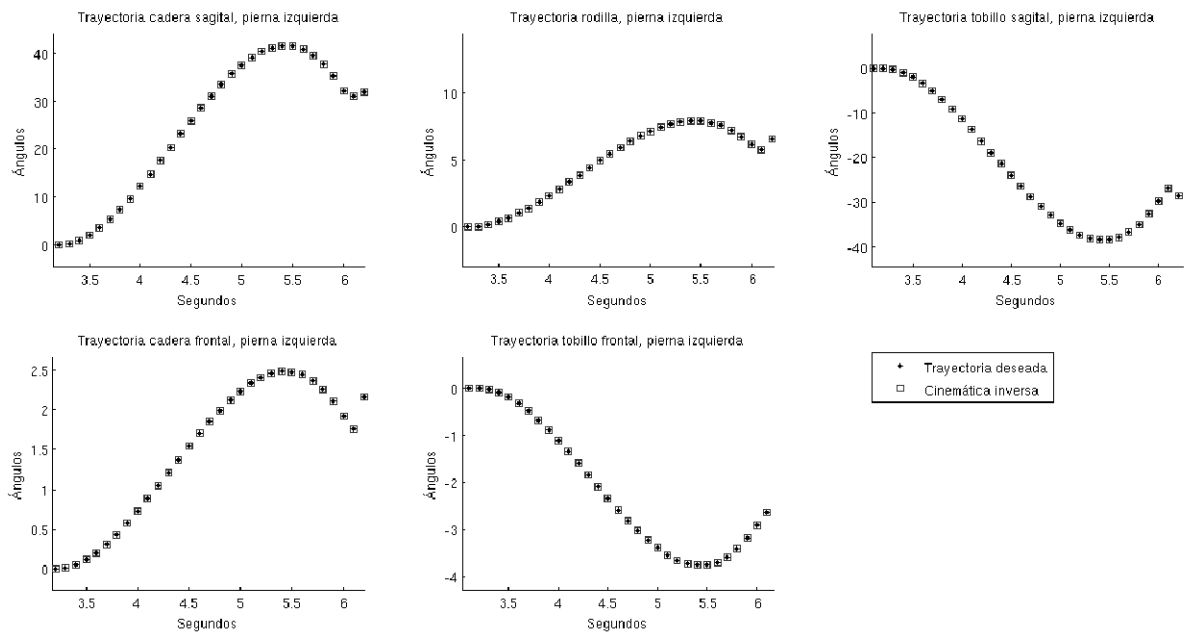


Figura 5.12: Seguimiento de trayectorias, pierna izquierda (fase b)

En el plano de 2 dimensiones para esta fase (b) se muestran las gráficas 5.13 para la pierna derecha y 5.14 para la pierna izquierda.

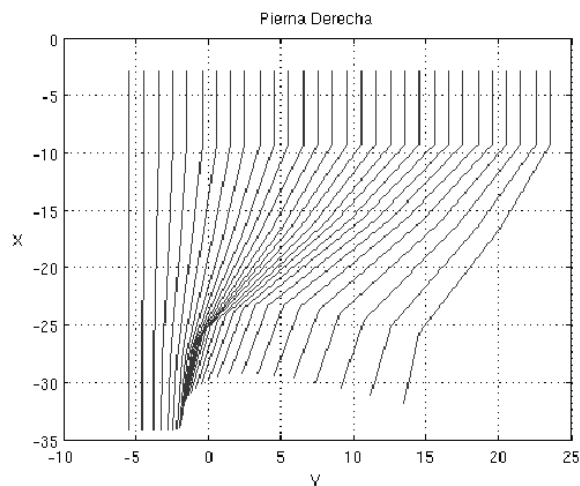


Figura 5.13: Movimiento en 2D de la pierna derecha (fase b).

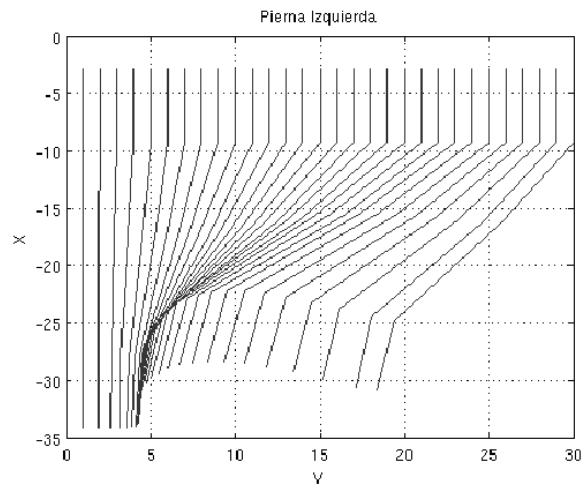


Figura 5.14: Movimiento en 2D de la pierna izquierda (fase b).

Para el ciclo de marcha (fase c-d), es decir, donde el robot da el paso (se toma solo un paso) se muestran en la Figura 5.15 el movimiento de la pierna derecha y en 5.16 el movimiento de la pierna izquierda.

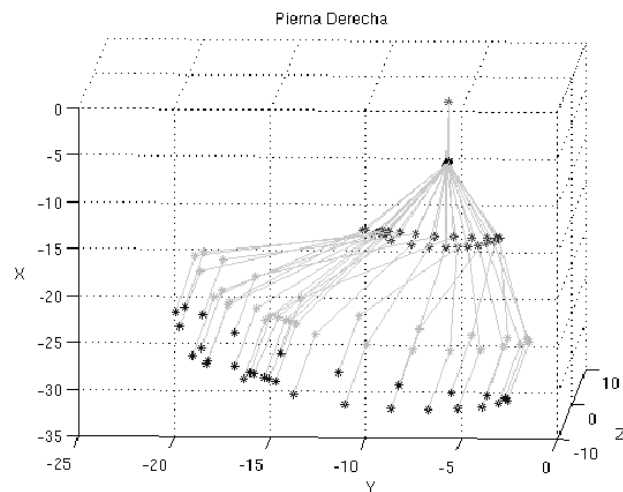


Figura 5.15: Representación del movimiento que realiza la pierna derecha (fase c-d).

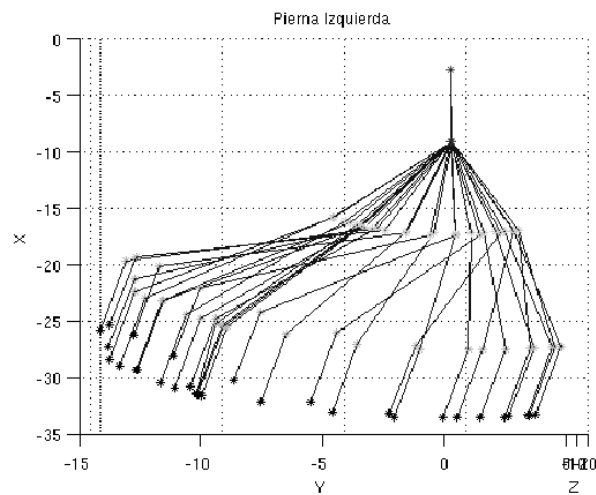


Figura 5.16: Representación del movimiento que realiza la pierna izquierda (fase c-d).

Sus trayectorias correspondientes en la Figura 5.17 (pierna derecha) y en la Figura 5.18 (pierna izquierda).

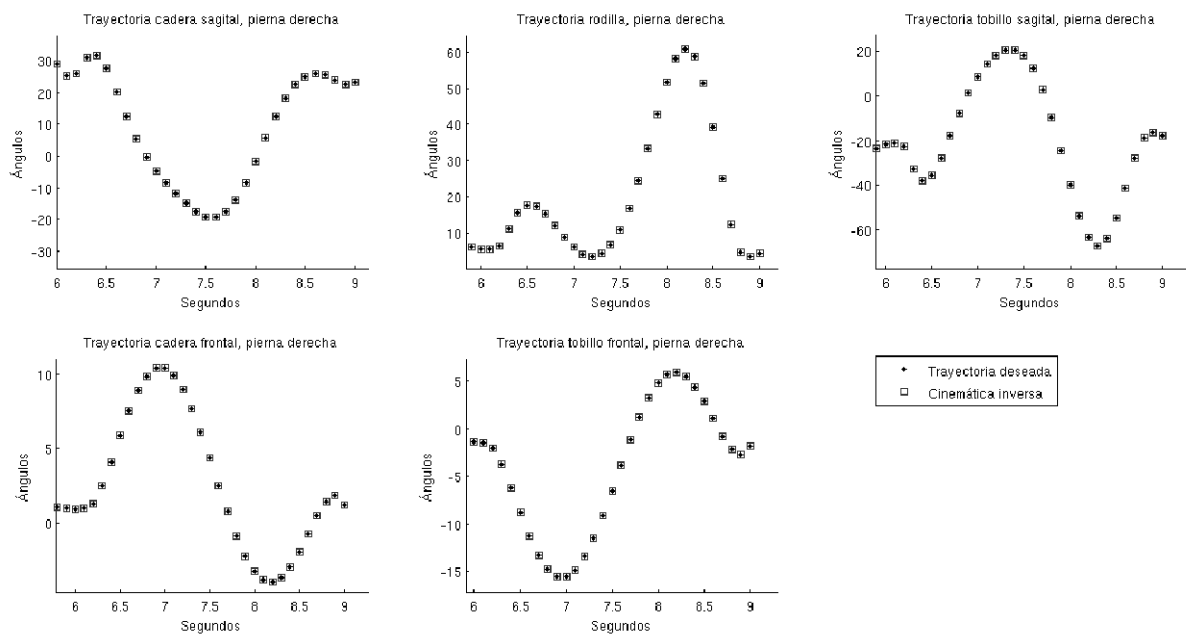


Figura 5.17: Seguimiento de trayectorias, pierna derecha (fase c-d).

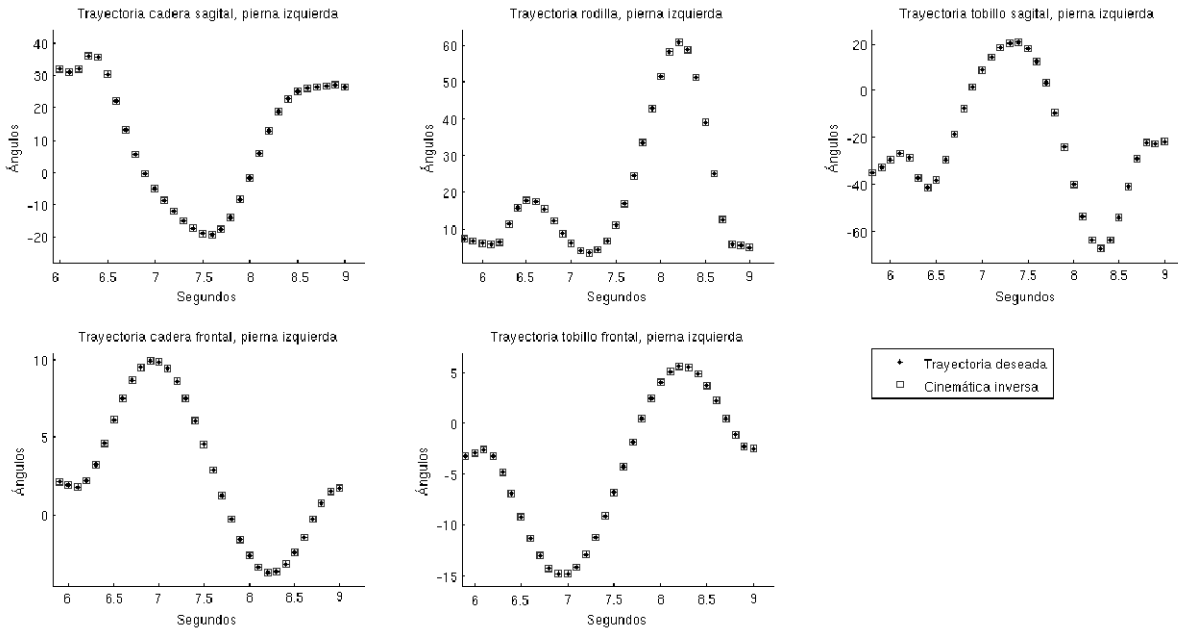


Figura 5.18: Seguimiento de trayectorias, pierna izquierda (fase c-d).

Las gráficas en 2 dimensiones para esta etapa de la locomoción para cada pierna se muestran en las Figuras 5.19 y 5.20.

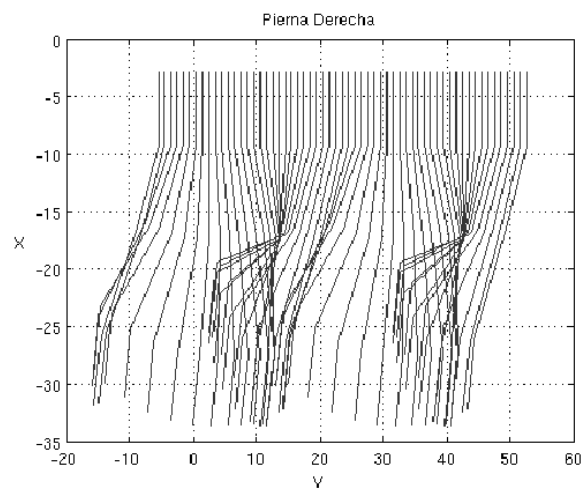


Figura 5.19: Movimiento en 2D de la pierna derecha (fase c-d).

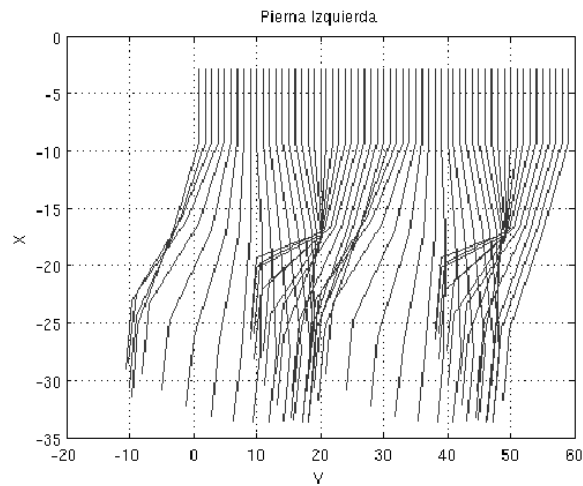


Figura 5.20: Movimiento en 2D de la pierna izquierda (fase c-d).

Con la ayuda de herramientas del programa Matlab se realizó una estructura en 3 dimensiones la cual presenta un movimiento de acuerdo a las trayectorias deseadas, en la Figura 5.21 se observa una secuencia de pasos que muestran la locomoción del robot. En esta figura se aprecia con más claridad el movimiento de la pierna derecha. Como el movimiento de la pierna izquierda es prácticamente el mismo se consideró una sola figura para mostrar el comportamiento.

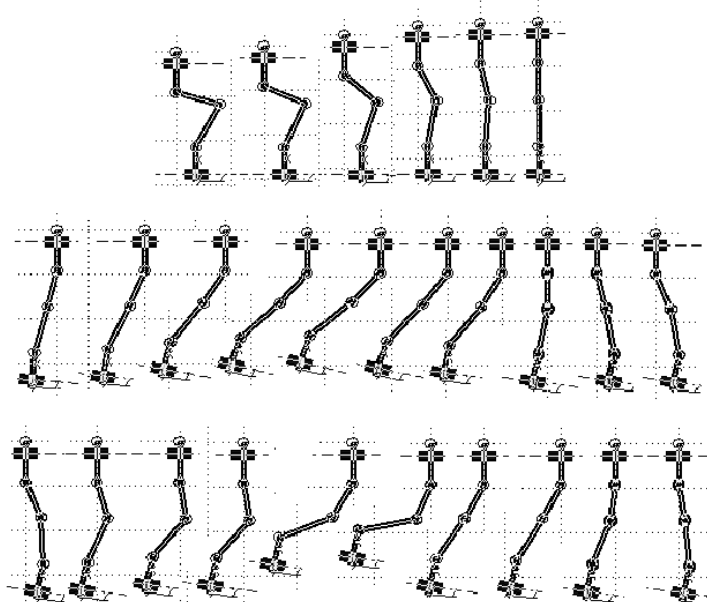


Figura 5.21: Secuencia de pasos de la locomoción de la pierna derecha de la estructura del robot.

Estos datos se obtuvieron con los programas presentados en el Apéndice B, se realizó también un programa en simulink donde se incluyen la cinemática directa e inversa a lazo abierto. La Figura 5.22 muestra una representación a bloques del lazo abierto y las gráficas presentadas en la Figura 5.23 muestran el seguimiento de trayectorias a lazo abierto.

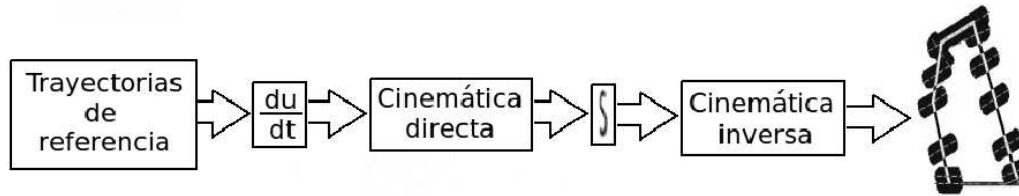


Figura 5.22: Esquema del lazo abierto

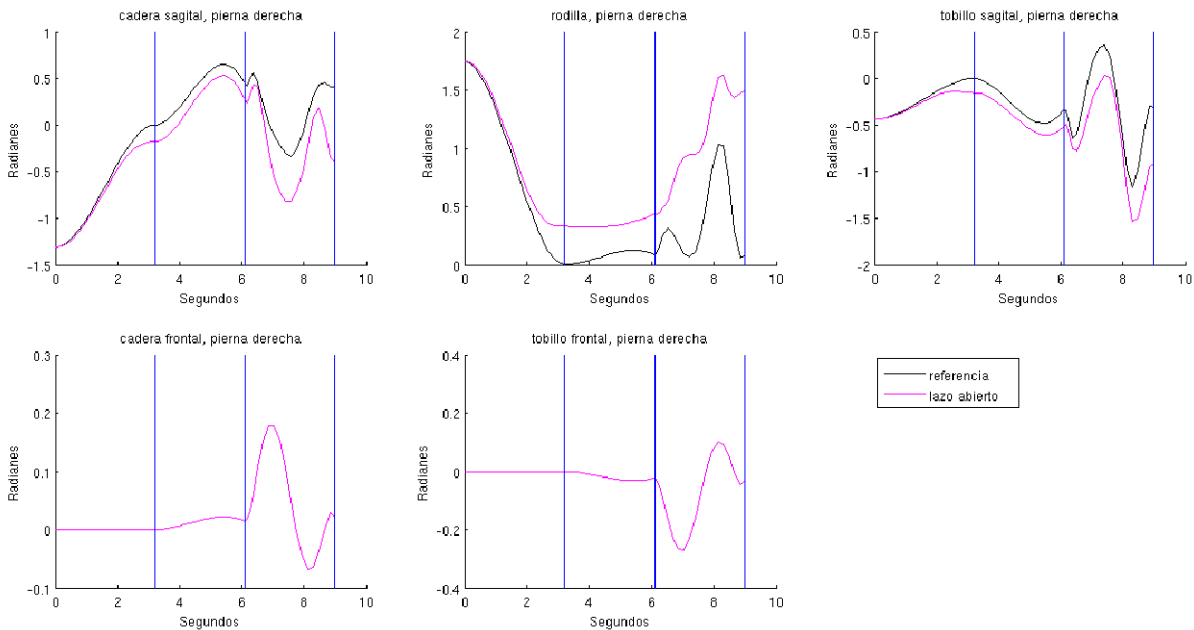


Figura 5.23: Seguimiento de trayectorias lazo abierto

Con la verificación de las trayectorias y de la cinemática inversa, se procede a la implementación del controlador. Para los resultados de simulación de la pierna derecha en lazo cerrado, se considera el diagrama 5.24 con 5 controladores PI+D difusos con las ganancias de la Tabla 5.1, cuyos valores se escogieron heurísticamente a partir de los ya asignados al control PID convencional. Estos valores de  $K$  presentaron un buen desempeño para todas las juntas, debido a esto, solo se modificó la variable  $L$  en base a la máxima flexión que cada articulación podrá tener.

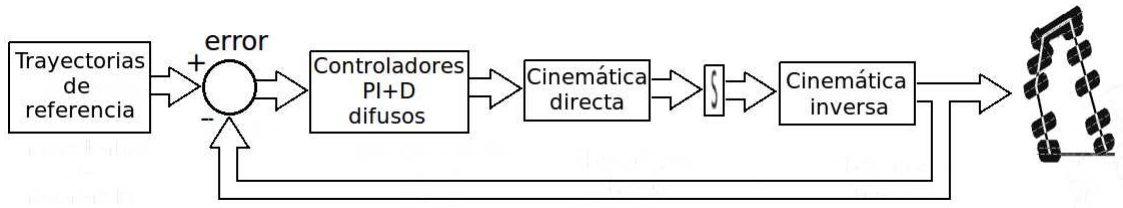


Figura 5.24: Esquema a bloques del controlador PI+D difuso lazo cerrado

Articulación	$K_p$	$K_i$	$K_d$	$L$
Tobillo frontal	5	0.7	0.3	60
Tobillo sagital	5	0.7	0.3	80
Rodilla	5	0.7	0.3	150
Cadera sagital	5	0.7	0.3	100
Cadera frontal	5	0.7	0.3	40

Tabla 5.1: Valores de  $K_p$ ,  $K_i$ ,  $K_d$  y  $L$  para el control difuso

Para realizar una comparación se consideran 5 PID clásicos, la sintonización del controlador PID se realizó mediante el método empírico de respuesta última de Ziegler-Nichols[21]. En este método, el sistema a lazo cerrado solo con control proporcional (es decir, las acciones de control integral y derivativa se encuentran desconectadas) se analiza para determinar la ganancia última  $K_u$  y el periodo de oscilación correspondiente (periodo final)  $P_u$  del proceso de respuesta. La ganancia última es la ganancia del controlador en el que el sistema a lazo cerrado es marginalmente estable, es decir, cuando la respuesta del sistema no presente variaciones notables. Los ajustes de Ziegler-Nichols se determinan a través de  $K_u$  y  $P_u$ , de la siguiente manera:  $K_p = 0.6K_u$ ,  $K_i = 0.5P_u$  y  $K_d = 0,125P_u$ .

Con  $K_u = 33.3$  y  $P_u = 0.24$  se obtienen las ganancias  $K_p = 20$ ,  $K_i = 0.12$  y  $K_d = 0.03$ . La elección final de las ganancias se realizó heurísticamente partiendo de los valores arrojados por el método anterior, por lo que se efectuaron repetitivas pruebas sobre los eslabones de la estructura del robot. Las variaciones en las ganancias  $K_p$ ,  $K_i$  y  $K_d$  fueron mínimas por lo que las 10 articulaciones presentan las mismas ganancias, siendo estas:  $K_p = 15$ ,  $K_i = 0,01$  y  $K_d = 0,5[1]$ .

En la Figura 5.25 se presentan las respuestas de seguimiento de las 5 articulaciones de la pierna derecha, como se puede notar el desempeño en estas gráficas de ambos conjuntos de

controladores (PID, PI+D difuso) es similar. Sin embargo, se realizó un acercamiento considerando solamente posición inicial - posición vertical. Este acercamiento se presenta en la Figura 5.26.

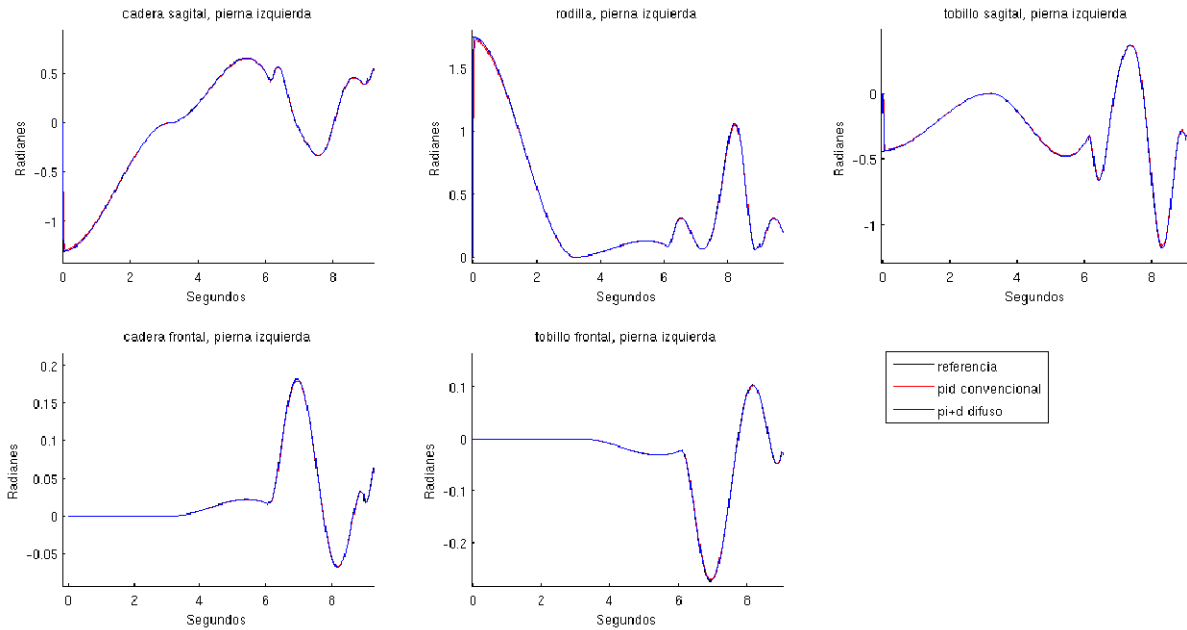


Figura 5.25: Comparación con la señal de referencia el PID convencional y el PI+D difuso

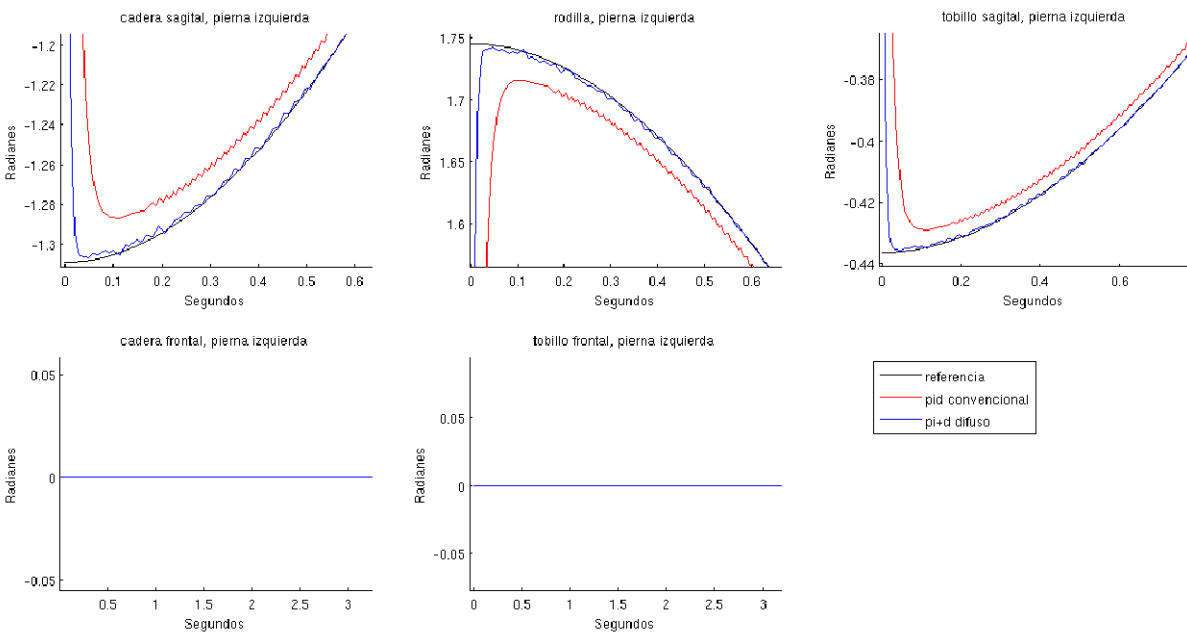


Figura 5.26: Acercamiento de la señal de referencia, el PID convencional y el PI+D difuso



En la Figura 5.27 se muestran los errores de seguimiento en la pierna derecha en cada una de las articulaciones.

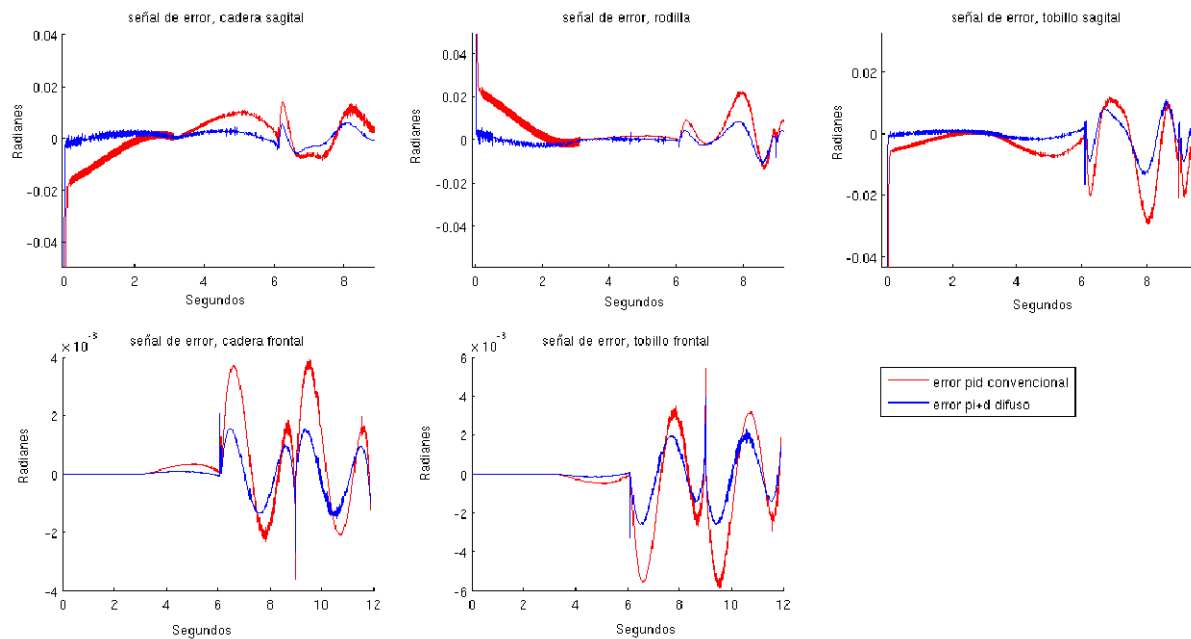


Figura 5.27: Comparación del error que presentan los dos controladores, pierna derecha

Debido a que la diferencia del error entre los controladores es mínima, se consideraron dos criterios de desempeño. Los criterios de desempeño permiten evaluar el comportamiento de los sistemas y son la base para optimizarlos respecto del comportamiento de su error. Para el primer caso (Figura 5.28) se utilizó el criterio integral de error cuadrático ( $ISE^1$ ), definido por el criterio:

$$ISE = \int_0^{\infty} e^2(t)dt \quad (5.1)$$

Para el segundo caso (Figura 5.29) se utilizó el criterio integral del tiempo multiplicado por el valor absoluto del error ( $ITAE^2$ ), que es insensible a los errores iniciales pero penaliza fuertemente los errores que permanecen a lo largo del tiempo. Esta definido por la ecuación:

$$ITAE = \int_0^{\infty} t|e(t)|dt \quad (5.2)$$

---

<sup>1</sup>Integral Squared Error

<sup>2</sup>Integral Time Absolute Error

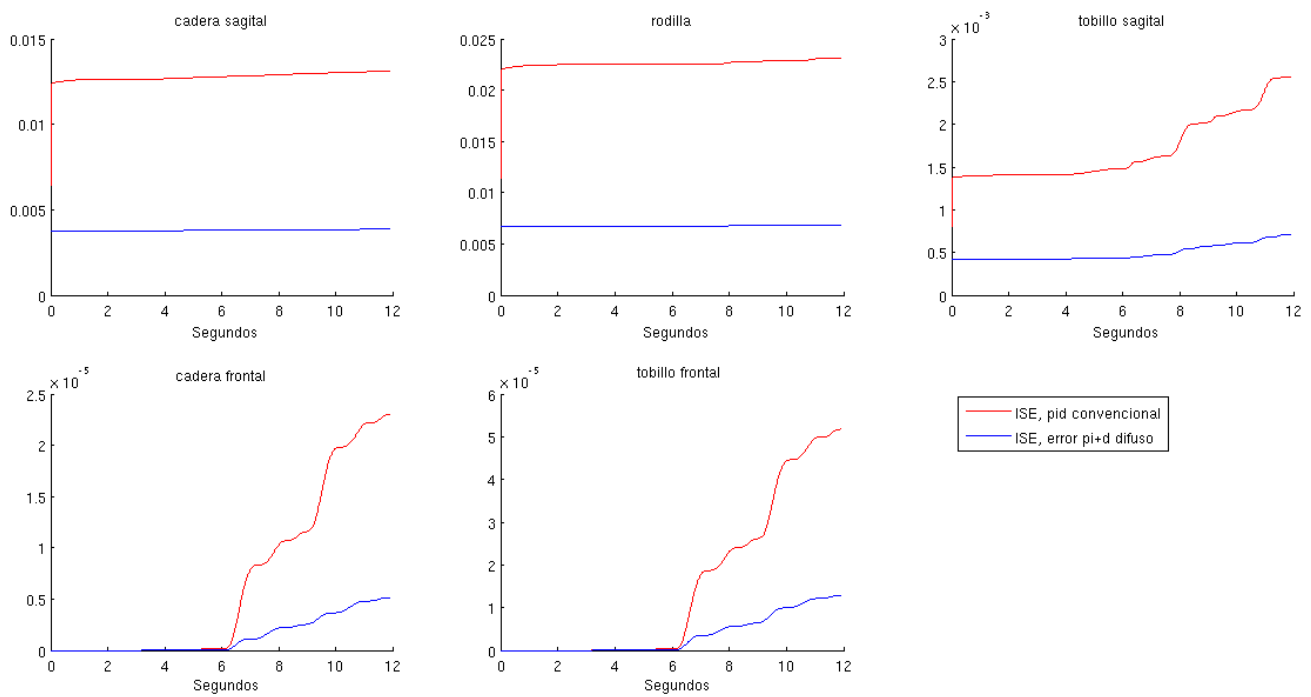


Figura 5.28: Criterio integral de error cuadrático ISE

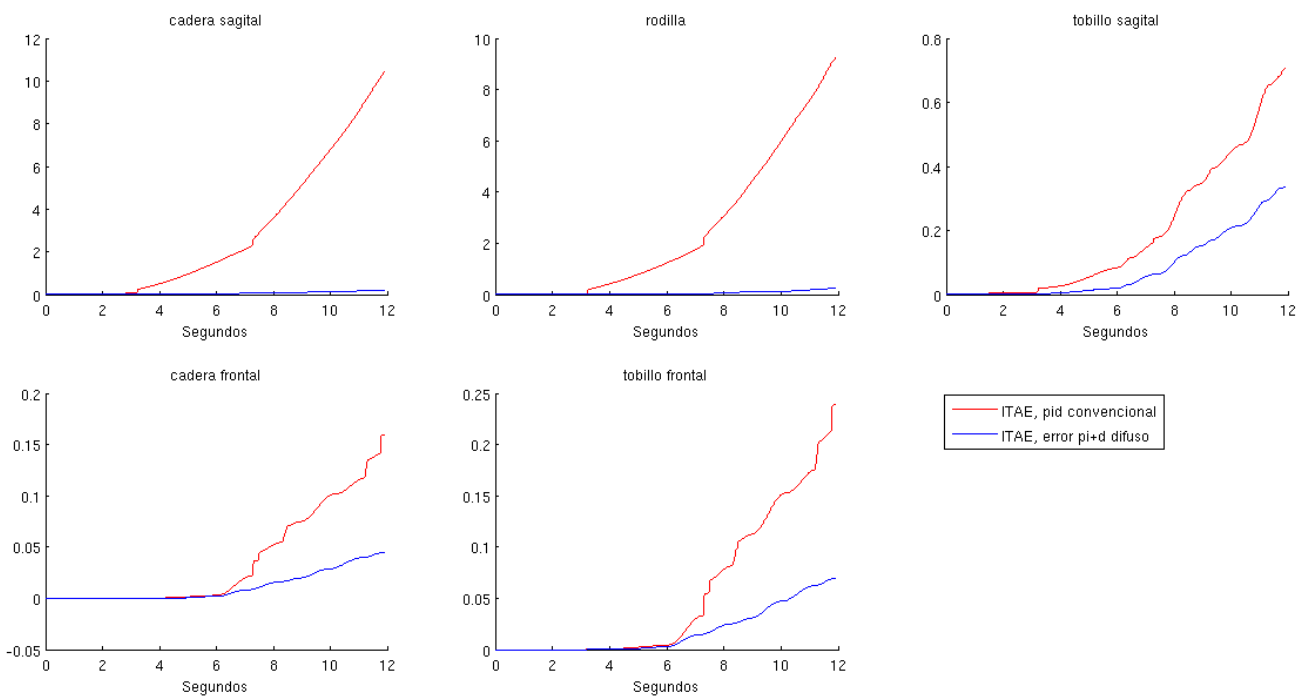


Figura 5.29: Criterio integral del tiempo multiplicado por el valor absoluto del error

Para la pierna izquierda se consideran las mismas ganancias tanto para el conjunto de PIDs como para los controladores PI+D difusos, debido a que los resultados de simulación del seguimiento de trayectorias para la pierna izquierda son similares a las de la pierna derecha solo se presentaron estos últimos.

Los valores de  $K_{uD}$  y  $K_{uPI}$  se modificaron para obtener una mejor respuesta en el controlador difuso quedando de la siguiente manera:

$$K_{uD} = 0,01 \quad (5.3)$$

$$K_{uPI} = 0,25 \quad (5.4)$$

## 5.2. Conclusiones

Como se pudo ver en las imágenes ambas piernas presentan un comportamiento parecido. Respecto al control, no se notó una gran diferencia entre los controladores, por lo que se realizó un análisis más profundo a la señal de error de cada controlador utilizando criterios de desempeño.

El criterio ISE es relativamente insensible a pequeños errores pero los grandes errores contribuyen fuertemente al valor de la integral, este criterio de comportamiento dará como resultado una respuesta con pequeños sobrepasamientos pero largos tiempos de estabilización, puesto que los pequeños errores a lo largo del tiempo contribuyen muy poco a la integral. Para el caso de este controlador el error en el control PID clásico es acumulable en cada paso del robot, sin embargo el incremento no es tan notable.

El criterio ITAE mostrará tiempos cortos de respuesta total y mayores sobrepasamientos que otros criterios. Lo que se puede apreciar en este criterio es que en el controlador PID convencional a diferencia del PI+D difuso, prevalecen los errores a lo largo del tiempo.

Con esto se puede ver que a pesar de no ser evidente la ventaja del controlador PI+D difuso sobre el PID convencional, se tiene un mejor desempeño con el transcurrir del tiempo.



# Conclusiones y trabajo futuro

Las conclusiones de este trabajo de tesis son las siguientes:

Dentro del estado del arte, se revisaron distintos tipos de algoritmos utilizados para el control de bípedos y algunos trabajos en donde ya se ha aplicado el control difuso. Se realizó un estudio suficiente para el entendimiento del caminado humano, este consistió en un análisis de las trayectorias de locomoción bípeda, se estudió acerca del equilibrio que se necesitan para mantener la estructura de pie específicamente el CoM y el ZMP, etapas consideradas en el ciclo de marcha y los planos en los que suele estudiarse la marcha humana.

Además, Se obtuvo un modelo matemático del robot bípedo tomando en cuenta la cinemática directa e inversa, así como las condiciones de equilibrio. Para la cinemática directa se utilizó el método de Denavit-Hartenberg y se obtuvo un modelo dinámico en función de coordenadas cartesianas del sistema, esto permitió realizar una simulación dinámica del robot. Con una introducción al control difuso se presentaron los beneficios y desventajas de utilizar este tipo de control, se optó entonces por comparar el controlador PI+D difuso con el PID clásico realizado en [1] y así tener una mejor visión de las ventajas que presenta el controlador difuso. Conforme a lo anterior se estudiaron y diseñaron leyes de control para el seguimiento de trayectorias. Para poder comprobar el desempeño del controlador difuso, se realizó una plataforma de simulación con la ayuda de Matlab mostrada en el apéndice B, como resultado de estas simulaciones el robot bípedo fue capaz de desarrollar una secuencia de marcha de forma autónoma, manteniendo el equilibrio durante su desplazamiento. A partir de los resultados obtenidos mediante las pruebas realizadas en estos programas se puede concluir que el controlador PI+D difuso tiene un menor índice de error en el seguimiento de señales y su respuesta es más rápida que el controlador PID convencional.

Los controladores PI+D difusos son una extensión natural de sus versiones convencionales, los cuales conservan las estructuras lineales de los controladores PID, con fórmulas de análisis sencillos y convencionales como resultado del diseño. Así, aunque el desarrollo de los controladores parece complejo, el resultado son formulas que pueden sustituir directamente los

controladores PID convencionales en cualquier sistema de control.

Como trabajo a futuro se planea implementar el controlador a un modelo físico experimental donde las perturbaciones reales (contacto con el suelo, comportamiento de los servomotores, tiempo de procesamiento de la tarjeta empleada, etc.) serán en teoría mejor manejadas por el control difuso.

# Bibliografía

- [1] “Integración y planificación de marcha en un robot bípedo de 10 grados de libertad (gdl)”. Daniel Alejandro Vite Téllez, Dr. Carlos Alberto Cruz Villar, Dr. Alejandro Rodríguez Ángeles, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. México, Agosto 2010.
- [2] “Robots móviles”. J. Ruiz del Solar, R. Salazar, Universidad de Chile. Chile, 2002 pp. 3
- [3] “Modelado y simulación del robot PASIBOT. Estudio de la rigidez y mejora en la prevención del vuelco lateral”, Víctor Espantoso Miranda, Jesús Meneses Alonso, Universidad Carlos III de Madrid. España, 2009, pp. 16-18.
- [4] “Mecatrónica bioinspirada de robots humanoides de tamaño natural”. Luis Maria Cabás Ormaechea, Carlos Balaguer Bernaldo de Quiroz, Universidad Carlos III de Madrid. España, Junio 2009, pp 35-40.
- [5] “Locomoción bípeda del robot humanoide Nao”. Samuel Fernández Iglesias, Josep M. Fuertes i Armengol, Universitat Politècnica de Catalunya. España, Noviembre 2009, pp. 15-30.
- [6] “Modelamiento y Control de un Robot Bípedo Usando un Sistema Operativo Implementado en un DSP”. Elvis O. Jara, Edison E. Alfaro, David R. Achancaray and Ricardo R. Rodríguez, Universidad Nacional de Ingeniería. Lima-Perú, Septiembre 2010.
- [7] “Locomotion control of a biped robot for stair-climbing by fuzzy stabilization tuning approach”. J. Lin, J. Chang, S.M. Lyu, S.W. Wang and Y.W. Lin, 2010 IEEE International Conference on Control Applications. Korea, September 2010, pp. 1590-1595.
- [8] “Algoritmos genéticos modificados para el balance de robots humanoides”, Edgar Eduardo Chavolla Canal, Daniel Zaldivar Navarro, Universidad de Guadalajara. México, Febrero 2010.

- [9] “Fuzzy Posture Control for Biped Walking Robot Based on Force Sensor for ZMP ”, K.C. Choi, M.C Lee, J. M Lee, Pusan National University. Korea, Marzo 2011.
- [10] “Neuro-Fuzzy ZMP Control of a Biped Robot”, Joao Paulo Ferreira, Manuel Crisóstomo, A. Paulo Coimbra, Superior Institute of Coimbra & University of Coimbra. Portugal, September 2006.
- [11] “Neuro-Fuzzy Algorithm for a Biped Robotic System”, Hataitep Wongsuwarn, Djitt Lao-wattana, World Academy of Science, Engineering and Technology. Artículo científico, 2006.
- [12] “Introduction to robotics: mechanics and control”, Craig J.J., Addison-Wesey Longman. 1989
- [13] “Development of a biped walking robot actuated by a closed-chain mechanism”, Hyeung-Sik Choi, Yong-Heon Park, volume 24. January 2006
- [14] “Notas de curso de robótica”, Emilio Gonzalez, CIEP-FI, 2011
- [15] “ Dynamic Balance Concept and the Maintenance of the Dynamic balance in Humanoid Robotics”, M. Vukobratović y B. Borovac, Intelligent Systems and Informatics. Serbia 2008, pp. 1-11.
- [16] “Springer Handbook of Robotics”, Springer S. Brunoy y K. Oussama, Verlag Berlin Heidelberg. 2008.
- [17] “Mechanics and Dynamical Systems with Mathematica”, Nicola Bellomo, Luigi Preziosi, Antonio Romano. Birkhauser, 2000.
- [18] “Planning Walking Patterns for a Biped Robot”, Quiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Arai, Noriho Koyachi y Kazuo Tanie. IEEE Transactions on robotics and automation, vol. 17 no. 3 pp. 280-289. 2001.
- [19] “Natural Gait Generation of Biped Robot based in Analysis of Human’s Gait”, Seung-Suk Ha, Jae-Hyoung Yu, Young-Joo Han, Her-Soo Hahn, IEEE International Conference on Smart Manufacturing Application, pp. 30-34. 2008.
- [20] “Introduction to Fuzzy Sets, fuzzy logic, and fuzzy control systems”, Guanrong Chen, Trung Tat Pham, CRC Press. 2000.
- [21] “Mechatronics An Integrated Approach”, Clarence W. de Silva. CRC Press, 2005.
- [22] “Montaje y Puesta a Punto de un Robot Humanoide”, Loreto Gutiérrez, Itziar Ribacoba, Víctor Etxebarria, Alfredo García, Universidad del País Vasco. España 2011.



- [23] “Diseño y Construcción de un Bípedo Experimental”, José Larriva, Edyson Guillermo, José Trelles, Oscar Vele. Madrid, Abril 2000.
- [24] “Curso Biomecánica”, Universidad de Santander. 2010.
- [25] “Biomecánica de la Marcha Humana”, Capítulo 3. Universidad de las Américas, Puebla 2011.
- [26] “Information power machine with senses and limbs. In Proc. CISM-IFTToMM Symp. on Theory and Practice of Robots and Manipulators”, Kato, I., Ohteru, S., Kobayashi, H., Shirai, K. Uchiyama. Italia 1973, pp. 12 - 24.
- [27] “Cinemática” Tema 4, parte 3. Universidad de Sevilla. España 2011.
- [28] “Equilibrio y Movilidad con personas mayores”, Debra J. Rose. Barcelona 2005.
- [29] “Sistemas Expertos y Lógica Difusa”, capítulo 2, Universidad de las Américas, Puebla 2011.
- [30] “Lógica difusa”, Capítulo 3, Universidad de las Américas, Puebla 2011.
- [31] “La lógica difusa”, Yuliana Corzo. Porlamar, Venezuela, Mayo 2001.
- [32] “Conceptos fundamentales de lógica difusa”, Capítulo 2, Consultado 2011.
- [33] “Control difuso una alternativa para aplicaciones de alta precisión”, Ferreyra A., Fuentes R., SOMI XIII Congreso de Instrumentación, pp 239-244, Octubre 1998, Ensenada B.C.N, México 2011.
- [34] “Estudio comparativo entre control PID y Difuso”, A. Ferreyra, R. Fuentes, UAM Azcapotzalco, México 2011.
- [35] “Robótica manipuladores y robots móviles”, Aníbal Ollero Baturone, marcombo boixareu editores. España 2001.
- [36] “Gait Analysis - Normal and Pathological Fuction”, Jacqueline Perry, SLACK. USA 1992.
- [37] “Gait Analysis An Introduction, Michael W. Whittle, 4a Edición. Butterworth Heinemann, USA 2007.
- [38] “Dynamics of Human Gait”, Chistopher L. Vaughan, Brian L. Davis, Jeremy C. O’Connor, 2a Edición. Kiboho Publishers, 1992.

- [39] “Lecciones básicas de biomecánica del aparato locomotor”, A. Viladot Voegeli y colaboradores, 1a Edición. Springer 2000.
- [40] “Bioinformática. Simulación, vida artificial e inteligencia artificial”, Rafael Lahoz Beltrá, 1a Edición. Díaz de Santos 2004.

# Apéndice A

## Ecuaciones

Aquí se presentan los cálculos para el desarrollo de las leyes de control PI+D difusos para la región 1 (Figura A.1) se tienen 4 reglas:

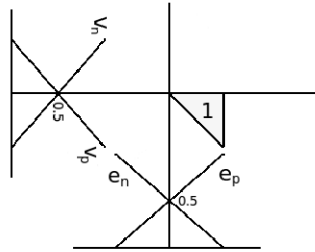


Figura A.1: Función de pertenencia, región 1

- Regla 1:  $e_n < 0,5$       $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n > 0,5$
- Regla 2:  $e_n < 0,5$       $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p < 0,5$
- Regla 3:  $e_p > 0,5$       $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n > 0,5$
- Regla 4:  $e_p > 0,5$       $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p < 0,5$

con estos datos obtenemos la la ley de control de esta región (ecuación A.1),

$$\begin{aligned}
 \Delta u_{PI} &= \frac{e_n o_n + e_n o_z + v_n o_z + v_p o_p}{2e_n + v_n + v_p} \\
 &= \frac{-L e_n + L v_p}{2e_n + v_n + v_p} = \frac{-L \frac{L-k_i e(nt)}{2L} + L \frac{L+k_p v(nt)}{2L}}{2 \frac{L-k_i e(nt)}{2L} + \frac{L-k_p v(nt)}{2L} + \frac{L+k_p v(nt)}{2L}} \\
 &= \frac{-L+k_i e(nt)+L+K_p v(nt)}{2(L-k_i e(nt))+L-K_p v(nt)+L+K_p v(nt)} = \frac{k_i e(nt)+K_p v(nt)}{2L-2k_i e(nt)+2L} \\
 \Delta u_{PI} &= \frac{L(k_i e(nt) + K_p v(nt))}{2(2L - k_i e(nt))} \tag{A.1}
 \end{aligned}$$

este procedimiento se realiza para obtener la ley de control de cada región. Para la región 2 se tienen las siguientes reglas:

- Regla 1:  $e_n < 0,5$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n < 0,5$
- Regla 2:  $e_n < 0,5$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p > 0,5$
- Regla 3:  $e_p > 0,5$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n < 0,5$
- Regla 4:  $e_p > 0,5$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p > 0,5$

$$\Delta u_{PI} = \frac{L(k_i e(nt) + K_p v(nt))}{2(2L - k_i e(nt))} \quad (\text{A.2})$$

La ley de control para la región 2 es la misma que la región 1 (ecuación A.1). La región 3 se rige por las siguientes reglas, teniendo como ley de control la ecuación A.3:

- Regla 1:  $e_n < 0,5$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n < 0,5$
- Regla 2:  $e_n < 0,5$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p > 0,5$
- Regla 3:  $e_p > 0,5$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n < 0,5$
- Regla 4:  $e_p > 0,5$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p > 0,5$

donde la ley de control se obtiene de la siguiente manera:

$$\begin{aligned} \Delta u_{PI} &= \frac{v_n o_n + e_n o_z + v_n o_z + e_p o_p}{2v_n + e_n + e_p} \\ &= \frac{-L v_n + L e_p}{2v_n + e_n + e_p} = \frac{-L \frac{L - k_p v(nt)}{2L} + L \frac{L + k_i e(nt)}{2L}}{2 \frac{L - k_p v(nt)}{2L} + \frac{L - k_i e(nt)}{2L} + \frac{L + k_i e(nt)}{2L}} \\ &= \frac{-L + k_p v(nt) + L + K_i e(nt)}{2(L - k_p v(nt)) + L - K_i e(nt) + L + K_i e(nt)} = \frac{2L(k_p v(nt) + K_i e(nt))}{(4L - 2k_p v(nt))} \\ \Delta u_{PI} &= \frac{L(k_p v(nt) + K_i e(nt))}{2(2L - k_p v(nt))} \end{aligned} \quad (\text{A.3})$$

La región 4 tiene la misma ley de control que la región 3 y presenta estas reglas:

- Regla 1:  $e_n > 0,5$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n < 0,5$
- Regla 2:  $e_n > 0,5$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p > 0,5$
- Regla 3:  $e_p < 0,5$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n < 0,5$
- Regla 4:  $e_p < 0,5$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p > 0,5$

$$\Delta u_{PI} = \frac{L(k_p v(nt) + K_i e(nt))}{2(2L - k_p v(nt))} \quad (\text{A.4})$$

En la región 5 se obtiene la ley de control con las reglas que se muestran a continuación:

- Regla 1:  $e_n > 0,5$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n < 0,5$
- Regla 2:  $e_n > 0,5$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p > 0,5$

- Regla 3:  $e_p < 0,5$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n < 0,5$
- Regla 4:  $e_p < 0,5$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p > 0,5$

$$\begin{aligned}
\Delta u_{PI} &= \frac{v_n o_n + v_p o_z + e_p o_z + e_p o_p}{2e_p + v_n + v_p} \\
&= \frac{-L v_n + L e_p}{2e_p + v_n + v_p} = \frac{-L \frac{L - k_p v(nt)}{2L} + L \frac{L + k_i e(nt)}{2L}}{2 \frac{L + k_i e(nt)}{2L} + \frac{L - k_p v(nt)}{2L} + \frac{L + k_p v(nt)}{2L}} \\
&= \frac{2L(K_p v(nt) + k_i e(nt))}{2(4L + 2k_i e(nt))} \\
\Delta u_{PI} &= \frac{L(k_i e(nt) + K_p v(nt))}{2(2L + k_i e(nt))} \tag{A.5}
\end{aligned}$$

Para la ley de control de la región 6 se tiene la ecuación A.5 en base a las siguientes reglas

- Regla 1:  $e_n > 0,5$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n > 0,5$
- Regla 2:  $e_n > 0,5$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p < 0,5$
- Regla 3:  $e_p < 0,5$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n > 0,5$
- Regla 4:  $e_p < 0,5$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p < 0,5$

$$\Delta u_{PI} = \frac{L(k_i e(nt) + K_p v(nt))}{2(2L + k_i e(nt))} \tag{A.6}$$

Para la región 7 se tienen las siguientes reglas

- Regla 1:  $e_n > 0,5$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n > 0,5$
- Regla 2:  $e_n > 0,5$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p < 0,5$
- Regla 3:  $e_p < 0,5$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n > 0,5$
- Regla 4:  $e_p < 0,5$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p < 0,5$

$$\begin{aligned}
\Delta u_{PI} &= \frac{e_n o_n + v_p o_z + e_p o_z + v_p o_p}{2v_p + e_n + e_p} \\
&= \frac{-L e_n + L v_p}{2v_p + e_n + e_p} = \frac{-L \frac{L - k_i e(nt)}{2L} + L \frac{L + k_p v(nt)}{2L}}{2 \frac{L + k_p v(nt)}{2L} + \frac{L + k_i e(nt)}{2L} + \frac{L - k_i e(nt)}{2L}} \\
\Delta u_{PI} &= \frac{L(k_i e(nt) + K_p v(nt))}{2(2L + k_i e(nt))} \tag{A.7}
\end{aligned}$$

La región 8 tiene por regla la misma que la región anterior

- Regla 1:  $e_n < 0,5$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n > 0,5$
- Regla 2:  $e_n < 0,5$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p < 0,5$
- Regla 3:  $e_p > 0,5$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n > 0,5$
- Regla 4:  $e_p > 0,5$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p < 0,5$

$$\Delta u_{PI} = \frac{L(K_i e(nt) + K_p v(nt))}{2(2L + K_i e(nt))} \quad (\text{A.8})$$

Las siguientes reglas corresponden a la región 9

- Regla 1:  $e_n = 0$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n > 0,5$
- Regla 2:  $e_n = 0$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p < 0,5$
- Regla 3:  $e_p = 1$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n > 0,5$
- Regla 4:  $e_p = 1$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p < 0,5$

donde su ley de control es:

$$\begin{aligned} \Delta u_{PI} &= \frac{e_n o_n + e_n o_z + v_n o_z + v_p o_p}{2e_n + v_n + v_p} \\ &= \frac{-L e_n + L v_p}{2e_n + v_n + v_p} = \frac{L \frac{L + K_p v(nt)}{2L}}{\frac{L - K_p v(nt)}{2L} + \frac{L + K_p v(nt)}{2L}} \\ \Delta u_{PI} &= \frac{1}{2}(L + k_p v(nt)) \end{aligned} \quad (\text{A.9})$$

Esta ley de control es la misma que para la región 10, obtenida por las reglas:

- Regla 1:  $e_n = 0$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n < 0,5$
- Regla 2:  $e_n = 0$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p > 0,5$
- Regla 3:  $e_p = 1$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n < 0,5$
- Regla 4:  $e_p = 1$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p > 0,5$

$$\Delta u_{PI} = \frac{1}{2}(L + k_p v(nt)) \quad (\text{A.10})$$

La región 11 tiene las reglas siguientes:

- Regla 1:  $e_n < 0,5$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n = 0$
- Regla 2:  $e_n < 0,5$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p = 1$
- Regla 3:  $e_p > 0,5$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n = 0$
- Regla 4:  $e_p > 0,5$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p = 1$

donde su ley de control es:

$$\begin{aligned} \Delta u_{PI} &= \frac{v_n o_n + e_n o_z + v_n o_z + e_p o_p}{2v_n + e_n + e_p} \\ &= \frac{L \frac{L + K_i e(nt)}{2L}}{\frac{L - k_i e(nt)}{2L} + \frac{L + K_i e(nt)}{2L}} \\ \Delta u_{PI} &= \frac{1}{2}(L + k_i e(nt)) \end{aligned} \quad (\text{A.11})$$

Región 12, con las siguientes reglas se obtiene la misma ley de control que la ecuación A.11

- Regla 1:  $e_n > 0,5$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n = 0$
- Regla 2:  $e_n > 0,5$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p = 1$
- Regla 3:  $e_p < 0,5$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n = 0$
- Regla 4:  $e_p < 0,5$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p = 1$

$$\Delta u_{PI} = \frac{1}{2}(L + k_i e(nt)) \quad (\text{A.12})$$

La ley de control para la región 13 se obtiene con las reglas:

- Regla 1:  $e_n = 1$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n > 0,5$
- Regla 2:  $e_n = 1$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p < 0,5$
- Regla 3:  $e_p = 0$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n > 0,5$
- Regla 4:  $e_p = 0$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p < 0,5$

$$\begin{aligned} \Delta u_{PI} &= \frac{v_n o_n + v_p o_z + e_p o_z + e_p o_p}{2e_p + v_n + v_p} \\ &= \frac{-L \frac{L - K_p v(nt)}{2L}}{\frac{L - k_p v(nt)}{2L} + \frac{L + K_p v(nt)}{2L}} \\ \Delta u_{PI} &= \frac{1}{2}(-L + K_p v(nt)) \end{aligned} \quad (\text{A.13})$$

La región 14 presenta la misma ley que la ecuación A.13

- Regla 1:  $e_n = 1$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n > 0,5$
- Regla 2:  $e_n = 1$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p < 0,5$
- Regla 3:  $e_p = 0$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n < 0,5$
- Regla 4:  $e_p = 0$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p < 0,5$

$$\Delta u_{PI} = \frac{1}{2}(-L + K_p v(nt)) \quad (\text{A.14})$$

La ley de control de la región 15 se obtiene con las siguientes reglas:

- Regla 1:  $e_n > 0,5$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n = 1$
- Regla 2:  $e_n > 0,5$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p = 0$
- Regla 3:  $e_p < 0,5$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n = 1$
- Regla 4:  $e_p < 0,5$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p = 0$

$$\begin{aligned}
\Delta u_{PI} &= \frac{e_n o_n + v_p o_z + e_p o_z + v_p o_p}{2v_p + e_n + e_p} \\
&= \frac{-L \frac{L - K_i e(nt)}{2L}}{\frac{L - k_i e(nt)}{2L} + \frac{L + K_i e(nt)}{2L}} \\
\Delta u_{PI} &= \frac{1}{2}(-L + k_i e(nt))
\end{aligned} \tag{A.15}$$

La región 16 tiene la misma ley que la región anterior y sus reglas son:

- Regla 1:  $e_n < 0,5$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n = 1$
- Regla 2:  $e_n < 0,5$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p = 0$
- Regla 3:  $e_p > 0,5$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n = 1$
- Regla 4:  $e_p > 0,5$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p = 0$

$$\Delta u_{PI} = \frac{1}{2}(-L + k_i e(nt)) \tag{A.16}$$

La región 17 basada en las reglas:

- Regla 1:  $e_n = 0$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n = 0$
- Regla 2:  $e_n = 0$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p = 1$
- Regla 3:  $e_p = 1$      $\min \{e_p, v_n\} = v_n \rightarrow o_z$   
 $v_n = 0$
- Regla 4:  $e_p = 1$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p = 1$

presenta la ley de control:

$$\begin{aligned}
\Delta u_{PI} &= \frac{e_n o_n + e_n o_z + v_n o_z + e_p o_p}{2e_n + v_n + e_p} \\
&= \frac{L e_p}{e_p} \\
\Delta u_{PI} &= L
\end{aligned} \tag{A.17}$$

Con las siguientes reglas se obtiene la ley de control de la región 18

- Regla 1:  $e_n = 1$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n = 0$
- Regla 2:  $e_n = 1$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p = 1$
- Regla 3:  $e_p = 0$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n = 0$
- Regla 4:  $e_p = 0$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p = 1$

$$\begin{aligned}
\Delta u_{PI} &= \frac{v_n o_n + e_n o_z + e_p o_z + e_p o_p}{2e_p + e_n + v_n} \\
\Delta u_{PI} &= 0
\end{aligned} \tag{A.18}$$

La ley de la región 19 (ecuación A.19) se consigue con las reglas:

- Regla 1:  $e_n = 1$      $\min \{e_n, v_n\} = v_n \rightarrow o_n$   
 $v_n = 1$



- Regla 2:  $e_n = 1$      $\min \{e_n, v_p\} = v_p \rightarrow o_z$   
 $v_p = 0$
- Regla 3:  $e_p = 0$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n = 1$
- Regla 4:  $e_p = 0$      $\min \{e_p, v_p\} = e_p \rightarrow o_p$   
 $v_p = 0$

$$\begin{aligned}\Delta u_{PI} &= \frac{v_n o_n + v_p o_z + e_p o_z + e_p o_p}{2e_p + v_n + v_p} \\ &= \frac{-Lv_p}{v_p} \\ \Delta u_{PI} &= -L\end{aligned}\tag{A.19}$$

La región 20 tiene como ley la misma que la región 18

- Regla 1:  $e_n = 0$      $\min \{e_n, v_n\} = e_n \rightarrow o_n$   
 $v_n = 1$
- Regla 2:  $e_n = 0$      $\min \{e_n, v_p\} = e_n \rightarrow o_z$   
 $v_p = 0$
- Regla 3:  $e_p = 1$      $\min \{e_p, v_n\} = e_p \rightarrow o_z$   
 $v_n = 1$
- Regla 4:  $e_p = 1$      $\min \{e_p, v_p\} = v_p \rightarrow o_p$   
 $v_p = 0$

$$\Delta u_{PI} = 0\tag{A.20}$$

Para el control D Difuso, el procedimiento es parecido al anterior. También se tienen las regiones de combinaciones de entrada mostradas en la Figura 4.6, y dividido por regiones se tienen las siguientes reglas y leyes de control correspondientes.

Región 1

- Regla 1:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p < 0,5$
- Regla 2:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n > 0,5$
- Regla 3:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p < 0,5$
- Regla 4:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n > 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L - K|y_d(nT)|)}\tag{A.21}$$

Región 2

- Regla 1:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p > 0,5$
- Regla 2:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n < 0,5$
- Regla 3:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p > 0,5$
- Regla 4:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n < 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L - K|y_d(nT)|)} \quad (\text{A.22})$$

## Región 3

- Regla 1:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p > 0,5$
- Regla 2:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n < 0,5$
- Regla 3:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p > 0,5$
- Regla 4:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n < 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L - K|y_d(nT)|)} \quad (\text{A.23})$$

## Región 4

- Regla 1:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p > 0,5$
- Regla 2:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n < 0,5$
- Regla 3:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p > 0,5$
- Regla 4:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n < 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L + K|y_d(nT)|)} \quad (\text{A.24})$$

## Región 5

- Regla 1:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p > 0,5$
- Regla 2:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n < 0,5$
- Regla 3:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p > 0,5$
- Regla 4:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n < 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L + K|y_d(nT)|)} \quad (\text{A.25})$$

## Región 6

- Regla 1:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p < 0,5$
- Regla 2:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n > 0,5$
- Regla 3:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p < 0,5$
- Regla 4:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n > 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L + K|y_d(nT)|)} \quad (\text{A.26})$$

### Región 7

- Regla 1:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p < 0,5$
- Regla 2:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n > 0,5$
- Regla 3:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p < 0,5$
- Regla 4:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n > 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L + K|y_d(nT)|)} \quad (\text{A.27})$$

### Región 8

- Regla 1:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p < 0,5$
- Regla 2:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n > 0,5$
- Regla 3:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p < 0,5$
- Regla 4:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n > 0,5$

$$\Delta u_D(nT) = \frac{L[Ky_d(nT) - K_d|\Delta y(nT)]}{2(2L - K|y_d(nT)|)} \quad (\text{A.28})$$

### Región 9

- Regla 1:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p < 0,5$
- Regla 2:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n > 0,5$
- Regla 3:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p < 0,5$
- Regla 4:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n > 0,5$

$$\Delta u_D(nT) = \frac{1}{2}[L - K_d\Delta y(nT)] \quad (\text{A.29})$$

### Región 10

- Regla 1:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p > 0,5$
- Regla 2:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n < 0,5$
- Regla 3:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p > 0,5$
- Regla 4:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n < 0,5$

$$\Delta u_D(nT) = \frac{1}{2}[L - K_d \Delta y(nT)] \quad (\text{A.30})$$

## Región 11

- Regla 1:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p = 1$
- Regla 2:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n = 0$
- Regla 3:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p = 1$
- Regla 4:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n = 0$

$$\Delta u_D(nT) = \frac{1}{2}[-L + K y_d(nT)] \quad (\text{A.31})$$

## Región 12

- Regla 1:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p = 1$
- Regla 2:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n = 0$
- Regla 3:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p = 1$
- Regla 4:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n = 0$

$$\Delta u_D(nT) = \frac{1}{2}[-L + K y_d(nT)] \quad (\text{A.32})$$

## Región 13

- Regla 1:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p > 0,5$
- Regla 2:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n < 0,5$
- Regla 3:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p > 0,5$
- Regla 4:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n < 0,5$

$$\Delta u_D(nT) = \frac{1}{2}[-L - K_d \Delta y(nT)] \quad (\text{A.33})$$

## Región 14

- Regla 1:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p < 0,5$
- Regla 2:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n > 0,5$
- Regla 3:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p < 0,5$
- Regla 4:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n > 0,5$

$$\Delta u_D(nT) = \frac{1}{2}[-L - K_d \Delta y(nT)] \quad (\text{A.34})$$

### Región 15

- Regla 1:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p = 0$
- Regla 2:  $y_{dp} < 0,5$      $\min \{y_{dp}, \Delta y_n\} = y_p \rightarrow o_p$   
 $\Delta y_n = 1$
- Regla 3:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p = 0$
- Regla 4:  $y_{dn} > 0,5$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n = 1$

$$\Delta u_D(nT) = \frac{1}{2}[L + K y_d(nT)] \quad (\text{A.35})$$

### Región 16

- Regla 1:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p = 0$
- Regla 2:  $y_{dp} > 0,5$      $\min \{y_{dp}, \Delta y_n\} = y_p \rightarrow o_p$   
 $\Delta y_n = 1$
- Regla 3:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p = 0$
- Regla 4:  $y_{dn} < 0,5$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n = 1$

$$\Delta u_D(nT) = \frac{1}{2}[L + K y_d(nT)] \quad (\text{A.36})$$

### Región 17

- Regla 1:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p = 1$
- Regla 2:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n = 0$
- Regla 3:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p = 1$
- Regla 4:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n = 0$

$$\Delta u_D(nT) = 0 \quad (\text{A.37})$$

### Región 18

- Regla 1:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p = 1$
- Regla 2:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n = 0$
- Regla 3:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p = 1$
- Regla 4:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n = 0$

$$\Delta u_D(nT) = L \quad (\text{A.38})$$

## Región 19

- Regla 1:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_p\} = y_{dp} \rightarrow o_z$   
 $\Delta y_p = 0$
- Regla 2:  $y_{dp} = 0$      $\min \{y_{dp}, \Delta y_n\} = y_{dp} \rightarrow o_p$   
 $\Delta y_n = 1$
- Regla 3:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_p\} = \Delta y_p \rightarrow o_n$   
 $\Delta y_p = 0$
- Regla 4:  $y_{dn} = 1$      $\min \{y_{dn}, \Delta y_n\} = \Delta y_n \rightarrow o_z$   
 $\Delta y_n = 1$

$$\Delta u_D(nT) = 0 \quad (\text{A.39})$$

## Región 20

- Regla 1:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_p\} = \Delta y_p \rightarrow o_z$   
 $\Delta y_p = 0$
- Regla 2:  $y_{dp} = 1$      $\min \{y_{dp}, \Delta y_n\} = \Delta y_n \rightarrow o_p$   
 $\Delta y_n = 1$
- Regla 3:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_p\} = y_{dn} \rightarrow o_n$   
 $\Delta y_p = 0$
- Regla 4:  $y_{dn} = 0$      $\min \{y_{dn}, \Delta y_n\} = y_{dn} \rightarrow o_z$   
 $\Delta y_n = 1$

$$\Delta u_D(nT) = -L \quad (\text{A.40})$$

# Apéndice B

## Programas

El siguiente programa se utiliza para obtener los polinomios que definen las trayectorias deseadas para la pierna derecha.

```
h = 0;
for k = 0 : 0,1 : 11,9
h = h + 1;
if k <= 3,2
ak = (100/3,2) * k;
tcadera = -(0,00015 * ak.^3 - 0,0225 * ak.^2 + 75);
trodilla = (0,0002 * ak.^3 - 0,03 * ak.^2 + 100);
ttobillo = -(0,00005 * ak.^3 - 0,0075 * ak.^2 + 25);
tcaderaf = 0; ttobillof = 0;
else
if k > 3,2 & k <= 6,1
ak = (100/2,9) * (k - 3,2);
tcadera = (-0,435 * 10^-5 * ak.^3 + 0,503 * 10^-3 * ak.^2) * 37,5;
trodilla = (-1,064 * 10^-5 * ak.^3 + 1,229 * 10^-3 * ak.^2) * 3;
ttobillo = (-0,402 * 10^-5 * ak.^3 + 0,465 * 10^-3 * ak.^2) * -30;
tcaderaf = (-0,363 * 10^-5 * ak.^3 + 0,4194 * 10^-3 * ak.^2) * 1,5;
ttobillof = (-0,622 * 10^-5 * ak.^3 + 0,719 * 10^-3 * ak.^2) * -1,3;
else
am = 0,01;
xc = [05,815,927,35405066,378,983,1587,3597,35100]' * am;
yc = [25,529,7523,3750 - 13,75 - 19,375022,52526,2522,523,4]' * am;
```

```

pc = polyfit(xc, yc, 10)';
xr = [0 3 10 12,7 20 25 30 39 50 60 72 80 89 96,5 100]' * am;
yr = [5,65 6,5 13,3 20 15 11 7,22 4,44 12,22 38,88 59 50,5 13,88 44,44]' * am;
pr = polyfit(xr, yr, 10)';
xcf = [0 15 37 50 70 72 80 89 100]' * am;
ycf = [1 6,5 9 4 0 - 11 1,7 - 1,5 1,5]' * am;
pcf = polyfit(xcf, ycf, 5)';
if k > 6,1 & k <= 9
ak = am * ((100/2,9) * (k - 6,1));
tcadera = 100 * (pc(1) * ak.10 + pc(2) * ak.9 + pc(3) * ak.8 + pc(4) * ak.7 + pc(5) * ak.6 + pc(6) * ak.5 + pc(7) * ak.4 + pc(8) * ak.3 + pc(9) * ak.2 +
pc(10) * ak + pc(11));
trodilla = 100 * (pr(1) * ak.10 + pr(2) * ak.9 + pr(3) * ak.8 + pr(4) * ak.7 + pr(5) * ak.6 + pr(6) * ak.5 + pr(7) * ak.4 + pr(8) * ak.3 + pr(9) * ak.2 +
pr(10) * ak + pr(11));
ttobillo = -(tcadera + trodilla) + 10;
tcaderaf = 100 * (pcf(1) * ak.5 + pcf(2) * ak.4 + pcf(3) * ak.3 + pcf(4) * ak.2 + pcf(5) * ak + pcf(6));
ttobillof = -1,5 * tcaderaf;
else
tcadera = 0; trodilla = 0; ttobillo = 0; tcaderaf = 0; ttobillof = 0;
end
end
end
end
pcadera(h) = tcadera; pcaderaf(h) = tcaderaf; prodilla(h) = trodilla; ptobillo(h) = ttobillo; ptobillof(h) = ttobillof;
end

```

**Para la simulación de una extremidad del robot y la comprobación de la cinemática inversa se utilizó el siguiente programa**

```

%constantes
h = 0;
l = [1,656,210,58,36,42,86,45];
l1 = l(1); l2 = l(2); l3 = l(3); l4 = l(4); l5 = l(5); l6 = l(6); l7 = l(7);
%trayectorias
%cinematicainversa
t1 = deg2rad(tcaderaf); t2 = deg2rad(tcadera); t3 = deg2rad(trodilla); t4 = deg2rad(ttobillo); t5 = deg2rad(ttobillof);
a = deg2rad([0909000 - 900]);
ai = [0l(6)l(5)l(4)l(3)l(2)l(1)]; d = [0l(7)00000];
theta = [0t1t2t3t4t5];

```



```

A1 = [cos(theta(1))-sin(theta(1))0ai(1); sin(theta(1))*cos(a(1))cos(theta(1))*cos(a(1))-sin(a(1))-sin(a(1))*d(1); sin(theta(1))*sin(a(1))cos(theta(1))*
sin(a(1)) - cos(a(1)) - cos(a(1)) * d(1); 0001];

A2 = [cos(theta(2))-sin(theta(2))0ai(2); sin(theta(2))*cos(a(2))cos(theta(2))*cos(a(2))-sin(a(2))-sin(a(2))*d(2); sin(theta(2))*sin(a(2))cos(theta(2))*
sin(a(2)) - cos(a(2)) - cos(a(2)) * d(2); 0001];

A3 = [cos(theta(3))-sin(theta(3))0ai(3); sin(theta(3))*cos(a(3))cos(theta(3))*cos(a(3))-sin(a(3))-sin(a(3))*d(3); sin(theta(3))*sin(a(3))cos(theta(3))*
sin(a(3)) - cos(a(3)) - cos(a(3)) * d(3); 0001];

A4 = [cos(theta(4))-sin(theta(4))0ai(4); sin(theta(4))*cos(a(4))cos(theta(4))*cos(a(4))-sin(a(4))-sin(a(4))*d(4); sin(theta(4))*sin(a(4))cos(theta(4))*
sin(a(4)) - cos(a(4)) - cos(a(4)) * d(4); 0001];

A5 = [cos(theta(5))-sin(theta(5))0ai(5); sin(theta(5))*cos(a(5))cos(theta(5))*cos(a(5))-sin(a(5))-sin(a(5))*d(5); sin(theta(5))*sin(a(5))cos(theta(5))*
sin(a(5)) - cos(a(5)) - cos(a(5)) * d(5); 0001];

A6 = [cos(theta(6))-sin(theta(6))0ai(6); sin(theta(6))*cos(a(6))cos(theta(6))*cos(a(6))-sin(a(6))-sin(a(6))*d(6); sin(theta(6))*sin(a(6))cos(theta(6))*
sin(a(6)) - cos(a(6)) - cos(a(6)) * d(6); 0001];

T06 = A1 * A2 * A3 * A4 * A5 * A6; T05 = A1 * A2 * A3 * A4 * A5; T04 = A1 * A2 * A3 * A4; T03 = A1 * A2 * A3; T02 = A1 * A2; T01 = A1;

px6 = T06(1, 4); py6 = T06(2, 4); pz6 = T06(3, 4); plot3(pz6, py6, -px6, 'b*'); grid; hold on;

px5 = T05(1, 4); py5 = T05(2, 4); pz5 = T05(3, 4); plot3(pz5, py5, -px5, 'g*');

px4 = T04(1, 4); py4 = T04(2, 4); pz4 = T04(3, 4); plot3(pz4, py4, -px4, 'r*');

px3 = T03(1, 4); py3 = T03(2, 4); pz3 = T03(3, 4); plot3(pz3, py3, -px3, 'k*');

px2 = T02(1, 4); py2 = T02(2, 4); pz2 = T02(3, 4); plot3(pz2, py2, -px2, 'm*');

x = -[px6px5px4px3px2]; y = [py6py5py4py3py2]; z = [pz6pz5pz4pz3pz2]; plot3(z, y, x, 'c');

px = cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) - sin(theta(1)) * (sin(theta(5)) * l(6) -
cos(theta(5)) * l(7)) + cos(theta(1)) * cos(theta(2) + theta(3)) * l(4) + cos(theta(1)) * cos(theta(2)) * l(3) + cos(theta(1)) * l(2) + l(1);

py = sin(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) + cos(theta(1)) * (sin(theta(5)) * l(6) -
cos(theta(5)) * l(7)) + sin(theta(1)) * (cos(theta(2) + theta(3)) * l(4) + cos(theta(2)) * l(3) + l(2));

pz = sin(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) + sin(theta(2) + theta(3)) * l(4) + sin(theta(2)) * l(3);

T6 = [cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * cos(theta(5)) - sin(theta(1)) * sin(theta(5))cos(theta(1)) * sin(theta(2) + theta(3) +
theta(4))cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * sin(theta(5)) + sin(theta(1)) * cos(theta(5))px;

sin(theta(1))*cos(theta(2)+theta(3)+theta(4))*cos(theta(5))+cos(theta(1))*sin(theta(5))sin(theta(1))*sin(theta(2)+theta(3)+theta(4))sin(theta(1))*
cos(theta(2) + theta(3) + theta(4)) * sin(theta(5)) - cos(theta(1)) * cos(theta(5))py;

sin(theta(2) + theta(3) + theta(4)) * cos(theta(5)) - cos(theta(2) + theta(3) + theta(4))sin(theta(2) + theta(3) + theta(4)) * sin(theta(5))pz; 0001];

tp234 = acos(-T6(3, 2));

if tp234 == 0

thetapd1 = 0; thetapd5 = 0; thetapd234 = 0;

else

tp5 = asin(T6(3, 3)/sin(tp234)); tp1 = asin(T6(2, 2)/sin(tp234)); w1 = sin(tp234) * cos(tp5); w2 = sin(tp234) * cos(tp1);

if w2 == T6(1, 2)

thetapd234 = tp234; thetapd1 = tp1;

```

```

else
thetapd234 = tp234 * ((asin(T6(1, 2)/cos(tp1)))/(abs(asin(T6(1, 2)/cos(tp1))))); thetapd1 = asin(T6(2, 2)/sin(thetapd234));
end

if w1 == T6(3, 1)
thetapd234 = tp234; thetapd5 = tp5;
else
thetapd234 = tp234 * ((asin(T6(3, 1)/cos(tp5)))/(abs(asin(T6(3, 1)/cos(tp5))))); thetapd5 = asin(T6(3, 3)/sin(thetapd234));
end

end

alpha = cos(thetapd5) * l(6) + sin(thetapd5) * l(7) + l(5); beta = sin(thetapd5) * l(6) - cos(thetapd5) * l(7);
gama = (T6(1, 4) - cos(thetapd1) * cos(thetapd234) * alpha + sin(thetapd1) * beta - cos(thetapd1) * l(2) - l(1))/cos(thetapd1); delta = T6(3, 4) -
sin(thetapd234) * alpha;
tp3 = acos((gama^2 + delta^2 - l(4)^2 - l(3)^2)/(2 * l(3) * l(4)));
a = cos(tp3) * l(4) + l(3); b = sin(tp3) * l(4);
tp2a = asin((a * delta - b * gama)/(a^2 + b^2)); tp2b = acos((a * gama + b * delta)/(a^2 + b^2));
if abs(tp2a * tp2b) == (tp2a * tp2b)
thetapd2 = tp2a; thetapd3 = tp3;
else
thetapd3 = -tp3; a = cos(thetapd3) * l(4) + l(3); b = sin(thetapd3) * l(4);
thetapd2 = asin((a * delta - b * gama)/(a^2 + b^2));
end

thetapd4 = thetapd234 - thetapd2 - thetapd3;

if h >= 70 & h <= 72
thetapd3 = -thetapd3; a = cos(thetapd3) * l(4) + l(3); b = sin(thetapd3) * l(4);
thetapd2 = asin((a * delta - b * gama)/(a^2 + b^2)); thetapd4 = thetapd234 - thetapd2 - thetapd3;
end

if h >= 82 & h <= 87
thetapd3 = -thetapd3; a = cos(thetapd3) * l(4) + l(3); b = sin(thetapd3) * l(4);
thetapd2 = asin((a * delta - b * gama)/(a^2 + b^2)); thetapd4 = thetapd234 - thetapd2 - thetapd3;
end

theta = [t1t2t3t4t5];

px = cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) - sin(theta(1)) * (sin(theta(5)) * l(6) -
cos(theta(5)) * l(7)) + cos(theta(1)) * cos(theta(2) + theta(3)) * l(4) + cos(theta(1)) * cos(theta(2)) * l(3) + cos(theta(1)) * l(2) + l(1));
py = sin(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) + cos(theta(1)) * (sin(theta(5)) * l(6) -
cos(theta(5)) * l(7)) + sin(theta(1)) * (cos(theta(2) + theta(3)) * l(4) + cos(theta(2)) * l(3) + l(2));
pz = sin(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) + sin(theta(2) + theta(3)) * l(4) + sin(theta(2)) * l(3);

```

```

T6 = [cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * cos(theta(5)) - sin(theta(1)) * sin(theta(5))cos(theta(1)) * sin(theta(2) + theta(3) +
theta(4))cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * sin(theta(5)) + sin(theta(1)) * cos(theta(5))px;
sin(theta(1))*cos(theta(2)+theta(3)+theta(4))*cos(theta(5))+cos(theta(1))*sin(theta(5))sin(theta(1))*sin(theta(2)+theta(3)+theta(4))sin(theta(1))*
cos(theta(2) + theta(3) + theta(4)) * sin(theta(5)) - cos(theta(1)) * cos(theta(5))py
sin(theta(2) + theta(3) + theta(4)) * cos(theta(5)) - cos(theta(2) + theta(3) + theta(4))sin(theta(2) + theta(3) + theta(4)) * sin(theta(5))pz; 0001];
tp234 = acos(-T6(3, 2));
if tp234 == 0
thetapd1 = 0; thetapd2 = 0; thetapd234 = 0;
else
tp5 = asin(T6(3, 3)/sin(tp234)); tp1 = asin(T6(2, 2)/sin(tp234)); w1 = sin(tp234) * cos(tp5); w2 = sin(tp234) * cos(tp1);
if w2 == T6(1, 2)
thetapd234 = tp234; thetapd1 = tp1;
else
thetapd234 = tp234 * ((asin(T6(1, 2)/cos(tp1)))/(abs(asin(T6(1, 2)/cos(tp1))))); thetapd1 = asin(T6(2, 2)/sin(thetapd234));
end
if w1 == T6(3, 1)
thetapd234 = tp234; thetapd2 = tp5;
else
thetapd234 = tp234 * ((asin(T6(3, 1)/cos(tp5)))/(abs(asin(T6(3, 1)/cos(tp5))))); thetapd2 = asin(T6(3, 3)/sin(thetapd234));
end
end
thetapd = rad2deg([thetapd1; thetapd3; thetapd4; thetapd5; thetapd2]);

```

**Con el siguiente programa se realiza la visualización de la estructura del robot en 3D, con ayuda del toolbox de robótica para MATLAB**

```

l = [1,656,210,58,36,42,86,45]./100;
a = deg2rad([090 - 90180 - 180]);
Ld7 = link([a(1) - l(1)00], 'standard');
Ld6 = link([a(3) - l(2)00], 'standard');
Ld5 = link([a(1) - l(3)00], 'standard');
Ld4 = link([a(1) - l(4)00], 'standard');
Ld3 = link([a(2) - l(5)00], 'standard');
Ld2 = link([a(2) - l(6)00], 'standard');
Ld1 = link([a(1)00 - l(7)], 'standard');
%piernaizquierda
Li1 = link([a(1)00l(7)], 'standard');
Li2 = link([a(2)l(6)00], 'standard');

```

```

Li3 = link([a(3)l(5)00], 'standard');
Li4 = link([a(1)l(4)00], 'standard');
Li5 = link([a(1)l(3)00], 'standard');
Li6 = link([a(2)l(2)00], 'standard');
Li7 = link([a(1)l(1)00], 'standard');

%robotpiernaderecha@link
r1 = robot(Ld1Ld2Ld3Ld4Ld5Ld6Ld7);

r1.name = 'bipedo';

fork = 0 : h - 1

th5 = 0; %plantadepie(eslabonextra)basedestructura
th0 = ptobillof(k + 1); %tobillofrontalderecha
th1 = ptobillo(k + 1); %tobillolateralderecha
th2 = prodilla(k + 1); %rodilladerecha = 40
th3 = pcadera(k + 1); %caderalateralderecha
th4 = pcaderaf(k + 1); %caderafrontalderecha
th6 = 0; %eslabonextra

%asignaciondevaloresparagraficacion
q1 = deg2rad([th6th5th4th3th2th1th0]);

plot(r1, q1);

holdon;

pause(0,5)

end

```

**El programa siguiente realiza los cálculos de la cinemática directa**

```

function salida = cinematicaN(j)

%entradas

theta1 = j(1); theta2 = j(2); theta3 = j(3); theta4 = j(4); theta5 = j(5); %theta
dtheta1 = j(6); dtheta2 = j(7); dtheta3 = j(8); dtheta4 = j(9); dtheta5 = j(10); %thetaderivada

%constantes

l = [1,656,210,58,36,42,86,45]./100;

a = deg2rad([0 - 90009090]);

ai = [l(1)l(2)l(3)l(4)l(5)l(6)];

d = [00000l(7)];

l1 = l(1); l2 = l(2); l3 = l(3); l4 = l(4); l5 = l(5); l6 = l(6); l7 = l(7);

t1 = theta1; t2 = theta2; t3 = theta3; t4 = theta4; t5 = theta5;

xpd1 = l2 * cos(t1);

ypd1 = l2 * sin(t1);

```

```

zpd1 = 0;

theta = [t1t2t3t4t5];

px = cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) - sin(theta(1)) * (sin(theta(5)) * l(6) -
cos(theta(5)) * l(7)) + cos(theta(1)) * cos(theta(2) + theta(3)) * l(4) + cos(theta(1)) * cos(theta(2)) * l(3) + cos(theta(1)) * l(2) + l(1);

py = sin(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) + cos(theta(1)) * (sin(theta(5)) * l(6) -
cos(theta(5)) * l(7)) + sin(theta(1)) * (cos(theta(2) + theta(3)) * l(4) + cos(theta(2)) * l(3) + l(2));

pz = sin(theta(2) + theta(3) + theta(4)) * (cos(theta(5)) * l(6) + sin(theta(5)) * l(7) + l(5)) + sin(theta(2) + theta(3)) * l(4) + sin(theta(2)) * l(3);

T6 = [cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * cos(theta(5)) - sin(theta(1)) * sin(theta(5))cos(theta(1)) * sin(theta(2) + theta(3) +
theta(4))cos(theta(1)) * cos(theta(2) + theta(3) + theta(4)) * sin(theta(5)) + sin(theta(1)) * cos(theta(5))px;

sin(theta(1))*cos(theta(2)+theta(3)+theta(4))*cos(theta(5))+cos(theta(1))*sin(theta(5))sin(theta(1))*sin(theta(2)+theta(3)+theta(4))sin(theta(1))*
cos(theta(2) + theta(3) + theta(4)) * sin(theta(5)) - cos(theta(1)) * cos(theta(5))py;

sin(theta(2) + theta(3) + theta(4)) * cos(theta(5)) - cos(theta(2) + theta(3) + theta(4))sin(theta(2) + theta(3) + theta(4)) * sin(theta(5))pz; 0001];

j11 = (cos(t5) * l7 - sin(t5) * l6) * cos(t1) + (-cos(t2 + t3 + t4) * sin(t5) * l7 - cos(t2 + t3 + t4) * l5 - cos(t2 + t3 + t4) * cos(t5) * l6 - cos(t2 + t3) *
l4 - cos(t2) * l3 - l2) * sin(t1);

j12 = (-sin(t2 + t3 + t4) * cos(t5) * l6 - sin(t2 + t3 + t4) * sin(t5) * l(7) - sin(t2 + t3 + t4) * l5 - sin(t2 + t3) * l4 - sin(t2) * l3) * cos(t1);

j13 = (-sin(t2 + t3 + t4) * cos(t5) * l6 - sin(t2 + t3 + t4) * sin(t5) * l(7) - sin(t2 + t3 + t4) * l5 - sin(t2 + t3) * l4) * cos(t1);

j14 = -cos(t1) * sin(t2 + t3 + t4) * cos(t5) * l6 - cos(t1) * sin(t2 + t3 + t4) * sin(t5) * l7 - cos(t1) * sin(t2 + t3 + t4) * l5;

j15 = -cos(t1) * cos(t2 + t3 + t4) * sin(t5) * l6 + cos(t1) * cos(t2 + t3 + t4) * cos(t5) * l7 - sin(t1) * cos(t5) * l6 - sin(t1) * sin(t5) * l7;

j21 = (cos(t2 + t3 + t4) * cos(t5) * l6 + cos(t2 + t3 + t4) * sin(t5) * l7 + cos(t2 + t3 + t4) * l5 + l2 + cos(t2 + t3) * l4 + cos(t2) * l3) * cos(t1) + (cos(t5) *
l7 - sin(t5) * l6) * sin(t1);

j22 = (-sin(t2 + t3 + t4) * cos(t5) * l6 - sin(t2 + t3 + t4) * sin(t5) * l7 - sin(t2 + t3 + t4) * l5 - sin(t2 + t3) * l4 - sin(t2) * l3) * sin(t1);

j23 = (-sin(t2 + t3 + t4) * cos(t5) * l6 - sin(t2 + t3 + t4) * sin(t5) * l7 - sin(t2 + t3 + t4) * l5 - sin(t2 + t3) * l4) * sin(t1);

j24 = -sin(t1) * sin(t2 + t3 + t4) * cos(t5) * l6 - sin(t1) * sin(t2 + t3 + t4) * sin(t5) * l7 - sin(t1) * sin(t2 + t3 + t4) * l5;

j25 = -sin(t1) * cos(t2 + t3 + t4) * sin(t5) * l6 + sin(t1) * cos(t2 + t3 + t4) * cos(t5) * l7 + cos(t1) * cos(t5) * l6 + cos(t1) * sin(t5) * l7;

j31 = 0;

j32 = cos(t2 + t3 + t4) * cos(t5) * l6 + cos(t2 + t3 + t4) * sin(t5) * l(7) + cos(t2 + t3 + t4) * l5 + cos(t2 + t3) * l4 + cos(t2) * l3;

j33 = cos(t2 + t3 + t4) * cos(t5) * l6 + cos(t2 + t3 + t4) * sin(t5) * l(7) + cos(t2 + t3 + t4) * l5 + cos(t2 + t3) * l4;

j34 = cos(t2 + t3 + t4) * cos(t5) * l6 + cos(t2 + t3 + t4) * sin(t5) * l(7) + cos(t2 + t3 + t4) * l5;

j35 = -sin(t2 + t3 + t4) * sin(t5) * l6 + sin(t2 + t3 + t4) * cos(t5) * l7;

%Matriz J

J = [j11j12j13j14j15; j21j22j23j24j25; j31j32j33j34j35];

dT = [dtheta1; dtheta2; dtheta3; dtheta4; dtheta5];

dX = J * dT;

dx1 = dX(1, 1);

dx2 = dX(2, 1);

dx3 = dX(3, 1);

```

```
%vectordesalida
```

```
salida = [dx1; dx2; dx3; T6(1, 2); T6(2, 2); T6(3, 1); T6(3, 2); T6(3, 3)];
```

## Programa para la solución de la cinemática inversa

```
functionsinv = cinversa(einv)
```

```
x = einv(1); y = einv(2); z = einv(3); T612 = einv(4);
```

```
T622 = einv(5); T631 = einv(6); T632 = einv(7); T633 = einv(8);
```

```
l = [1,656,210,58,36,42,86,45]./100;
```

```
n = 1;
```

```
tp234 = acos(-T632);
```

```
if tp234 == 0
```

```
thetapd1 = 0; thetapd5 = 0; thetapd234 = 0;
```

```
else
```

```
tp5 = asin(T633/sin(tp234)); tp1 = asin(T622/sin(tp234));
```

```
w1 = sin(tp234) * cos(tp5); w2 = sin(tp234) * cos(tp1);
```

```
if w2 == T612
```

```
thetapd234 = tp234; thetapd1 = -tp1;
```

```
else
```

```
thetapd234 = tp234 * ((asin(T612/cos(tp1)))/(abs(asin(T612/cos(tp1))))); thetapd1 = asin(T622/sin(thetapd234));
```

```
end
```

```
if w1 == T631
```

```
thetapd234 = tp234; thetapd5 = -tp5;
```

```
else
```

```
thetapd234 = tp234 * ((asin(T631/cos(tp5)))/(abs(asin(T631/cos(tp5))))); thetapd5 = asin(T633/sin(thetapd234));
```

```
end
```

```
end
```

```
alpha = cos(thetapd5) * l(6) + sin(thetapd5) * l(7) + l(5);
```

```
beta = sin(thetapd5) * l(6) - cos(thetapd5) * l(7);
```

```
gama = (x - cos(thetapd1) * cos(thetapd234) * alpha + sin(thetapd1) * beta - cos(thetapd1) * l(2) - l(1))/cos(thetapd1);
```

```
delta = z - sin(thetapd234) * alpha;
```

```
tp3 = acos((gama2 + delta2 - l(4)2 - l(3)2)/(2 * l(3) * l(4)));
```

```
a = cos(tp3) * l(4) + l(3); b = sin(tp3) * l(4);
```

```
tp2a = asin((a * delta - b * gama)/(a2 + b2));
```

```
tp2b = acos((a * gama + b * delta)/(a2 + b2));
```

```
if abs(tp2a * tp2b) == (tp2a * tp2b)
```

```
thetapd2 = tp2a; thetapd3 = -tp3;
```

```
else
```

```

thetapd3 = -tp3;

a = cos(thetapd3) * l(4) + l(3);

b = sin(thetapd3) * l(4);

thetapd2 = asin((a * delta - b * gama)/(a^2 + b^2));

end

thetapd4 = thetapd234 - thetapd2 - thetapd3;

if thetapd3 < 0,0005 & thetapd3 > -0,0005

n = 1;

end

if n == 1

thetapd3 = -thetapd3;

a = cos(thetapd3) * l(4) + l(3);

b = sin(thetapd3) * l(4);

thetapd2 = asin((a * delta - b * gama)/(a^2 + b^2));

thetapd4 = thetapd234 - thetapd2 - thetapd3;

end

thetapd = rad2deg([thetapd1; thetapd2; thetapd3; thetapd4; thetapd5]);

sinv = [thetapd1; thetapd2; thetapd3; thetapd4; thetapd5];

```

**El programa siguiente contiene el control PI+D difuso con las reglas vistas en el Apéndice A. Donde  $K_P$ ,  $K_I$ ,  $K_D$  y  $L$  pueden ser variables**

```

function[sys,x0, str, ts] = todo(t,x,u,flag)

switch flag,

case 0

case 3

sys = mdlOutputs(t,x,u);

case 1, 2, 4, 9

sys = [];

otherwise

error(['Unhandled flag =', num2str(flag)]);

end

function[sys,x0, str, ts] = mdlInitializeSizes()

sizes = simsizes;

sizes.NumContStates = 0;

sizes.NumDiscStates = 0;

sizes.NumOutputs = -1;

sizes.NumInputs = -1;

```

```

sizes.DirFeedthrough = 1;

sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

str = [];

x0 = [];

ts = [-10];

functionsys = mdlOutputs(t,x,u)

T = 0,1;

K = 1,0;

Kp, Kd, Ki, L

ref = 5,0;

Kud = 0,01;

Kupi = 0,2;

vnt = (u(1) - u(3))/T;

a = L * (Ki * u(1) + Kp * vnt)/(2 * (2 * L - Ki * abs(u(1))));

aa = L * (Ki * u(1) + Kp * vnt)/(2 * (2 * L + Ki * abs(u(1))));

b = L * (Ki * u(1) + Kp * vnt)/(2 * (2 * L - Kp * abs(vnt)));

bb = L * (Ki * u(1) + Kp * vnt)/(2 * (2 * L + Kp * abs(vnt)));

c = (1/2) * (L + Kp * vnt);

d = (1/2) * (L + Ki * u(1));

e = (1/2) * (-L + Kp * vnt);

f = (1/2) * (-L + Ki * u(1));

g = 0;

h = L;

i = -L;

upi = u(6) + Kupi * (a + aa + b + bb + c + d + e + f + g + h + i);

ynt = (u(4) - u(5))/T;

a1 = L * (K * u(2) - Kd * ynt)/(2 * (2 * L - K * abs(u(2))));

b1 = L * (K * u(2) - Kd * ynt)/(2 * (2 * L + K * abs(u(2))));

c1 = (1/2) * (L - Kd * ynt);

d1 = (1/2) * (-L + K * u(2));

e1 = (1/2) * (-L - Kd * ynt);

f1 = (1/2) * (L + K * u(2));

g1 = 0;

h1 = -L;

i1 = L;

```



$ud = u(7) + Kud * (a1 + b1 + c1 + d1 + e1 + f1 + g1 + h1 + i1);$   
 $upid = upi - ud;$   
 $sys = [upid, upi, ud, a, b, c, d];$

La siguiente figura muestra el diagrama elaborado en simulink para el sistema a lazo abierto

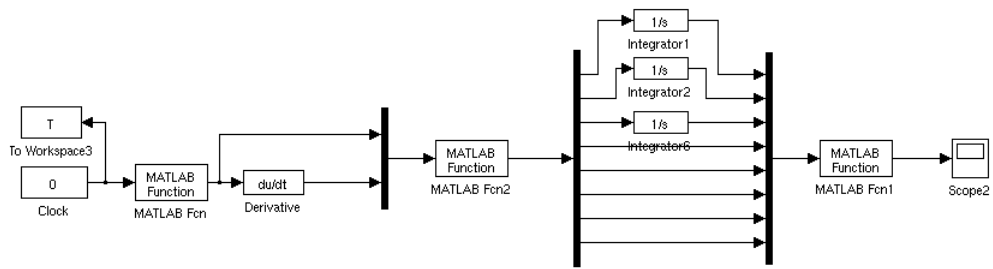


Figura B.1: Sistema lazo abierto

La siguiente figura muestra el diagrama elaborado en simulink para el control PID convencional

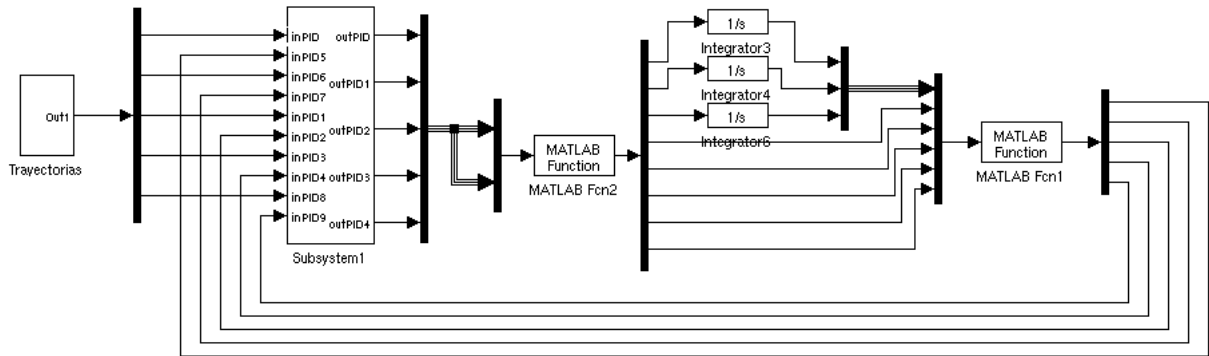


Figura B.2: PID convencional

La siguiente figura representa el subsystem1 de la imagen anterior que corresponde al control PID convencional

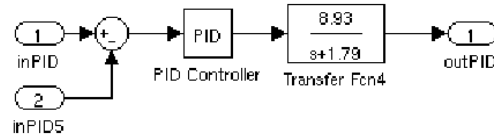


Figura B.3: control PID convencional [1]

La siguiente figura muestra el diagrama del controlador PI+D difuso

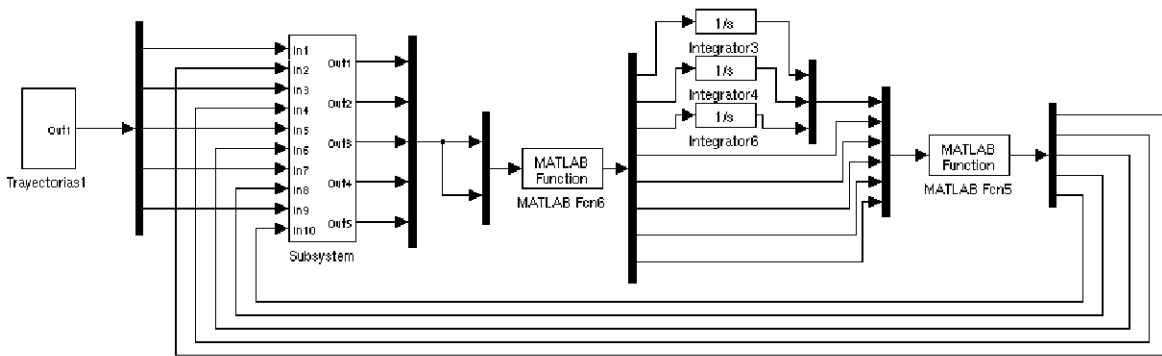


Figura B.4: PI+D difuso

La siguiente figura representa el subsystem de la imagen anterior que corresponde al control PI+D difuso

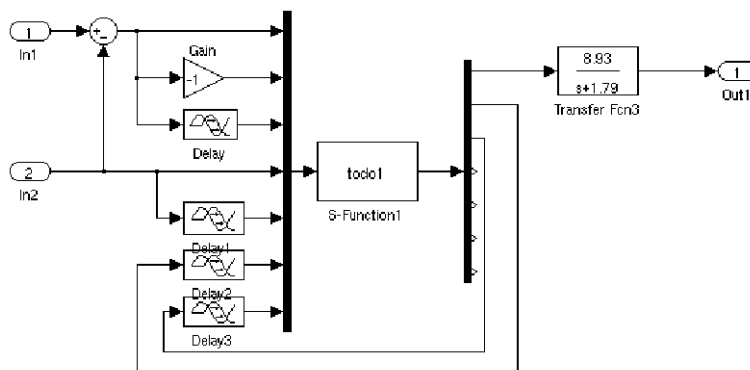


Figura B.5: PI+D difuso