



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

PLANIFICACIÓN Y OPTIMIZACIÓN
DE LA CAMINATA DE UN ROBOT BÍPEDO

T E S I S

QUE PARA OPTAR POR EL GRADO DE
INGENIERO MECATRÓNICO

PRESENTA

RAFAEL LÓPEZ GARCÍA

TUTOR:

DR. EDMUNDO GABRIEL ROCHA CÓZATL



2012

Generación de patrones de marcha dinámicamente estables
para un robot bípedo antropomórfico

Rafael López García

20 de febrero de 2012

Agradecimientos

A papá y mamá, por todo su amor.

A mi hermana, por su amistad incondicional.

A mi tía Norma, que tanto me quiere, que tanto la quiero.

A mis tías Laura, Martha, Angélica y Alicia, por todo su afecto y confianza.

A mi abuela Lore, por su cariño.

A Saraí y Gilberto, que siempre están aquí.

A todos mis amigos de la facultad, en especial a Luis, Irving y Sahel, con quienes he compartido momentos tan agradables.

A Edmundo y Octavio, cuya orientación y ayuda hizo posible este trabajo.

Índice general

1. Introducción	12
1.1. Motivación	12
1.2. Objetivos	13
1.3. Trabajo previo	13
1.4. Organización y contenido de la tesis	14
2. La robótica bípeda	16
2.1. Panorama histórico	16
2.1.1. Las primeras investigaciones de robots bípedos en universidades y laboratorios	16
2.1.1.1. El proyecto WABOT	16
2.1.1.2. Robots gimnásticos del LegLab del MIT	18
2.1.2. Desarrollo de robots bípedos humanoides y la exploración de sus aplicaciones . .	19
2.1.2.1. Los humanoides de Honda	19
2.1.2.2. Avances en la Universidad de Waseda posteriores al proyecto WABOT .	20
2.1.2.3. El proyecto de humanoides HRP	21
2.2. Panorama actual	21
2.2.1. Proyecto de Robots de Compañía de Toyota	21
2.2.2. Los humanoides MAHRU y AHRA	22
2.2.3. Surena II	23
2.2.4. PAL Robotics	23
2.2.5. Concursos y torneos	24
2.2.6. La robótica bípeda en México	24
2.3. Marco conceptual	25
2.3.1. La caminata pasiva y energizada	25
2.3.2. Arquitecturas mecánicas de los robots bípedos	26
2.3.3. Otras clasificaciones	27
2.3.4. Breve descripción de la marcha	28
2.3.5. Sistema de referencia	29
2.4. Robótica bípeda: razones y objetivos	29
2.5. Temas actuales en la robótica bípeda	31
2.5.1. Estabilidad	32
2.5.2. Generación de patrones de marcha	34
2.5.3. Control	34

3. La locomoción bípeda	36
3.1. La marcha	36
3.1.1. Definición	36
3.1.2. El ciclo de marcha	36
3.1.3. Tareas y fases de la marcha	37
3.1.4. Nomenclatura tradicional	38
3.1.5. Elementos funcionales	39
3.1.6. Características del paso	40
3.2. La marcha humana	40
3.2.1. Actividad en el sistema locomotor durante la marcha	41
3.2.2. Actividad en el sistema pasajero durante la marcha	43
3.2.3. Acción muscular durante la marcha	45
4. Descripción del banco de pruebas	46
4.1. El prototipo de pruebas	46
4.1.1. El robot bípedo Scout	46
4.1.2. Arquitectura del robot	47
4.2. Interfaz gráfica de usuario para el accionamiento y manipulación del prototipo de pruebas	48
4.2.1. Concepto y trabajo relacionado	48
4.2.2. La interfaz gráfica de usuario	49
4.2.3. Descripción de la interfaz gráfica	49
4.2.4. Accionamiento de los servomotores	51
4.2.5. Ensamble y programación del prototipo virtual	52
4.2.6. Resultados	53
5. Planificación de la caminata	55
5.1. Modelos para la planificación de la caminata	55
5.2. Métodos para la planificación de la caminata	56
5.3. Planificación de la marcha mediante la generación de trayectorias para sus efectores finales	57
5.3.1. Trayectorias de los efectores finales	59
5.3.2. Efectos por la reducción del número de trayectorias y justificación	59
5.3.3. Función para definir a las trayectorias	60
5.3.4. Parámetros de tiempo para definir a la marcha	62
5.3.5. Parámetros para definir las trayectorias del pie flotante	62
5.3.6. Parámetros para definir las trayectorias de la cadera	63
5.3.7. Ecuaciones para definir las trayectorias de la cadera y del pie flotante	64
5.3.8. Trayectorias de inicio y de paro de la marcha	68
5.3.9. Algoritmo	68
5.4. Generación de un ciclo de marcha	70
5.4.1. Selección de parámetros	70
5.4.2. Graficas de las trayectorias de los pies y torso	71
5.4.3. Simulación del ciclo de marcha	71

6. Optimización de la marcha	76
6.1. Propuesta empleada y justificación	76
6.2. Los algoritmos genéticos	77
6.2.1. Bases biológicas	77
6.2.2. Elementos y operadores de un algoritmo genético	78
6.2.3. Evaluación y condiciones de paro	79
6.3. Algoritmo genético	80
6.3.1. Aclaraciones	80
6.3.2. Parámetros	81
6.3.3. Representación	81
6.3.4. Generación de la población inicial	82
6.3.5. Límites de los parámetros para la trayectoria de la cadera	82
6.3.6. Función objetivo	84
6.3.6.1. Definición del área de seguridad	86
6.3.6.2. Cálculo del ZMP	87
6.3.7. Selección	91
6.3.8. Reproducción	92
6.3.9. Mutación	93
6.3.10. Algoritmo	94
7. Experimentos y simulaciones	96
7.1. Descripción de las pruebas	96
7.1.1. Selección de parámetros	96
7.1.2. Trayectorias de los eslabones L61 y L62	99
7.2. Resultados	99
7.2.1. Desempeño del algoritmo	99
7.2.2. Selección de los cromosomas	100
7.2.3. Trayectorias de la cadera optimizadas	102
7.2.4. Trayectorias del ZMP de los patrones optimizados	103
7.3. Simulación de un ciclo de marcha	103
7.3.1. Trayectorias de posición y velocidad angular	103
7.3.2. Simulaciones en Mathematica y LabVIEW	107
7.3.3. Prueba con el prototipo físico	110
8. Conclusiones y trabajo a futuro	112
8.1. Conclusiones	112
8.2. Conclusiones	114
8.3. Conclusiones	116
8.4. Conclusiones	118
8.5. Trabajo a futuro	120
8.6. Trabajo a futuro	124
8.7. Trabajo a futuro	127
8.8. Trabajo a futuro	129
8.9. Organización y contenido de la tesis	132

9. La robótica bípeda	133
9.1. Panorama histórico	133
9.1.1. Las primeras investigaciones de robots bípedos en universidades y laboratorios	133
9.1.1.1. El proyecto WABOT	133
9.1.1.2. Robots gimnásticos del LegLab del MIT	135
9.1.2. Desarrollo de robots bípedos humanoides y la exploración de sus aplicaciones . .	136
9.1.2.1. Los humanoides de Honda	136
9.1.2.2. Avances en la Universidad de Waseda posteriores al proyecto WABOT .	137
9.1.2.3. El proyecto de humanoides HRP	138
9.2. Panorama actual	138
9.2.1. Proyecto de Robots de Compañía de Toyota	138
9.2.2. Los humanoides MAHRU y AHRA	139
9.2.3. Surena II	140
9.2.4. PAL Robotics	140
9.2.5. Concursos y torneos	141
9.2.6. La robótica bípeda en México	141
9.3. Marco conceptual	142
9.3.1. La caminata pasiva y energizada	142
9.3.2. Arquitecturas mecánicas de los robots bípedos	143
9.3.3. Otras clasificaciones	144
9.3.4. Breve descripción de la marcha	145
9.3.5. Sistema de referencia	146
9.4. Robótica bípeda: razones y objetivos	146

Índice de figuras

2.1. Robots construidos como parte del proyecto WABOT. a) WL-5, b) WABOT-1, c) WL-10RD, d) WL-12.	17
2.2. Robots construidos en el MIT. a) Planar One-Leg Hooper. b) 3D One Leg-Hooper. c) Planar Biped.	18
2.3. Bipedos y humanoides desarrollados por Honda	19
2.4. Honda ASIMO descendiendo escaleras.	20
2.5. Humanoide WABIAN-2R de la Universidad de Waseda.	21
2.6. El robot humanoide HRP-2. Puede caminar sobre superficies irregulares o sobre caminos estrechos.	22
2.7. Humanoides de la familia de robots de compañía de Toyota.	22
2.8. Surena II, un humanoide desarrollado con el propósito de explorar aspectos teóricos y prácticos de la caminata bípeda.	23
2.9. El bípedo REEM-A de PAL Robotics. Participó en la RoboCup del 2006, donde la competencia de caminata.	24
2.10. Bípedo bidimensional de McGeer. Duplicado construido en la Universidad de Cornell.	26
2.11. A la izquierda, QRIO, un humanoide de entretenimiento que fue cancelado en el 2006. A la derecha, NAO, el reemplazo de AIBO como plataforma estandar en la RoboCup.	27
2.12. Secuencia de un paso. Cambio de fase de soporte simple a fase de soporte doble en un bípedo bidimensional. CHEVALEREAU. Bipedal Robots, 2009.	28
2.13. Planos del movimiento humano. BEHNKE. Kinetic Anatomy, 2000.	29
2.14. Prototipo del bípedo Atlas. Según sus diseñadores, una vez terminado, será capaz de caminar sobre terrenos desiguales.	31
2.15. ASIMO realizando algunas tareas de servicio.	32
2.16. Fuerzas actuantes en la planta de un robot bípedo. Adaptado de VUKOBRATOVIC. ZMP -Thirty Five Years of its Life, 2004.	33
3.1. Ciclo de marcha de un niño de 8 años (edad promedio a la que se alcanza el patrón propio del adulto) descrito —en este caso para la pierna derecha— por las 8 unidades funcionales propuestas por Jaquelin Perry. Adaptado de L VAUGHAN C. Dynamics of human gait, 1992.	37
3.2. Nomenclatura tradicional para describir el ciclo común de la caminata bípeda. L VAUGHAN C. Dynamics of human gait, 1992.	38
3.3. Características del paso a través de las huellas. Adaptado de L VAUGHAN C. Dynamics of human gait, 1992.	40

3.4. Movimientos de las articulaciones del sistema locomotor en el plano sagital. La línea punteada representa las trayectorias de un adulto, y la línea continua representa las trayectorias de un niño de 7 años. La línea vertical punteada separa al periodo de postura y oscilación. Adaptado de GANLEY. Gait kinematics and kinetics of 7-year-old children, 2005.	41
3.5. Secuencias de las áreas de soporte del pie durante la postura. Adaptado de PERRY JACQUELIN. Gait analysis, 1992.	42
3.6. El movimiento de los brazos y hombros al caminar. La imagen de en medio muestra el ciclo de marcha desde una vista superior. De igual manera, la imagen de abajo muestra una vista superior del ciclo de marcha, pero con un corte transversal en la cadera. FERNÁNDEZ IGLESIAS S. Locomoción bípeda del robot humanoide Nao, 2009.	43
3.7. Vistas lateral y posterior de siete grupos musculares de las extremidades inferiores, mostrando la actividad muscular de éstos durante las fases de la marcha. El sombreado negro representa el mayor nivel de actividad muscular, y el blanco, el estado inactivo de los músculos.	44
4.1. El robot bípedo Scout.	46
4.2. Modelo simplificado del bípedo Scout. Los ángulos θ_{ni} se asocian a las rotaciones de los servomotores en las juntas.	47
4.3. Panel frontal. Interacción directa con Scout.	50
4.4. Panel frontal. Manipulación del bípedo mediante patrones de caminata.	50
4.5. Panel frontal. Modelo virtual del bípedo.	51
4.6. Relación ciclo de trabajo - posición angular del servomotor	52
4.7. Árboles jerárquicos para el modelo virtual.	52
4.8. Ciclo de marcha implementado en el prototipo real.	53
4.9. Simulación del ciclo de marcha.	54
5.1. En la planificación mediante parámetros se proponen trayectorias para los pies y cadera. 56	
5.2. Esquema de robot bípedo con un mecanismo compensador usado para seguir una trayectoria preestablecida del ZMP. Adaptado de Springer Handbook of Robotics, 2008.	57
5.3. Generación de una trayectoria del ZMP dadas las huellas de los pies. Adaptado de Springer Handbook of Robotics, 2008.	58
5.4. Función cúbica con $T = 10$ y $q(0) = q(T) = 0$	61
5.5. Esquema del ciclo de marcha.	62
5.6. Parámetros para la generación de trayectorias. Vista sagital. La pierna de color azul es la oscilante, y la pierna roja es la de soporte.	63
5.7. Trayectorias de la cadera.	64
5.8. Parámetros para la generación de trayectorias. Vista frontal.	65
5.9. En la imagen izquierda, el inicio de la marcha, y en la imagen derecha, el paro.	69
5.10. Diagrama de flujo para la planificación de la caminata.	72
5.11. Trayectorias para definir la posición y orientación del eslabón L_B según los parámetros seleccionados.	73
5.12. Trayectorias para definir la posición de los eslabones L_{61} y L_{62} según los parámetros seleccionados.	74

5.13. Simulación de la caminata del robot bípedo Scout. a) ($t=0$) corresponde a la postura inicial del robot, justo cuando comienza su puesta en marcha. En b) ($t=10$) comienza el ciclo de marcha del robot. c) ($t=20s$) y e) ($t=30s$) son las etapas de soporte doble de los pasos izquierdo y derecho del robot. e) ($t=40$) corresponde al paro del robot, donde regresa a su postura inicial. 75

6.1. Elementos básicos de un algoritmo genético 79

6.2. Diferentes trayectorias de la cadera sobre el eje x para un semiciclo de marcha. Los parámetros para la curva a) son $F_c = 20$, $V_{fc_o} = -1$ y $V_{fc_m} = 6.5$. Para la curva b) son $F_c = 20$, $V_{fc_o} = 2$ y $V_{fc_m} = 6.5$. Y para la curva c) son $F_c = 15$, $V_{fc_o} = 0$ y $V_{fc_m} = 6.5$. 83

6.3. Áreas de interés para determinar la estabilidad del bipedo de acuerdo al punto donde esté ubicado el ZMP. 84

6.4. Trayectorias del ZMP sobre el eje x para dos ciclos de marcha diferentes. 86

6.5. Vista inferior de la planta de un pie del bípedo Scout. El borde rojo representa los límites de la planta, y el borde azul, los límites del área de seguridad. 87

6.6. Modelo del péndulo invertido. 90

6.7. Cruza monopunto. La línea punteada marca el punto por donde se dividirá a los cromosomas padres. 92

6.8. Grupos de genes. Cada cromosoma se compone de estos grupos, conteniendo así todos los genes. 93

6.9. Cruza en un solo punto modificada. La línea punteada asigna la división al cromosoma virtual donde se cambia el origen de sus genes. Los bordes negros redondeados dividen a los cromosomas por grupos. 94

6.10. Diagrama de flujo para la optimización de la marcha. 95

7.1. Trayectorias para los eslabones L_{61} y L_{62} de los tres patrones de marcha. Las únicas que cambian son las del desplazamiento sobre el eje y 98

7.2. Valor del ajuste, o *fitness*, promedio del algoritmo genético durante cada generación para cada conjunto de parámetros. 100

7.3. Datos sobre el ajuste, o *fitness*, de la última generación del algoritmo genético para cada conjunto de parámetros. El primer renglón muestra su valor promedio, los siguientes 2 muestran el valor mínimo o máximo del individuo peor o mejor evaluado, y los últimos dos, el número de individuos con dicha calificación. 100

7.4. Parámetros de posición de los cromosomas seleccionados para generar las trayectorias de la cadera. 100

7.5. Parámetros de velocidad de los cromosomas seleccionados para generar las trayectorias de la cadera. 101

7.6. Trayectorias optimizadas a seguir por el eslabón L_B dadas las trayectorias a seguir por los eslabones L_{61} y L_{62} 101

7.7. Trayectorias del ZMP para los patrones de caminata optimizados. En a) $L_p = 100 [mm]$, b) $L_p = 150 [mm]$ y c) $L_p = 200 [mm]$. Los círculos verdes representan a la trayectoria del ZMP durante el soporte doble, y los verdes, a la trayectoria del ZMP durante el soporte simple. Los polígonos negros dibujados con línea continua representan los límites del área de seguridad durante cada paso, y los dibujados con línea discontinua representan al pie que se levanta al inicio de la marcha o al pie que aterriza durante el paro de la marcha. 102

7.8. Valores adoptados por los ángulos $\theta_{11}, \theta_{21}, \theta_{31}, \theta_{41}, \theta_{51}, \theta_{61}$ durante el ciclo de marcha propuesto. 104

7.9. Valores adoptados por los ángulos $\theta_{12}, \theta_{22}, \theta_{32}, \theta_{42}, \theta_{52}, \theta_{62}$ durante el ciclo de marcha propuesto. 105

7.10. Velocidades de las juntas rotacionales $\theta_{11}, \theta_{21}, \theta_{31}, \theta_{41}, \theta_{51}, \theta_{61}$ durante el ciclo de marcha propuesto. 106

7.11. Velocidades de las juntas rotacionales $\theta_{12}, \theta_{22}, \theta_{32}, \theta_{42}, \theta_{52}, \theta_{62}$ durante el ciclo de marcha propuesto. 107

7.12. Configuraciones adoptadas por el bípedo durante la simulación del ciclo de marcha propuesto. a), b) y c) corresponden a la etapa de inicio de la marcha. c), d) y e) corresponden al primer semiciclo de la etapa cíclica de la marcha. La esfera naranja representa la posición del centro de gravedad; el círculo que se traslada sobre la superficie de desplazamiento representa a la posición del ZMP. 108

7.13. Configuraciones adoptadas por el bípedo durante la simulación del ciclo de marcha propuesto. a), b) y c) corresponden al segundo semiciclo de la etapa cíclica de la marcha. c), d) y e) corresponden a la etapa de paro de la marcha. La esfera naranja representa la posición del centro de gravedad; el círculo que se traslada sobre la superficie de desplazamiento representa a la posición del ZMP. 109

7.14. Configuraciones adoptadas por el bípedo durante la simulación del ciclo de marcha propuesto. a), b) y c) corresponden a la etapa de inicio de la marcha. c), d) y e) corresponden al primer semiciclo de la etapa cíclica de la marcha. e), f) y g) corresponden al segundo semiciclo de la etapa cíclica de la marcha. h), i) y j) corresponden a la etapa de paro de la marcha. 110

7.15. Caminata del robot bípedo Scout para el ciclo de marcha propuesto. 111

9.1. Robots construidos como parte del proyecto WABOT. a) WL-5, b) WABOT-1, c) WL-10RD, d) WL-12. 134

9.2. Robots construidos en el MIT. a) Planar One-Leg Hooper. b) 3D One Leg-Hooper. c) Planar Biped. 135

9.3. Bípedos y humanoides desarrollados por Honda 136

9.4. Honda ASIMO descendiendo escaleras. 137

9.5. Humanoide WABIAN-2R de la Universidad de Waseda. 138

9.6. El robot humanoide HRP-2. Puede caminar sobre superficies irregulares o sobre caminos estrechos. 139

9.7. Humanoides de la familia de robots de compañía de Toyota. 139

9.8. Surena II, un humanoide desarrollado con el propósito de explorar aspectos teóricos y prácticos de la caminata bípeda. 140

9.9. El bípedo REEM-A de PAL Robotics. Participó en la RoboCup del 2006, donde la competencia de caminata.	141
9.10. Bípedo bidimensional de McGeer. Duplicado construido en la Universidad de Cornell. .	143
9.11. A la izquierda, QRIO, un humanoide de entretenimiento que fue cancelado en el 2006. A la derecha, NAO, el reemplazo de AIBO como plataforma estandar en la RoboCup. .	144
9.12. Secuencia de un paso. Cambio de fase de soporte simple a fase de soporte doble en un bípedo bidimensional. CHEVALEREAU. Bipedal Robots, 2009.	145
9.13. Planos del movimiento humano. BEHNKE. Kinetic Anatomy, 2000.	146

Capítulo 1

Introducción

1.1. Motivación

Durante varios años, la necesidad de construir máquinas capaces de realizar actividades monótonas, de gran precisión o de alto riesgo para los obreros, motivó al desarrollo de autómatas con objetivos particulares y características muy distintas a las de los seres humanos, tales como los manipuladores robóticos. Si bien su estudio aún persiste, el propio progreso en la robótica clásica, así como la inquietud por construir mecanismos semejantes a los organismos biológicos y los beneficios que éstos implican, han motivado la investigación en robots biológicamente inspirados. Los investigadores, persuadidos por la idea de que en un futuro los robots convivirán con humanos y, por lo tanto, deberán ser capaces de desenvolverse en los mismos entornos, han mantenido gran interés por el desarrollo de robots bípedos. Gracias al impacto de humanoides como ASIMO y HRP-2, a la necesidad de contar con robots capaces de encargarse de tareas donde nuestra sociedad tiene deficiencias y a las aportaciones en el estudio y análisis de la locomoción bípeda humana, el desarrollo de la robótica bípeda no ha cesado y ha motivado a la definición de nuevos conceptos y a la generación de nuevos métodos para solucionar los problemas de diseño, estabilidad y control de robots bípedos, donde han emergido nuevos desafíos; algunos, incluso, imposibles de resolver usando métodos tradicionales, impulsando la adopción de enfoques multidisciplinarios, que abarcan tanto campos de la ingeniería —que van desde la robótica, el control y la electrónica, hasta la inteligencia artificial y computación gráfica— como campos de otras áreas, entre ellas la biología, la psicología y las neurociencias [52].

Mientras las primeras investigaciones en la robótica bípeda y humanoide se enfocaban en la generación de ciclos de marcha que guardaran una estabilidad estática, en la actualidad los temas se han ampliado con el propósito de emplear a los humanoides como robots de servicio, que, en un futuro, transformarán la vida de las personas: realizarán una gran cantidad de tareas en beneficio de la humanidad y ayudarán a resolver problemas sociales y de salud: cuidarán de personas de edad avanzada, ayudarán y asistirán a discapacitados, vigilarán y patrullarán zonas públicas y, además, colaborarán en actividades de la vida cotidiana ([52] y [63]).

Se espera que los desafíos que implica el desarrollo de la robótica humanoide estimulen el progreso tecnológico: la necesidad de los robots humanoides por contar con sistemas mecánicos lo suficientemente sofisticados como para desenvolverse en entornos humanos, incluso interactuar con ellos y emplear sus mismas herramientas; con algoritmos avanzados para la generación de movimientos ver-

satélites y sincronizados; con sistemas inteligentes y esquemas de control robustos que actúen frente a perturbaciones y situaciones imprevistas, y con sistemas artificiales que simulen el comportamiento del cerebro humano, producirá un importante avance científico y tecnológico y ampliará el conocimiento que tenemos de nosotros mismos ([52] y [63]).

La robótica humanoide, al igual que la industria automotriz y de las computadoras, vista como tal, tendrá en el futuro un efecto multiplicador, generando industrias proveedoras y consumidoras que impulsarán el desarrollo de los países que inviertan y se coloquen a la vanguardia de ella [63].

El desarrollo de robots bípedos y humanoides tiene grandes beneficios y consecuencias: estimula la investigación científica e impulsa el progreso tecnológico, motiva la sinergia entre diversos campos de la ingeniería y de otras ciencias, propone alternativas a problemas sociales y de salud. Los robots humanoides, por la capacidad de adaptarse a entornos humanos y de emplear sus mismas herramientas, por la capacidad de interacción y convivencia con otros robots y con los seres humanos, además de sus habilidades y destrezas motoras, no solo representan una área de investigación, sino una nueva industria que invierte en su desarrollo y promueve su comercialización, tanto de robots de servicio como de entretenimiento.

1.2. Objetivos

1. Diseñar una interfaz gráfica de usuario que facilite la interacción con el robot bípedo Scout y funja, junto con el prototipo físico, como banco de pruebas. Sus funciones serán:
 - a) El accionamiento del robot a partir de la incidencia directa sobre sus articulaciones.
 - b) El accionamiento del robot a partir de trayectorias generadas para sus juntas rotacionales, de manera que se puedan implementar ciclos de marcha.
 - c) La visualización de un modelo virtual en el que se muestre la disposición espacial del robot y sobre el cual se puedan realizar las dos acciones anteriores.
2. Llevar a cabo la planificación de la marcha, restringida al desplazamiento en el plano sagital y sobre una superficie plana horizontal, en sus etapas de puesta en marcha, fase cíclica y paro del robot. La formulación de las funciones de la juntas rotacionales del robot para la generación de ciclos de marcha, se llevará a cabo a partir de las trayectorias de sus efectores finales, definidas por funciones polinómicas cúbicas.
3. Optimizar la marcha del robot en base al criterio de estabilidad del ZMP. Se desarrollará un algoritmo genético que determinará la trayectoria óptima de la cadera dadas las trayectorias que definen la posición espacial y orientación de ambos pies durante un ciclo de marcha.

1.3. Trabajo previo

Previo al trabajo de esta tesis se desarrollaron los modelos de cinemática y dinámica espacial para el robot bípedo de 12 GDL internos Scout [43]. Además, se realizó un programa en Mathematica para el cálculo de la cinemática inversa y la simulación de un ciclo de marcha [43].

Aunque hay otras alternativas, asumiendo que se tiene un robot bípedo ya construido, el proceso para que desarrolle un ciclo de marcha dinámicamente estable puede dividirse en tres partes:

1. Modelado cinemático y dinámico del robot. Dependiendo de la técnica a usar en las siguientes partes es necesario o no el modelo dinámico.
2. Generación de patrones de caminata. El objetivo de los robots bípedos es desplazarse, lo cual se logra —dada su configuración mecánica— mediante el desarrollo de trayectorias para cada una de las juntas articulares del robot, que, como resultado de sus movimientos sincronizados, producen ciclos de marcha. Dado que los robots bípedos son dinámicamente inestables, los ciclos de marcha se generan obedeciendo criterios de estabilidad, siendo el ZMP el más común.
3. Diseño e implementación de un controlador para estabilizar al bípedo ante perturbaciones. Debido a que los ciclos de marcha son generados empleando un modelo ideal, y considerando un entorno con condiciones también ideales, es necesario contar con algún control que modifique las trayectorias de las juntas rotacionales del robot ante fallas por las idealizaciones en el modelo o en el entorno y que estabilice al robot bípedo ante perturbaciones internas o externas.

Mientras uno de los propósitos de la interfaz gráfica es ayudar a probar un ciclo de marcha en el prototipo físico (descrito en [43]), con la planificación y optimización de la marcha —dado el modelo cinemático y el programa en Mathematica para su cálculo— se pretende continuar y profundizar la investigación en torno a la robótica bípeda empleando al robot bípedo Scout.

1.4. Organización y contenido de la tesis

El segundo capítulo de esta tesis ofrece una breve exploración de la robótica bípeda; incluye una revisión histórica a través de los robots bípedos más significativos y un recorrido por el panorama actual, con la selección de algunos humanoides que permiten analizar el presente y dar una idea del futuro de la robótica bípeda. Además, se describen algunos conceptos básicos que serán empleados en capítulos siguientes. También se ofrece una breve reseña sobre los temas de investigación actuales en la robótica bípeda y se proporciona una noción del ZMP.

El tercer capítulo se adentra en el estudio de la locomoción bípeda: ofrece un análisis descriptivo basado en los objetivos y fases de la marcha humana y, además, describe tanto las funciones del sistema pasajero y locomotor como la influencia del sistema nervioso y muscular durante la marcha.

El cuarto capítulo ofrece una descripción del banco de pruebas. La primera subsección proporciona información general sobre el prototipo de pruebas y define la arquitectura del robot en base a la nomenclatura empleada en [43]. La segunda subsección presenta la interfaz gráfica de usuario desarrollada para permitir la interacción con el robot bípedo Scout y muestra algunos resultados en los que se observa su utilidad y funcionamiento.

El quinto capítulo trata sobre la planificación de la caminata. Ofrece algunos modelos y métodos empleados para la generación de patrones de marcha y describe punto por punto el método empleado. Al final se presentan los resultados e imágenes de la simulación de un ciclo de marcha generado.

El sexto capítulo profundiza en la optimización de la marcha. Ofrece una breve reseña de los algoritmos genéticos: sus bases biológicas, su funcionamiento y sus operadores, y describe el método, que mediante el uso de algoritmos genéticos y la estrategia empleada para la síntesis de marcha en el capítulo anterior, proporciona la trayectoria óptima de la cadera dadas las trayectorias de los pies según un ciclo de marcha, siendo el ZMP el criterio para la optimización.

El séptimo capítulos se dan a conocer los experimentos y las simulaciones que se realizaron en torno a la planificación y optimización de la marcha.

El octavo capítulo da a conocer las conclusiones del trabajo y propone algunas direcciones futuras de la investigación.

Capítulo 2

La robótica bípeda

2.1. Panorama histórico

2.1.1. Las primeras investigaciones de robots bípedos en universidades y laboratorios

En Japón, alrededor de la década de 1970, creció notablemente el interés por una nueva clase de robots: los robots antropomórficos. Los investigadores comenzaron su desarrollo inspirados por algunos personajes de ciencia ficción, como Astroboy, de la caricatura homónima, o Marius, de la obra de teatro *Robots universales de Rossum*. A diferencia de los manipuladores robóticos, estos personajes guardaban una apariencia física similar a la de los seres humanos. Y no solo eso, también eran capaces de aprender, discernir y socializar, e incluso tenían sentimientos y modelos de conducta. Tales características provocaron una gran fascinación, llena de preguntas e inquietudes, por los humanoides: ¿En verdad sería posible construir robots con capacidades físicas e intelectuales semejantes a las del ser humano? ¿Qué herramientas y tecnologías se necesitarían? ¿Cómo sería convivir con ellos? Persuadidos por las ventajas y aplicaciones que tendrían esta clase de robots, y con el propósito de un día hacer realidad aquellos individuos que solo eran concebibles en obras de ciencia ficción, varios científicos decidieron enfocar sus investigaciones en lo que hoy resultan las bases de la robótica humanoide y que en su tiempo dieron origen a los primeros robots antropomórficos.

2.1.1.1. El proyecto WABOT

En 1970, con el objetivo de construir robots similares a los seres humanos, y que auxiliaran en la medicina, enfermería y cuidado diario, cuatro laboratorios de la Universidad de Waseda se unieron para fundar el grupo de Bioingeniería (*The Bio-engineering group*), y juntos emprendieron el proyecto WABOT (abreviación de *Waseda Robot*), en donde forjaron una prolífica familia de robots bípedos y humanoides [20].

Uno de los primeros robots desarrollados como parte del proyecto WABOT fue el WL-5 (figura 2.1), un bípedo de 11 grados de libertad —5 en cada pierna y uno en la cadera— que era capaz de caminar con estabilidad estática al mover su centro de gravedad en el plano frontal; tardaba 45 segundos en dar un paso, y sus movimientos estaban controlados por una minicomputadora [19]. En 1973, con la experiencia adquirida en el diseño y construcción de extremidades robóticas, como pies y manos,

comenzó —bajo la dirección del profesor Ichiro Kato— el desarrollo del primer robot antropomórfico a gran escala. El WABOT-1 (figura 2.1), también considerado como el primer robot con inteligencia artificial, tenía la capacidad de comunicarse con los seres humanos, de calcular distancias y de agarrar y transportar objetos; se consideraba que tenía la facultad mental de un niño de un año y medio. Cabe destacar que para la elaboración de sus extremidades inferiores se empleó la configuración del bípedo WL-5 [20].

Otra de las grandes contribuciones del proyecto WABOT de la Universidad de Waseda a la robótica humanoide fue el lograr reproducir el primer ciclo de marcha dinámicamente estable con el robot bípedo WL-10RD (figura 2.1) en el año de 1984 [59]. Éste podía caminar sobre una superficie plana o, bien, sobre terrenos irregulares, como superficies inclinadas o escaleras. Además, contaba con sensores de torque en las juntas de los tobillos y en la cadera para controlar las transiciones dadas por los cambios en el pie de apoyo. Sobre una superficie plana era capaz de caminar a 1.3 segundos por paso; en cambio, sobre una superficie irregular caminaba a 2.5 segundos por paso [19]. El método empleado para asegurar que su caminata fuera dinámicamente estable se sustentó en el concepto del ZMP, que fue propuesto por Miomir Vukobratovic a inicios de la década de 1970. En la actualidad, el concepto del ZMP sigue vigente y es uno de los más empleados para la generación de patrones de caminata con estabilidad dinámica en robots bípedos y humanoideos.

Posterior al logro de desarrollar una caminata dinámicamente estable, surgió la familia WL-12, una serie de robots bípedos con actuadores hidráulicos, cuyo propósito era incrementar la estabilidad y reproducir una caminata semejante a la marcha humana. Todos los robots de esta familia se caracterizaron por contar con un tronco de 2 GDL que asistía en la estabilización de la marcha. El primer robot de la serie, el WL-12 (figura 2.1), era capaz de evadir obstáculos y de adaptar su trayectoria a terrenos irregulares [35]. Se desplazaba a 2.6 segundos por paso, y empleaba un algoritmo que generaba la trayectoria del cuerpo en función de las trayectorias de las extremidades inferiores y la trayectoria del ZMP. En 1989, con el bípedo WL-12III, se probó exitosamente un algoritmo capaz de estabilizar al robot ante perturbaciones externas. El algoritmo determinaba la posición de aterrizaje del pie flotante dada la trayectoria del torso, que era generada en función de las fuerzas externas y momentos.

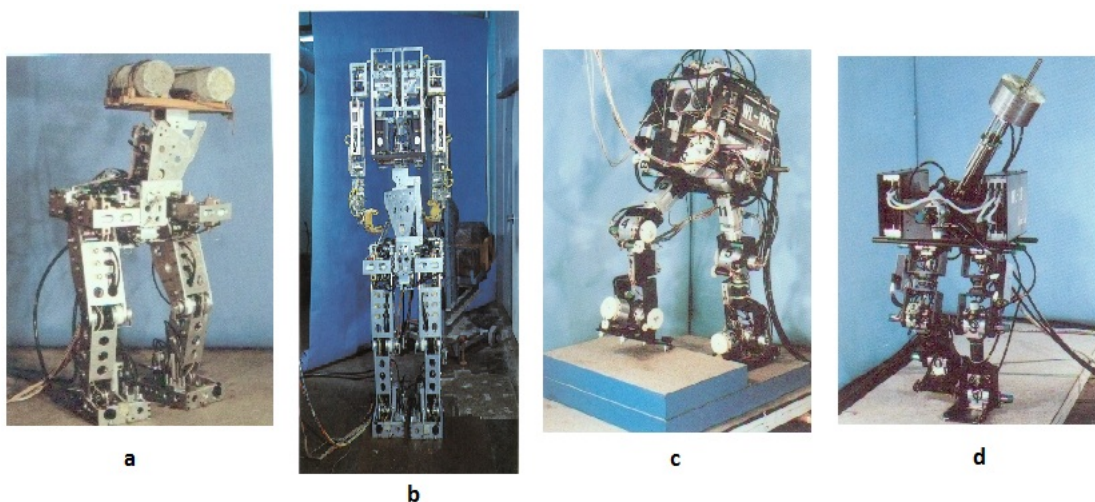


Figura 2.1: Robots construidos como parte del proyecto WABOT. a) WL-5, b) WABOT-1, c) WL-10RD, d) WL-12.

2.1.1.2. Robots gimnásticos del LegLab del MIT

Mientras en Japón empleaban el criterio del ZMP con el propósito de generar ciclos de marcha con la velocidad y armonía características de la locomoción bípeda humana, en el *Leg Laboratory* del MIT construyeron, en la década de 1980, robots con una, dos y cuatro piernas, que en vez de caminar, corrían o saltaban. Los primeros robots que construyeron eran de una sola pierna, pues consideraban que de esta forma se simplificaba su análisis y que el conocimiento adquirido en el estudio de su comportamiento y en el diseño de sus controladores podría ser empleado como base para el análisis de robots con más piernas. Utilizando los robots Planar One-Leg Hooper (figura 2.2) [34] y 3D One-Leg Hooper (figura 2.2) [32], los investigadores del *LegLab* demostraron que era posible controlar el salto de mecanismos de una sola pierna con un conjunto simple de algoritmos; el sistema de control estaba dividido en tres partes: la primera controlaba el salto; la segunda, la velocidad y aceleración, y la tercera, la postura de la estructura [46].

Dados los resultados positivos en el problema de locomoción básico (robots con una sola pierna), el siguiente reto consistió en aplicar los mismos conceptos de equilibrio, pero ahora para un robot bípedo. Para esto se analizó a la estructura como un sistema periódico donde la función de la pierna se va alternando, y donde solo una puede estar activa a la vez; es decir, en un instante solo una pierna (pierna activa) está tocando tierra, está impulsándose y está ejerciendo un torque sobre la estructura del robot. Asumiendo que la otra pierna no interfiere en la marcha, es posible emplear el algoritmo diseñado para robots con una sola pierna en robots bípedos. Esta idea se empleó para hacer caminar al robot Planar Biped (figura 2.2) [33]. Es posible emplear el mismo enfoque para controlar robots con cualquier número de piernas, siempre y cuando solo una pierna esté activa. El problema es que los cuadrúpedos y los robots con aun más piernas suelen tener más de una pierna en estado activo. Por lo cual, para aplicar el mismo algoritmo, empleado en robots con una y dos piernas, en robots cuadrúpedos, se propuso un método de control que coordina grupos de piernas que están activas simultáneamente, y cuyo comportamiento es similar al de una sola pierna [46].

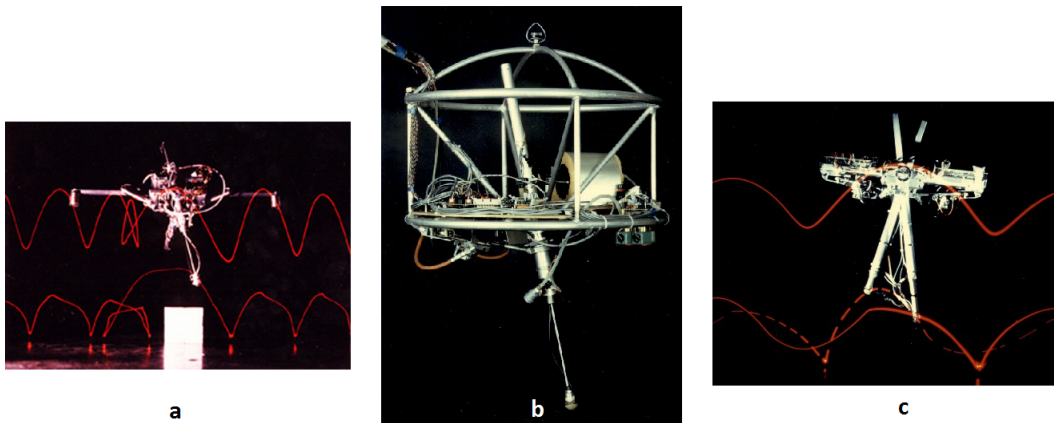


Figura 2.2: Robots construidos en el MIT. a) Planar One-Leg Hooper. b) 3D One Leg-Hooper. c) Planar Biped.

2.1.2. Desarrollo de robots bípedos humanoides y la exploración de sus aplicaciones

2.1.2.1. Los humanoides de Honda

En 1986 Honda se propuso el desarrollo de un humanoide que pudiera vivir en armonía con los humanos; desde entonces han producido una gran cantidad de robots bípedos y humanoides. Su primer objetivo consistió en el desarrollo de un robot capaz de caminar, de subir y bajar escaleras y de adaptarse a diferentes ambientes. Así nació la serie de bípedos E. El primero, el E0 (figura 2.3), caminaba con estabilidad estática y tomaba 30 segundos para dar un solo paso. En los siguientes modelos, E1, E2 y E3 (figura 2.3), se mejoró la velocidad de la caminata al emplear algoritmos que simulaban la marcha humana [16]. El E2 fue el primer robot de la compañía con estabilidad dinámica; podía caminar sobre una superficie plana o subir y bajar escaleras. En los bípedos E4, E5 y E6 (figura 2.3) se trabajó con la estabilización de la caminata, para lo cual emplearon el criterio del ZMP [2]. El último bípedo de la serie, el E6, fue terminado en 1993, y en él se integraron todas las funciones de caminata que se habían implementado en los demás robots de la serie. Después de la serie E, siguió la serie P (figura 2.3), que, durante los años de 1993 y 1997, dio origen a tres diferentes prototipos de humanoides. El P2, con una altura de $182 [cm]$ y un peso de $210 [Kg]$, era capaz de compensar las fuerzas externas, de evitar colisiones y de caminar sobre superficies irregulares [16]. Contaba con cámaras, acelerómetros, giroscopios y sensores de fuerza en los tobillos y muñecas. Las mejoras más significativas del último robot de la serie, el P3, con respecto al P2 fueron su tamaño ($160 [cm]$) y peso ($130 [Kg]$). El desempeño de estos humanoides sorprendió a la comunidad científica, y más tarde se supo que su éxito se debía a una nueva idea donde se usaban demostraciones de ciclos de marcha realizados por humanos para generar las trayectorias de referencia que debía seguir el robot [15]. El método consistía en grabar patrones de caminata, optimizarlos y reproducirlos en un robot, que se mantenía estable de acuerdo a una ley de control que tomaba como entradas la dinámica del movimiento del robot y los torques que se generaban en sus tobillos.



Figura 2.3: Bípedos y humanoides desarrollados por Honda

Después de la serie P, en el año 2000, se construyó el primer prototipo de ASIMO (acrónimo de *Advanced Step in Innovative Mobility*), un robot cuya configuración y diseño fueron pensadas para que se desarrollara sin problemas en un ambiente habitado por humanos; por ejemplo, el alcance de sus manos se eligió en base a un análisis que proponía, respectivamente, la altura y longitud óptimas de los hombros y brazos del robot para que pudiese auxiliar en tareas de la vida diaria, como abrir y cerrar una puerta o encender y apagar un foco. ASIMO (figura 2.4) puede caminar a $0.75 [m/s]$ y correr con una rapidez de $1.67 [m/s]$. A lo largo de los años, se han desarrollado nuevos prototipos de ASIMO cada vez con diseños más sofisticados, funciones más complejas y con mayor flexibilidad y estabilidad

en la caminata. La capacidad de predecir los movimientos del controlador del modelo más actual le permiten al humanoide cambiar suavemente su patrón de caminata, sin necesidad de hacer paros o cambios bruscos en su trayectoria. El método consiste en equilibrar las fuerzas centrífugas cambiando la posición del centro de gravedad del robot, para lo cual se hacen cálculos predictivos antes de que el bípedo modifique su trayectoria. Los cálculos determinan en cada instante el cambio óptimo de la posición del centro de gravedad mientras se genera el nuevo patrón de marcha en tiempo real [3].



Figura 2.4: Honda ASIMO descendiendo escaleras.

2.1.2.2. Avances en la Universidad de Waseda posteriores al proyecto WABOT

En 1992, para dar continuidad al proyecto WABOT, inicia el Proyecto Humanoide, con el fin de desarrollar humanoides con usos prácticos, que pudieran salir del laboratorio; para lo cual, los robots tenían que ser inteligentes: comunicarse y expresarse, comprender los deseos de los humanos y desenvolverse con naturalidad. Bajo estas consideraciones, se desarrolló el robot WABIAN, un humanoide tamaño humano capaz de caminar con el ritmo promedio de un humano, empleando motores eléctricos. Así fue surgiendo una nueva serie de robots con mejores prestaciones; por ejemplo, el WABIAN-R fue creado para explorar la interacción del robot con su medio ambiente, y en el WABIAN-RII se introdujeron diferentes movimientos corporales para transmitir emociones. El último humanoide de la serie, terminado en el 2006, es el WABIAN-2R (figura 2.5). Tiene una altura de $150 [cm]$ y un peso de $64 [Kg]$. A diferencia de la mayoría de robots, que caminan con las rodillas flexionadas, éste camina estirándolas, de la misma manera que lo hace un humano [31].

A pesar del interés por construir humanoides, el desarrollo de robots bípedos cada vez mejores no ha cesado. Por ejemplo, los bípedos W-15 y W-16, construidos a partir del año 2000, fueron diseñados empleando un mecanismo paralelo que elimina los problemas de rigidez, potencia y precisión de los mecanismos seriales. El robot bípedo W-16 fue el primero en ser capaz de cargar a una persona con un peso de hasta $60 [Kg]$ [36].

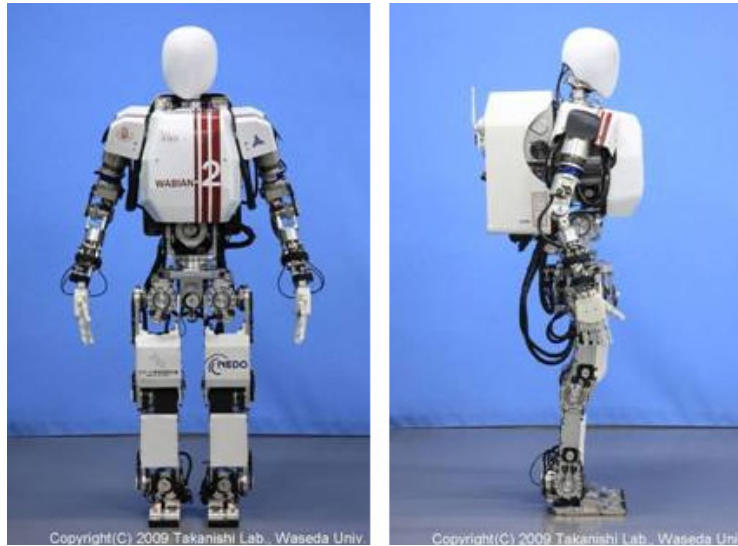


Figura 2.5: Humanoide WABIAN-2R de la Universidad de Waseda.

2.1.2.3. El proyecto de humanoides HRP

Entre 1998 y 2002 el Ministerio de Economía, Negocio e Industria de Japón promovió el proyecto HRP (por sus siglas en inglés *Humanoid Robotics Project*). Sus objetivos eran el desarrollo de una plataforma de robots humanoides y la exploración de las aplicaciones de los humanoides. Su primer robot, HRP-1, desarrollado por Honda, ha sido empleado en tareas de mantenimiento de plantas industriales y en el cuidado del hogar y la oficina [17]. Tenía un peso de 120 [Kg] y una estatura de 160 [cm]. Debido a que el controlador de Honda impedía el control simultáneo de piernas y brazos, fue reemplazado por uno con mayor flexibilidad. La combinación del robot HRP-1 y del nuevo controlador dio origen al humanoide HRP-1S; con éste se exploró el cuidado de pacientes en cama. Con el HRP-2, un humanoide mucho más ligero (58 [Kg]) y con una estatura de 154 [cm], continuaron investigando las aplicaciones de los humanoides, esta vez, en trabajos colaborativos con humanos [17].

2.2. Panorama actual

Gracias al impacto causado por humanoides como ASIMO y WABIAN, en la actualidad, el interés por la robótica bípida se ha expandido alrededor del mundo. Aunque con diferentes enfoques y alcances, diversas empresas privadas, laboratorios financiados por instituciones gubernamentales y universidades de Europa, Asia y América han pasado, de ser espectadores, a participar activamente en el desarrollo y análisis de robots bípedos y humanoides.

2.2.1. Proyecto de Robots de Compañía de Toyota

En el 2001 Toyota, una empresa japonesa, estableció su Proyecto de Robots de Compañía (*Partner Robot Project*) con el propósito de construir robots que pudieran ayudar a las personas y convivir en armonía con ellas. Esta familia cuenta con tres categorías: humanoides bípedos, humanoides con llantas y bípedos montables; todos estos tienen características en común; están diseñados para transmitir afecto y amabilidad y para ser capaces de manipular herramientas que les permitan asistir en el hogar



Figura 2.6: El robot humanoide HRP-2. Puede caminar sobre superficies irregulares o sobre caminos estrechos.

o realizar trabajos de manufactura [56]. Dentro de la categoría de humanoides caminantes, hay dos modelos (figura 2.7); uno toca la trompeta, y otro, el violín. Ambos son de tamaño humano, y el más reciente, el que toca el violín, fue dado a conocer en el 2007.

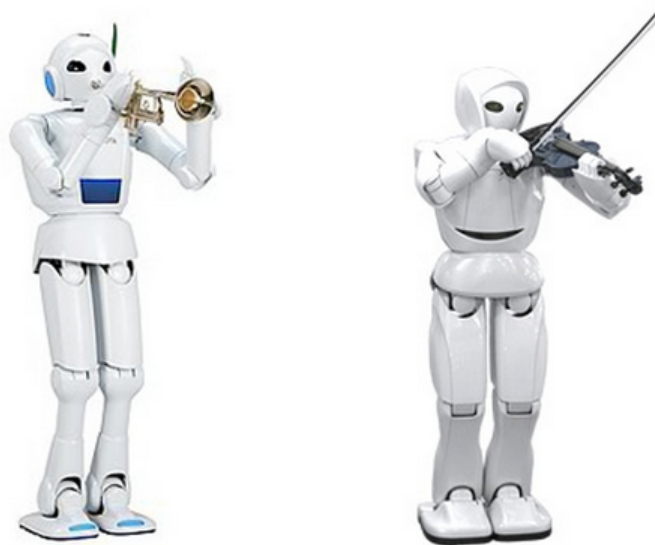


Figura 2.7: Humanoides de la familia de robots de compañía de Toyota.

2.2.2. Los humanoides MAHRU y AHRA

En el año de 2004 inició el desarrollo de los humanoides MAHRU y AHRA, un proyecto cooperativo entre Samsung y el Instituto de Ciencia y Tecnología de Corea. MAHRU es una humanoide de 150 [cm] de altura y 67 [Kg] de peso; puede caminar a 0.9 [Km/h] hacia adelante, atrás o hacia los lados. Además,

tiene un sistema de reconocimiento de voz y es capaz de manipular objetos.

La característica más peculiar de estos humanoides, y que los distingue de otros, es que son teleoperados, por lo cual el CPU principal de la unidad puede estar fuera del entorno en el que se desenvuelve el robot [41]. Para lograrlo, el humanoide emplea sensores que le ayudan a determinar su estado y el del ambiente que lo rodea; después, a través de una red inalámbrica, envía esta información a un servidor externo que procesa todos los datos y determina los comandos a ser ejecutados por el robot; finalmente, el servidor se vuelve a comunicar con el robot y éste ejecuta las acciones que le son indicadas.

2.2.3. Surena II

Surena II (figura 2.8) es el segundo humanoide manufacturado por la Universidad de Tehran de Irán con la cooperación del Ministerio de Industrias y Minas del mismo país, y terminó de ser construido en julio del 2010. Tiene una estatura de 145 [cm] y un peso de 45 [Kg] y puede ser operado con un control remoto. A pesar de que su desempeño es notablemente inferior al de humanoides como ASIMO, el progreso del proyecto ha sido rápido; por ejemplo, tan solo en los primeros dos años, construyeron dos humanoides —el primero con llantas, Surena I— y lograron reproducir un ciclo de marcha dinámicamente estable en su segundo humanoide, Surena II. El objetivo de los integrantes del proyecto para el prototipo de tercera generación, aún en desarrollo, es mejorar el comportamiento y añadir nuevas funciones al robot para que pueda ser empleado en actividades industriales o en el cuidado del hogar [13].



Figura 2.8: Surena II, un humanoide desarrollado con el propósito de explorar aspectos teóricos y prácticos de la caminata bípeda.

2.2.4. PAL Robotics

PAL Robotics es una empresa que se dedica a la investigación y desarrollo, junto con su comercialización, de robots humanoides. El origen de esta empresa, ubicada en Barcelona, data del año 2004, cuando inició el desarrollo de su primer humanoide, el REEM-A (figura 2.9), capaz de caminar y jugar ajedrez, el cual fue terminado en tan solo un año. Dos años después, fue presentado su segundo,

y más reciente, humanoide bípedo. Tiene una altura de 147 [cm] y un peso de 60 [Kg]. Cuenta con un sistema de cámaras y sensores que le permiten desenvolverse con autonomía e, incluso, caminar evitando obstáculos a 1.5 [Km/h] [48].

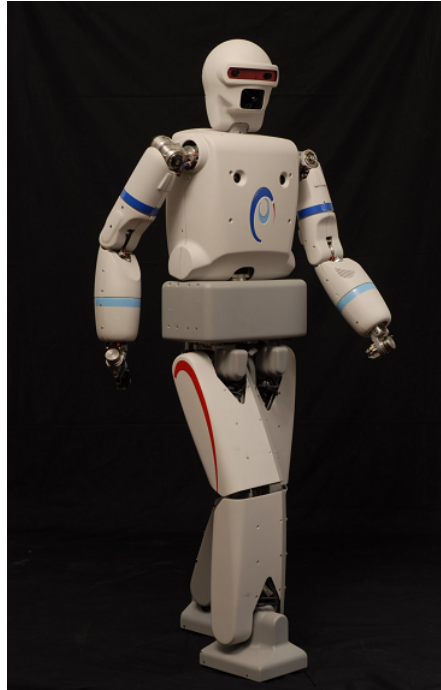


Figura 2.9: El bípedo REEM-A de PAL Robotics. Participó en la RoboCup del 2006, donde la competencia de caminata.

2.2.5. Concursos y torneos

Los concursos de robótica se han convertido en eventos comunes y habituales alrededor del mundo, donde diferentes organizaciones y universidades tienen la oportunidad de competir y ganar; pero también, de compartir y difundir sus más recientes investigaciones. Uno de los torneos más populares es la RoboCup, una iniciativa internacional cuyo objetivo es promover el desarrollo de robots inteligentes. Actualmente, cuenta con concursos divididos en las categorías de fútbol, robots de rescate y robots para el hogar. Fundada en 1997, con el propósito de construir, para el 2050, un equipo de humanoides capaces de vencer a los ganadores de la Copa Mundial de Fútbol, la RoboCup ha impulsado y popularizado la investigación en la robótica bípeda y humanoide [47]. La variedad de categorías y ligas que maneja, permiten participar a universidades de diferentes partes del mundo y con diferentes presupuestos y alcances. La próxima RoboCup, a organizarse en julio del 2012, tendrá como sede a la Ciudad de México.

2.2.6. La robótica bípeda en México

En cuanto a México, en el área de la robótica bípeda, se puede mencionar que, aunque los avances y la inversión están por debajo de los correspondientes de los líderes internacionales en el área, sí existe un interés en diversas universidades y demás centros de investigación por colaborar en el desarrollo

de robots bípedos. Estudiantes e investigadores de instituciones como el IPN o el ITESM y de centros como el cenidet, han publicado sus resultados y logros en los temas de diseño, construcción y análisis de robots bípedos, todos en prototipos a escala. Por ejemplo, un grupo de estudiantes de UPIITA creó a Nicanor, un robot bípedo experimental que simula la marcha y el equilibrio del ser humano; además, pretenden que sirva como base para el diseño de un exoesqueleto; en el ITAM y en ITESM han comprado robots humanoides NAO para ser programados por sus alumnos y participar en competencias internacionales. Por otra parte, en el Laboratorio de Biorrobótica de la UNAM, se han adquirido robots bípedos para formar un equipo de fútbol de humanoides, los cuales son operados mediante un software, ViRbot, desarrollado por alumnos y profesores del laboratorio; también han construido y programado un robot de servicio, AL-ITA, que, empleando conceptos de inteligencia artificial, es capaz de entender el lenguaje natural y de recordar y reconocer rostros. AL-ITA participó en el último torneo de la RoboCup, realizado en Estambul.

2.3. Marco conceptual

Los robots bípedos, aunque varios en su tipo, son aquellos cuyo aparato locomotor consiste, básicamente, en dos piernas. Su configuración mecánica está formada por una cadena cinemática abierta constituida por dos subcadenas idénticas, piernas, que se conectan a través de un eslabón común, la cadera; algunos constan de un elemento más, el torso, cuyos movimientos coordinados con las piernas permiten incrementar la estabilidad del robot durante la marcha. Los humanoides con piernas pertenecen a la categoría de robots bípedos, y su estructura parte de un símil con la configuración anatómica humana; por tanto, pueden emular algunos gestos y aspectos físicos del ser humano. En muchos casos, los robots humanoides representan la evolución de investigaciones que iniciaron con el desarrollo de robots únicamente con extremidades inferiores. Por otra parte, en varios artículos, se emplean humanoides para discutir temas de robótica bípeda, como estabilidad, control o generación de trayectorias. Debido a ello, en la literatura, es común encontrar que a los humanoides se les nombre como robots bípedos o, simplemente, bípedos.

2.3.1. La caminata pasiva y energizada

Las investigaciones en la robótica bípeda se pueden dividir en dos categorías: el análisis de la caminata pasiva —donde la caminata se da como efecto de la acción gravitatoria sobre el cuerpo del robot— y el análisis y control de la caminata energizada, que requiere de una fuente de energía externa [61].

En la **caminata pasiva**, un mecanismo, sin actuadores, se desplaza sobre una superficie inclinada; de esta forma, se aprovecha la energía potencial para compensar la pérdida de energía debido a los impactos con la superficie. La virtud de estos robots es que no necesitan de energía externa ni de ningún tipo de control para caminar [61]. El concepto de locomoción pasiva fue introducido por McGeer en 1988, quien construyó un bípedo bidimensional (figura 2.10) que era propulsado por la gravedad. El modelo actuaba como dos péndulos acoplados, donde mientras uno ejercía la función de pierna de soporte actuando como un péndulo invertido, el otro consistía en un péndulo normal que desarrollaba la función de pierna oscilante. Más recientemente, grupos de investigación en caminata pasiva de la Universidad de Cornell, la Universidad Tecnológica de Delf y del MIT, han extendido los resultados de McGeer para la creación de robots con la capacidad de caminar sobre superficies horizontales, y,

al igual que el modelo de McGeer, reproducen una locomoción pasiva gobernada por la dinámica del mecanismo. Estos robots se caracterizan por constar con sensores de contacto, con una cantidad mínima de actuadores que son activados mediante señales *on/off* y con un sistema de control primitivo. Debido a que estos prototipos están basados en caminantes pasivos y emplean actuadores para reemplazar a la gravedad como fuente de energía, se les denomina híbridos.

El interés por la caminata pasiva se debe, principalmente, a la similitud con la marcha humana, pues tienen eficiencias y movimientos parecidos a los humanos.

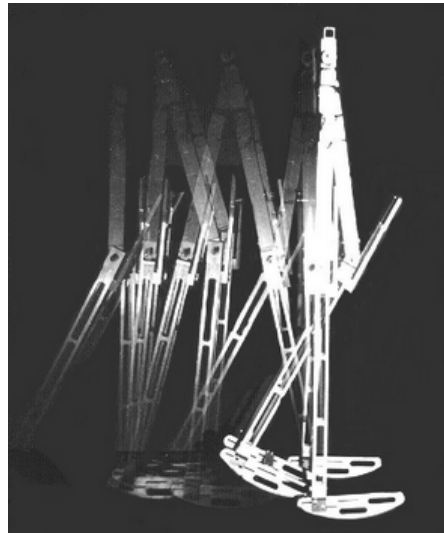


Figura 2.10: Bípido bidimensional de McGeer. Duplicado construido en la Universidad de Cornell.

En la **caminata energizada**, el robot —relativamente, con un número grande de actuadores— logra desplazarse al ser controlada la posición de los actuadores de cada uno de sus eslabones en cada instante, generando así una serie de movimientos coordinados entre sus articulaciones, que están determinados tanto por la zona en la que se desea posicionar al pie flotante, como por algún criterio que asegure su estabilidad durante la marcha, por lo general, el ZMP [61]. Estos robots, a comparación de los bípedos pasivos, tienen un alto grado de complejidad mecánica: se caracterizan por ser antropomórficos —por lo general, con 6 GDL o más—, por constar de una gran cantidad de actuadores y por tener una estructura rígida. Debido a ello, su confiabilidad, eficiencia y robustez mecánica es menor. Por otra parte, son estas mismas características las que les permiten modificar su patrón de caminata y ajustar ciertas variables en su movimiento para mantener su balance, lo cual los vuelve viables para un sinnúmero de aplicaciones. En la actualidad, la mayoría de prototipos pertenecen a esta categoría; sin embargo, debido a su complejidad matemática y a la necesidad de un conocimiento profundo de su dinámica, temas como el desarrollo de controladores para bípedos no se han profundizado al mismo nivel que el diseño y construcción de estos prototipos. Ejemplos de robots energizados son los desarrollados en la Universidad de Waseda o los humanoides de Honda, entre ellos, ASIMO. A esta categoría también pertenecen los humanoides miniatura QRIO de Sony y NAO de Alderaban Robotics (figura 2.11).

2.3.2. Arquitecturas mecánicas de los robots bípedos

En la robótica bípida, es posible establecer una distinción entre los diferentes tipos de prototipos de acuerdo a su arquitectura mecánica [51]; dentro de esta clasificación, existen las siguientes categorías:



Figura 2.11: A la izquierda, QRIO, un humanoide de entretenimiento que fue cancelado en el 2006. A la derecha, NAO, el reemplazo de AIBO como plataforma estandar en la RoboCup.

“Static” walkers. Su proyección del centro de masa siempre se mantiene dentro del conjunto convexo formado por los puntos de soporte del robot con el plano de desplazamiento. Esta estrategia fue implementada por el profesor Kato y su grupo de investigadores en el robot WL-5, el primer robot caminante que se conoce. Debido a que en la actualidad existen métodos más sofisticados para la generación de patrones de marcha, las investigaciones en la caminata estática son pocas, y solo se justifican si representan un avance en alguno de los problemas actuales de la robótica bípeda; por ejemplo, en [8], desarrollan un algoritmo para que un bípodo con marcha estática adapte su caminata sobre una superficie con una pendiente desconocida; en [45], prueban una estrategia para economizar la energía empleada en los tobillos, aplicada a la caminata estática.

“Dynamic” walkers. Su estabilidad no está asegurada por la posición de la proyección del centro de masa, sino por otros criterios, como el ZMP, el Centro de Presión o el FRI (*Foot-Rotation-Indicator* por sus siglas en inglés).

“Purely dynamic” walkers. Son aquellos cuyos tobillos están inactuados o, bien, no tienen pies; en este caso, el polígono de contacto entre el robot y la superficie se reduce a un solo punto en la fase de soporte simple, y a un segmento de recta en la fase de soporte doble. A esta categoría pertenecen los robots pasivos activados por la fuerza de la gravedad, así como los robots híbridos basados en las investigaciones de McGeer. Se caracterizan por contar con pocos grados de libertad y por tener piernas telescópicas.

2.3.3. Otras clasificaciones

Los robots bípedos, además de ser clasificados por su arquitectura mecánica y por la forma en la que son energizados, también pueden ser clasificados:

- Por su tipo de movimiento en bidimensionales —cuyos eslabones solo se mueven dentro del plano sagital— y tridimensionales, cuya configuración mecánica les permite trasladarse en un espacio tridimensional.

- Por su número de grados de libertad en minimalistas, con menos de 6 GDL en sus piernas, y antropomórficos, con 6 o más grados de libertad en sus piernas.

2.3.4. Breve descripción de la marcha

La marcha es un modo de locomoción bípeda en la que las extremidades inferiores, con actividad alternada, ejecutan los mismos movimientos elementales. La marcha normal se caracteriza por el hecho de que el cuerpo jamás abandona totalmente el contacto con el suelo. El ciclo de marcha puede ser dividido en dos fases:

Fase de soporte simple. La fase de soporte simple se define como la fase de la locomoción donde solo un pie está en contacto con el piso, mientras que el otro se balancea en el aire. A la pierna que está en contacto se le denomina pierna de soporte, y a la otra, pierna flotante u oscilante.

Fase de soporte doble. La fase de soporte doble se da cuando ambos pies están en contacto con la superficie. En esta etapa hay una pierna anterior —sobre la que recae el peso del cuerpo— y otra posterior.

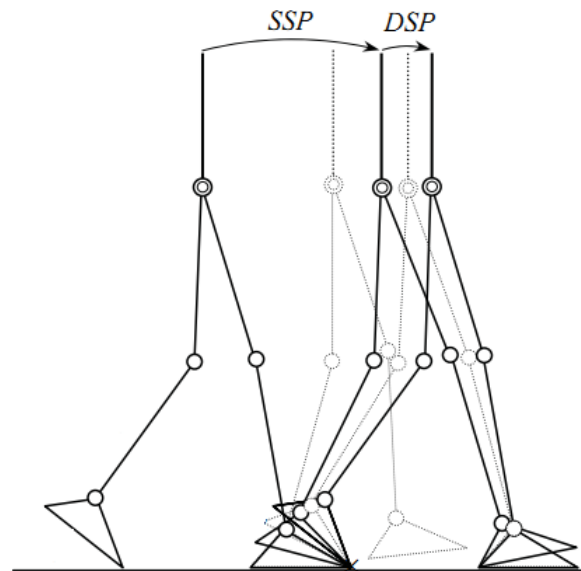


Figura 2.12: Secuencia de un paso. Cambio de fase de soporte simple a fase de soporte doble en un bípido bidimensional. CHEVALEREAU. Bipedal Robots, 2009.

Mientras que la fase de soporte simple constituye alrededor del 70 % de la marcha, la fase de soporte doble abarca aproximadamente el 30 % de ella [22]. Si bien la fase de soporte simple es parte del correr y del andar, el soporte doble desaparece al correr, que en cambio, se describe por la fase de apoyo —en la que el pie toca el suelo (funge como una plataforma estable para el movimiento del cuerpo) y se propulsa nuevamente— y la fase de oscilación, en la que el mismo pie no toca el suelo.

Desde el punto de vista dinámico, la marcha es una sucesión de impulsos y frenados, en los que el impulso se sitúa a nivel del miembro inferior posterior, y el frenado, en el miembro anterior. En la marcha humana, uno de los objetivos principales del aparato locomotor es el de ahorrar energía durante la caminata.

2.3.5. Sistema de referencia

Para la descripción y análisis de la marcha se establece un sistema de referencia (figura 2.13) que consta de tres planos y tres ejes respecto al cuerpo; cada plano o eje es perpendicular a cada uno de los otros dos.

Plano sagital. Es el plano longitudinal que divide al cuerpo en secciones izquierda y derecha. Pasa desde la parte anterior del cuerpo hasta la posterior, dividiendo a éste en dos mitades, izquierda y derecha.

Plano frontal. Es el plano paralelo al eje del cuerpo y perpendicular al plano sagital que separa al cuerpo en porciones frontal y posterior. Pasa desde un extremo lateral del cuerpo hasta el otro, dividiendo al cuerpo en dos mitades, anterior y posterior.

Plano transversal. Pasa horizontalmente al cuerpo, dividiéndolo en mitades superior e inferior.

Los ejes de movimiento representan aquella línea imaginaria alrededor de la cual se realiza el movimiento articular de un segmento corporal.

Eje frontal. Pasa horizontalmente de lado a lado. Se encuentra dispuesto en ángulo recto con el eje sagital.

Eje sagital. Se dirige horizontalmente desde el frente hacia atrás. Se encuentra dispuesto en ángulo recto con el eje frontal.

Eje vertical. Se ubica perpendicular al suelo. Se encuentra situado perpendicularmente a la línea de gravedad.

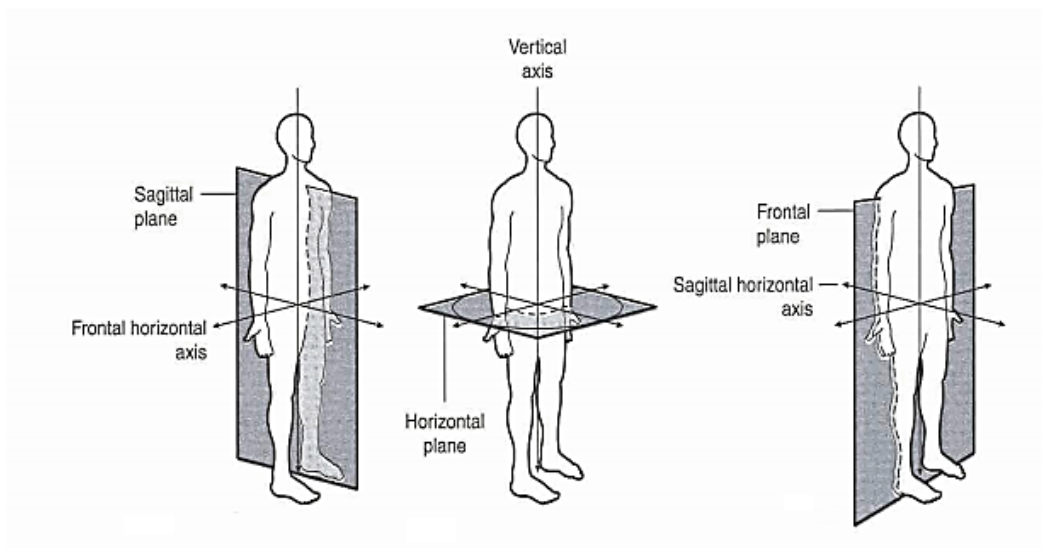


Figura 2.13: Planos del movimiento humano. BEHNKE. Kinetic Anatomy, 2000.

2.4. Robótica bípida: razones y objetivos

Si bien la robótica bípida es, por su complejidad, un estímulo para investigadores y alumnos; por su futura comercialización, una oportunidad de inversión para empresas y gobierno, y por su diseño y peculiar sistema locomotor, de interés para las personas en general, no se deben perder de vista ni las razones de su origen ni las metas y objetivos bajo los que se ha planteado su desarrollo.

Gran parte de los avances en la ingeniería se han dado por la necesidad de construir nuevos aparatos que representen un beneficio para la humanidad. Por ejemplo, en el área de la robótica, de inicio, los manipuladores robóticos se concibieron con el fin de poder manejar material radioactivo sin que representara un riesgo para el operador; más tarde, gracias al ritmo de su desarrollo y a lo funcional que resultó su configuración, su uso se extendió al sector industrial, militar y espacial. De manera similar, los robots bípedos también tienen sus porqués de su concepción, y aunque a día de hoy se han propuesto una gran variedad de aplicaciones, éstos aún están en etapa de investigación; sin embargo, se espera que en algunos años se incremente su comercialización y puedan emplearse en actividades que resulten en beneficio para los seres humanos, como hasta ahora lo han hecho los manipuladores robóticos, pero en tareas aun más diversas.

Aparte de la tendencia emocional y cultural que implica construir una máquina que se parezca y se desenvuelva como un ser humano, existen tres razones principales por las cuales son importantes los robots bípedos.

La primera razón es debido a su movilidad. Al carecer de un eslabón fijo a tierra, los robots bípedos pueden desplazarse de un lado a otro. A pesar de que los robots con llantas, como los *Ballbots*, también son capaces de alcanzar dicho objetivo, su funcionalidad se restringe a terrenos planos o pavimentados; en cambio, los robots bípedos tienen mayor flexibilidad y maniobrabilidad.

Para trasladarse, los robots bípedos emplean puntos de apoyo aislados y discontinuos; por su parte, la trayectoria de desplazamiento de un robot con llantas es continua. Por lo tanto, mientras que la ruta de un robot con llantas está condicionada por la topografía del camino, el movimiento de un bípedo está restringido por su configuración y por su espacio de trabajo. Además, las piernas proveen al cuerpo, o chasis, del robot, de un sistema de suspensión que le permite desacoplar su trayectoria de la ruta trazada por los pies del robot, por lo cual un robot bípedo puede evitar o, bien, disminuir las fuerzas de impacto o vibraciones que, inevitablemente, se producen en un robot con llantas de acuerdo a las particularidades de la superficie [46].

En teoría, los robots bípedos pueden desplazarse sobre terracería, subir y bajar escaleras o caminar sobre entre obstáculos sin tener que rodearlos; también (teóricamente) son capaces de acceder a zonas de difícil acceso, como arroyos, superficies agrietadas o terrenos fangosos. Aunque aún no existe un robot cuyo desempeño le permita realizar todas estas actividades, el progreso y los logros conseguidos desde los primeros robots bípedos motiva a pensar en que un día habrá bípedos con una gran destreza. Actualmente, Boston Dynamics está desarrollando el humanoide Atlas (figura 2.14) que, de acuerdo a sus diseñadores, será capaz de caminar sobre terrenos irregulares, gatear cuando sea necesario y caminar de lado para desplazarse sobre terrenos angostos.

Aunque los robots bípedos representan grandes ventajas ante los robots con llantas, su progreso y desarrollo depende de sub tareas que aún están siendo investigadas, como la generación de ciclos de marcha dinámicamente estables, el diseño de un sistema de control para sus articulaciones o el desarrollo de un algoritmo que determine la secuencia de movimientos de los pies en terrenos irregulares y desconocidos. Los robots bípedos implican grandes retos desde su concepción hasta su puesta en marcha, pues sus diseños deben ser robustos; es recomendable construirlos con materiales ligeros, que suelen ser costosos, y, por lo regular, se emplean técnicas de control avanzadas para probar su funcionamiento [46].

La segunda razón es porque el diseño, la construcción y el análisis de robots bípedos son útiles para comprender el sistema de locomoción humano [46]. A pesar de la naturalidad



Figura 2.14: Prototipo del bípedo Atlas. Según sus diseñadores, una vez terminado, será capaz de caminar sobre terrenos desiguales.

y destreza con la que empleamos nuestras piernas para desplazarnos, aún no se terminan por entender los principios que gobiernan la marcha humana. Las teorías y algoritmos originados a partir del estudio de robots bípedos pueden orientar las investigaciones biológicas en el tema, sugiriendo modelos para pruebas de experimentación y validación. Por ejemplo, se ha empleado un robot bípedo como banco de pruebas para analizar el comportamiento de una prótesis por debajo de la rodilla. El robot, que una de sus piernas ha sido reemplazada por la prótesis, reproduce un patrón de marcha humano, mientras se evalúan las características dinámicas de la prótesis, así como su influencia en el desempeño general de la marcha. Por su parte, Honda ha construido un exoesqueleto para personas con debilidad muscular en extremidades inferiores, basado en las investigaciones que se dieron en torno al desarrollo de ASIMO.

La última razón se debe al creciente interés por los robots de servicio. Se predice que estos robots transformarán en buena medida la vida de las personas en un futuro próximo, ya que realizarán una gran cantidad de tareas en beneficio de la humanidad; desde labores cotidianas, como cortar el pasto, hasta tareas que requieren cierto entrenamiento especializado, como cuidar personas de edad avanzada o enfermas. Y dentro de la gran gama de robots de servicio que se pueden desarrollar, son los robots bípedos humanoides los que ofrecen mayores beneficios, pues pueden ir y estar donde las personas, trabajar en ambientes pensados para los humanos y utilizar todas las herramientas que el hombre ha inventado a lo largo de su historia. *El impulsar el desarrollo de robots humanoides en México ayudará a mejorar la competitividad del país y nos colocará en ventaja competitiva ante una previsible invasión de una nueva industria que se vislumbra inminente [63].*

2.5. Temas actuales en la robótica bípeda

La investigación en la robótica bípeda puede clasificarse en tres áreas principales: el análisis en la estabilidad de la postura del robot, la generación de patrones de marcha y el diseño e implementación de

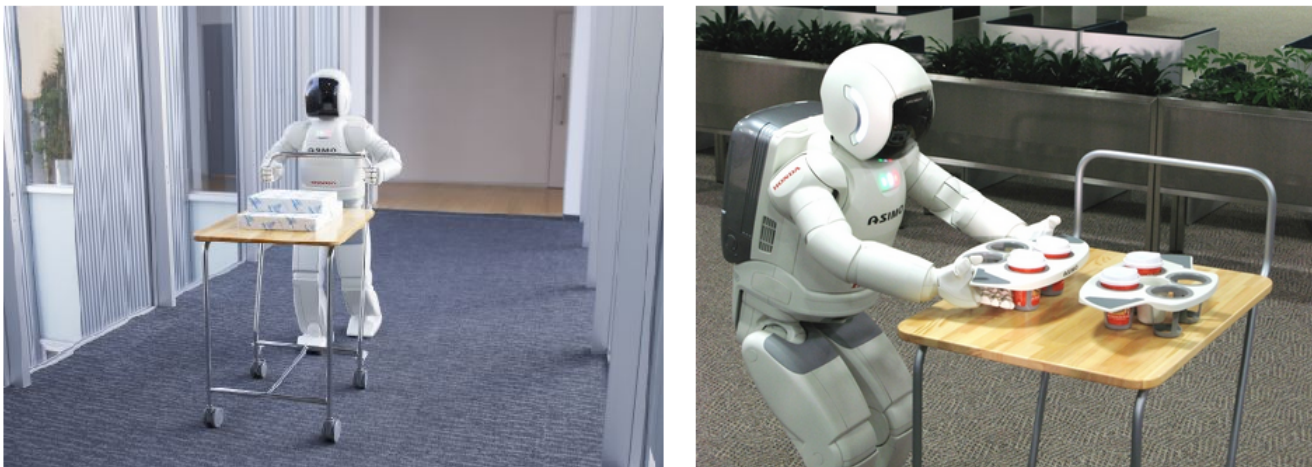


Figura 2.15: ASIMO realizando algunas tareas de servicio.

controladores para estabilizar al robot [11]. La estabilidad es importante dado que permite determinar si el robot bípedo es capaz de mantenerse en pie y conservar su postura durante la marcha. Por otra parte, se le llama generación de patrones de marcha al cómputo de las trayectorias angulares para cada una de las juntas rotacionales que posee el robot, cuyos movimientos sincronizados hacen posible la locomoción bípeda. Los ciclos de marcha son implementados produciendo las entradas de control adecuadas para los actuadores de acuerdo a los perfiles de posición y velocidad establecidos. Dado que los ciclos de marcha son generados considerando idealizaciones tanto en el robot como en el entorno, es necesario el diseño e implementación de controladores que permitan estabilizar al bípedo ante posibles perturbaciones.

2.5.1. Estabilidad

La estabilidad de los robots bípedos depende de la forma y tamaño de sus pies. El criterio más común para el análisis de estabilidad es el punto de momento cero, ZMP por sus siglas en inglés *Zero-Moment-Point*. Este concepto, que toma en cuenta ciertos parámetros dinámicos y de la estructura mecánica del robot, determina la estabilidad del bípedo durante la marcha a través de la posible rotación de la planta del pie, la cual se examina en relación a la posición de un punto de referencia sobre la superficie por la que camina el bípedo. Un concepto importante en el tema de estabilidad es el polígono de soporte, que es el conjunto convexo que forman los puntos de soporte del robot con el piso.

El ZMP es un concepto empleado para asegurar la estabilidad dinámica en la locomoción bípeda, y tiene una gran importancia para el análisis de marcha, síntesis y control. Debido a la naturaleza de un robot bípedo, el contacto entre el pie de soporte y la superficie es determinante para lograr una caminata efectiva; sin embargo, esta situación no puede ser controlada de manera directa, por lo cual se debe recurrir a un método indirecto que determine la dinámica apropiada para el mecanismo sostenido por el pie. El indicador general del comportamiento del mecanismo es el punto donde todas las fuerzas que actúan sobre el mismo pueden ser reemplazadas por una sola. A este punto se le llama ZMP.

Para dar una noción del ZMP, en [59] se considera al robot durante la fase de soporte simple, donde las fuerzas y momentos que actúan [figura 2.16 (a)] son las siguientes:

- La fuerza \mathbf{F}_A y momento \mathbf{M}_A del mecanismo sostenido por el pie de soporte. Actúan en el tobillo del pie de soporte.
- El peso del pie de soporte \mathbf{W} . Actúa en el centro de gravedad del pie.
- La fuerza de reacción \mathbf{R} y momento de reacción \mathbf{M} de la superficie. Actúan en alguna región del polígono de soporte.

Considerando que la fuerza de fricción actúa en el área de contacto del pie con la superficie y que el pie está inmóvil, se puede establecer que las componentes horizontales de la fuerza de reacción \mathbf{R} representan a la fuerza de fricción, las cuales están en equilibrio con las fuerzas horizontales de \mathbf{F}_A . El momento de reacción \mathbf{M} representa el momento de la fuerza de fricción; cuando la marcha del bípedo es estable, sus componentes horizontales son iguales a cero, quedando únicamente la componente vertical M_z , la cual está en equilibrio con la componente vertical de \mathbf{M}_A [figura 2.16 (b)]. La componente vertical de la fuerza de reacción \mathbf{R} representa la reacción de la superficie, la cual está en equilibrio con la componente vertical de \mathbf{F}_A y el peso del pie de soporte \mathbf{W} . Hasta aquí, la fuerza \mathbf{F}_A , el peso \mathbf{W} y la componente vertical de \mathbf{M}_A han sido equilibradas. Sin embargo, la única forma de compensar las componentes horizontales del momento \mathbf{M}_A es cambiando la posición de la fuerza de reacción \mathbf{R} ; es decir, la posición de \mathbf{R} depende de la magnitud de las componentes horizontales de \mathbf{M}_A . El problema: si el polígono de soporte del pie no es lo suficientemente grande para abarcar a dicha fuerza, ésta actuará sobre los bordes de la planta del pie, y los momentos no compensados de \mathbf{M}_A provocarán una rotación sobre dicho borde, lo que puede generar que el bípedo se vuelque.

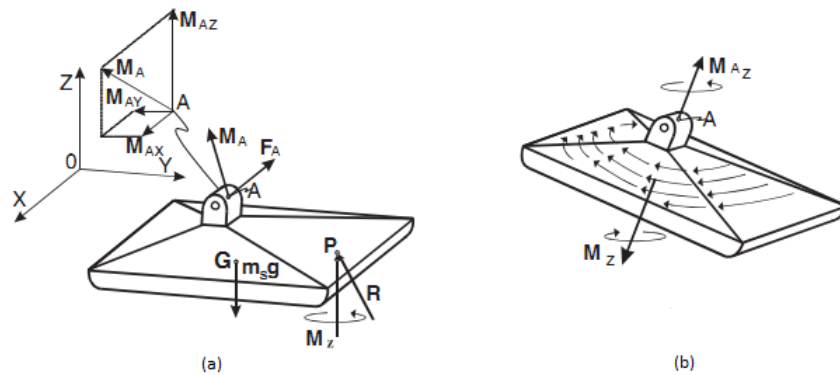


Figura 2.16: Fuerzas actuantes en la planta de un robot bípedo. Adaptado de VUKOBRATOVIC. ZMP -Thirty Five Years of its Life, 2004.

De acuerdo a la proposición anterior, la condición necesaria para mantener estabilidad dinámica es que las componentes horizontales de los momentos que actúan frente a los momentos de fricción sean cero. Se puede decir también que si la posición de \mathbf{R} de acuerdo a las ecuaciones de equilibrio se encuentra dentro del polígono de soporte, el robot es estable; a dicha posición se le llama ZMP.

2.5.2. Generación de patrones de marcha

Los temas actuales en la generación de patrones de marcha son el desarrollo de trayectorias que aseguren la estabilidad del bípedo, el estudio de métodos biológicamente inspirados para producir patrones de marcha y el estudio de técnicas para la optimización de dichos patrones. Dado que es posible asegurar la estabilidad de un robot bípedo manteniendo al ZMP dentro del polígono de soporte, la manera más común de generar patrones de marcha es mediante el desarrollo de trayectorias para cada una de las juntas articulares del robot, que, como resultado de sus movimientos sincronizados, produzcan un ciclo de marcha que cumpla con el criterio del ZMP. Debido a la complejidad que requiere generar patrones de caminata dinámicamente estables, los primeros métodos que se emplearon fueron en modo *offline*; es decir, primero, se planificaba la caminata del bípedo, y una vez que bajo ciertas idealizaciones del robot y del ambiente se había logrado generar un ciclo de marcha que cumpliera con algún criterio de estabilidad, éste se reproducía en el prototipo físico. Gracias al surgimiento de computadoras cada vez más rápidas y al desarrollo de métodos de planificación de caminata bípeda más sofisticados, en la actualidad es posible la generación de patrones de marcha en modo *online*, cuya principal ventaja es que el ciclo de marcha puede ser actualizado en tiempo real para que el robot se adapte a cambios en el ambiente.

La forma más directa de generar las trayectorias para las juntas articulares del robot es mediante la planificación de la posición y orientación de sus efectores finales —pies y torso—, lo cual requiere de la resolución de la cinemática inversa [65]; sin embargo, debido al alto costo computacional que requiere resolver dichas ecuaciones —y que va incrementando conforme al número de grados de libertad del robot—, esta estrategia solo resulta práctica en la generación de trayectorias en modo *offline*. Un método que no requiere análisis ni procesos matemáticos complejos consiste en la planificación de la caminata mediante la incidencia directa en las articulaciones del robot; sin embargo, se pierde el control sobre la posición, velocidad y aceleración de los efectores finales del robot [65].

De acuerdo al método empleado para la generación de patrones de marcha, es posible que se requiera el modelo dinámico del robot; sin embargo, debido a su complejidad matemática y a la dificultad que implica su análisis, se suelen emplear modelos simplificados. El modelo del péndulo invertido es el más empleado; en éste, la estructura del bípedo se reemplaza por un punto en el que se concentra toda la masa del robot (centro de masa) [64]. El modelo del péndulo invertido es útil para el análisis de estabilidad por medio del criterio del ZMP, de ahí su uso para la generación de ciclos de marcha y el diseño de controladores.

Por otra parte, se han empleado métodos biológicamente inspirados para la generación de ciclos de marcha y empleado osciladores neuronales para la elaboración de algoritmos que aprendan a generar patrones de caminata dinámicamente estables sobre superficies con condiciones desconocidas [54]. Finalmente, se han empleado algoritmos genéticos para optimizar los ciclos de marcha producidos por los osciladores neuronales [21].

2.5.3. Control

Un problema canónico en la robótica bípeda es el diseño de controladores que generen movimientos en lazo cerrado, que sean periódicos y que sean estables. En la mayoría de casos, el control se hace a dos niveles: bajo y alto. El control a nivel bajo garantiza que cada junta actuada siga una referencia. Por otra parte, el control a nivel alto modifica las trayectorias de referencia de las juntas angulares actuadas

para asegurar que el ZMP se mantenga dentro del polígono de soporte. Debido a que la trayectoria del ZMP durante la marcha del bípedo difiere con la del patrón de caminata generado, es necesario emplear el control a nivel alto con el fin de compensar este error. Para el diseño de controladores de nivel alto, se suelen emplear dos estructuras: control monoarticular —donde se considera que las articulaciones del robot están desacopladas, por lo cual hay un controlador para cada articulación, y los efectos de los otros componentes son tratados como perturbaciones— y control multiarticular, que suele emplear un modelo dinámico simplificado del robot.

Recientemente, se ha popularizado la investigación en el área de los controladores biológicamente inspirados. De acuerdo a diversos estudios sobre la locomoción en sistemas biológicos —en los que se observa que existe una sincronización global entre las actividades rítmicas del sistema nervioso y los movimientos rítmicos del aparato locomotor— los patrones motrices son generados por la interacción del aparato locomotor, el sistema nervioso y los sistemas sensitivos bajo parámetros variables en el ambiente. Los patrones rítmicos motrices son coordinados por circuitos neuronales (*Central Pattern Generators*) [54]. Los métodos biológicamente inspirados emplean modelos de redes neuronales para simular el funcionamiento de los circuitos neuronales, y utilizan modelos matemáticos para representar a la estructura del robot. En [54] se emplearon osciladores neuronales para el control de un modelo computacional de un bípedo y se probó que era posible adaptar el patrón de caminata sobre diferentes superficies a partir de la interacción entre el controlador, el cuerpo del robot y el ambiente.

Capítulo 3

La locomoción bípeda

3.1. La marcha

3.1.1. Definición

El caminar o andar se define como la repetición de una serie de movimientos articulares simultáneos que son ocasionados por la actividad muscular y que desplazan al cuerpo, manteniendo una postura estable, sobre una línea de progresión. Para ello, los músculos deben actuar en el momento preciso y con la intensidad adecuada. El objetivo principal del aparato locomotor durante la marcha es el ahorro de energía. Tal es así, que en [23] se define a la caminata como la traslación del centro de masa a través del espacio, de tal manera que se emplee la menor cantidad de energía. Cualquier tipo de caminata bípeda requiere de dos características básicas [22]:

1. Movimientos periódicos de cada pie para llevarlo de una posición de soporte a otra siguiente.
2. Fuerzas de reacción suficientes, aplicadas a través del pie a la superficie de apoyo, para soportar el peso del cuerpo.

El movimiento periódico de la pierna es la esencia de la naturaleza cíclica de la marcha.

3.1.2. El ciclo de marcha

Periodos

A la secuencia desarrollada por cada una de las extremidades inferiores se le conoce como ciclo de marcha o ciclo común, que consta de dos periodos: uno de apoyo (al que se le conoce como postura, y que abarca alrededor del 60 % de dicho ciclo) y uno de oscilación o avance, que abarca alrededor del 40 % del mismo ciclo. La caminata se da como resultado de la repetición del ciclo común. Dado que éste puede dividirse en eventos cuya continuidad debe ser preservada, cualquiera de estos eventos puede considerarse como el punto de partida de la marcha.

Fases

La fase de soporte doble se repite dos veces durante el ciclo común: al inicio y al final del periodo de apoyo. Cada fase de soporte doble abarca alrededor del 15 % del ciclo común. La fase de soporte simple, aquella en la que solo un pie está en contacto con la superficie de apoyo, abarca el porcentaje restante, aproximadamente 70 %.

3.1.3. Tareas y fases de la marcha

Incluidas en los periodos de apoyo y oscilación, existen tres tareas que constituyen el ciclo de marcha: el avance (*limb advancement*), la aceptación de peso (*weight acceptance*) y el soporte simple (*single limb support*). Dentro del periodo de oscilación se lleva a cabo el avance, y dentro del periodo de apoyo se llevan a cabo las tareas de aceptación de peso y de soporte simple. Cada una de estas tareas está compuesta por fases; en total resultan 8 unidades funcionales conocidas como fases de la marcha (figura 3.1), e inicialmente, fueron propuestas por Jaquelin Perry [24]. La aceptación de peso, que es la primera tarea, está compuesta por las fases de contacto inicial y de respuesta a la carga. La tarea de soporte simple se compone por las fases de postura intermedia y postura final. Adicionalmente, la pre-oscilación es la fase de transición entre el soporte simple y el avance. Por último, el avance, la tercera tarea, está compuesta por la oscilación inicial, la oscilación intermedia y la oscilación final. En esta tarea, el cuerpo se traslada para adquirir la postura del siguiente contacto inicial.

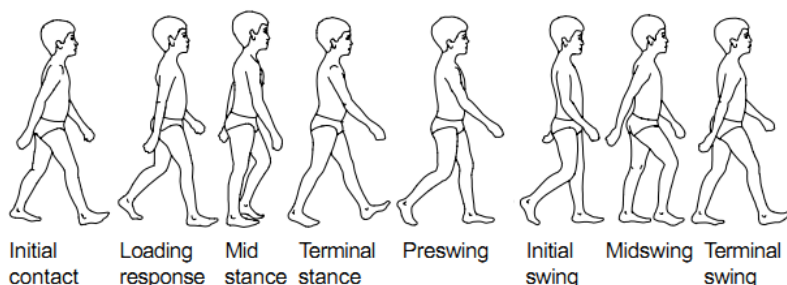


Figura 3.1: Ciclo de marcha de un niño de 8 años (edad promedio a la que se alcanza el patrón propio del adulto) descrito —en este caso para la pierna derecha— por las 8 unidades funcionales propuestas por Jaquelin Perry. Adaptado de L VAUGHAN C. Dynamics of human gait, 1992.

Relacionándolas con el movimiento de la pierna derecha, las fases del ciclo de marcha son:

- Contacto inicial (*initial contact*). Inicia cuando el pie toca el suelo. Conforme el talón va girando, el área de contacto entre la planta del pie y la superficie aumenta. En esta fase se logra ubicar a la pierna en la posición de inicio del periodo de postura.
- Respuesta a la carga (*loading response*). Esta fase marca el inicio de la fase de soporte doble. Comienza cuando el pie toca la superficie de apoyo, y continúa hasta que el otro pie se levanta. En esta fase se reduce el impacto causado por la superficie sobre el cuerpo al flexionar la rodilla, se estabiliza la marcha y se permite seguir con el avance.
- Postura intermedia (*midstance*). Esta fase representa la primera mitad de la fase de soporte simple. Inicia cuando el otro pie se levanta y continúa hasta que el centro de masa se alinea con la parte delantera del pie de apoyo. Mientras la pierna derecha se encuentra en esta fase, la pierna izquierda está en la fase de oscilación inicial (figura 3.1). En la fase de postura intermedia se consigue el apoyo completo del pie y se logra estabilizar al tronco y a las extremidades.
- Postura final (*terminal stance*). En esta unidad funcional se completa el movimiento realizado en la fase de soporte simple. Inicia cuando el talón del pie de apoyo asciende y continúa hasta que el otro pie toca la superficie de contacto. En esta fase se logra desplazar al centro de masa del cuerpo por delante del pie de apoyo.

- Pre-oscilación (*preswing*). Ésta es la última fase perteneciente al periodo de postura; representa a la fase de soporte doble del pie contrario. Inicia cuando la otra pierna hace contacto con la superficie. Durante esta fase se realiza una transferencia de carga entre las extremidades y se logra ubicar a la pierna en la posición de inicio del periodo de oscilación.
- Oscilación inicial (*initial swing*). Corresponde, aproximadamente, a una tercera parte del periodo de oscilación. Inicia cuando el pie abandona la superficie y se extiende hasta que éste se encuentra al lado del pie de apoyo. El pie se eleva mediante la flexión de la cadera y rodilla. Mientras un pie se encuentra en esta fase, el otro está en la fase de postura intermedia.
- Oscilación intermedia (*midswing*). Inicia cuando la pierna oscilante se encuentra a un lado de la pierna de apoyo, y continúa hasta que la pierna rebasa la posición del pie flotante (plano lateral) y la posición de la tibia es ortogonal con respecto a la superficie de apoyo. La pierna se extiende como efecto a la gravedad, y el tobillo continúa su flexión para llegar a su posición natural.
- Oscilación final (*terminal swing*). La fase final del periodo de oscilación inicia cuando la tibia es ortogonal a la superficie de apoyo; se extiende hasta que la pierna establece contacto con la superficie. En esta fase se logra completar el avance de la pierna y se le prepara para entrar al periodo de postura.

3.1.4. Nomenclatura tradicional

Tradicionalmente, el ciclo de marcha se divide en ocho eventos basados en el movimiento del pie (figura 3.2).

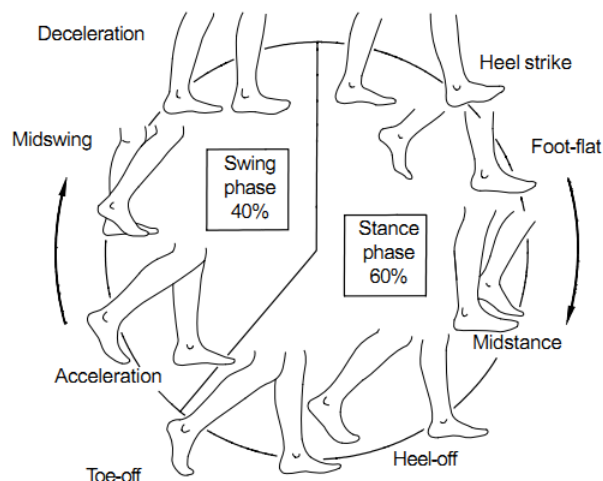


Figura 3.2: Nomenclatura tradicional para describir el ciclo común de la caminata bípeda. L VAUGHAN C. Dynamics of human gait, 1992.

Los eventos que ocurren durante el periodo de postura son los siguientes:

- Ataque del talón en el suelo (*heel strike*). Representa el punto en el que el centro de masa está en su posición más baja.
- Pie plano en el suelo (*foot-flat*). Es el momento en el que el polígono de soporte del pie toca la superficie.

- Postura intermedia (*midstance*). Sucede cuando el pie flotante rebasa al pie de apoyo y el centro de masa está en su posición más alta.
- Despegue del talón (*heel-off*). El talón pierde contacto con la superficie.
- Despegue de los dedos (*toe-off*). Es el punto en el que termina el periodo de postura: cuando el pie abandona la superficie.

Los eventos que ocurren durante el periodo de oscilación son los siguientes:

- Aceleración (*acceleration*). Los músculos de la cadera se contraen para incrementar la velocidad del pie flotante y llevar la pierna hacia adelante.
- Oscilación intermedia (*midswing*). Ocurre cuando el pie pasa por debajo del cuerpo. Este evento coincide al de postura intermedia (*midstance*) del otro pie.
- Desaceleración (*deceleration*). Describe la acción de los músculos mientras desaceleran la pierna y estabilizan al pie.

Esta clasificación es útil para describir el ciclo de marcha ideal de una persona; sin embargo, para el análisis de ciclos de marcha de cualquier tipo —por ejemplo, el de personas con patologías en el pie— se requiere hacerlo con la propuesta de Jaquelin Perry, que es lo suficientemente general para estudiarlos.

3.1.5. Elementos funcionales

La marcha se puede dividir en tres elementos funcionales: progresión o avance, estabilidad y conservación de energía [24].

Progresión o avance. Consiste en trasladar al cuerpo de un lugar a otro. El avance es ocasionado por el desplazamiento del centro de masa del cuerpo hacia adelante y por la oscilación del pie flotante. Las fuerzas ocasionadas por estas acciones generan un momento de inercia que es preservado al inicio del siguiente periodo de postura, durante el cual el pie y el tobillo sirven como mecedora para permitir que el cuerpo avance balanceándose sobre el pie de apoyo.

La fuerza básica para desplazar al cuerpo hacia adelante es proporcionada por el peso corporal. Al inicio y al final del periodo de postura, el cuerpo se deja caer de forma rápida hacia el suelo, provocando fuerzas de reacción mayores a la magnitud del peso. En efecto, en la marcha humana, el pie flotante cae sobre la superficie de apoyo; sin embargo, en la robótica humanoide, este comportamiento debe ser evitado con el fin de proteger a la estructura y a los sensores de fuerza localizados en la planta de los pies [45].

Estabilidad. La estabilidad en posición erguida es el resultado del balance entre la alineación del cuerpo y la actividad muscular en cada articulación. Cada parte del cuerpo es un peso que caerá, a menos que sea frenado. Evitar la caída del cuerpo es una tarea difícil debido a varias razones. Por una parte, por la superficie reducida que ocupa el polígono de soporte durante la fase de soporte simple y por la diferencia de áreas que abarca la planta del pie tomando como referencia al tobillo, pues mientras la zona de apoyo del talón es tan solo mayor a un centímetro, la zona de apoyo anterior al tobillo promedia alrededor de 10 [cm] en adultos. Por otra parte, la mayor concentración de masa en la parte superior del cuerpo —que implica una ubicación del centro de masa superior a la cadera— con respecto a la masa de las extremidades inferiores, complica aún más la estabilización de la marcha.

Conservación de la energía. Una forma de tener idea de la cantidad de energía gastada durante la marcha es conociendo la cantidad de oxígeno empleada. Dado que el oxígeno es consumido cuando los músculos se contraen, para mejorar la eficiencia de la marcha, es preciso disminuir el esfuerzo muscular al caminar, lo cual se logra sustituyéndolo (el esfuerzo muscular) por el uso de los momentos de inercia producidos en la marcha y mediante la coordinación de los movimientos de la rodilla, pelvis y tobillo para mantener una distancia constante entre las extremidades durante el periodo de postura.

3.1.6. Características del paso

El paso es el conjunto de fenómenos que se producen entre el apoyo de un talón y el apoyo sucesivo del talón contralateral. Al ángulo formado por la línea de marcha y el eje del pie se le llama ángulo de paso. La longitud de la zancada es la distancia que se traslada el sujeto durante un ciclo, y es medida calculando la distancia entre las huellas de un mismo pie. Dos pasos forman una zancada. En una marcha normal, la longitud de paso de una pierna es aproximadamente igual a la longitud de paso de la otra. La amplitud de paso es la distancia que separa al talón de la línea de marcha (figura 3.3).

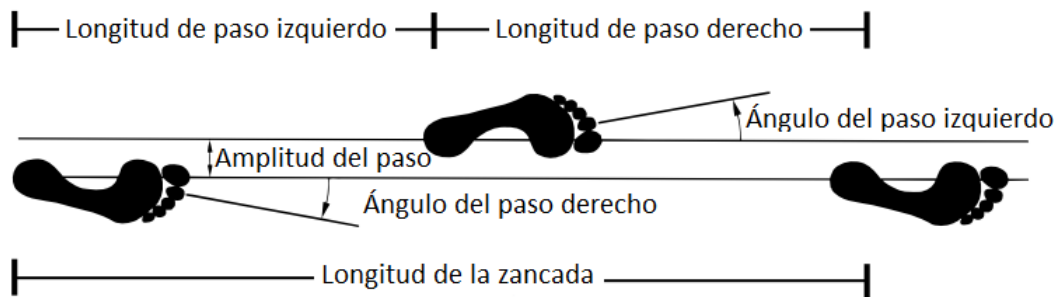


Figura 3.3: Características del paso a través de las huellas. Adaptado de L VAUGHAN C. Dynamics of human gait, 1992.

3.2. La marcha humana

La marcha bípeda ha sido adquirida por el hombre a través de millones de años de evolución y ha sido posible por profundas modificaciones anatómicas; ésta liberó al hombre del uso de las manos para otras funciones de supervivencia [62]. La marcha es realizada por medio de la acción alternada de las extremidades inferiores, sin que el cuerpo abandone el suelo; durante ésta el equilibrio se encuentra amenazado en cada instante y el cuerpo se balancea al filo de una catástrofe [62].

La marcha es el resultado de un proceso que empieza como un impulso nervioso en el sistema nervioso central y finaliza con la producción de fuerzas de reacción del suelo. La programación de los movimientos del aparato locomotor ocurre en los centros supraespinales, e implica la conversión de una idea en un patrón para la actividad muscular con la que se desarrolla un ciclo de marcha. Los comandos de esta programación son transmitidos al tronco del encéfalo y a la médula espinal, y la ejecución de estos comandos implica la generación de los patrones para la actividad muscular y la retroalimentación sensorial de los músculos y articulaciones [30].

La marcha es un proceso aprendido y no el desarrollo de un reflejo innato [58]. El ciclo de marcha de cada persona tiene características particulares, determinadas por factores como la masa y longitud

de los distintos segmentos corporales. El ser humano necesita un período prolongado para lograr la estabilidad bípeda y la marcha independiente. El desarrollo de la marcha se da como un proceso de aprendizaje. Al inicio, la marcha carece de armonía y de una adecuada coordinación. Para conseguir una mayor base de sustentación, la amplitud del paso es muy amplia; los brazos del cuerpo se separan para mejorar el equilibrio. Posteriormente, el sujeto experimenta con su sistema neurológico y musculoesquelético: altera las trayectorias de las diferentes articulaciones de su cuerpo durante la marcha hasta conseguir un completo control neuronal. Finalmente, a la edad de entre 7 y 9 años, se alcanza el ciclo de marcha propio del adulto.

De acuerdo a la cinética, la marcha se da como resultado de los movimientos generados por las fuerzas intrínsecas y extrínsecas. Las fuerzas de mayor influencia en el movimiento del cuerpo durante la marcha, son aquellas debidas a la acción de la gravedad, la contracción muscular, la inercia y las reacciones del suelo (resultantes de las fuerzas que ejerce el pie sobre el suelo).

Durante la caminata, el cuerpo se divide en dos sistemas: el locomotor y el pasajero. El sistema locomotor está conformado por las extremidades inferiores y la pelvis, que son las partes que contribuyen directamente a la locomoción bípeda. Por su parte, el sistema pasajero está conformado por el tronco, las extremidades superiores y la cabeza.

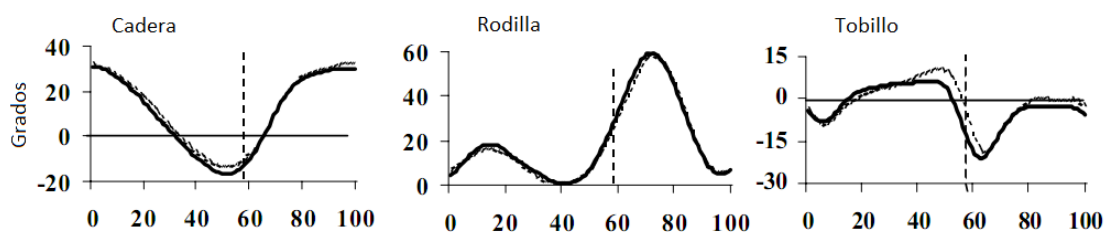


Figura 3.4: Movimientos de las articulaciones del sistema locomotor en el plano sagital. La línea punteada representa las trayectorias de un adulto, y la línea continua representa las trayectorias de un niño de 7 años. La línea vertical punteada separa al periodo de postura y oscilación. Adaptado de GANLEY. Gait kinematics and kinetics of 7-year-old children, 2005.

3.2.1. Actividad en el sistema locomotor durante la marcha

El tobillo. En la unión entre la tibia y el pie se presenta una situación especial: a través del tobillo, las fuerzas verticales del cuerpo son transmitidas a un sistema de soporte horizontal. Si bien el rango de los movimientos angulares del tobillo durante la marcha no es grande (en promedio 30° [40]), su movimiento sí es crítico para el avance y la absorción del impacto [24]. Dos veces durante el ciclo común se alternan los movimientos de flexión plantar (FP) y flexión dorsal (FD). Tres cambios ocurren dentro del período de postura (FP, FD, FP), y únicamente se realiza un movimiento de flexión dorsal durante el período de oscilación (figura 3.4). Dentro del período de postura, el movimiento del tobillo obedece a la carga impuesta sobre la pierna de apoyo y al desplazamiento del vector de peso. Mientras tanto, dentro del período de oscilación, el movimiento del tobillo se ve influenciado por el peso del pie flotante y la velocidad de su desplazamiento.

El pie. Hay tres articulaciones principales dentro del pie relacionadas con la mecánica de la caminata: las articulaciones subastragalina, mediotarsiana y metatarsfalángica. La articulación subastragalina añade movilidad en los planos transversal y coronal al tobillo. La articulación mediotarsiana

contribuye a la absorción del impacto. Finalmente, la articulación metatarsofalángica permite al pie girar sobre las cabezas del metatarso, y provee al cuerpo de una área de soporte en la parte delantera del pie.

En el período de postura existe un patrón del contacto entre la planta del pie (bóveda plantar) y la superficie de desplazamiento, que está dividido en tres áreas de soporte: talón, pie plano y pie delantero (figura 3.5). La última área de soporte del pie delantero es el primer metatarsiano (figura 3.5), que ocurre mientras el peso del cuerpo se transfiriere hacia el otro pie.

A través del pie se producen las fuerzas de reacción del suelo, que colaboran con el desarrollo de la marcha. Existe una fuerza de reacción vertical —que varía según los desplazamientos verticales del centro de masa—, una fuerza de reacción longitudinal —que traduce las fuerzas de frenado y empuje que se producen en el choque y despegue del pie— y una fuerza lateral, según los desplazamientos laterales del centro de masa.

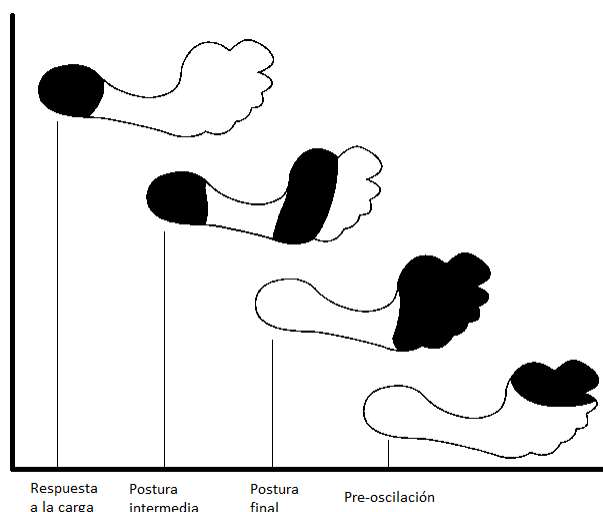


Figura 3.5: Secuencias de las áreas de soporte del pie durante la postura. Adaptado de PERRY JACQUELIN. *Gait analysis*, 1992.

La rodilla. Es la articulación que une al fémur y a la tibia. Debido a la longitud de éstos, pequeñas variaciones angulares en la rodilla resultan en cambios significantes en las posición del pie o del cuerpo; por lo tanto, el movimiento y estabilidad de la rodilla es determinante para el desarrollo de la marcha [24]. El movimiento sagital de la rodilla es usado, dentro del período de postura, para el avance y, dentro del período de oscilación, para el desplazamiento del pie. El movimiento de la rodilla en el plano coronal ayuda en el balance vertical y la rotación transversal de la rodilla adapta la alineación¹ mientras el centro de masa del cuerpo se traslada, pasando de estar detrás del pie de soporte a estar por delante de él. La rodilla, al igual que el tobillo, alterna dos veces los movimientos de flexión y extensión. Durante la fase de oscilación llega a flexionarse en promedio hasta 60° (figura 3.4).

La cadera. Representa la unión entre los sistemas locomotor y pasajero; por ello, su configuración provee movimientos en tercera dimensión con músculos específicos para cada actividad. Dentro del período de postura, el movimiento de la cadera permite que la pelvis y el tronco permanezcan erectos. La cadera presenta un solo período de extensión y flexión en cada ciclo. En el plano sagital, cuando

¹Se le denomina alineación a la configuración que adquieren los elementos del miembro inferior en condiciones normales. Dicha alineación minimiza el trabajo muscular y el desgaste articular durante la locomoción.

el pie toca el piso, comienza flexionada alrededor de 30° y se extiende hasta alcanzar 10° . Al final de la fase de postura la cadera vuelve a flexionarse hasta alcanzar los 30° iniciales (figura 3.4). El rango total de movimiento de la cadera en el plano sagital es de 40° a 50° [25].

La pelvis. En la pelvis se produce un continuo cambio postural por el movimiento de las caderas; primero desciende alrededor de una articulación de la cadera y luego, alrededor de la otra. Cuando el peso del cuerpo se transfiere de una pierna a otra, hay una desviación de la pelvis hacia el otro lado o extremidad en la que se apoya el cuerpo. Además, la pelvis rota hacia adelante, aproximadamente 8° , en el periodo de oscilación, con lo que es posible dar un paso ligeramente más largo.

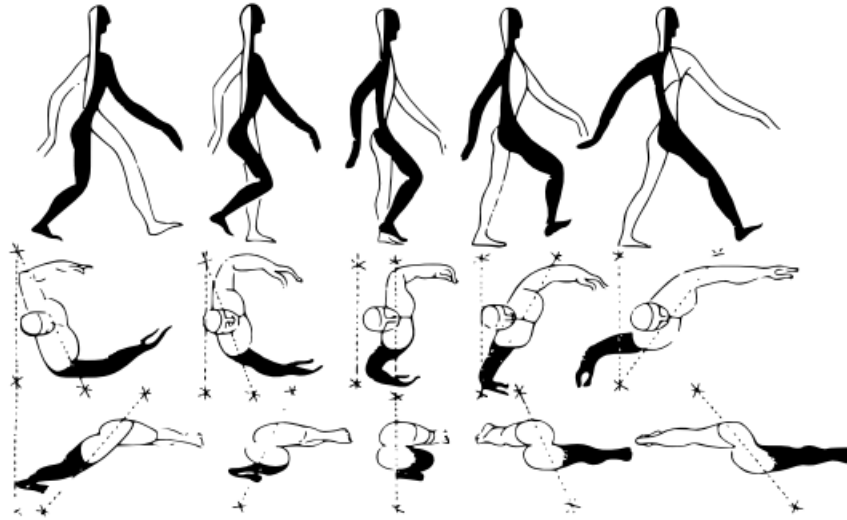


Figura 3.6: El movimiento de los brazos y hombros al caminar. La imagen de en medio muestra el ciclo de marcha desde una vista superior. De igual manera, la imagen de abajo muestra una vista superior del ciclo de marcha, pero con un corte transversal en la cadera. FERNÁNDEZ IGLESIAS S. Locomoción bípeda del robot humanoide Nao, 2009.

3.2.2. Actividad en el sistema pasajero durante la marcha

El tronco. Durante la marcha normal la cabeza y el tronco se comportan como una unidad; sus movimientos son de torsión, inclinación y oscilación [24], y resultan secundarios para la actividad realizada por las extremidades inferiores.

Las extremidades superiores. Con el fin de separar lo menos posible el centro de gravedad de su eje de progresión y disminuir la oscilación transversal del tronco, es preciso que los hombros y brazos realicen el giro balanceándose en forma sincrónica con los miembros inferiores, pero en sentido inverso (figura 3.6). El balanceo de los miembros superiores no solo es un movimiento pendular pasivo², sino que también depende de las acciones musculares. Las acciones musculares para el control del balanceo de los brazos durante la marcha es utilizada para vencer la inercia. Al caminar con los brazos pegados al tronco, el torso oscila mucho en el eje transversal, y aunque resulta más difícil, se puede continuar con el desarrollo con la marcha [50].

²Se le llama movimientos pasivos a aquellos donde la estructura no tiene intención de generar las fuerzas que los producen.

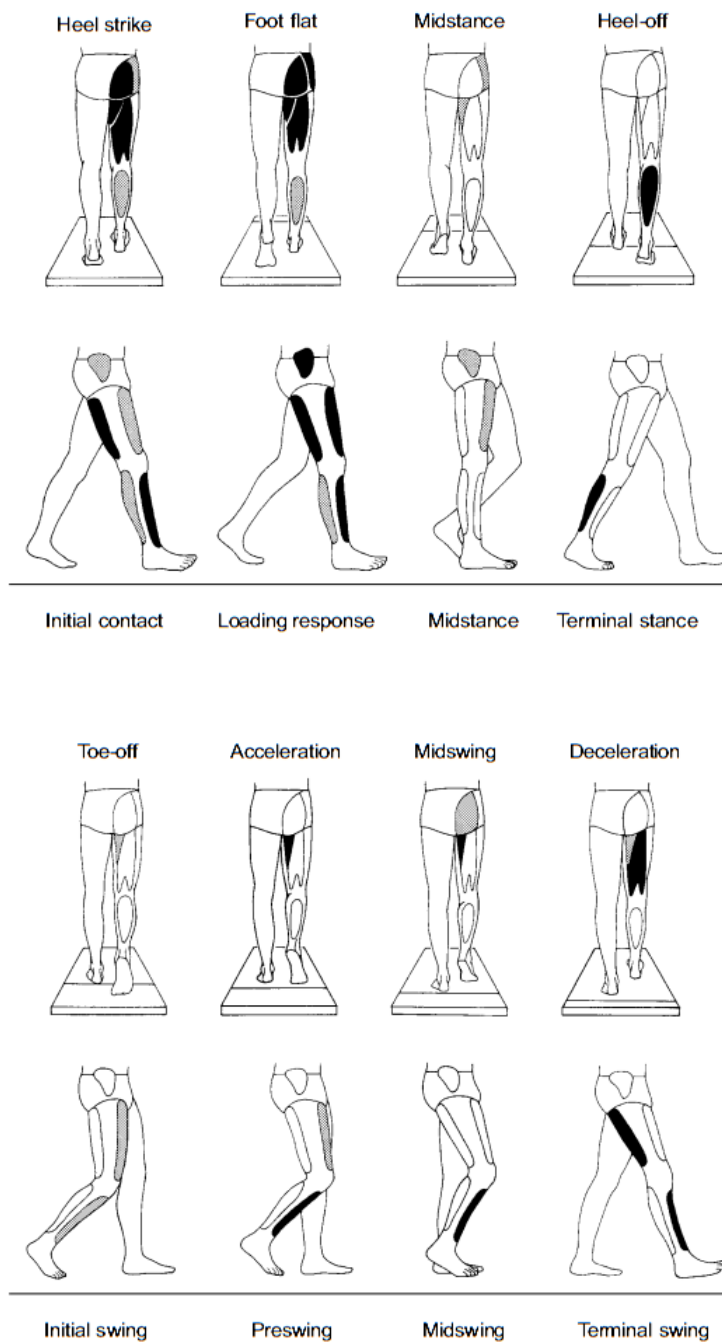


Figura 3.7: Vistas lateral y posterior de siete grupos musculares de las extremidades inferiores, mostrando la actividad muscular de éstos durante las fases de la marcha. El sombreado negro representa el mayor nivel de actividad muscular, y el blanco, el estado inactivo de los músculos.

3.2.3. Acción muscular durante la marcha

A través de la electromiografía³ se ha comprobado que la mayoría de los músculos principales de las extremidades inferiores están activos durante el ciclo de marcha. Estos músculos permanecen especialmente activos al principio y al final del período de postura y al principio del período de oscilación debido a las aceleraciones y a la transferencia de peso de un pie a otro [60].

En [22] se introdujo un método para la ilustración de la actividad muscular de las extremidades inferiores durante la marcha, que consiste en la organización de modelos de madera del sistema locomotor en serie dispuestos con la configuración de la etapa de la caminata que representan. En base a fotografías tomadas de estos modelos, se realizan dibujos de las extremidades con los grupos musculares superpuestos y se colorean los músculos de acuerdo a su nivel de actividad 3.7.

³La electromiografía es una técnica para la evaluación y registro de la actividad eléctrica producida por los músculos esqueléticos, donde se colocan electrodos en las extremidades del cuerpo, dando estímulos repetidos de corriente eléctrica.

Capítulo 4

Descripción del banco de pruebas

4.1. El prototipo de pruebas

4.1.1. El robot bípedo Scout

Scout es un robot bípedo desarrollado por Lynxmotion (figura 4.1). Tiene una altura de $23[cm]$ y un peso de $0.9[kg]$. Su estructura, constituida por eslabones de aluminio anodizado, se encuentra restringida por la geometría de sus servomotores. Los 12 GDL internos con los que cuenta lo dotan de una gran versatilidad, haciéndolo capaz de caminar hacia adelante, hacia atrás o girar con patrones de marcha similares a los de la locomoción humana.



Figura 4.1: El robot bípedo Scout.

4.1.2. Arquitectura del robot

El robot bípedo Scout está constituido por piernas de seis eslabones unidas entre sí mediante un eslabón central al que, equiparando la arquitectura del robot con la anatomía del ser humano, se le denomina torso. Sus trece eslabones se conectan en serie a través de juntas rotacionales actuadas por servomotores.

Para identificar a cada eslabón de manera inequívoca se establece la siguiente nomenclatura:

El torso se identifica con la letra B y los eslabones de las piernas con las etiquetas ni ; donde $1 \leq n \leq 6$ se encuentra determinada por su distribución en relación al torso, correspondiendo $n = 1$ a los eslabones unidos a éste y $n = 6$ a los que cumplen el papel de pies. Se asigna $i = 1$ a los eslabones de la pierna izquierda e $i = 2$ a los eslabones de la pierna derecha. La rotación de los ejes de los servomotores se describe por medio de los ángulos θ_{ni} , siendo ni la etiqueta del eslabón sujeto a su eje y la de su barra correspondiente en el modelo simplificado que se muestra en la figura 4.2. Dicha figura representa una simplificación del modelo CAD del bípedo, que resultó necesaria para la obtención del modelo cinemático y dinámico del bípedo Scout en [43]. En esta simplificación las juntas se representan por medio de cilindros cuyos ejes coinciden con los de los servomotores y se unen entre sí a través de barras.

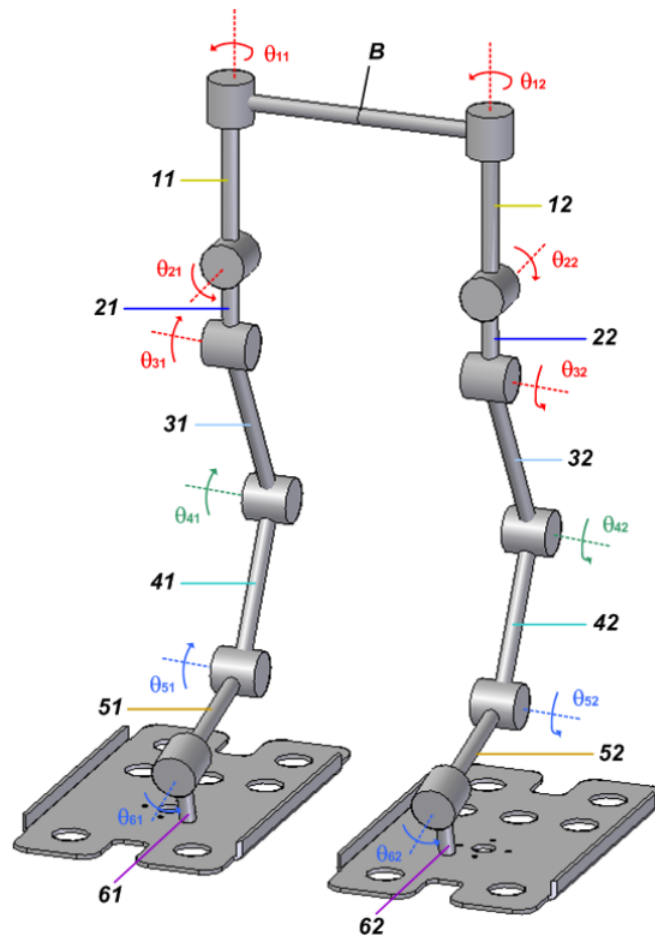


Figura 4.2: Modelo simplificado del bípedo Scout. Los ángulos θ_{ni} se asocian a las rotaciones de los servomotores en las juntas.

4.2. Interfaz gráfica de usuario para el accionamiento y manipulación del prototipo de pruebas

La interfaz que a continuación se describe fue realizada para facilitar el accionamiento de los servomotores del bípedo Scout desde la computadora.

Una vez que se tiene construido al robot bípedo —en este caso, a Scout—, el proceso para que logre una caminata dinámicamente estable, puede dividirse en tres partes:

- Modelado cinemático del robot. En algunos casos, dependiendo de las técnicas a usar en los siguientes puntos, también se requiere el modelado dinámico.
- Generación de patrones de marcha.
- Diseño e implementación de un controlador para estabilizar al bípedo ante perturbaciones.

Considerados estos puntos, la interfaz debe, como primer paso, permitir validar el modelo cinemático del robot; como siguiente paso, demostrar la validez de los patrones de marcha y poder evaluar e implementar patrones de marcha dinámicamente estables. En un futuro, se pretende que la interfaz facilite la implementación de los controladores que se diseñarán para estabilizar al bípedo.

4.2.1. Concepto y trabajo relacionado

Una interfaz gráfica de usuario (GUI, por sus siglas en inglés *Graphical User Interface*) aprovecha las capacidades gráficas de un ordenador para permitir el uso de un programa de una manera más intuitiva. En ingeniería es común usarlas tanto para facilitar el accionamiento y la programación de robots, como para simular procesos en los que se aprovechen las visualizaciones de espacios tridimensionales virtuales; tales como las rutinas de un sistema de manufactura flexible [39].

Aunque no es necesario contar con interfaces gráficas para manipular robots bípedos, éstas facilitan la interacción con ellos, permitiendo realizar una gran variedad de pruebas en menor tiempo. Éstas son usadas para manipular a los bípedos en tiempo real, para programar rutinas o para obtener datos sobre su funcionamiento.

Durante la generación de patrones de caminata se deben tomar en cuenta las restricciones en los movimientos angulares de las juntas del robot para evitar choques entre sus eslabones. En tanto que en [11] se obtienen experimentalmente los límites angulares para la generación de trayectorias, es necesario contar con un sistema que permita modificar la posición angular de cada uno de los eslabones de manera directa y en tiempo real. Debido a los métodos existentes para la generación de ciclos de marcha dinámicamente estables, también resulta necesario permitir el accionamiento del robot con trayectorias angulares previamente calculadas. Por las razones anteriores, la interfaz desarrollada cuenta con las siguientes prestaciones:

- Variación de las posiciones angulares de los servomotores del bípedo en tiempo real.
- Variación de las posiciones angulares de los servomotores de acuerdo a secuencias previamente calculadas y almacenadas en un archivo.
- Prototipo virtual para la asistencia en la realización de las acciones anteriores, mostrando la configuración espacial del bípedo.

Debido a las ventajas de contar con una GUI en aplicaciones de robótica, en la actualidad existen varias empresas dedicadas a su diseño. Por ejemplo, ALDERABAN robotics ha creado la plataforma NAOqi para programar a su humanoide NAO. Cyberbotics Ltd. creó Webots [38], un software para robots móviles que permite modelar y simular cuadrúpedos, bípedos y humanoides, entre otros; además de transferir rutinas de movimiento a robots comerciales.

A pesar de las facilidades ofrecidas por estos y otros programas, también cuentan con varias desventajas. Además de tener un gran costo, los programas anteriores presentan las siguientes limitaciones: no pueden ser usados para interactuar con cualquier bípedo ni pueden ser modificados para adaptarlos a las necesidades del usuario. Con una interfaz propia es posible contar con algunas características de las plataformas anteriores y modificar el código de acuerdo a las necesidades y evolución del proyecto.

Se pretende que esta herramienta sea de gran apoyo para la realización de pruebas y experimentos en el bípedo Scout, facilitando la investigación en torno a la estabilidad dinámica de la marcha.

4.2.2. La interfaz gráfica de usuario

La interfaz gráfica para el robot bípedo Scout tiene como propósito ser una herramienta intuitiva y fácil de usar, mediante la cual el usuario pueda interactuar con el robot de una manera más rápida y efectiva, facilitándole la realización de pruebas y experimentos. Debido a las partes con las que debe contar este software: menús y botones, modelo virtual, comunicación entre la computadora y el robot Scout, es necesario realizarlo sobre una plataforma que permita programar todas estas características.

LabVIEW es un ambiente de programación gráfica que, además de ofrecer las mismas funciones que otros lenguajes de programación, cuenta con una gran cantidad de módulos que permiten su comunicación con dispositivos externos como microcontroladores y sensores para su control y lectura. Además, cuenta con los módulos 3D Picture Control que permite crear escenas en 3D y MATLAB Script que permite ejecutar código escrito bajo la sintaxis de MATLAB. Estas características marcaron el diseño de la GUI para el robot bípedo en LabVIEW.

4.2.3. Descripción de la interfaz gráfica

Con el objetivo de separar las funciones de la interfaz y distribuir los elementos del panel frontal de manera homogénea se usó el contenedor *Tab Control*, el cual permite mostrar diferente información de acuerdo a la pestaña seleccionada por el usuario. Se crearon tres pestañas:

1. *Interacción directa con Scout* (figura 4.3). Los controles contenidos en esta pestaña permiten manipular las juntas rotacionales del bípedo de manera directa, obteniendo una respuesta instantánea. Cuenta con un arreglo de doce controles numéricos, cada uno vinculado con la posición angular del servomotor que mueve al eslabón *ni*. La posición de los servomotores se puede modificar introduciendo el ángulo deseado en grados o deslizando el cursor sobre las barras.

2. *Manipulación del bípedo mediante patrones de caminata* (figura 4.4). Esta pestaña cuenta con cuatro controles y dos indicadores, y también sirve como pestaña de configuración. El primer control, *Archivo*, sirve para seleccionar la ruta y nombre del archivo que contiene los datos de cada una de las trayectorias a seguir por los servomotores. En la parte inferior, se encuentran los botones para modificar el tiempo en el que transcurre la secuencia de posiciones, y para activar o desactivar la transferencia de estos datos al robot bípedo. Para comunicarse con los servomotores se usa como intermediario a un microcontrolador que envía las señales de referencia de acuerdo a los datos que

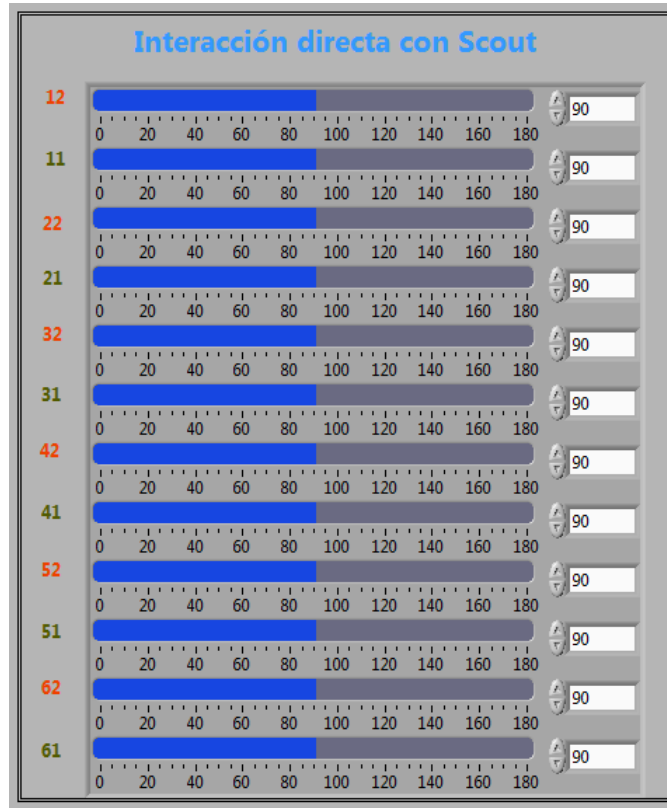


Figura 4.3: Panel frontal. Interacción directa con Scout.

recibe de la computadora. Por ello, existe un control para elegir el puerto serial en el que se conecta el microcontrolador. El indicador de la parte central es un arreglo donde se muestran los datos extraídos del archivo seleccionado.

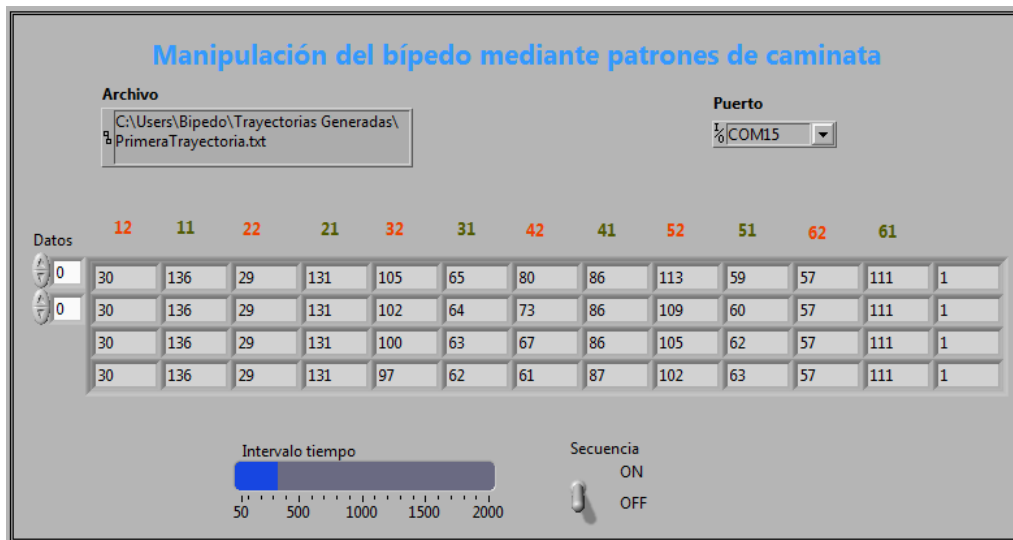


Figura 4.4: Panel frontal. Manipulación del bípido mediante patrones de caminata.

3. Modelo virtual del robot Scout (figura 4.5). Esta pestaña cuenta con dos opciones: interacción

directa o accionamiento mediante el archivo cargado en la pestaña anterior. En la parte izquierda de ambas opciones hay una ventana donde se muestra la configuración espacial del bípedo. En la parte derecha, la opción de interacción directa cuenta con el mismo arreglo de la primera pestaña, pero, en este caso, sirve para manipular al modelo virtual. En la parte derecha de la segunda opción, existen varios botones para seleccionar y configurar el tipo de cámara virtual, y un botón para iniciar o activar el ciclo de marcha. La cámara cuenta con tres configuraciones: libre, fijar cámara a robot o fijar objetivo a cámara. Debido a que los parámetros de las dos últimas opciones están descritos en coordenadas esféricas, se cuenta con un control tipo “rueda” con dos perillas para seleccionar los ángulos, y con un control de deslizamiento para seleccionar la distancia entre la cámara y el objetivo.

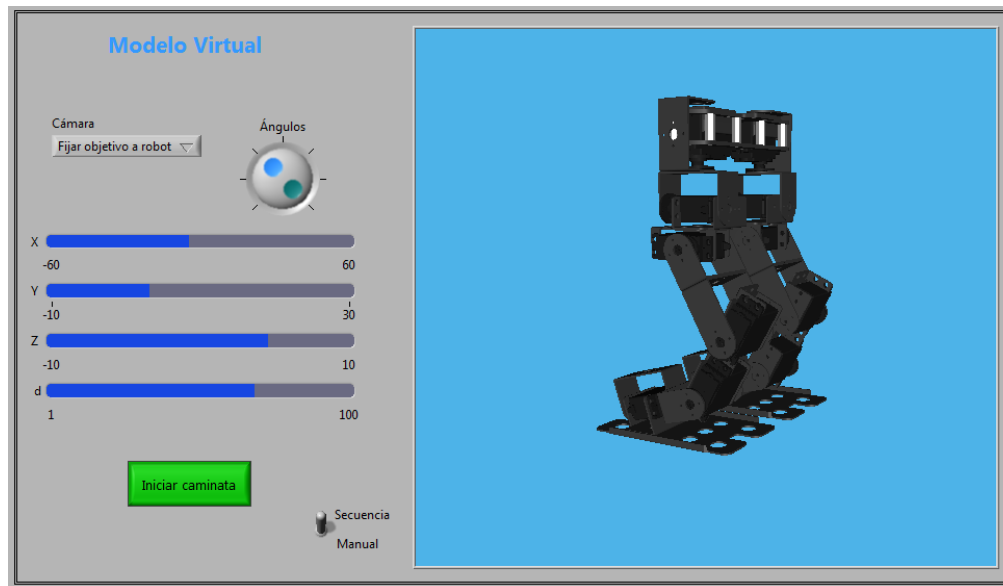


Figura 4.5: Panel frontal. Modelo virtual del bípedo.

4.2.4. Accionamiento de los servomotores

El movimiento de Scout es proporcionado por la acción de doce servomotores digitales HS5475 de la marca Hitec. Éstos tienen un torque de $5.47[\text{kg} - \text{cm}]$ a $6[\text{V}]$, un peso de $39.7[\text{g}]$ y cuentan con un sistema de engranajes de larga duración. Además, por el hecho de ser digitales, sus características son configurables, se puede cambiar su velocidad, precisión y el rango de giro. Los servomotores tienen un sistema en lazo cerrado para controlar su posición angular de acuerdo a una señal de referencia.

En los motores de aeromodelismo, como los del bípedo, es posible modificar su señal de referencia mediante la modulación por ancho de pulso (PWM, por sus siglas en inglés *Pulse-Width Modulation*). En el caso de los servomotores Hitec, su frecuencia es de $50[\text{Hz}]$ y su ciclo de trabajo se encuentra entre 4.5% y 10.5% , correspondiendo estos valores a las posiciones angulares de sus extremos y variando de manera lineal (figura 4.6).

Para generar las señales de referencia que marcan el movimiento de los servomotores se escribió un programa en el ambiente de desarrollo de código abierto Arduino para después cargarlo en el microcontrolador ATmega1280 de la tarjeta Arduino MEGA. Usando la librería *SoftwareServo*, incluida en Arduino, se crearon y configuraron doce salidas de PWM, una para cada servomotor. La comunicación

entre el microcontrolador y la interfaz gráfica se estableció mediante el protocolo RS-232. De esta manera, el usuario puede modificar la posición angular de los servomotores desde la computadora.



Figura 4.6: Relación ciclo de trabajo - posición angular del servomotor

4.2.5. Ensamble y programación del prototipo virtual

Una de las áreas de la ingeniería que encuentra aplicación en la robótica bípeda y humanoide es la computación gráfica. Empleando conceptos básicos de ésta es posible modelar y simular robots móviles con gran realismo. Para modelar a Scout se usó Solid Edge, en el cual se dibujaron todas las piezas con gran nivel de detalle. El ensamble del prototipo físico se realizó empleando modelado jerárquico, y se programó usando el módulo 3D Picture Control de LabVIEW.

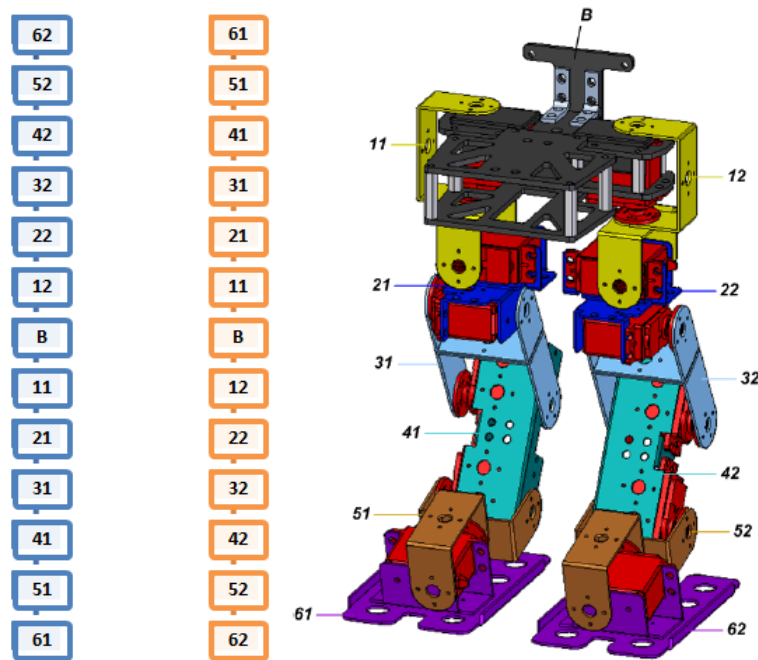


Figura 4.7: Árboles jerárquicos para el modelo virtual.

El modelado jerárquico consiste en la organización de los elementos del objeto a modelar en una estructura de árbol en donde cada nodo es una secuencia ordenada de transformaciones geométricas [14]. Por lo cual, si se realiza una rotación o traslación a un elemento padre, a todos los elementos hijos se les aplicará automáticamente la misma transformación.

Con el objetivo de importar a LabVIEW el modelo virtual del bípedo, se usó Solid Edge para generar el ensamble de las piezas que no tienen movimiento relativo entre sí, y se exportaron al formato VRML.

Cabe destacar que para la simulación del bípedo se construyeron dos árboles jerárquicos. En uno, árbol jerárquico azul de la figura 4.7, cuando el pie de apoyo es el de la pierna derecha, la pieza con mayor jerarquía es el eslabón 62, y el elemento que se encuentra hasta abajo del árbol es el eslabón 61. Cuando el pie derecho es el pie de apoyo, árbol jerárquico naranja de la figura 4.7, la jerarquía se invierte. Por lo tanto, además de los datos de la secuencia de las posiciones angulares de los servomotores, el archivo que se encuentra en la pestaña “Manipulación del bípedo mediante patrones de caminata” contiene una columna extra para indicar el cambio de jerarquía en las piezas.

4.2.6. Resultados

Al interactuar con Scout mediante la pestaña de interacción directa, se observó el comportamiento de los servomotores, lo cuales vibraban mucho. Para mejorar su desempeño, se ajustaron los parámetros de velocidad y de ancho de banda muerta, y aunque aún lo hacen, vibran menos.

La interacción con Scout mediante la segunda pestaña, “Manipulación del bípedo mediante patrones de caminata”, permitió validar los resultados de la cinemática inversa del bípedo presentada en [43]. El proceso para realizar este experimento fue el siguiente: primero, en un modelo del bípedo realizado en Solid Edge, se manipularon directamente sus juntas para que el robot obtuviera diversas configuraciones que representaran su posición espacial durante un momento preciso de la marcha, y de cada una de ellas se obtuvieron las posiciones espaciales de los eslabones B, 61 y 62. Para obtener una trayectoria continua de estos eslabones, se interpolaron dichas posiciones usando splines de segundo orden. Posteriormente, estas trayectorias se ingresaron en un programa realizado en Mathematica que resuelve la cinemática inversa del bípedo, y se obtuvieron los datos de las trayectorias angulares de todas las juntas rotacionales del bípedo Scout que, posteriormente, se guardaron en un archivo. Finalmente, se cargaron estas trayectorias en la GUI, se seleccionó el puerto al que estaba conectado el microcontrolador y se inició la comunicación con el prototipo físico. Los resultados pueden observarse en la liga [44], y en la figura 4.8 se muestran algunas capturas del funcionamiento del bípedo.

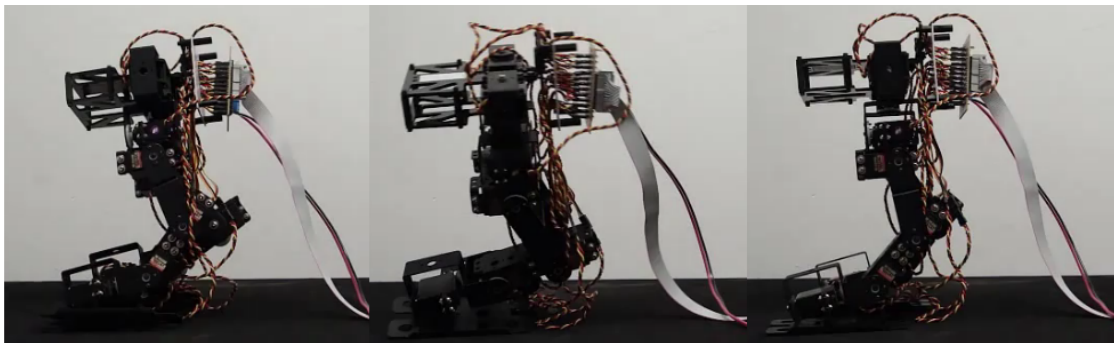


Figura 4.8: Ciclo de marcha implementado en el prototipo real.

Probando el funcionamiento del modelo virtual con el patrón de caminata de la prueba anterior, se puede apreciar que entre más datos haya para definir una trayectoria, más fino es su movimiento, aunque también más lento. Si se usan menos datos, se observan saltos en la simulación, como si su caminata no fuera continua. Esto permite explicar por qué en el modelo real una secuencia con menos datos es más inestable. Debido a que el movimiento del modelo real es continuo, no pueden existir saltos en su trayectoria. Los cambios abruptos en la posición angular de los servomotores en tiempos pequeños generan cambios bruscos en la velocidad de los eslabones que desestabilizan al bípedo. En la figura 4.9 se muestran algunas capturas del funcionamiento del modelo virtual.

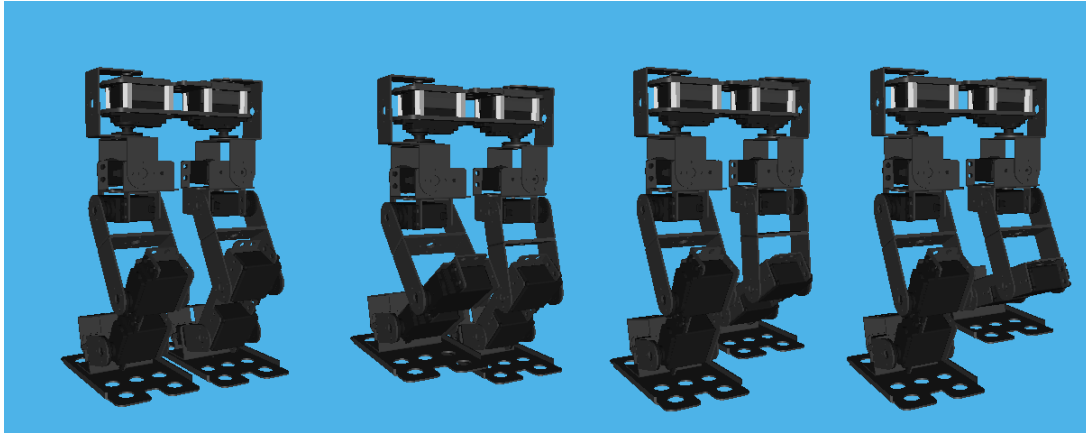


Figura 4.9: Simulación del ciclo de marcha.

Capítulo 5

Planificación de la caminata

En la robótica bípeda, a la formulación sistemática de las funciones angulares para cada junta rotacional, con el fin de obtener un movimiento deseado, es llamada síntesis de la marcha [42]. El principio de movimiento en los robots bípedos consiste en el seguimiento de estas funciones o trayectorias. A pesar de que éste no es un problema nuevo, aún no está cerrado, y hay varias propuestas para resolverlo; unas utilizan el análisis dinámico del robot, y otras emplean técnicas de inteligencia artificial. Los sistemas biológicos, que han sido de inspiración para la elaboración y construcción de robots bípedos, también constituyen una fuente de inspiración para atender problemas de movimiento. Por ejemplo, una de las técnicas más populares para la generación de patrones de marcha en robots bípedos, consiste en simular o imitar los movimientos realizados durante la marcha por el ser humano. Pero no solamente se han elaborado modelos basados en sus movimientos, sino también, se han propuesto modelos basados en los principios neurofisiológicos que gobiernan la marcha [9].

5.1. Modelos para la planificación de la caminata

Si bien es posible realizar el análisis dinámico del robot considerando a cada uno de los eslabones que lo conforman como elementos separados, su uso se restringe, debido a su alto costo computacional, a la generación de patrones de marcha en modo *offline*. El modelo del péndulo invertido, que considera a toda la masa del robot concentrada en un solo punto, reduce el tiempo de cálculo, y permite su uso en la generación de patrones de marcha en modo *online*; sin embargo este modelo carece de exactitud. Basados en el modelo del péndulo invertido y con el propósito de simular el desempeño de los robots bípedos con mayor exactitud, se han propuesto los modelos del péndulo invertido de doble masa y del péndulo invertido de múltiples masas, que consideran movimientos pendulares separados y en más eslabones; sin embargo, lo que se gana en exactitud, se pierde en rapidez, pues requieren de más tiempo para su cómputo.

El modelo de las fuerzas virtuales es un control intuitivo para el desarrollo de movimientos de caminata, que controla los torques en las juntas del robot basado en las fuerzas de reacción. En el MIT, para implementarlo en un robot bípedo, simularon sus componentes mecánicos para calcular fuerzas virtuales, que son aplicadas a través de las juntas del robot [26].

5.2. Métodos para la planificación de la caminata

Planificación mediante la incidencia directa en las articulaciones. También llamada planificación en el espacio articular, consiste en generar la caminata bípeda mediante la asignación de secuencias predeterminadas modificando las posiciones de las juntas angulares directamente. Este método se considera aplicable en robots que no rebasan los 12 grados de libertad [65], y solo es recomendable cuando el tamaño del bípedo o humanoide está debajo de los 0.50 [m], debido a que su velocidad no es muy alta y sus efectos dinámicos no son tan críticos [45].

Planificación mediante parámetros. Mediante este método se incide directamente en las articulaciones del robot, ya que se trabaja con la posición y orientación de los efectores finales [65]. Se proponen funciones para las trayectorias de los pies y la cadera, que están en función de ciertos parámetros, como la longitud del paso, su altura o la inclinación de la cadera 5.1. Como la etapa cíclica de la marcha es simétrica, se emplean las mismas trayectorias para el pie izquierdo y derecho. Las trayectorias de las juntas articulares se obtienen mediante la solución numérica de las ecuaciones cinemáticas de posición.

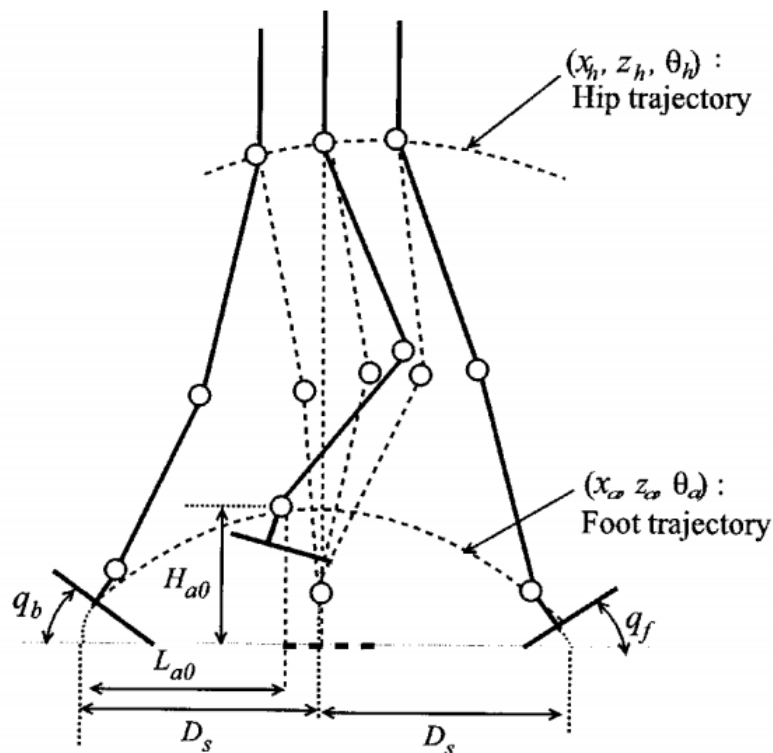


Figura 5.1: En la planificación mediante parámetros se proponen trayectorias para los pies y cadera.

Planificación en base al criterio del ZMP. Consiste en la generación de una secuencia de movimientos angulares con el fin de que el robot siga una trayectoria preestablecida del ZMP. En este método se requiere tanto del cálculo de la cinemática del robot, como el de la dinámica. Originalmente, se empleaba un mecanismo (figura 5.2), que en su parte superior tenía una estructura móvil —también conocida como tronco—, para que el robot cumpliera con la trayectoria preestablecida del ZMP [53]. Las juntas angulares del robot seguían ciertas trayectorias prescritas, mientras que el ZMP era controlado cambiando la posición de la estructura móvil.

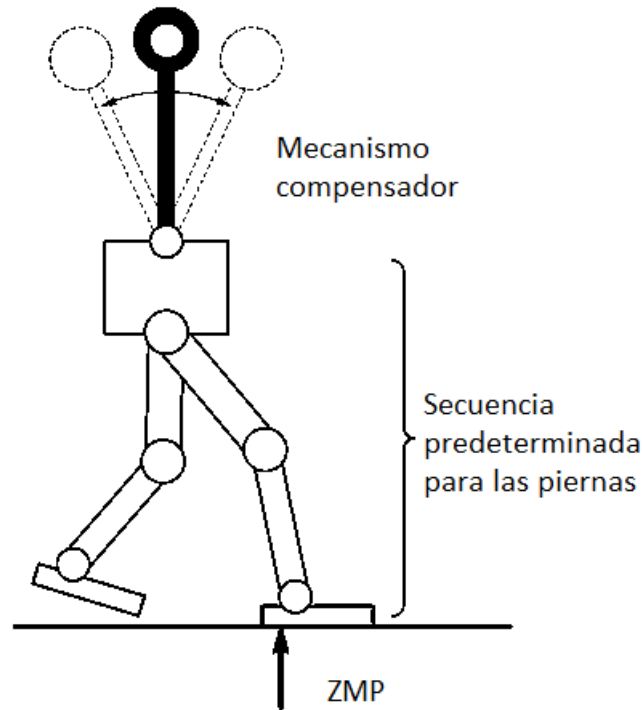


Figura 5.2: Esquema de robot bípedo con un mecanismo compensador usado para seguir una trayectoria preestablecida del ZMP. Adaptado de Springer Handbook of Robotics, 2008.

En las implementaciones actuales, más modernas, en vez de usar trayectorias prescritas para las juntas angulares del robot, se emplean las trayectorias a seguir por los pies, y en torno a éstas, se define el movimiento del cuerpo. Las trayectorias de todas las juntas angulares se obtienen mediante la cinemática inversa, dadas las posiciones espaciales de los pies y del cuerpo. En la figura 5.3 se ilustra este método, donde se dibujan las huellas de los pies sobre la superficie a través del tiempo; es decir, la zona sobre la cual se puede trazar la trayectoria del ZMP, el polígono de soporte, asegurando la estabilidad dinámica del robot.

5.3. Planificación de la marcha mediante la generación de trayectorias para sus efectores finales

Para la generación de los patrones de marcha, se optó por el método de planificación mediante parámetros. Si bien la marcha humana es el resultado de una sincronización global entre las actividades rítmicas del sistema nervioso y los movimientos rítmicos del aparato locomotor —incluyendo la interacción con el ambiente— que son generados por el efecto de fuerzas extrínsecas e intrínsecas, tales como la inercia o las contracciones musculares, de manera que se emplee la menor cantidad de energía posible; aquí el único objetivo es realizar la planificación de la marcha; es decir, definir las trayectorias angulares que debe seguir el robot bípedo Scout para que éste logre desplazarse por medio de la repetición de un ciclo de marcha.

Para la generación de las trayectorias, se tienen en cuenta las siguientes consideraciones:

- El robot se desplaza sobre una superficie plana y horizontal.

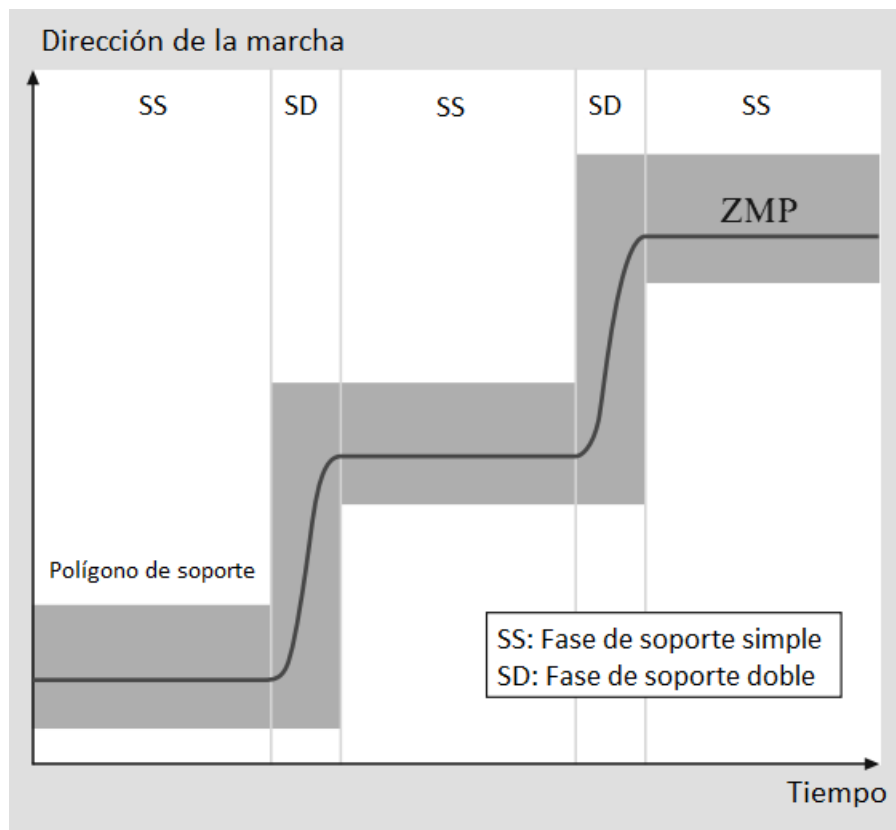


Figura 5.3: Generación de una trayectoria del ZMP dadas las huellas de los pies. Adaptado de Springer Handbook of Robotics, 2008.

- La dirección de desplazamiento del robot durante la marcha siempre es la misma.
- La marcha es un modo de locomoción bípeda donde las extremidades inferiores, con actividad alternada, ejecutan los mismos movimientos elementales.
- La marcha es el resultado de un evento cíclico, al que se le llama ciclo común o ciclo de marcha, que consta de dos fases: soporte doble y soporte simple.
- La secuencia del pie izquierdo y del pie derecho es la misma, aunque desfasada: mientras uno tiene la función de pie de apoyo, el otro funge como el pie flotante, y viceversa; por tanto, los parámetros que definen a las trayectorias de los pies, son los mismos.

La implementación en un prototipo físico de un ciclo de marcha generado mediante el método de planificación mediante parámetros, no asegura su estabilidad dinámica durante la marcha. Sin embargo, sí es un paso hacia la generación de ciclos de marcha dinámicamente estables, que, incluso, llega a formar parte de dicho proceso. Mediante este método se pueden manipular, amén de las posiciones, las velocidades y aceleraciones de los efectores finales, aspectos que son muy importantes en las fases de transición, y determinantes para que el robot no pierda el equilibrio. Además, como se verá en, mediante el empleo de este método y un algoritmo genético, se optimizará la marcha de modo que sea lo más estable posible.

5.3.1. Trayectorias de los efectores finales

Como se mencionó en 5.2, la planificación mediante parámetros consiste en la generación de trayectorias para los efectores finales del robot.

El robot bípedo Scout cuenta con tres efectores finales: los eslabones L_{61} (pie izquierdo), L_{62} (pie derecho) y L_B (torso), y para obtener la configuración espacial de cada uno de estos se requiere definir su posición y orientación. Por lo cual se requeriría de una función vectorial, o 3 funciones escalares, para definir la posición de cada eslabón, y de otra función vectorial, o tres funciones escalares, para definir su orientación. En total, se requerirían de 18 trayectorias (más de las que se requieren en la planeación en el espacio articular, 12). Pero el fin de trabajar con los efectores finales, es reducir la complejidad en la generación de trayectorias, no aumentarla. Por tanto, para reducir el número de trayectorias a generar, y por tanto el número de variables y parámetros que se pueden manipular, se considerará que los pies, eslabones L_{61} y L_{62} , permanecen con la misma orientación durante el ciclo de marcha: siempre paralelos a la superficie de apoyo. Por su parte, la cadera solo podrá cambiar su orientación con dos rotaciones representadas por los ángulos de Euler ϕ_B y ψ_B . El ángulo ϕ_B corresponde a una rotación sobre el eje y_B y el ángulo ψ_B corresponde a una rotación sobre el eje z_{ϕ_B} , referidos al marco local del eslabón L_B . En total, pasan de ser 18 trayectorias a solo 11.

Debido a que se considera que la función (vectorial) del pie derecho e izquierdo es la misma, aunque desfasada, ésta se puede considerar como una sola, como la función para la trayectoria del pie flotante. Y en total, pasan de ser 10 funciones (escalares) a solo 7.

Trayectorias para el pie flotante¹

- Función $x_{pf}(t)$ que define el movimiento del pie flotante sobre el eje x con respecto al tiempo.
- Función $y_{pf}(t)$ que define el movimiento del pie flotante sobre el eje y con respecto al tiempo.
- Función $z_{pf}(t)$ que define el movimiento del pie flotante sobre el eje z con respecto al tiempo.

Trayectorias para la cadera

- Función $x_B(t)$ que define el movimiento de la cadera sobre el eje x con respecto al tiempo.
- Función $y_B(t)$ que define el movimiento de la cadera sobre el eje y con respecto al tiempo.
- Función $z_B(t)$ que define el movimiento de la cadera sobre el eje z con respecto al tiempo.
- Función $\phi_B(t)$ que define la rotación de la cadera alrededor del eje y_B con respecto al tiempo.
- Función $\psi_B(t)$ que define la rotación de la cadera alrededor del eje z_{ϕ_B} con respecto al tiempo.

5.3.2. Efectos por la reducción del número de trayectorias y justificación

En la marcha humana, el pie cumple con distintas funciones: funge como base de apoyo y transmite las fuerzas al suelo; disminuye el impacto causado por el peso del cuerpo y las fuerzas de reacción del suelo; ayuda a mantener la estabilidad durante la marcha, y permite la progresión a través del cambio de apoyo entre el talón y la parte delantera del pie. Todas estas acciones, que se logran a través del control muscular y el movimiento que éste produce, son imposibles de imitar para el robot bípedo Scout debido a su arquitectura. Como Scout solo cuenta con un eslabón para representar al pie, no es

¹Donde pf puede referirse el eslabón 61 o 62, aunque con un desfasamiento en el tiempo.

posible copiar el patrón de postura de la boveda plantar visto en 3.2.1, lo que impide realizar la función de frenado —que tiene el pie en el momento del choque del talón con el suelo— y la función de empuje, que ocurre en el momento de despegue del pie. Asimismo, como el movimiento de los eslabones del robot es proporcionado y controlado mediante servomotores, tampoco es posible copiar las acciones musculares en los pies, que se detallan en [24], y que sirven para mantener la estabilidad y absorber el impacto.

En la marcha humana, el cambio de orientación del pie con rotación en el eje frontal, es necesario para que pueda cumplir con las funciones señaladas en el párrafo anterior. El cambio de orientación del pie con rotación en el eje vertical, sirve para cambiar la dirección de la marcha.

Debido a que el ciclo de marcha a generar tiene el propósito de que el robot camine sobre una superficie plana y sin cambiar de dirección, y a que la funcionalidad de la rotación del pie sobre el eje frontal se restringe a la fase de soporte, en la cual, por las mismas características del pie de robot, no es posible imitar; resulta innecesaria la generación de trayectorias para cambiar la orientación de los pies. Aunque con Scout, como con muchos otros bípedos, no es posible reproducir algunas de las funciones del pie humano, hay otras que no se pierden. El pie de Scout, sí funge como base de apoyo y transmite las fuerzas de la estructura al suelo, y también ayuda a mantener la estabilidad durante la marcha.

Finalmente, se consideró necesario introducir una trayectoria para definir la orientación del eslabón L_B . Incluir ϕ_B permite incrementar la longitud del paso. ψ_B permite incrementar la altura del paso e incrementar la movilidad del eslabón L_B sobre el plano frontal.

5.3.3. Función para definir a las trayectorias

Ya se han seleccionado las trayectorias que se emplearán para definir el movimiento del bípedo. Ahora, falta seleccionar la función para definir a estas trayectorias con respecto al tiempo. Para ello, se empleará la interpolación cúbica segmentaria (*spline*).

La interpolación consiste en estimar valores intermedios entre cada uno de los puntos que componen una serie de datos [7]. La interpolación segmentaria, en lugar de emplear un solo polinomio para la aproximación de un conjunto de datos, asigna un polinomio a cada par de puntos, los cuales, unidos, forman una trayectoria continua [4]. A diferencia de la interpolación lineal segmentaria, la interpolación cúbica segmentaria sí asegura la continuidad de la curva en los extremos de los intervalos. Además, la primera y segunda derivada de este polinomio, por ser de tercer grado, es cuadrática y lineal respectivamente, lo que le confiere gran flexibilidad y permite el trazo de curvas suaves.

La ecuación de un polinomio de tercer orden se define como

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (5.1)$$

Donde $q(t)$ representa a cualquiera de las trayectorias de los efectores finales.

La primera derivada de dicho polinomio es

$$\dot{q}(t) = a_1 + a_2t + a_3t^2 \quad (5.2)$$

En donde a_i son los coeficientes de la ecuación y $t \in [0, T]$.

Este polinomio requiere de cuatro restricciones. Dos se obtienen de las condiciones iniciales del movimiento, y dos, de las condiciones finales.

Evaluando al polinomio $q(t)$ y a su derivada $\dot{q}(t)$ en los extremos del intervalo tenemos

$$\begin{aligned} q(0) &= a_0 \\ \dot{q}(0) &= a_1 \\ q(T) &= a_0 + a_1T + a_2T^2 + a_3T^3 \\ \dot{q}(T) &= a_1 + a_2T + a_3T^2 \end{aligned} \tag{5.3}$$

Si se denota a la posición inicial como q_i y a la final como q_f , entonces

$$\begin{aligned} q(0) &= q_i \\ \dot{q}(0) &= \dot{q}_i \\ q(T) &= q_f \\ \dot{q}(T) &= \dot{q}_f \end{aligned} \tag{5.4}$$

Resolviendo el sistema de ecuaciones 5.3

$$\begin{aligned} a_0 &= q_i \\ a_1 &= \dot{q}_i \\ a_2 &= \frac{3(q_f - q_i) - (2\dot{q}_i + \dot{q}_f)T}{T^2} \\ a_3 &= \frac{2(q_i - q_f) + (\dot{q}_i + \dot{q}_f)T}{T^3} \end{aligned} \tag{5.5}$$

Finalmente, sustituyendo 5.5 en 5.1

$$q(t) = q_i + \dot{q}_i t + \frac{3(q_f - q_i) - (2\dot{q}_i + \dot{q}_f)T}{T^2} t^2 + \frac{2(q_i - q_f) + (\dot{q}_i + \dot{q}_f)T}{T^3} t^3 \tag{5.6}$$

En la figura 5.4 se muestra un ejemplo de una curva que se puede obtener empleando la ecuación 5.6.

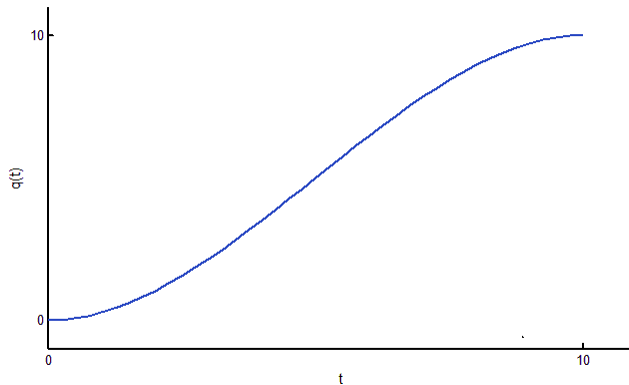


Figura 5.4: Función cúbica con $T = 10$ y $q(0) = q(T) = 0$.

5.3.4. Parámetros de tiempo para definir a la marcha

Para la generación del patrón de caminata, se considera a la marcha como un proceso con actividad alternada entre las piernas, que cambian entre el soporte doble y el soporte simple. Debido a que la función cúbica (5.6) que define las trayectorias de los efectores finales requiere de un período, éste se propone en torno a la duración de un paso, y a las duraciones del soporte simple y soporte doble 5.5:

$$\begin{aligned}
 T &\triangleq \text{duración del paso} \\
 T_{ss} &\triangleq \text{duración del soporte simple} \\
 T_{sd} &\triangleq \text{duración del soporte doble} \\
 \Delta t &\triangleq \text{intervalo de muestreo}
 \end{aligned}$$

Mientras que la posición de ambos pies permanece constante durante el soporte doble, la cadera siempre está en movimiento, por lo cual, en las funciones de las trayectorias del pie flotante se emplea al periodo del soporte simple (T_{ss}); en cambio, las *splines* cúbicas que definen el movimiento de la cadera emplean el período completo de un paso (T). $\frac{T}{\Delta t}$ indica el número de muestras en T segundos. Una cantidad reducida del número de muestras produce cambios en la estructura del bípodo, que son pocos pero grandes; mientras un valor alto de $\frac{T}{\Delta t}$ produce trayectorias para el desplazamiento del robot más suaves, aunque también implica un mayor costo computacional.

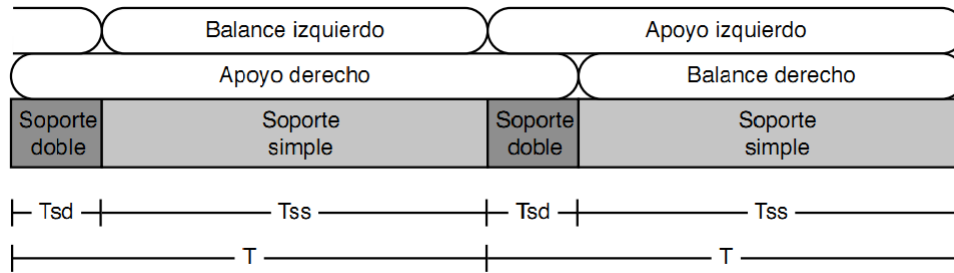


Figura 5.5: Esquema del ciclo de marcha.

5.3.5. Parámetros para definir las trayectorias del pie flotante

La función del pie se alterna entre pierna de soporte —donde permanece sobre el suelo— y pierna oscilante, donde se desplaza hacia adelante. Cuando cambia entre estas funciones, también ocurre un cambio significativo en su velocidad. Al cambiar de pierna de soporte a pierna oscilante, el pie pasa de permanecer estático a estar en movimiento; por tanto, la velocidad inicial del pie flotante es cero.

Un aspecto muy importante para lograr que el robot bípodo desarrolle una marcha estable y efectiva, es cuidar el aterrizaje del pie flotante; momento en el cual se puede llegar a producir un cambio abrupto en la velocidad del pie, provocando fuerzas de reacción en el suelo que llegan a desestabilizar la marcha. Entre más suave sea el contacto entre la planta del pie y la superficie, mejor para la estabilidad del robot. Es por eso que se considera a la velocidad final del pie flotante como cero.

Debido a que la velocidad inicial y final del pie flotante es nula, la función 5.6 se reduce a

$$q(t) = q_i + \frac{3(q_f - q_i)}{T_{ss}^2} t^2 + \frac{2(q_i - q_f)}{T_{ss}^3} t^3 \tag{5.7}$$

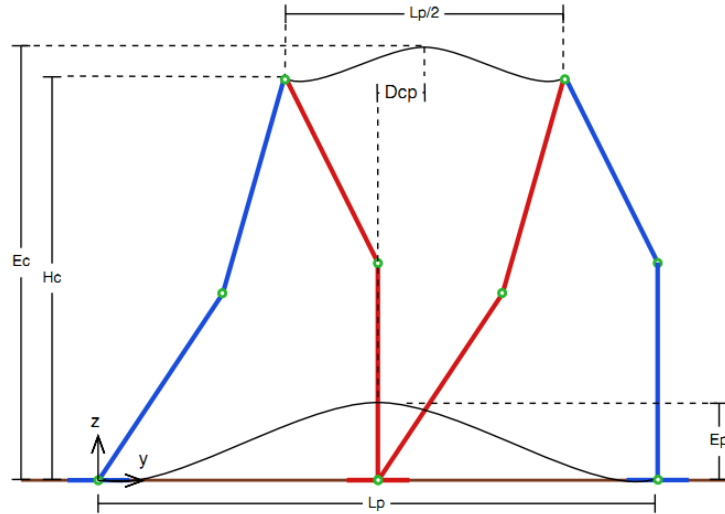


Figura 5.6: Parámetros para la generación de trayectorias. Vista sagital. La pierna de color azul es la oscilante, y la pierna roja es la de soporte.

Por lo cual, los parámetros para definir la trayectoria del pie flotante (figuras 5.6 y 5.8) se reducen a parámetros de posición; éstos son:

- $L_p \triangleq$ longitud del paso
- $E_p \triangleq$ elevación del paso
- $F_p \triangleq$ desplazamiento frontal del pie
- $P_p \triangleq$ posición frontal del pie

5.3.6. Parámetros para definir las trayectorias de la cadera

A diferencia de los pies, el movimiento de la cadera se considera constante: se mantiene en movimiento durante la fase de soporte simple y la fase de soporte doble. Para definir su posición durante la marcha se tienen a las funciones $x_B(t)$, $y_B(t)$ y $z_B(t)$, y para definir a su orientación, $\phi_B(t)$ y $\psi_B(t)$.

A pesar de que para todas las funciones se emplean *splines* cúbicas, cada una tiene sus propias características: la curva $x_B(t)$, que expresa el movimiento lateral de la cadera, guarda similitud con una curva senoidal, y la curva $z_B(t)$, que determina la altura de la cadera, se asemeja más a una señal senoidal rectificada. La función $y_B(t)$, que marca el avance en el plano sagital de la cadera, siempre se mantiene con una pendiente positiva (figura 5.7). El desplazamiento de la cadera y de los pies sobre el plano sagital es el mismo por cada ciclo de marcha. Los parámetros de posición y orientación para definir a la trayectoria de la cadera (figuras 5.6 y 5.8) son:

- $H_c \triangleq$ altura de la cadera
- $E_c \triangleq$ elevación de la cadera
- $D_{cp} \triangleq$ desplazamiento de la cadera con respecto al pie de apoyo
- $F_c \triangleq$ desplazamiento frontal de la cadera
- $\phi_c \triangleq$ rotación cadera sobre el eje y_B
- $\psi_c \triangleq$ rotación cadera sobre el eje z_{ϕ_B}

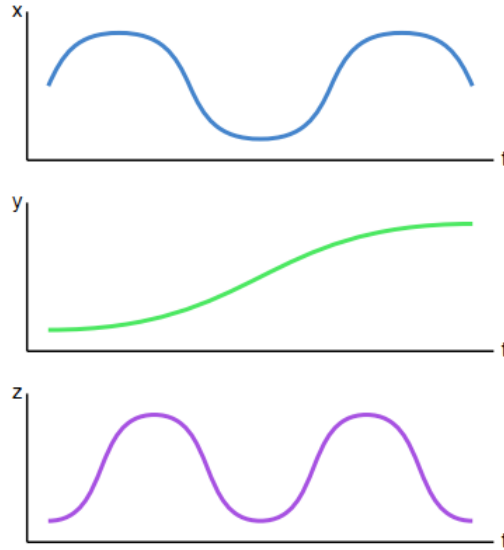


Figura 5.7: Trayectorias de la cadera.

Además, para cada una de las tres funciones que describen la trayectoria de la cadera, se cuenta con un parámetro para establecer su velocidad intermedia en $t = \frac{T}{2}$. Las funciones $x_B(t)$ y $y_B(t)$, cuyas velocidades al inicio y final del ciclo de marcha pueden ser diferentes de cero, cuentan con un parámetro más para definir la velocidad en dichos puntos. Por lo tanto, los parámetros de velocidad para definir a la trayectoria de la cadera son:

$V_{fc_o} \triangleq$ velocidad inicial y final en el desplazamiento frontal de la cadera

$V_{fc_m} \triangleq$ velocidad intermedia en el desplazamiento frontal de la cadera

$V_{ic_o} \triangleq$ velocidad inicial y final en el avance sagital de la cadera

$V_{ic_m} \triangleq$ velocidad intermedia en el avance sagital de la cadera

$V_{ec_m} \triangleq$ velocidad intermedia en la elevación de la cadera

5.3.7. Ecuaciones para definir las trayectorias de la cadera y del pie flotante

En el pie flotante, el punto en el que llega a su elevación y desplazamiento lateral máximos, resulta a la mitad del período de soporte simple. El movimiento de la cadera es constante durante la marcha, por lo cual, si se escriben sus ecuaciones de movimiento en base al período T mostrado en la figura 5.5, su elevación y desplazamiento lateral máximos no estarían a la mitad del período de soporte simple, punto que también divide a un semiciclo de marcha por la mitad. Lo cual representaría un desfase entre los movimientos de la cadera y del pie flotante, una falta de sincronía. Para evitar este problema, se desplaza hacia la derecha una distancia T_{sd} al período T , con lo que los ejes que dividen a la mitad al período T y T_{ss} coinciden y, por tanto, también los puntos en que su elevación y desplazamiento lateral son máximos. En base a la consideración anterior, se escribieron las ecuaciones de movimiento de los efectores finales.

Funciones para el pie flotante

- Función que define el movimiento del pie flotante sobre el eje x con respecto al tiempo

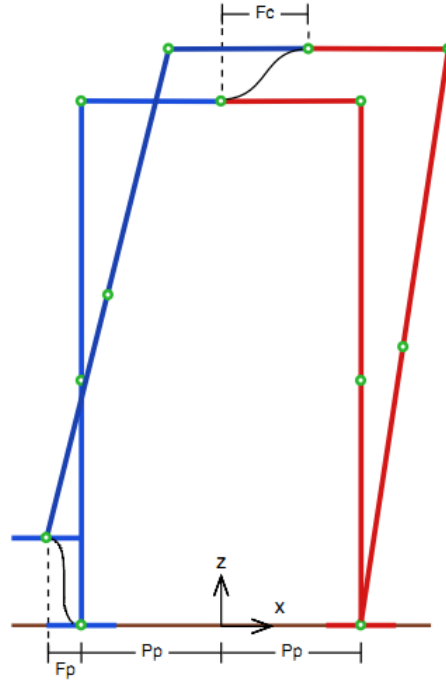


Figura 5.8: Parámetros para la generación de trayectorias. Vista frontal.

Parámetros

$F_p \triangleq$ desplazamiento frontal del pie

$P_p \triangleq$ posición frontal del pie

Ecuación

$$x_{pf}(t) = \begin{cases} P_p + 12 \frac{F_p - P_p}{T_{ss}^2} t^2 + 16 \frac{P_p - F_p}{T_{ss}^3} t^3 & \text{Si } T_{sd} \leq t < \frac{T_{ss}}{2} \\ F_p + 12 \frac{P_p - F_p}{T_{ss}^2} t^2 + 16 \frac{F_p - P_p}{T_{ss}^3} t^3 & \text{Si } \frac{T_{ss}}{2} \leq t < T_{sd} + T_{ss} \end{cases} \quad (5.8)$$

- Función que define el movimiento del pie flotante sobre el eje y con respecto al tiempo

Parámetros

$L_p \triangleq$ longitud del paso

Ecuación

$$y_{pf}(t) = \begin{cases} 0 & \text{Si } t < T_{sd} \\ 3 \frac{L_p}{T_{ss}^2} t^2 - 2 \frac{L_p}{T_{ss}^3} t^3 & \text{Si } T_{sd} \leq t < T_{sd} + T_{ss} \\ L_p & \text{Si } t \geq T_{sd} + T_{ss} \end{cases} \quad (5.9)$$

- Función que define el movimiento del pie flotante sobre el eje z con respecto al tiempo

Parámetros

$E_p \triangleq$ elevación del paso

Ecuación

$$z_{pf}(t) = \begin{cases} 0 & t < T_{sd} \\ 12 \frac{E_p}{T_{ss}^2} t^2 - 16 \frac{E_p}{T_{ss}^3} t^3 & \text{Si } T_{sd} \leq t < \frac{T_{ss}}{2} \\ E_p - 12 \frac{E_p}{T_{ss}^2} t^2 + 16 \frac{E_p}{T_{ss}^3} t^3 & \text{Si } \frac{T_{ss}}{2} \leq t < T_{sd} + T_{ss} \\ 0 & \text{Si } t \geq T_{sd} + T_{ss} \end{cases} \quad (5.10)$$

Funciones para la cadera

- Función que define el movimiento de la cadera sobre el eje x con respecto al tiempo

Parámetros

$F_c \triangleq$ desplazamiento frontal de la cadera

$V_{fco} \triangleq$ velocidad inicial y final en el desplazamiento frontal de la cadera

$V_{fcm} \triangleq$ velocidad intermedia en el desplazamiento frontal de la cadera

Ecuación

$$x_B(t) = \begin{cases} V_{fco} t + 4 \frac{3F_c - (2V_{fco} + V_{fcm})T}{T^2} t^2 - 8 \frac{2F_c - (V_{fco} + V_{fcm})T}{T^3} t^3 & \text{Si } 0 \leq t < \frac{T}{2} \\ F_c + V_{fcm} t - 4 \frac{3F_c + (2V_{fcm} + V_{fco})T}{T^2} t^2 + 8 \frac{2F_c + (V_{fcm} + V_{fco})T}{T^3} t^3 & \text{Si } \frac{T}{2} \leq t < T \end{cases} \quad (5.11)$$

- Función que define el movimiento de la cadera sobre el eje y con respecto al tiempo

Parámetros

$L_p \triangleq$ longitud del paso

$D_{cp} \triangleq$ desplazamiento de la cadera con respecto al pie de apoyo

$V_{lco} \triangleq$ velocidad inicial y final en el avance sagital de la cadera

$V_{lcm} \triangleq$ velocidad intermedia en el avance sagital de la cadera

Ecuación

$$y_B(t) = \begin{cases} D_{cp} + V_{lc_o}t + \frac{3L_p - 4(2V_{lc_o} + V_{lc_m})T}{T^2}t^2 - 4\frac{L_p - 2(V_{lc_o} + V_{lc_m})T}{T^3} & \text{Si } 0 \leq t < \frac{T}{2} \\ D_{cp} + \frac{L_p}{4} + V_{lc_m}t - \frac{3L_p + 4(2V_{lc_m} + V_{lc_o})T}{T^2}t^2 + 4\frac{L_p + 2(V_{lc_m} + V_{lc_o})T}{T^3}t^3 & \text{Si } \frac{T}{2} \leq t < T \end{cases} \quad (5.12)$$

- Función que define el movimiento de la cadera sobre el eje z con respecto al tiempo

Parámetros

$$\begin{aligned} H_c &\triangleq \text{altura de la cadera} \\ E_c &\triangleq \text{elevación de la cadera} \\ V_{ec_m} &\triangleq \text{velocidad intermedia en la elevación de la cadera} \end{aligned}$$

Ecuación

$$z_B(t) = \begin{cases} H_c + V_{ec_m}t + 6\frac{E_c - H_c}{T^2}t^2 - 8\frac{H_c - E_c}{T^3}t^3 & \text{Si } 0 \leq t < \frac{T}{2} \\ E_c + V_{ec_m}t - 6\frac{E_c - H_c}{T^2}t^2 + 8\frac{H_c - E_c}{T^3}t^3 & \text{Si } \frac{T}{2} \leq t < T \end{cases} \quad (5.13)$$

- Función que define la rotación de la cadera alrededor del eje y con respecto al tiempo.

Parámetros

$$\phi_c \triangleq \text{rotación cadera sobre el eje } z$$

Ecuación

$$\phi_B(t) = \begin{cases} 0 & t < T_{sd} \\ 12\frac{\phi_c}{T_{ss}^2}t^2 - 16\frac{\phi_c}{T_{ss}^3}t^3 & \text{Si } T_{sd} \leq t < \frac{T_{ss}}{2} \\ \phi_c - 12\frac{\phi_c}{T_{ss}^2}t^2 + 16\frac{\phi_c}{T_{ss}^3}t^3 & \text{Si } \frac{T_{ss}}{2} \leq t < T_{sd} + T_{ss} \\ 0 & \text{Si } t \geq T_{sd} + T_{ss} \end{cases} \quad (5.14)$$

- Función que define la rotación de la cadera alrededor del eje z con respecto al tiempo.

Parámetros

$$\psi_c \triangleq \text{rotación cadera sobre el eje } z$$

Ecuación

$$\psi_B(t) = \begin{cases} \psi_c & \text{Si } t < T_{sd} \\ \psi_c + -6\frac{\psi_c}{T_{ss}^2}t^2 - 4\frac{\psi_c}{T_{ss}^3}t^3 & \text{Si } T_{sd} \leq t < T_{sd} + T_{ss} \\ -\psi_c & \text{Si } t \geq T_{sd} + T_{ss} \end{cases} \quad (5.15)$$

5.3.8. Trayectorias de inicio y de paro de la marcha

En el inicio y paro de la marcha (figura 5.9), mientras un pie permanece fijo, la cadera y el otro pie se mueven, en el inicio, desde su postura inicial, y en el paro, hacia su postura final. Aunque las coordenadas espaciales de la postura inicial y final son diferentes, se considera que sus coordenadas angulares son las mismas; es decir, la postura inicial y final es idéntica, solo que se encuentran ubicadas en posiciones distintas. La postura inicial y final del robot bípedo están dadas por las coordenadas espaciales de los eslabones L_B , L_{61} y L_{62} . Las coordenadas y y z de los eslabones L_{61} y L_{62} son las mismas, y sobre el eje x se encuentran separados por la misma con respecto al centro del eslabón L_B .

El objetivo del inicio de la marcha es llevar al bípedo de la postura inicial a la posición donde comienza la etapa cíclica de la marcha; el objetivo del paro de la marcha es romper con su etapa cíclica y conseguir que el bípedo se establezca en posición estática y estable. Para cumplir con dichos objetivos, algunos de los parámetros están restringidos.

El pie flotante, el que se mueve para llegar a la postura inicial o final, debe desplazarse sobre el plano sagital el equivalente a la mitad del desplazamiento que da por cada paso durante el ciclo de marcha ($\frac{L_p}{2}$). Los demás parámetros para el pie flotante no están restringidos.

En el inicio de la marcha, la cadera debe desplazarse (en el eje sagital) de tal manera que su distancia con respecto a la posición final del pie en movimiento sea D_{cp} . Es decir

$$\frac{L_p}{4} + D_{cp} - y_B$$

Para llegar a su postura final, en el paro de la marcha, su desplazamiento es de

$$\frac{L_p}{4} - D_{cp} + y_B$$

En el inicio de la marcha, la altura de la cadera debe ir de la altura de su postura inicial (z_B) a la altura que tiene en el inicio del ciclo de marcha H_c , y la rotación de la cadera va de 0° a ψ_c . En el paro la situación se invierte.

Además, durante el inicio o paro de la marcha, las velocidades iniciales de las funciones para definir las trayectorias para la cadera son diferentes a las finales. Sin embargo, el considerar la velocidad inicial o final como cero, reduce los límites de velocidad que estas curvas pueden tener en su otro extremo, lo que implica una reducción notable en la variedad de curvas que se pueden generar; por lo tanto, se considera que las velocidades iniciales y finales V_{fc_o} y V_{fc_m} se mantienen constantes incluso durante el inicio y paro de la marcha.

5.3.9. Algoritmo

Para la síntesis de la marcha basada en la planificación de trayectorias para los efectores finales del robot en base a parámetros, se tiene el algoritmo de la figura 5.10, en el cual se observa de forma muy simplificada y resumida todo el proceso necesario para la generación de un ciclo de marcha, que va

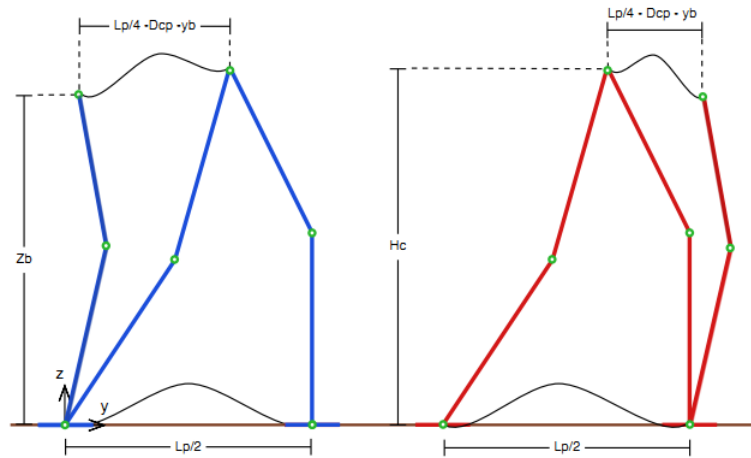


Figura 5.9: En la imagen izquierda, el inicio de la marcha, y en la imagen derecha, el paro.

desde la selección de los parámetros hasta la generación de las trayectorias angulares para cada una de las juntas rotacionales del bípodo. El proceso donde se lleva a cabo la generación de *splines*, se repite cuatro veces; dos son para la puesta en marcha y el paro del robot y dos más son para generar un ciclo de marcha completo.

5.4. Generación de un ciclo de marcha

5.4.1. Selección de parámetros

Empleando el algoritmo de la figura 5.10, se desarrolló en MATLAB un programa para la generación de ciclos de marcha. Para probarlo se usaron las siguientes entradas:

(Los parámetros de tiempo están en segundos; los de distancia, en milímetros; los de rotación en grados, y los de velocidad, en milímetros por segundo).

Posiciones iniciales

Pie izquierdo

$$x_{61} = 44.54$$

$$y_{61} = 0$$

$$z_{61} = 0$$

Pie derecho

$$x_{62} = 44.54$$

$$y_{62} = 0$$

$$z_{62} = 0$$

Cadera

$$x_B = 0$$

$$y_B = 38.5$$

$$z_B = 198.37$$

$$\phi_B = 0$$

$$\psi_B = 0$$

Parámetros

Parámetros de tiempo

$$T = 10$$

$$T_{sd} = 3$$

$$T_{ss} = 7$$

$$\Delta t = \frac{1}{3}$$

Parámetros para el pie flotante

$$L_p = 80$$

$$E_p = 50$$

$$F_p = 5$$

Parámetros de posición para la cadera

$$H_c = 198.37$$

$$E_c = 214.37$$

$$F_c = 50$$

$$D_{cp} = 0$$

$$\phi_c = 0$$

$$\psi_c = 50$$

Parámetros de velocidad para la cadera

$$V_{fc_o} = 30$$

$$V_{fc_m} = 0$$

$$V_{lc_o} = 5$$

$$V_{lc_m} = 0$$

$$V_{ec_m} = -5$$

5.4.2. Graficas de las trayectorias de los pies y torso

Las trayectorias que deben seguir los eslabones L_B , L_{61} y L_{62} del robot bípedo Scout se muestran en las figuras 5.11 y 5.12.

5.4.3. Simulación del ciclo de marcha

De acuerdo al algoritmo de la figura 5.10, una vez que han sido generadas las trayectorias para los pies y torso, se procede al cálculo de la cinemática inversa, paso en el que se obtienen las trayectorias angulares de las juntas rotacionales del bípedo, las cuales debe seguir el robot con el fin de que cumpla con el patrón de caminata generado. Aunque el algoritmo se programó en MATLAB, antes de empezar con este trabajo, el código para el cálculo de la cinemática inversa ya había sido escrito en Mathematica (éste se encuentra en [43], donde también fue empleado para la simulación de un ciclo de marcha), por lo que se empleó el programa *Mathematica Symbolic Toolbox for MATLAB* [6] mediante el cual se pueden ejecutar comandos de Mathematica desde MATLAB e importar sus resultados. La figura 5.13 muestra capturas con la configuración adoptada por el robot durante el patrón de caminata generado; éstas se extrajeron de una simulación que forma parte del programa realizado en Mathematica.

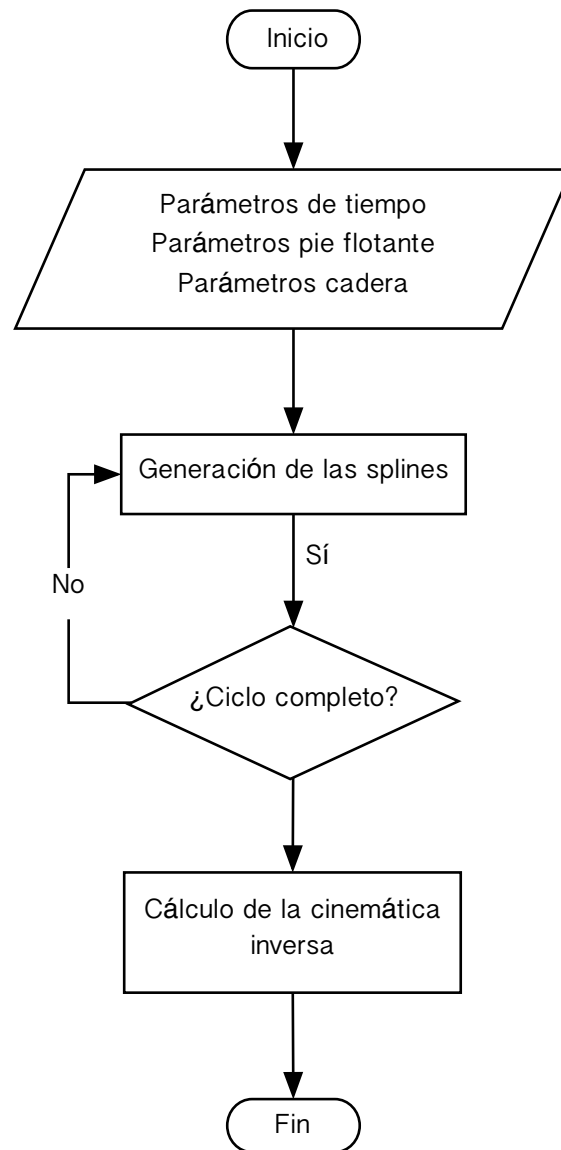


Figura 5.10: Diagrama de flujo para la planificación de la caminata.

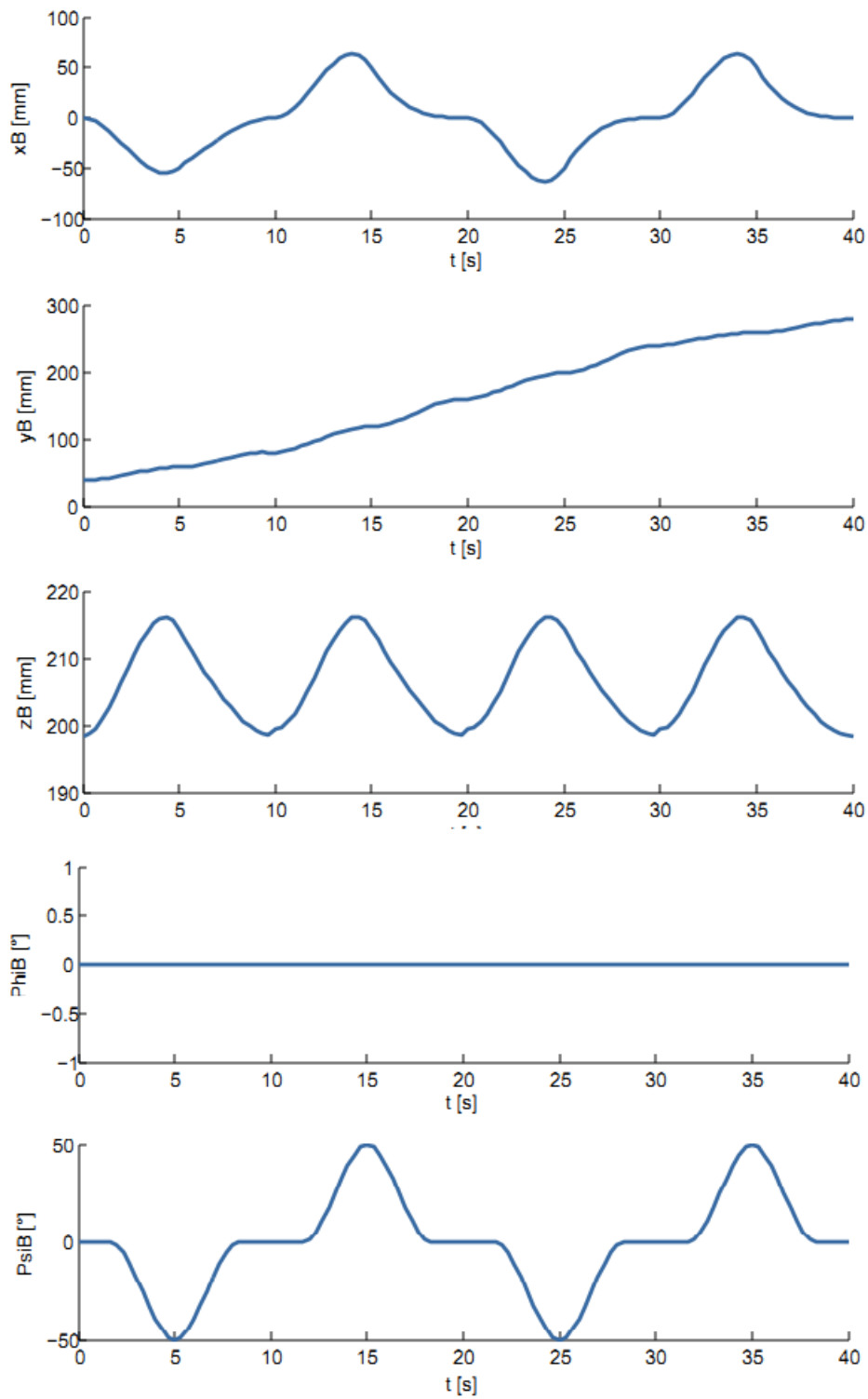


Figura 5.11: Trayectorias para definir la posición y orientación del eslabón L_B según los parámetros seleccionados.

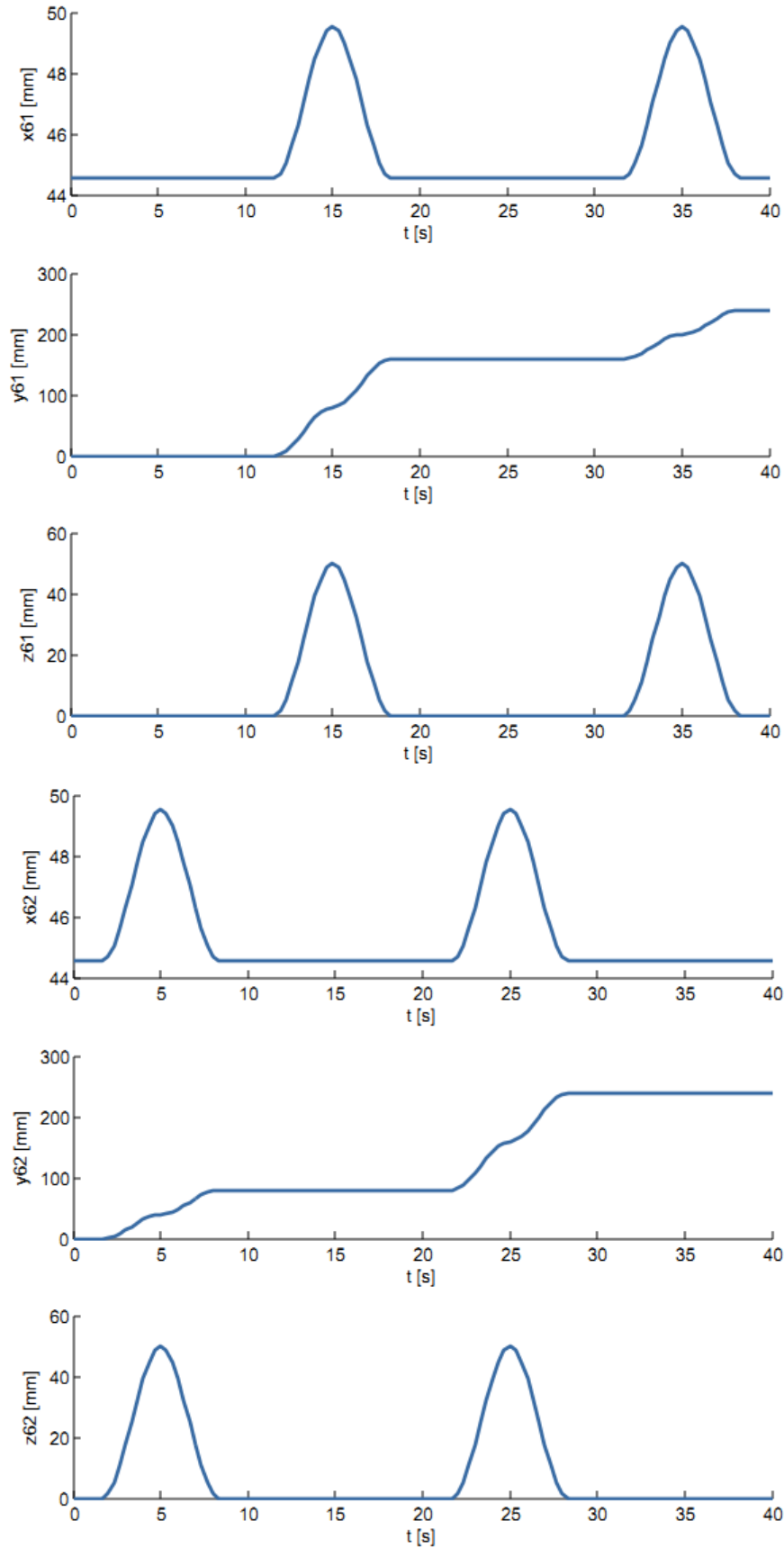


Figura 5.12: Trayectorias para definir la posición de los eslabones L_{61} y L_{62} según los parámetros seleccionados.

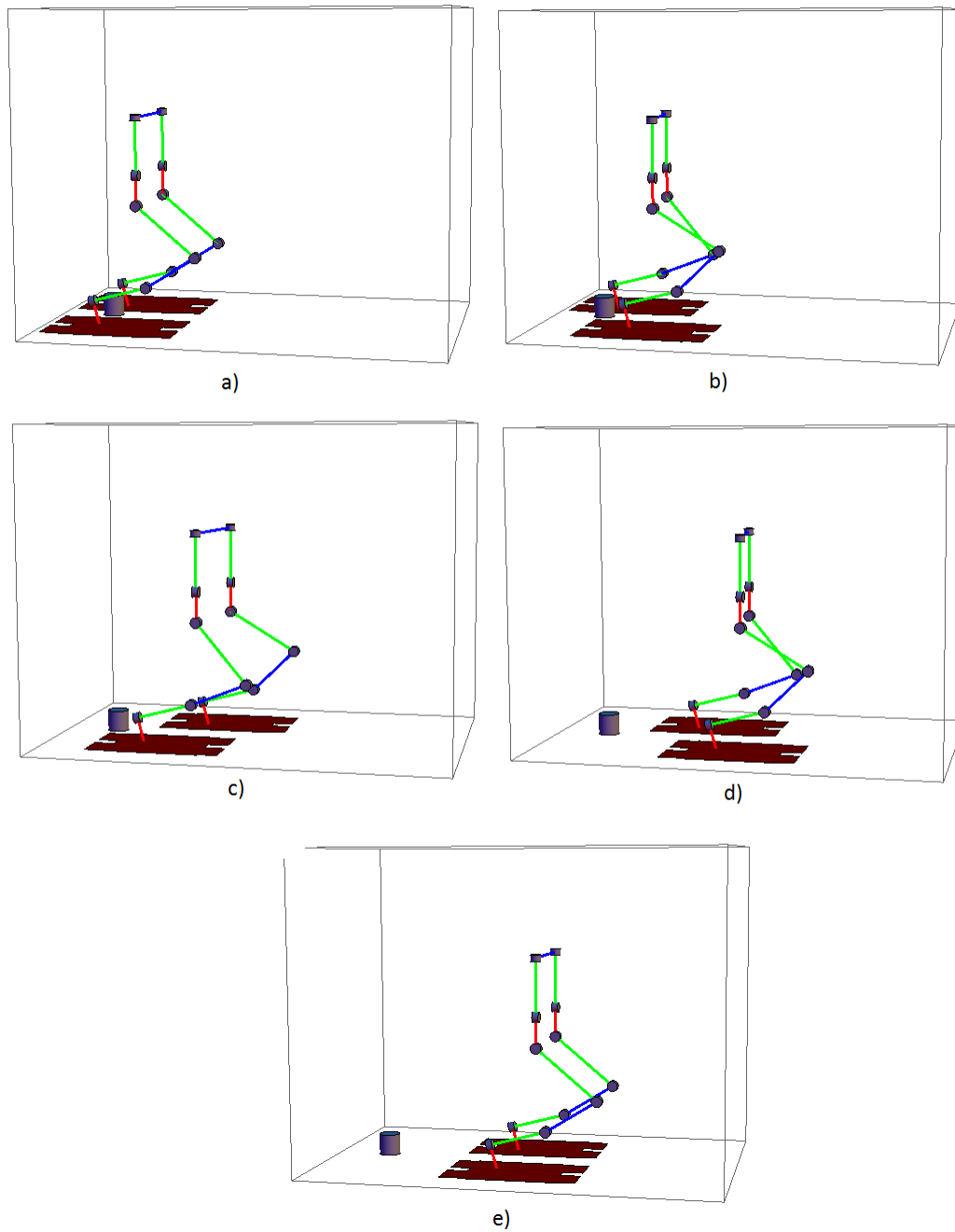


Figura 5.13: Simulación de la caminata del robot bípedo Scout. a) ($t=0$) corresponde a la postura inicial del robot, justo cuando comienza su puesta en marcha. En b) ($t=10$) comienza el ciclo de marcha del robot. c) ($t=20s$) y e) ($t=30s$) son las etapas de soporte doble de los pasos izquierdo y derecho del robot. e) ($t=40$) corresponde al paro del robot, donde regresa a su postura inicial.

Capítulo 6

Optimización de la marcha

Existen distintas técnicas para el aprendizaje y la optimización de los ciclos de marcha; éstas se basan en objetivos, tales como la minimización del gasto energético, la maximización del margen de estabilidad o la velocidad del aprendizaje. Para ello se emplean diversas técnicas de inteligencia artificial: redes neuronales, aprendizaje por reforzamiento, aprendizaje por imitación o algoritmos genéticos.

Los algoritmos genéticos son una técnica de la inteligencia artificial empleada para encontrar soluciones exactas o aproximadas a problemas de optimización y búsqueda. Dentro de la robótica bípeda, se les considera una herramienta bastante útil en la resolución de problemas relacionados con la optimización de la marcha. Para evaluar a los elementos individual y colectivamente, se emplea una función de evaluación según el objetivo; ésta puede considerar, por ejemplo, a la energía consumida durante la marcha o a la relación entre la posición del ZMP y del área de soporte.

Aquí, siendo la planificación mediante parámetros el método empleado para la generación de patrones de marcha, se emplean a los algoritmos genéticos para optimizarlos, de manera que los parámetros que definen a las trayectorias seguidas por los efectores finales del robot, sean aquellos que aseguren la mayor estabilidad de éste durante la ejecución del ciclo de marcha. (Este método parte de la propuesta hecha por Goswami Dip, publicada y detallada en [12].)

6.1. Propuesta empleada y justificación

Con el objetivo de asegurar la estabilidad dinámica de un robot, se han propuesto diversos métodos para la síntesis de la marcha basados en el criterio del ZMP. Debido a la cantidad de masa de la parte superior del cuerpo —lo cual origina que sus desplazamientos afecten en gran medida a la estabilidad del robot—, originalmente, los métodos para la generación de ciclos de marcha dinámicamente estables, consistían en diseñar la trayectoria del ZMP deseada y generar el movimiento del torso requerido para cumplir con tal trayectoria. En estos casos, entre la posición del ZMP está más cercana al centro de la región de estabilidad, área de soporte, la marcha es más estable. Sin embargo, hay una limitación: la capacidad de alterar la posición del ZMP cambiando la posición de la cadera, está limitada por las particularidades físicas del robot, tales como su arquitectura, dimensiones y peso. Otra propuesta para formular el problema de estabilidad dinámica, consiste en un método iterativo, el cual busca la trayectoria para la cadera que asegura, durante la marcha, el margen de estabilidad más amplio. En

este caso no se plantea una trayectoria del ZMP a seguir, pero sí se evalúa al movimiento de la cadera en base al efecto que tiene en la ubicación del ZMP. Esta idea no se limita a buscar la manera en la que se cumpla con una trayectoria del ZMP prescrita que, sí, asegura la estabilidad dinámica de la marcha, pero que, también, probablemente, ni siquiera sea posible llevar a cabo; en cambio, se elimina la necesidad de cumplir con una trayectoria del ZMP propuesta, que es canon para la estabilidad de la marcha, y se pasa a un problema abierto, donde el mismo espacio de búsqueda puede tener soluciones múltiples que representan diversas trayectorias del ZMP, pero todas dentro de un margen de estabilidad aceptable. Si se sigue esta propuesta, debido a la naturaleza del problema y a su planteamiento, los algoritmos genéticos son una alternativa viable para dar solución al problema, pues destacan por su capacidad de explorar el espacio de búsqueda amplia y eficientemente [29], haciendo una barrida al espacio de posibles soluciones válidas.

6.2. Los algoritmos genéticos

Los algoritmos genéticos están basados en los mecanismos de selección natural y genética natural; mediante ellos se navega por espacios de búsqueda amplios con el fin de encontrar soluciones óptimas. De acuerdo a los principios de la selección natural, a lo largo de las generaciones, las poblaciones evolucionan. Los algoritmos genéticos replican este proceso biológico mediante la computación en búsqueda de soluciones ante problemas del mundo real. Debido al uso de esta analogía, se valen de operadores aleatorios para —en vez de buscar un único punto— guiar la búsqueda de los mejores puntos de una población. Para ello emplean una función objetivo.

Un algoritmo genético consta de una población formada por un conjunto de “criaturas artificiales” (cromosomas (cadenas de bits)) que representan posibles soluciones ante algún problema [49]. Mediante un proceso que simula a la evolución biológica, se seleccionan, de acuerdo con su capacidad de reproducción, a algunos cromosomas de la población y se cruzan para producir nuevos elementos que dan paso a una nueva generación. A través del paso de generaciones, se obtienen individuos cada vez mejores, que ofrecen una mejor solución ante el problema planteado.

Cualquier solución potencial a un problema puede ser presentada dando valores a una serie de parámetros. El conjunto de todos éstos se codifica en una cadena de parámetros denominada cromosoma. En la codificación se asigna un determinado número de bits a cada parámetro y se realiza una discretización de la variable presentada por cada gen. Cada individuo tiene asociado un ajuste —*fitness*— de acuerdo a su efectividad con la que resuelve el problema.

Para la generación de los descendientes, el algoritmo cuenta con los operadores de cruce y mutación, inspirados en la genética. El operador de cruce extrae algunos elementos de dos cromosomas y los combina para formar uno nuevo. La mutación, de modo aleatorio, cambia el valor de los bits en algunas localizaciones del nuevo cromosoma.

Los algoritmos genéticos operan de forma simultánea con varias soluciones, lo que les facilita escapar de los máximos locales. Aunada a esta ventaja existe cierta incertidumbre sobre su convergencia, pues pueden tardar mucho en lograrlo e, incluso, nunca converger.

6.2.1. Bases biológicas

Los seres vivos estamos compuestos por aparatos, grupos de sistemas biológicos que desempeñan una función común. Por ejemplo, el aparato locomotor está integrado por los sistemas muscular, oseo,

articular y nervioso. Cada sistema es un grupo de órganos, los cuales son estructuras constituidas por varios tejidos que realizan una función conjunta. Por ejemplo, los huesos, que proporcionan sostén y ayudan a darle forma al cuerpo [55], son órganos formados por tejido que es, a la vez, fuerte, resistente y ligero; los músculos, formados con tejido elástico [55], también son órganos que, acortando su longitud como parte de la estimulación de impulsos provenientes del sistema nervioso, generan movimientos [18]. El nivel organizacional de los seres vivos sigue bajando: los tejidos son conjuntos de células especializadas en realizar una actividad determinada.

Las células son la base de todos los organismos; de su unión nacen los tejidos y órganos que forman a los seres vivos más evolucionados [5]. Cada célula contiene cromosomas (cadenas de ADN), que son los portadores de los factores hereditarios.

Conceptualmente, un cromosoma se divide en genes, bloques funcionales de ADN que determinan las características del individuo de acuerdo a la configuración que adoptan, alelo, y cuya información es transmitida de generación en generación. El conjunto de todos los genes de un individuo constituye el genotipo.

La mayoría de las especies reproductoras sexualmente almacenan sus cromosomas por parejas. En el ser humano, cada célula somática contiene 23 pares de cromosomas. Durante la reproducción sexual se produce una recombinación o cruce: en cada padre, se intercambian los genes entre cada par de cromosomas para formar uno solo (gameto). Después los gametos de los padres se emparejan, generando un conjunto completo de cromosomas diploides (pares de cromosomas). Durante la descendencia pueden ocurrir cambios en la información contenida en el ADN de la célula, que se conocen como mutaciones. A la frecuencia con la que se producen las mutaciones de un gen se le denomina tasa de mutación [1]. Una tasa de mutación alta implica un mayor potencial de adaptación, pues permite explorar más variantes genéticas, pero, a su vez, se aumenta el número de mutaciones perjudiciales, haciendo a los individuos menos adaptados. Las mutaciones representan una fuente de variabilidad; provocan un cambio gradual en la estructura genética de las poblaciones [5]. Sin embargo, las mutaciones son un evento estocástico: los cambios que producen no tienen ninguna relación directa con el éxito o eficacia de los organismos que la sufren [1]. En cambio, la selección es el proceso ordenado mediante el cual se escogen, de entre todas las variantes existentes, aquellas que sean más útiles para el organismo. La selección natural es un proceso acumulativo que permite incorporar pequeñas mejoras generación tras generación, y, como consecuencia, ocasiona un cambio en la proporción de las variantes genéticas, produciendo adaptaciones de acuerdo con el contexto ecológico de la población y generando un cambio directo en su capacidad de sobrevivencia y reproducción.

6.2.2. Elementos y operadores de un algoritmo genético

En los algoritmos genéticos, el término cromosoma se refiere a un candidato a la solución del problema, que, a menudo, se codifica como una cadena de bits. Los cromosomas se conforman por genes, que son bloques de bits adyacentes que codifican un elemento particular del candidato a la solución. Es simple: los cromosomas están formados por genes, y los genes, por bits. Al valor que puede adquirir cada bit, un 1 o un 0, se le llama alelo, y a la configuración de bits que adquiere un cromosoma se le llama genotipo (figura 6.1).

Los algoritmos genéticos cuentan con tres operadores básicos: selección, cruce y mutación [49].

El operador de selección escoge cromosomas entre la población para efectuar la reproducción. Los elementos participantes en el cruce se pueden seleccionar aleatoriamente o mediante algún proceso que

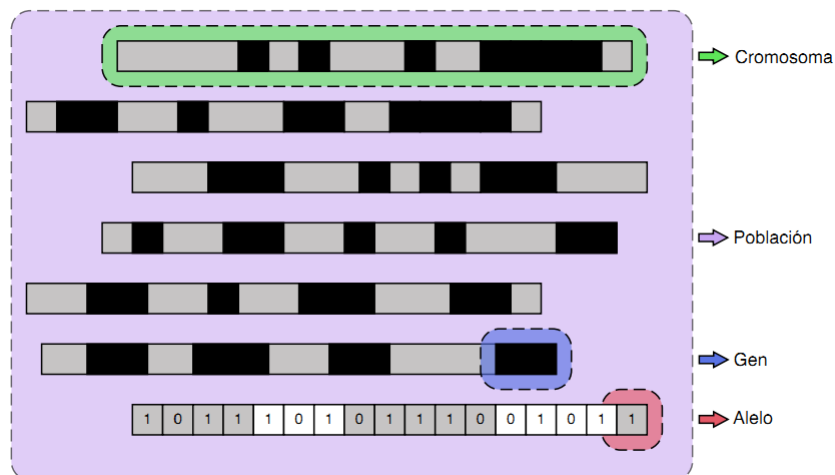


Figura 6.1: Elementos básicos de un algoritmo genético

haga uso del ajuste de los individuos. Entre los métodos de la segunda categoría, los más comunes son la selección por ruleta y la selección por torneo. En la selección por ruleta se le asigna a cada individuo de la población una parte de una ruleta proporcional a su ajuste, de tal forma que la suma de todos los porcentajes sea la unidad. Para seleccionar un individuo se genera un número aleatorio dentro del intervalo $[0, 1]$ y se devuelve al individuo que abarca dicho valor de la ruleta. Por su parte, la selección por torneo consiste en elegir a los individuos en base a comparaciones directas entre ellos: se seleccionan al azar a dos o más individuos de la población, y de entre éstos se selecciona al más apto para pasarlo a la siguiente generación.

El operador de cruce es una representación computacional de la reproducción sexual: genera un nuevo cromosoma a partir de dos que funcionan como progenitores. La idea principal del cruce se basa en que, si se toman dos individuos correctamente adaptados al medio y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes del ajuste de los padres [37]. Por tanto, al compartir características *buenas* de dos individuos, es posible que la descendencia tenga elementos con un ajuste mayor al de cada uno de los padres por separado. La técnica de cruce más sencilla, cruce de 1 punto, consiste en cortar a los cromosomas por un punto seleccionado aleatoriamente —de forma que se divida al cromosoma en dos segmentos—, seleccionar por cada cromosoma un segmento de manera que cada uno corresponda a parámetros distintos, y, finalmente, unir a dichos segmentos para formar un nuevo individuo. Otro método, muy similar al anterior, consiste en fraccionar a los cromosomas pero en más partes.

El operador de mutación realiza la permutación de un bit en algún lugar aleatorio del cromosoma. Generalmente, la probabilidad de mutación es muy baja, porque, regularmente, los individuos disminuyen su ajuste después de haber mutado. Sin embargo, las mutaciones garantizan la variabilidad, de manera que cualquier punto en el espacio de búsqueda tiene la probabilidad de ser analizado.

6.2.3. Evaluación y condiciones de paro

La clave en el éxito de un algoritmo genético radica en su capacidad de evaluar a los elementos de la población, de lo cual se encarga la función objetivo (también llamada función de evaluación), que establece una medida numérica, llamada ajuste o *fitness*, de la bondad de la solución [37]. El

ajuste permite controlar el número de selecciones, cruces y mutaciones llevadas a cabo. La función de evaluación se diseña en base al problema que se está tratando de resolver, y tiene más que ver con éste que con el propio algoritmo genético.

El algoritmo consta de un ciclo mediante el que se avanza de generación en generación. Éste para cuando los mejores individuos de la población presentan soluciones suficientemente buenas para el problema que se desea resolver, cuando la población ha convergido o cuando se ha alcanzado el número de generaciones máximo especificado.

6.3. Algoritmo genético

Para la optimización de la marcha, el algoritmo genético cuenta con los operadores básicos: selección, cruce y mutación. También cuenta con una función objetivo que evalúa a cada cromosoma según la estabilidad dinámica de su marcha. La población está compuesta por cromosomas cuyos genes contienen a los parámetros de posición y velocidad que definen a la trayectoria de la cadera. Los parámetros que definen a la trayectoria del pie flotante funcionan como entradas.

6.3.1. Aclaraciones

La propuesta a seguir no tiene la intención de buscar el movimiento de la cadera mediante el cual el bípedo sigue una trayectoria del ZMP, sino, simplemente, dadas las pisadas de los pies, busca una secuencia de movimientos para la cadera, con el propósito de que la trayectoria del ZMP se encuentre dentro de un margen de estabilidad considerable: es decir, el método empleado no asegura que se pueda generar un ciclo de marcha que en todo momento asegure la estabilidad dinámica del robot, pero al menos hace una búsqueda exhaustiva con el objetivo de encontrar dicho ciclo. Las razones por las que el algoritmo puede ser incapaz de encontrar este ciclo pueden tener dos orígenes. Por una parte, se debe recordar que el método solo trabaja con la variación del movimiento de la cadera dadas las trayectorias de los pies; por lo tanto, es posible que con las condiciones impuestas bajo las que se desea realizar el ciclo de marcha —las trayectorias de los pies— no exista ninguna solución posible que en todo momento asegure la estabilidad dinámica. Por otra parte, se encuentran las propias limitaciones físicas del mecanismo. Sin embargo, lo que este método sí asegura es encontrar, dadas ciertas condiciones, el ciclo de marcha óptimo. Éste es un método en el que el algoritmo empleado explora el espacio de soluciones con el fin de optimizar la marcha. Para ello, se cuenta con dos herramientas: primero, el algoritmo generado en el capítulo anterior para la síntesis de la marcha, y, segundo, los algoritmos genéticos.

En [11] proponen un método para la optimización de la marcha empleando algoritmos genéticos. Es importante notar que relacionan directamente a la velocidad del bípedo con el largo de su paso: entre más largo es el paso, mayor es su velocidad. Entonces, la propuesta consiste en realizar una compensación (*tradeoff*) entre el margen de estabilidad y la velocidad de la marcha, bajo el argumento de que ambas propiedades son dependientes y excluyentes: el incremento en la velocidad (tamaño del paso) produce un decremento en el margen de estabilidad y viceversa. En el algoritmo que propone, la única entrada cuantifica la valoración que se le da a la estabilidad en proporción con la que se le da a la velocidad.

La propuesta que aquí se plantea consiste en dar libertad al usuario en la elección de los parámetros para la generación de las trayectorias de los pies, y en función de ellas encontrar la trayectoria óptima

de la cadera. La ventaja con la que se cuenta es el número de parámetros para definir a la trayectoria de la cadera, mediante los cuales no solamente se puede definir la posición de la cadera en algunos puntos, sino también su velocidad, aumentando notablemente la gama de trayectorias posibles y, por tanto, el espacio de búsqueda para el algoritmo.

6.3.2. Parámetros

Los valores de entrada al algoritmo genético son aquellos que sirven para configurar al tiempo y para definir la trayectoria del pie flotante. A diferencia del capítulo 6, aquí se usan dos parámetros para la configuración del tiempo. Uno, T , representa el tiempo que tarda el robot en dar un paso, y el otro Tp_{sd} representa el porcentaje de T en el que el bípido se encuentra en la fase de soporte doble.

$T \triangleq$ *duración del paso*

$Tp_{sd} \triangleq$ *porcentaje soporte doble*

Los parámetros para el pie flotante siguen siendo los mismos. Vale la pena recordar que los ciclos de marcha que aquí se desarrollan, y bajo los cuales se planteó la generación de patrones de caminata del capítulo 6, cumplen con la característica de obedecer a movimientos sincronizados entre las piernas, cuyas fases son simétricas. Por lo tanto, los parámetros que definen al pie flotante son los mismos, independientemente de que éste represente a la pierna derecha o izquierda.

$L_p \triangleq$ *longitud del paso*

$E_p \triangleq$ *elevación del paso*

$F_p \triangleq$ *desplazamiento frontal del pie*

6.3.3. Representación

Uno de los problemas en el desarrollo de un algoritmo genético es el de definir de qué forma los cromosomas corresponden con la solución de algún problema [10]. Para ello, se deben encontrar parámetros que en conjunto representen a una solución, los cuales fungen como los genes de los cromosomas. En este problema, los genes de los cromosomas son el subconjunto de parámetros empleados en la generación de ciclos de marcha que definen al movimiento del eslabón L_B del bípido.

Parámetros de posición para la cadera

$H_c \triangleq$ *altura de la cadera*

$E_c \triangleq$ *elevación de la cadera*

$D_{cp} \triangleq$ *desplazamiento de la cadera con respecto al pie de apoyo*

$F_c \triangleq$ *desplazamiento frontal de la cadera*

$\phi_c \triangleq$ *rotación cadera sobre el eje y*

$\psi_c \triangleq$ *rotación cadera sobre el eje z*

Parámetros de velocidad para la cadera

$V_{f_{c_o}} \triangleq$ *velocidad inicial y final en el desplazamiento frontal de la cadera*

$V_{f_{c_m}} \triangleq$ *velocidad intermedia en el desplazamiento frontal de la cadera*

$V_{i_{c_o}} \triangleq$ *velocidad inicial y final en el avance sagital de la cadera*

$V_{i_{c_m}} \triangleq$ *velocidad intermedia en el avance sagital de la cadera*

$V_{e_{c_m}} \triangleq$ *velocidad intermedia en la elevación de la cadera*

Mediante la codificación, las soluciones son representadas empleando cadenas de bits con el propósito de facilitar la aplicación de los operadores de selección, cruce y mutación al cromosoma. Particularmente, en este problema, ni el operador de selección ni el de cruce requieren contar con codificación alguna; se puede trabajar directamente con los valores que representan a cada uno de los parámetros para definir la trayectoria de la cadera, los cuales son números reales enteros. Sin embargo, en la mutación sí se requiere contar con codificación, pero solo del individuo al que se le realizará una permutación en alguno de sus bits.

6.3.4. Generación de la población inicial

La población inicial es la principal fuente de material genético para el algoritmo, y debe contener cromosomas que estén bien dispersos en el espacio de soluciones [10]. La forma más fácil y rápida para generar la población inicial es seleccionando los parámetros o genes de los cromosomas al azar. Las restricciones mecánicas del robot limitan los valores de los parámetros a elegir. Por lo tanto, aunque los parámetros son generados al azar, éstos se deben encontrar dentro de cierto rango.

6.3.5. Límites de los parámetros para la trayectoria de la cadera

Los valores máximos y mínimos de los parámetros de posición se obtuvieron experimentalmente, manipulando al prototipo físico desde la interfaz gráfica, y los máximos y mínimos de los parámetros de velocidad se obtuvieron observando los efectos que causaban sus valores en la trayectoria a seguir. Por ejemplo, para la trayectoria de la cadera sobre el eje x , se pueden generar cualquier tipo de trayectorias, siempre y cuando éstas se encuentren del lado del pie de soporte. Cualquier valor menor a cero en la velocidad inicial del desplazamiento frontal de la cadera (V_{fc_o}) produce una pendiente negativa, con lo que se obtiene una trayectoria no deseada debido a que el centro de masa se aleja del área de soporte; entonces el límite inferior para V_{fc_o} es cero. Sin embargo, el problema no es tan simple para encontrar los máximos y mínimos de la velocidad intermedia en el desplazamiento frontal de la cadera (V_{fc_m}): como se puede observar en las imágenes b) y c) de la figura 6.2, los mismos valores para V_{fc_m} pueden producir una curva que cumple con las condiciones establecidas, o bien, una curva con características no deseadas.

El problema es que las curvas no solamente están en función de un parámetro, sino de varios; por lo tanto, no existen límites fijos para V_{fc_m} entre los cuales se pueda generar cualquier curva que cumpla con las condiciones establecidas. Una solución sería encontrar el rango para V_{fc_m} sobre el cual no hay ninguna curva con características no deseadas. Sin embargo, se reduciría la variedad de curvas que se pueden generar y, en consecuencia, el espacio de soluciones. Otra opción sería encontrar alguna función que determine el rango de los parámetros que no se han establecido en función de los que ya han sido definidos. Por ejemplo, para generar la curva que define el movimiento de la cadera sobre el eje x , se tienen (además de T) los parámetros F_c , V_{fc_o} y V_{fc_m} . Entonces, primero se establecería el valor del desplazamiento frontal de la cadera. Después, dado F_c , con una función, se sabría sobre qué rango puede estar algún parámetro de velocidad y se establecería su valor. Finalmente, dados dos parámetros, se determinarían los límites del parámetro faltante y se le asignaría algún valor. Esta opción abarcaría todo el espacio de soluciones posibles; sin embargo, encontrar tal función no es tarea sencilla; además, las condiciones para generar las trayectorias de la cadera sobre los ejes x , y y z difieren, por lo que se requeriría de tres funciones diferentes. La alternativa por la que se opta consiste en un método iterativo

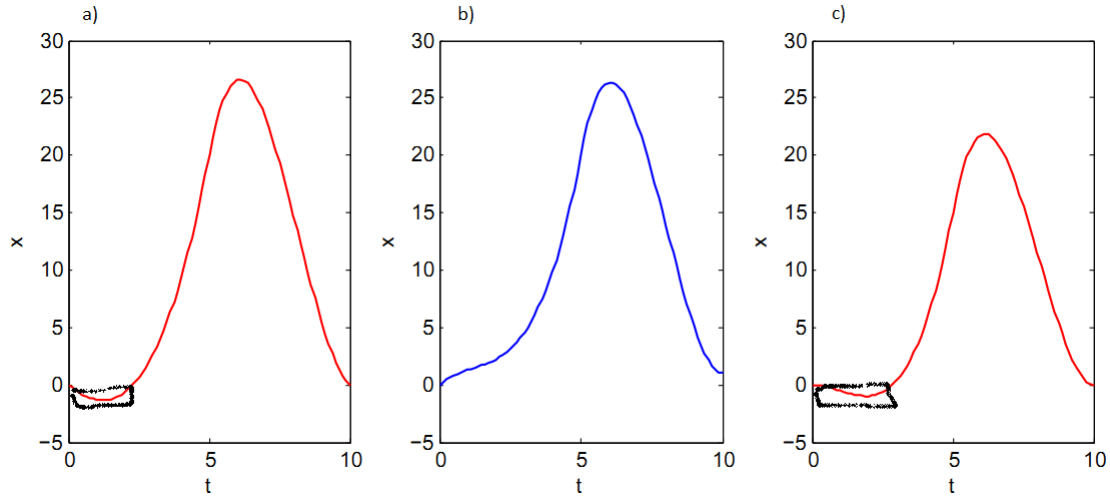


Figura 6.2: Diferentes trayectorias de la cadera sobre el eje x para un semiciclo de marcha. Los parámetros para la curva a) son $F_c = 20$, $V_{fc_o} = -1$ y $V_{fc_m} = 6.5$. Para la curva b) son $F_c = 20$, $V_{fc_o} = 2$ y $V_{fc_m} = 6.5$. Y para la curva c) son $F_c = 15$, $V_{fc_o} = 0$ y $V_{fc_m} = 6.5$.

donde los parámetros de las curvas se seleccionan al azar. Después se examina la curva: se contabiliza su número de pendientes y sus máximos y mínimos, y con los resultados obtenidos se determina si se está cumpliendo con las condiciones establecidas; entonces, sus parámetros se seleccionan como genes para formar cromosomas. En caso contrario, se vuelven a seleccionar nuevos parámetros aleatoriamente.

Así como los límites de los parámetros de posición se seleccionaron en base a las limitaciones físicas del robot, los límites de los parámetros de velocidad para cada curva se encontraron experimentalmente, después de haber generado varias funciones y observado sus gráficas, como las mostradas en la figura 6.2.

Límites para los parámetros de posición para la cadera
(Los parámetros, en milímetros. ϕ_c y ψ_c , en grados).

$$189 \leq H_c \leq 239$$

$$189 \leq E_c \leq 239$$

$$-25 \leq D_{cp} \leq 25$$

$$15 \leq F_c \leq 80$$

$$0 \leq \phi_c \leq 75$$

$$0 \leq \psi_c \leq 55$$

Límites parámetros de velocidad para la cadera

(Todos los parámetros, en milímetros por segundo).

$$0 \leq V_{fc_o} \leq 35$$

$$-10 \leq V_{fc_m} \leq 10$$

$$0 \leq V_{lc_o} \leq 35$$

$$0 \leq V_{lc_m} \leq 15$$

$$-15 \leq V_{ec_m} \leq 15$$

6.3.6. Función objetivo

La función objetivo evalúa a cada uno de los individuos y mide la calidad de la solución que representan. Es la herramienta que permite simular el concepto de individuos mejor adaptados [28]. Aquellos individuos cuya función de evaluación refleja un mayor nivel de adaptación, tienen más posibilidades de ser elegidos para construir a la siguiente población.

El concepto del ZMP se usa para garantizar que la base de soporte de un robot bípedo guarde contacto con el suelo y que un patrón de caminata pueda llevarse a cabo tal y como fue planeado. Teóricamente, si el ZMP está dentro del área convexa formada por los puntos de contacto del pie y del suelo, el robot puede caminar de manera estable. Sin embargo, si el ZMP se encuentra cercano a los límites del polígono de soporte, la sensibilidad del bípedo a perturbaciones es mayor. Para evitar este problema se suele recurrir a un área de seguridad, que es un subespacio del polígono de soporte que excluye a las zonas cercanas a sus límites (figura 6.3). La estabilidad de los sistemas bípedos se cuantifica por la distancia del ZMP con respecto al polígono de soporte [12].

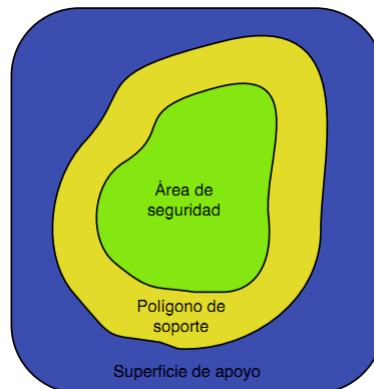


Figura 6.3: Áreas de interés para determinar la estabilidad del bipedo de acuerdo al punto donde esté ubicado el ZMP.

Entre más cercano está el ZMP al centro del polígono de soporte, mayor es su margen de estabilidad.

Por lo tanto, la función que se usa para evaluar a los individuos del algoritmo genético emplea como base a la posición del ZMP con respecto al polígono de soporte.

Sea la función

$$q = x_{zmp}^2 + y_{zmp}^2 \quad (6.1)$$

donde

$$x_{zmp} \triangleq \text{posición del ZMP en el eje } x$$

$$y_{zmp} \triangleq \text{posición del ZMP en el eje } y$$

y donde el origen desde el cual se mide tanto a x_{zmp} como a y_{zmp} corresponde al centro del polígono de soporte.

Por tanto, entre menor sea el valor de la función q , mayor es la estabilidad del robot.

Como el ZMP se evalúa durante todo el ciclo de marcha y no solamente en un punto, se define a la función

$$Q(t) = x(t)_{zmp}^2 + y(t)_{zmp}^2 \quad (6.2)$$

que calcula la trayectoria del ZMP durante el ciclo de marcha con respecto al centro del polígono de soporte. Entonces, el ciclo de marcha óptimo es aquel que minimiza a la función Q .

Aunque la función Q es una buena opción para evaluar a los individuos del algoritmo genético, el resultado puede ser engañoso. Por ejemplo, la figura 6.4 muestra dos trayectorias del ZMP sobre el eje x en las cuales al calcular la función Q se obtienen resultados bastante similares. Si estas trayectorias fueron generadas a partir de dos cromosomas —conjunto de parámetros (genes) para definir el movimiento de la cadera—, significa que ambos son igual de aptos. Sin embargo, se puede observar que en un momento la curva a) sale fuera de los límites del área de seguridad, y, de acuerdo al criterio empleado, el ciclo de marcha se vuelve dinámicamente inestable en dicho intervalo. Al contrario, la curva b), aunque cercana a sus límites, siempre se encuentra dentro del área de seguridad, por lo cual el ciclo de marcha es dinámicamente estable en todo momento. Si el valor de la función Q de ambas curvas fuera similar, se estaría evaluando a los individuos del algoritmo genético erróneamente, y aunque éste pudiera encontrar una solución, podría no ser la correcta, pues la función Q pone en el mismo nivel a cromosomas con diferente aptitud. Para resolver este problema se propone a Q como una función definida a trozos:

$$Q(t) = q_a(t) + q_b(t) \quad (6.3)$$

con

$$q_a(t) = \begin{cases} 0 & \text{Si } x_m \leq x_{zmp} \leq x_M \\ x(t)_{zmp}^2 & \text{Si } x_{zmp} < x_m \text{ o } x_{zmp} > x_M \end{cases} \quad (6.4)$$

$$q_b(t) = \begin{cases} 0 & \text{Si } y_m \leq y_{zmp} \leq y_M \\ y(t)_{zmp}^2 & \text{Si } y_{zmp} < y_m \text{ o } y_{zmp} > y_M \end{cases} \quad (6.5)$$

donde

$x_m \triangleq$ límite inferior del área de seguridad sobre el eje x

$x_M \triangleq$ límite superior del área de seguridad sobre el eje x

$y_m \triangleq$ límite inferior del área de seguridad sobre el eje y

$y_M \triangleq$ límite superior del área de seguridad sobre el eje y

Asimismo, con la nueva función, dos trayectorias del ZMP que se mantienen dentro del área de seguridad tienen el mismo nivel de adaptación.

Finalmente, la función objetivo, que proporciona el ajuste de cada cromosoma es

$$f = \frac{1}{Q(t) + 1} \quad (6.6)$$

cuyo valor máximo es 1. Entre más apto es un cromosoma, su nivel de adaptación se encuentra más cercano a 1. Los cromosomas menos aptos se encuentran más cercanos a 0.

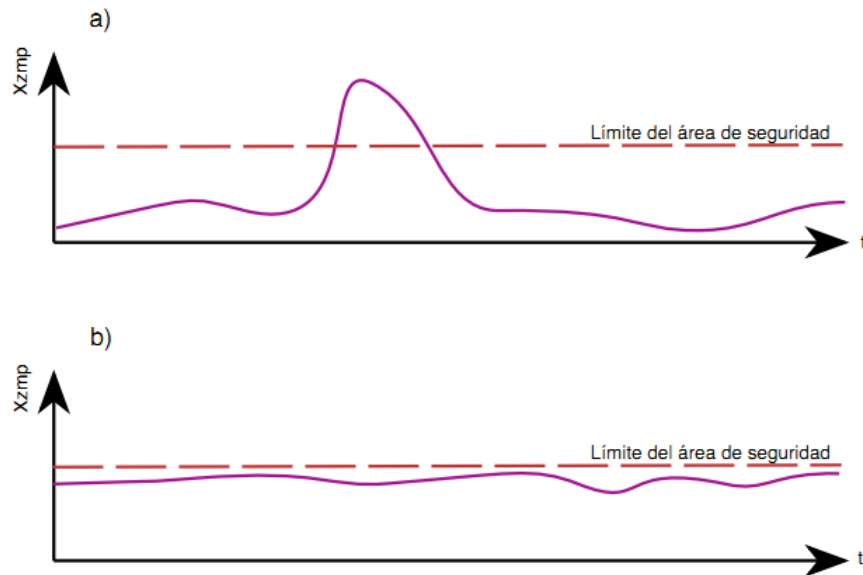


Figura 6.4: Trayectorias del ZMP sobre el eje x para dos ciclos de marcha diferentes.

6.3.6.1. Definición del área de seguridad

Aunque teóricamente el ciclo de marcha del robot bípedo es dinámicamente estable si la trayectoria del ZMP se mantiene siempre dentro del polígono de soporte, es probable que la planta del pie se despegue del suelo cuando la trayectoria del ZMP se encuentra en los límites del polígono de soporte,

sobre todo en presencia de perturbaciones. El área de seguridad es un subespacio del polígono de soporte sobre la cual, idealmente, debe ubicarse la trayectoria del ZMP con el propósito de aumentar la estabilidad durante la marcha. La figura 6.4 muestra una imagen del área de seguridad. Será dentro de esta área donde el valor de la función Q será siempre cero.

En milímetros, sus límites se establecen con los valores

$$x_a = 15$$

$$y_a = 25$$

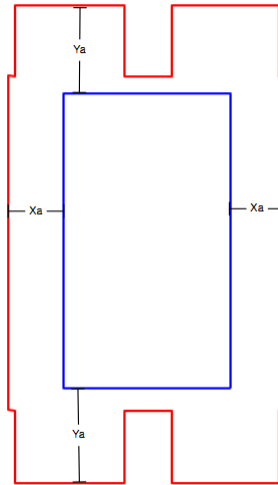


Figura 6.5: Vista inferior de la planta de un pie del bípido Scout. El borde rojo representa los límites de la planta, y el borde azul, los límites del área de seguridad.

6.3.6.2. Cálculo del ZMP

El ZMP es el punto sobre la superficie de desplazamiento donde la suma de todos los momentos, ocasionados por la gravedad y las fuerzas inerciales, es igual a cero [59]. Cuando el ZMP se ubica dentro del área convexa formada por los puntos de contacto entre el pie de soporte y el suelo, el robot puede desarrollar una marcha dinámicamente estable. El modelo del péndulo invertido describe de manera aproximada la dinámica del robot bípido durante la fase de soporte simple.

El modelo consiste en una partícula de masa ubicada en el centro de masa del robot y de un segmento imaginario que conecta al centro de masa del robot con un punto de contacto del robot con el suelo [57]. Bajo este modelo se asume que la fuerza de fricción en el suelo es lo suficientemente grande para prevenir deslizamientos y que las fuerzas de reacción del suelo actúan en un solo punto dentro de la zona de contacto del pie de soporte con la superficie. En la figura 6.6 se muestra al modelo del péndulo invertido.

Las fuerzas y momentos que actúan sobre el péndulo invertido son

$$\begin{aligned}
\sum \mathbf{T}_{gr} &\triangleq \text{torques creados por las fuerzas de reacción del suelo} \\
\mathbf{T}_{cm} &\triangleq \text{momento dinámico del centro de masa} \\
\mathbf{F}_{gr} &\triangleq \text{fuerza de gravedad}
\end{aligned}$$

La ecuación de equilibrio rotacional es

$$\sum \mathbf{T}_{gr} + \mathbf{r}_{cm} \times \mathbf{F}_{gr} = \mathbf{T}_{cm} \quad (6.7)$$

con

$$\sum \mathbf{T}_{gr} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^T \quad (6.8)$$

$$\mathbf{F}_{gr} = \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T \quad (6.9)$$

$$\mathbf{T}_{cm} = \frac{d}{dt} \mathbf{L}_{cm} \quad (6.10)$$

$$\mathbf{r}_{cm} = \begin{bmatrix} x & y & z \end{bmatrix}^T \quad (6.11)$$

y donde

$$\mathbf{r}_{cm} \triangleq \text{vector que va desde el punto de contacto hasta el centro de masa}$$

$$\mathbf{L}_{cm} \triangleq \text{momento angular del centro de masa}$$

$$m \triangleq \text{cantidad de masa ubicada en el centro de masa}$$

$$g \triangleq \text{aceleración gravitatoria}$$

Dado que \mathbf{L}_{cm} se refiere al momento angular de una masa puntual, la ecuación 6.10 se puede expresar como

$$\frac{d}{dt} \mathbf{L}_{cm} = \frac{d}{dt} (\mathbf{r}_{cm} \times \mathbf{p}_{cm}) \quad (6.12)$$

donde

$$\mathbf{p}_{cm} \triangleq \text{momentum del centro de masa}$$

Usando la regla de Leibniz para la derivación de un producto, se puede expresar a 6.12 como

$$\frac{d}{dt} \mathbf{L}_{cm} = \left(\frac{d}{dt} \mathbf{r}_{cm} \times \mathbf{p}_{cm} \right) + \left(\mathbf{r}_{cm} \times \frac{d}{dt} \mathbf{p}_{cm} \right) \quad (6.13)$$

Dado que

$$\mathbf{p}_{cm} = m \mathbf{v}_{cm} \quad (6.14)$$

$$\frac{d}{dt} \mathbf{r}_{cm} = \mathbf{v}_{cm} \quad (6.15)$$

donde

$$\mathbf{v}_{cm} \triangleq \text{velocidad del centro de masa}$$

ambos vectores son paralelos; por lo tanto, el producto cruz del primer sumando de 6.13 es 0. Entonces, la ecuación 6.13 queda como

$$\frac{d}{dt} \mathbf{L}_{cm} = \mathbf{r}_{cm} \times m \frac{d}{dt} \mathbf{v}_{cm} \quad (6.16)$$

con

$$\mathbf{v}_{cm} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix} \quad (6.17)$$

Sustituyendo 6.11 y 6.17 en 6.16, y después, 6.8, 6.9, 6.11 y 6.16 en 6.7, se obtiene

$$\left(T_{gr_x} \hat{i} + T_{gr_y} \hat{j} + T_{gr_z} \hat{k} \right) + \left((x \hat{i} + y \hat{j} + z \hat{k}) \times mg \hat{k} \right) = (x \hat{i} + y \hat{j} + z \hat{k}) \times \left(m \frac{d}{dt} (x \hat{i} + y \hat{j} + z \hat{k}) \right) \quad (6.18)$$

Al calcular los productos cruz, se tiene

$$\left(T_{gr_x} \hat{i} + T_{gr_y} \hat{j} + T_{gr_z} \hat{k} \right) + \left(-mgy \hat{i} + mgx \hat{j} \right) = m \left((y\ddot{z} - z\ddot{y}) \hat{i} + (z\ddot{x} - x\ddot{z}) \hat{j} + (x\ddot{y} - y\ddot{x}) \hat{k} \right) \quad (6.19)$$

Separando la ecuación vectorial 6.19 en dos ecuaciones escalares con los términos multiplicados por los vectores unitarios \hat{i} y \hat{j} , se obtienen las ecuaciones de movimiento

$$\sum T_{gr_x} - mgy = m(y\ddot{z} - z\ddot{y}) \quad (6.20)$$

$$\sum T_{gr_y} + mgx = m(z\ddot{x} - x\ddot{z}) \quad (6.21)$$

El ZMP puede ser usado para representar la suma de torques, resultado de las fuerzas de reacción del suelo, como sigue

$$\sum \mathbf{T}_{gr} - \mathbf{r}_{zmp} \times \mathbf{F}_{gr} = \begin{bmatrix} 0 & 0 & M_z \end{bmatrix}^T \quad (6.22)$$

con

$$\mathbf{r}_{zmp} = \begin{bmatrix} x_{zmp} & y_{zmp} & 0 \end{bmatrix} \quad (6.23)$$

donde

$$\mathbf{r}_{zmp} \triangleq \text{posición del ZMP}$$

$$M_z \triangleq \text{momento sobre el eje } z.$$

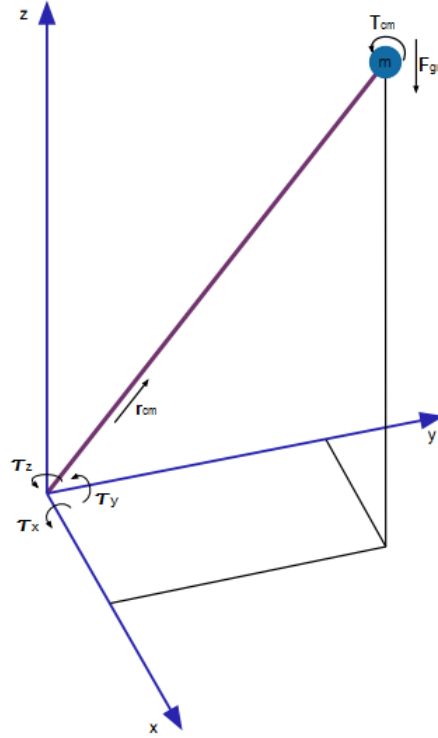


Figura 6.6: Modelo del péndulo invertido.

Sustituyendo 6.8, 6.9 y 6.23 en 6.22 se obtiene

$$\left(T_{gr_x} \hat{i} + T_{gr_y} \hat{j} + T_{gr_z} \hat{k} \right) + \left(\left(x_{zmp} \hat{i} + y_{zmp} \hat{j} \right) \times mg \hat{k} \right) = M_z \hat{k} \quad (6.24)$$

De tal ecuación vectorial se obtienen las dos ecuaciones vectoriales

$$T_{gr_x} = mgy_{zmp} \quad (6.25)$$

$$T_{gr_y} = -mgx_{zmp} \quad (6.26)$$

Al sustituir 6.25 en 6.20 y 6.26 en 6.21 se pueden obtener dos ecuaciones que relacionan a la posición del ZMP con la dinámica del centro de masa:

$$mgy_{zmp} - mgy = m(y\ddot{z} - z\ddot{y}) \quad (6.27)$$

$$-mgx_{zmp} + mgx = m(z\ddot{x} - x\ddot{z}) \quad (6.28)$$

Finalmente, reacomodando términos se obtiene

$$x_{zmp} = \frac{z\ddot{x} - x\ddot{z}}{g} + x \quad (6.29)$$

$$y_{zmp} = \frac{y\ddot{z} - z\ddot{y}}{g} + y \quad (6.30)$$

Éstas son las ecuaciones usadas para el cálculo de la función objetivo; como se puede observar, la trayectoria del ZMP depende de la posición y aceleración del centro de masa, que está en función de la posición y orientación de cada uno de los eslabones del bípido. Empleando la nomenclatura de la subsección 4.1.2, el vector del centro de masa del bípido es

$$\mathbf{r}_{cm} = \frac{m_B}{m_T} \mathbf{r}_B + \sum_{i=1}^2 \left(\sum_{n=1}^6 \frac{m_{ni}}{m_T} \mathbf{r}_{ni} \right) \quad (6.31)$$

donde m_T es la masa total del bípido, m_B es la masa del eslabón B y m_{ni} es la masa del eslabón ni ; \mathbf{r}_B es el vector que va del origen al centro de masa del eslabón B y r_{ni} es el vector que va del origen al centro de masa del eslabón ni .

Dado a que el modelo del péndulo invertido representa al bípido únicamente durante la fase de soporte simple, la estabilidad del sistema se analiza solamente durante dicha fase. Así como el análisis de la marcha durante la fase de soporte simple es un factor clave para generar patrones dinámicamente estables, el incluir a la fase de soporte doble durante la generación de dichos patrones, resulta clave para la transición tanto del peso como del ZMP de un pie a otro, factores que también ayudan a mantener la estabilidad [27]. Es por eso que aunque para la optimización de la marcha no se tome en cuenta a la fase de soporte doble, ésta sí se incluye para la síntesis de la marcha.

6.3.7. Selección

Para el algoritmo propuesto se usa el método de selección por ruleta, donde la posibilidad que tienen los individuos de formar a las nuevas generaciones es proporcional a su nivel de adaptación. Entre mayor sea el valor obtenido por un individuo en la función f , mayor es su probabilidad de ser seleccionado. La función que asigna la probabilidad (en porcentaje) a cada elemento es

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j} * 100$$

done

$p_i \triangleq$ probabilidad del elemento i

$f_i \triangleq$ fitness o ajuste del elemento i

$f_j \triangleq$ fitness o ajuste del elemento j

$n \triangleq$ tamaño de la población

6.3.8. Reproducción

El operador de cruce o reproducción transfiere el material genético de una población a otra; parte de dos cromosomas, padres, para formar a un cromosoma, hijo. En el algoritmo se usa un método basado en el operador de cruce en un solo punto. El método tradicional consiste en dividir en dos partes a cada uno de los cromosomas padres por el mismo punto, seleccionar una parte correspondiente de un cromosoma y unirla con su parte complementaria del otro cromosoma, de tal forma que se cree un nuevo cromosoma, hijo, con toda la materia genética necesaria para considerarlo una posible solución al problema (figura 6.7).

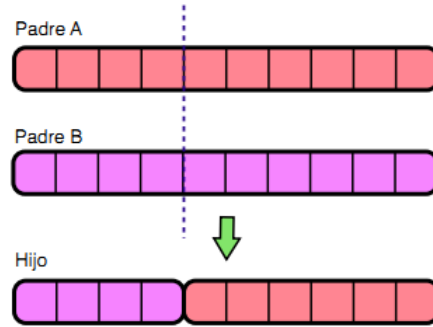


Figura 6.7: Cruza monopunto. La línea punteada marca el punto por donde se dividirá a los cromosomas padres.

Debido a que cada una de las curvas para definir la trayectoria de la cadera debe cumplir con ciertas condiciones preestablecidas y a que las características de cada curva están en función de varios parámetros (genes), la recombinación genética podría producir curvas *defectuosas* (subsección 6.3.5). Para evitar este problema, se organizan los cromosomas en grupos, donde cada uno contiene a los parámetros para definir a los polinomios de alguna curva de la cadera (figura 6.8). Los puntos de cruce posibles son aquellos que permiten a cada grupo de genes mantenerse completo.

El operador de cruce en un solo punto limita la recombinación genética. Con este método, por ejemplo, no sería posible formar a un cromosoma cuyos primeros y últimos genes pertenezcan a un mismo padre, mientras los genes que se encuentran a la mitad de la cadena pertenecen al otro padre. Para solucionar este problema se propone el uso de un cromosoma virtual al que se le aplica una reorganización genética aleatoria, como herramienta para alterar la ubicación de sus genes. La estructura natural del cromosoma es la que muestran los padres A y B de la figura 6.9: primero se encuentran los genes para la función que define el movimiento sobre el eje x , después están los que definen su movimiento sobre el eje y , posteriormente, los que definen su movimiento sobre el eje z y, finalmente, los parámetros ϕ_c y ψ_c , que definen a las rotaciones de la cadera sobre el eje y y z .

La recombinación genética ocurre de la siguiente forma: para crear al cromosoma virtual, se cambia aleatoriamente la organización de los grupos de genes. El punto de cruce ya no se le aplica a cada padre, sino al cromosoma virtual, el cual se ubica de tal manera que corte al cromosoma por grupos, dejándolos completos. Los genes que dentro de la cadena del cromosoma se encuentran a la izquierda del punto de cruce proceden de un mismo padre, mientras que los genes que se encuentran a la izquierda del mismo punto son procedentes del otro padre. Posteriormente, el gameto es creado con la información genética del cromosoma virtual pero con la estructura de sus cromosomas padres.

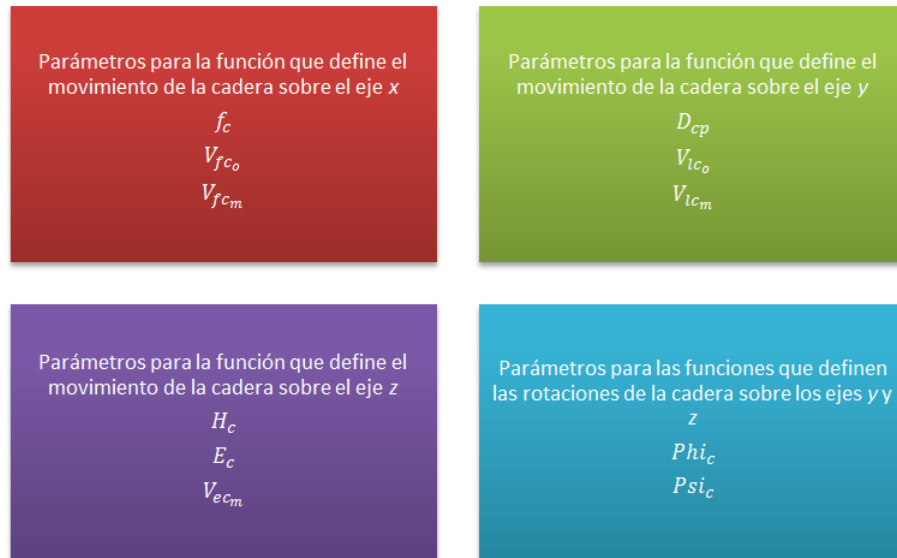


Figura 6.8: Grupos de genes. Cada cromosoma se compone de estos grupos, conteniendo así todos los genes.

Con esta modificación al operador monopunto, como se puede observar en el cromosoma hijo de la figura 6.9, es posible producir cualquier combinación genética.

Con el operador de cruza se generan nuevos individuos, los cuales pueden ser más, o menos, aptos. El algoritmo genético trabaja esperando que algunos de los nuevos individuos generados durante el proceso de reproducción sean más aptos. Sin embargo, así como el operador de cruza puede producir mejores individuos, también puede producir individuos con un ajuste menor al de sus cromosomas padre. Si durante una generación un cromosoma es seleccionado para participar en el proceso de reproducción y su descendencia tiene una adaptabilidad y capacidad de reproducción menor, la aleatoriedad del algoritmo estaría contribuyendo a alejarse de una posible solución, en vez de optimizarla. Para considerar este caso dentro del algoritmo genético, el operador de cruza tiene una probabilidad de 80 % de actuar, mientras el restante 20 % considera al caso anterior, seleccionando entre los dos cromosomas padre al más apto para usarlo como elemento de la nueva población; en este caso, el cromosoma hijo es idéntico a un cromosoma padre.

6.3.9. Mutación

El operador de mutación es el único que requiere trabajar con los genes codificados. La probabilidad de mutación es baja. En caso de que un elemento deba ser mutado, se selecciona al azar un gen para aplicarle una mutación. Una vez seleccionado, se codifica en una cadena de bits y se selecciona a alguno de éstos para permutar su valor. Posteriormente, se decodifica y se integra al cromosoma. Para ser aprobada la mutación, el cromosoma afectado debe cumplir con dos condiciones: el valor del gen mutado debe encontrarse dentro de los límites marcados en la subsección 6.3.5 y la curva del grupo al que pertenece el gen debe cumplir con las condiciones ya preestablecidas en la misma subsección. En caso de que no se cumplan ambas condiciones, se realiza otra mutación. En caso de aplicar sin éxito el operador más de 25 veces, se deja al cromosoma sin alterar, como si nunca hubiera mutado. La

Capítulo 7

Experimentos y simulaciones

Empleando el algoritmo de la figura 6.10, se desarrolló en MATLAB un programa para la optimización de la marcha en el que solamente se obtienen los parámetros para su etapa cíclica (no se consideran a las etapas de puesta en marcha ni de paro de la marcha). Para completar la planificación de la caminata se emplea al algoritmo de la figura 5.10 del capítulo 5, en el que se utilizan a los parámetros de entrada del algoritmo genético (posición inicial, parámetros de tiempo y parámetros para el pie flotante), además de los parámetros optimizados, por dicho algoritmo, para la cadera.

Con el algoritmo genético empleado también es posible obtener los parámetros optimizados para las etapas de puesta en marcha y paro de la marcha; sin embargo, se consideró que no era necesario obtenerlas, ya que el propósito de los experimentos y simulaciones es validar la funcionalidad y mostrar la utilidad de los algoritmos desarrollados; mientras el desempeño y los resultados del algoritmo para la optimización de la marcha quedan demostrados con los experimentos realizados, únicamente, para la etapa cíclica de la marcha, la funcionalidad del algoritmo para la planificación de la marcha queda demostrada al ser usada para desarrollar un ciclo de marcha completo, con todas sus etapas.

7.1. Descripción de las pruebas

Para la generación de ciclos de marcha dinámicamente estables, pero también para evaluar al algoritmo genético, éste se ejecuta tres veces, en las que, a excepción de la longitud del paso, se emplean los mismos parámetros de entrada. De acuerdo a los resultados obtenidos en [11], el margen de estabilidad del robot depende del tamaño del paso: entre más largo es el paso, menor es el margen de estabilidad y viceversa: entre más largo es el paso, mayor es la dificultad para encontrar ciclos de marcha dinámicamente estables. Por lo tanto, al correr varias veces el algoritmo, 3, en las que solamente se varía la longitud del paso, es posible observar su capacidad y determinar sus límites y alcances.

Aunque el algoritmo genético se puede detener una vez que se haya encontrado un individuo lo suficientemente apto, para comparar los resultados, según la longitud del paso, el número de generaciones se fija a 15.

7.1.1. Selección de parámetros

Los parámetros a utilizar en los tres ciclos de marcha son los mismos, a excepción de la longitud del paso.

(Los parámetros de tiempo están en segundos; los de distancia, en milímetros; los de rotación en grados, y los de velocidad, en milímetros por segundo).

Posiciones iniciales

Pie izquierdo

$$x_{61} = 44.54$$

$$y_{61} = 0$$

$$z_{61} = 0$$

Pie derecho

$$x_{62} = 44.54$$

$$y_{62} = 0$$

$$z_{62} = 0$$

Cadera

$$x_B = 0$$

$$y_B = 38.5$$

$$z_B = 234.37$$

$$\phi_B = 0$$

$$\psi_B = 0$$

Parámetros

Parámetros de tiempo

$$T = 8$$

$$Tsd = 35\%$$

$$\Delta t = \frac{4}{15}$$

Parámetros para el pie flotante

$$L_p = \begin{cases} 100 & \text{Patrón 1} \\ 150 & \text{Patrón 2} \\ 200 & \text{Patrón 3} \end{cases}$$

$$E_p = 85$$

$$F_p = 5$$

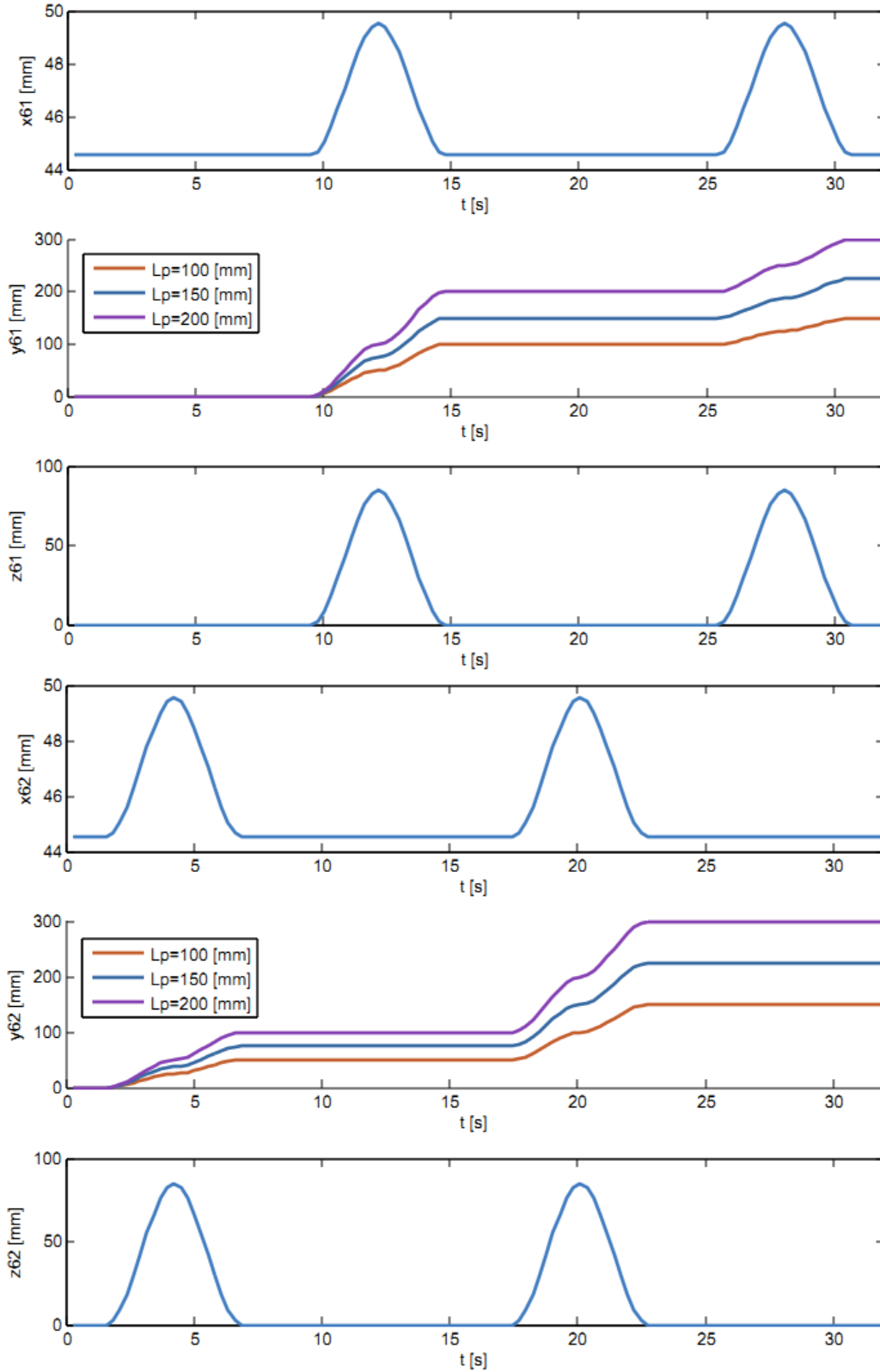


Figura 7.1: Trayectorias para los eslabones L_{61} y L_{62} de los tres patrones de marcha. Las únicas que cambian son las del desplazamiento sobre el eje y .

7.1.2. Trayectorias de los eslabones L61 y L62

De acuerdo a la posición inicial (y final) del bípido, a los parámetros de tiempo y a los parámetros para el pie flotante, se muestran, según las ecuaciones para definir las trayectorias del pie flotante expuestas en el capítulo 5, las trayectorias de los eslabones L_{61} y L_{62} en la figura 7.1. Éstas funcionan como parámetros de entrada para el algoritmo genético. La longitud de paso del primer ciclo de marcha es de $L_p = 100 [mm]$, la del segundo es de $L_p = 150 [mm]$, y la del último, de $L_p = 200 [mm]$. El periodo de cada paso se fijó en $T = 8 [s]$; por lo tanto, el tiempo que tarda el robot en ejecutar las tres etapas de la marcha (inicio, paro y una sola etapa cíclica) es de $32 [s]$.

7.2. Resultados

Empleando los parámetros de entrada mostrados en 7.1.1, el algoritmo para la optimización de la marcha se ejecutó una vez para cada conjunto de parámetros. Los parámetros con los que se corrió en cada caso son los siguientes:

$$\begin{aligned} \text{Población} &= 30 \\ \text{Generaciones} &= 15 \\ \text{Mutación} &= 20 [\%] \end{aligned}$$

7.2.1. Desempeño del algoritmo

Para evaluar a cada individuo de la población de acuerdo al periodo de muestreo, $\Delta t = \frac{4}{15} [s]$, se emplearon 30 unidades de muestreo. El tiempo aproximado para evaluar a cada individuo fue de $20 [s]$, el que se tardó en evaluar a toda una generación fue de $15 [min]$ y el evaluar a las 15 generaciones tardó alrededor de $2.5 [hr]$. La mayor parte de este tiempo fue consumido en el cálculo de la cinemática inversa.

En cada ejecución del algoritmo la función de costo convergió y se acercó a su valor máximo, 1. En todos los casos, la población inicial consta de un conjunto de individuos que son, en promedio, poco aptos. Con el avance generacional, mediante los procesos de selección natural y genética natural que el algoritmo genético simula, se obtienen individuos cada vez mejores, que ofrecen una mejor solución al problema de estabilidad dinámica de la marcha (figura 7.2), con lo que se considera haber demostrado la funcionalidad y utilidad del método de optimización.

Observando los datos de la última generación, se obtienen, para cada conjunto de parámetros, algunos individuos cuyo *fitness* promedio es 1, lo cual implica que en todo momento la trayectoria del ZMP está dentro del área de seguridad, y, por tanto, idealmente, la marcha es dinámicamente estable. A pesar de que para cada caso, el algoritmo genético encuentra diferentes parámetros para la trayectoria de la cadera que generan un ciclo de marcha que es, siempre, dinámicamente estable, cuando la longitud del paso es de $L_p = 200 [mm]$ solo encuentra 3 individuos con $f = 1$, mientras que cuando la longitud del paso es de $L_p = 100$ o de $L_p = 150$ encuentra más de 20 individuos con $f = 1$ (figura 7.3).

El ZMP depende de la posición del centro de masa, y en un robot bípido está determinada por la posición de sus pies. Se considera que el aplicar el algoritmo en un bípido con torso o en un humanoide,

mejoraría notablemente su desempeño, pues se tendría más control sobre la posición del centro de masa.

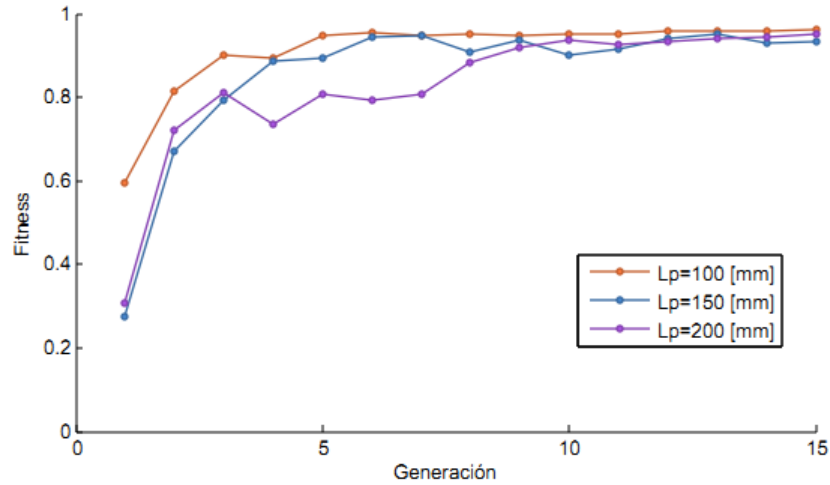


Figura 7.2: Valor del ajuste, o *fitness*, promedio del algoritmo genético durante cada generación para cada conjunto de parámetros.

Promedio y máximos y mínimos de la función de costo			
	Patrón 1	Patrón 2	Patrón 3
<i>f_{prom}</i>	0.96	0.95	0.93
<i>f_{mín}</i>	0.95	0.33	0.84
<i>f_{máx}</i>	1	1	1
# <i>f_{mín}</i>	1	1	1
# <i>f_{máx}</i>	22	24	3

Figura 7.3: Datos sobre el ajuste, o *fitness*, de la última generación del algoritmo genético para cada conjunto de parámetros. El primer renglón muestra su valor promedio, los siguientes 2 muestran el valor mínimo o máximo del individuo peor o mejor evaluado, y los últimos dos, el número de individuos con dicha calificación.

7.2.2. Selección de los cromosomas

Debido a que el programa para la optimización de la marcha encontró, para cada caso, más de un individuo con el máximo ajuste, se seleccionó únicamente uno para para cada patrón de marcha. Los parámetros de los individuos seleccionados se muestran en las figuras 7.4 y 7.5.

Parámetros de posición para la cadera				
	Patrón 1	Patrón 2	Patrón 3	Unidades
Hc	208.37	194.37	192.37	[mm]
Ec	214.37	205.37	194.37	[mm]
Fc	95	97	121	[mm]
Dcp	-14	-7	-15	[mm]
Phis	3	39	29	[°]
Psib	35	45	26	[°]

Figura 7.4: Parámetros de posición de los cromosomas seleccionados para generar las trayectorias de la cadera.

Parámetros de velocidad para la cadera				
	Patrón 1	Patrón 2	Patrón 3	Unidades
v_{fc0}	27	-1	2	[mm/s]
v_{fcm}	-5	19	19	[mm/s]
v_{lco}	12	15	4	[mm/s]
v_{lcm}	15	20	31	[mm/s]
v_{ecm}	-2	5	0	[mm/s]

Figura 7.5: Parámetros de velocidad de los cromosomas seleccionados para generar las trayectorias de la cadera.

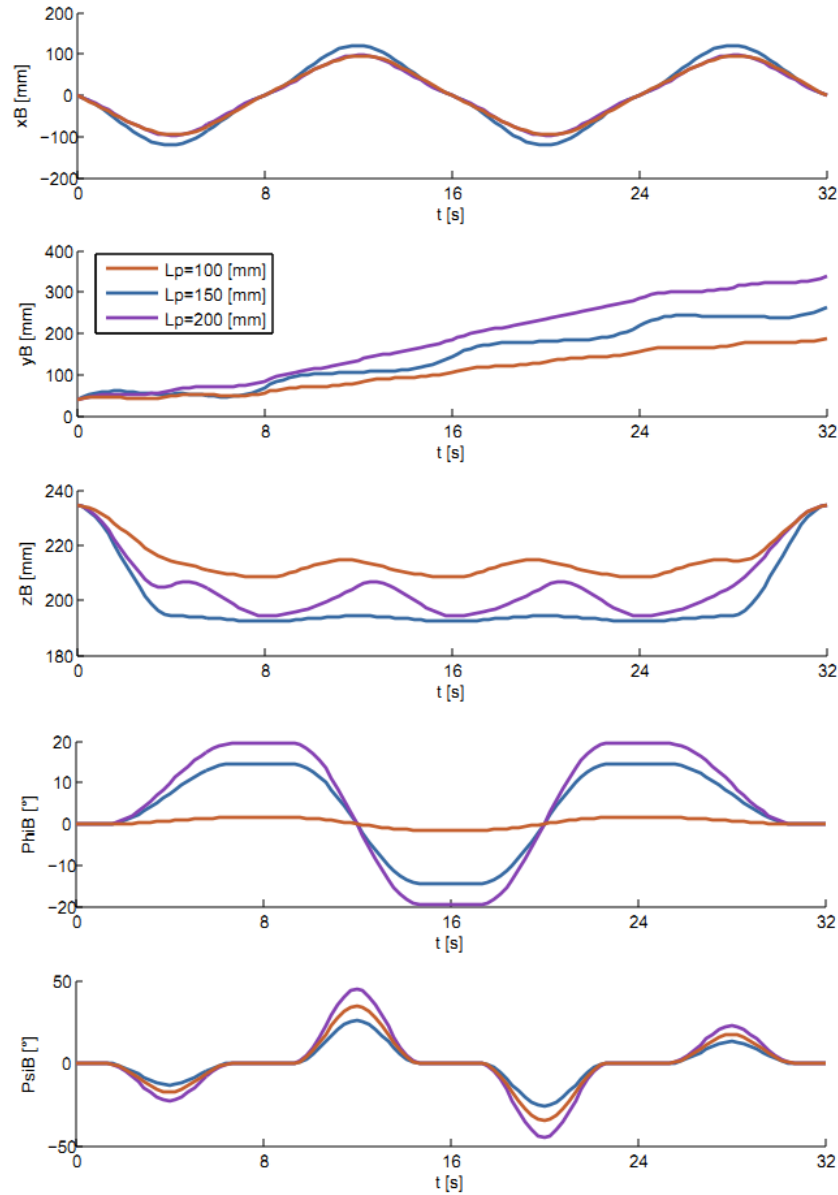


Figura 7.6: Trayectorias optimizadas a seguir por el eslabón L_B dadas las trayectorias a seguir por los eslabones L_{61} y L_{62} .

7.2.3. Trayectorias de la cadera optimizadas

En función de los parámetros obtenidos de los cromosomas mejor evaluados y de las funciones vistas en 5, se obtuvo, empleando el programa para la planificación de la caminata, la trayectoria a seguir por el eslabón L_B para cada conjunto de parámetros para definir la trayectoria del pie flotante (en éstos solo varía la longitud del paso) (figura 7.6). La trayectoria de la cadera únicamente se encuentra optimizada en la etapa cíclica de la marcha; para las etapas de inicio y paro de la marcha se usaron los mismos parámetros que para la etapa cíclica. Debido a que las trayectorias de la cadera fueron generadas empleando las mismas funciones, *splines*, en las que únicamente cambiaba el valor de sus parámetros, en las gráficas se puede observar que son muy similares.

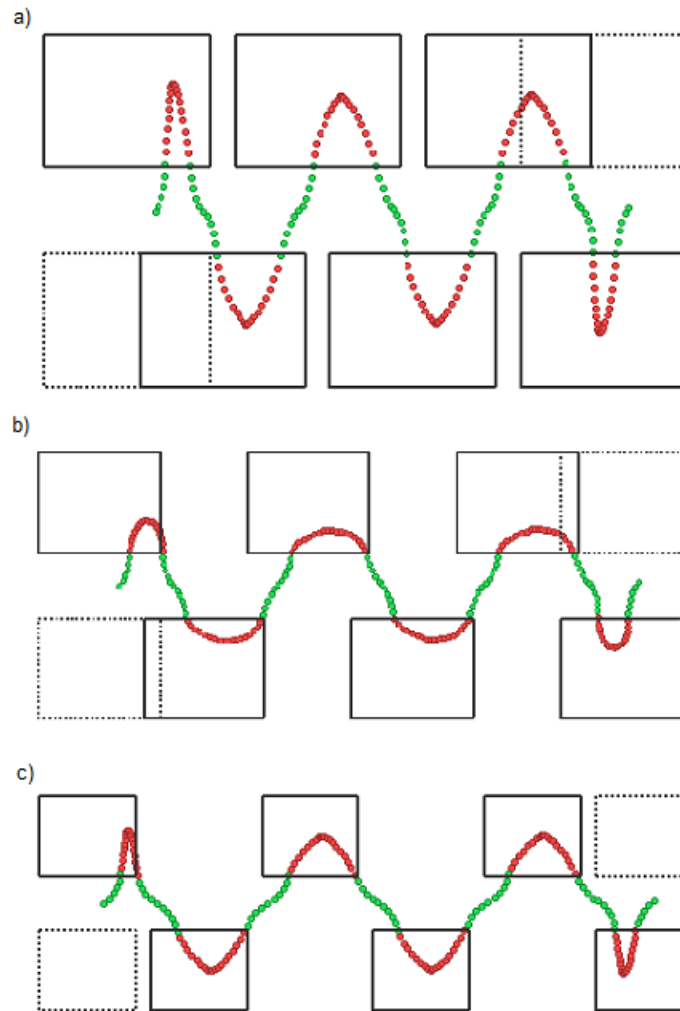


Figura 7.7: Trayectorias del ZMP para los patrones de caminata optimizados. En a) $L_p = 100 [mm]$, b) $L_p = 150 [mm]$ y c) $L_p = 200 [mm]$. Los círculos verdes representan a la trayectoria del ZMP durante el soporte doble, y los verdes, a la trayectoria del ZMP durante el soporte simple. Los polígonos negros dibujados con línea continua representan los límites del área de seguridad durante cada paso, y los dibujados con línea discontinua representan al pie que se levanta al inicio de la marcha o al pie que aterriza durante el paro de la marcha.

7.2.4. Trayectorias del ZMP de los patrones optimizados

De acuerdo a las trayectorias para los eslabones L_{61} y L_{62} (figura 7.1) y a la trayectoria del eslabón L_B (figura 7.6), es posible dibujar la trayectoria del ZMP para cada patrón de caminata. De acuerdo a los resultados del algoritmo genético, el ZMP se encuentra, siempre, durante la etapa cíclica de la marcha, dentro del área de seguridad, como se puede corroborar en la figura

7.3. Simulación de un ciclo de marcha

Para probar los resultados obtenidos por el algoritmo genético, se seleccionó el primer patrón de caminata, cuya longitud de paso es $L_p = 100 [mm]$.

7.3.1. Trayectorias de posición y velocidad angular

Con la solución numérica de las ecuaciones cinemáticas de posición (desarrolladas en [43]) para cada una de las configuraciones espaciales de los pies y torso mostradas en las figuras 7.1 y 7.6 se obtuvieron los valores ilustrados en las gráficas de las figuras 7.8 y 7.9, arrojados por el programa transcrito en el apéndice D de [43].

Con la solución numérica de las ecuaciones cinemáticas de velocidad (desarrolladas en [43]) para cada una de las configuraciones espaciales de los pies y torso mostradas en las figuras 7.1 y 7.6 se obtuvieron los valores ilustrados en las gráficas de las figuras 7.10 y 7.11, arrojados por el programa transcrito en el apéndice D de [43].

Resulta importante conocer las velocidades angulares que requieren las juntas rotacionales para efectuar el ciclo de marcha propuesto, puesto que los servomotores empleados únicamente pueden alcanzar una velocidad angular de $\dot{\theta} = 60 [\frac{^\circ}{s}]$ sin carga. De acuerdo a las figuras 7.10 y 7.11, para cualquier junta rotacional, se requiere, en cualquier momento, una velocidad menor a $\dot{\theta} = 25 [\frac{^\circ}{s}]$; sin embargo, se desconoce la carga producida en cada instante sobre cada una de las juntas. Para conocer si los servomotores lograrán producir los perfiles de velocidad requeridos sería necesario realizar el análisis dinámico, que escapa a los alcances de este trabajo.

Los resultados presentados, así como el análisis dinámico, son información útil para el diseño y construcción de un robot bípedo, por ejemplo, para la selección de los motores.

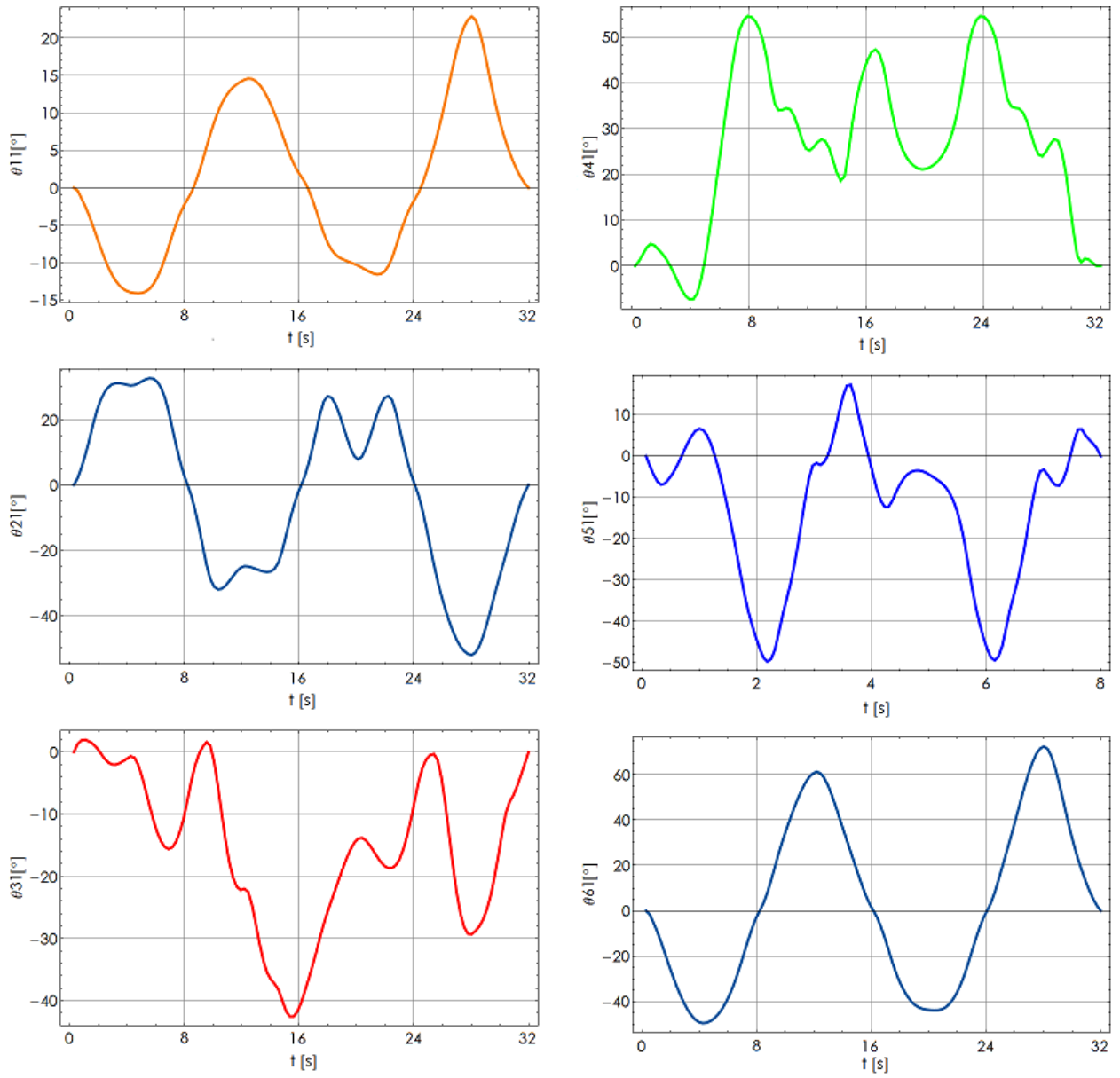


Figura 7.8: Valores adoptados por los ángulos θ_{11} , θ_{21} , θ_{31} , θ_{41} , θ_{51} , θ_{61} durante el ciclo de marcha propuesto.

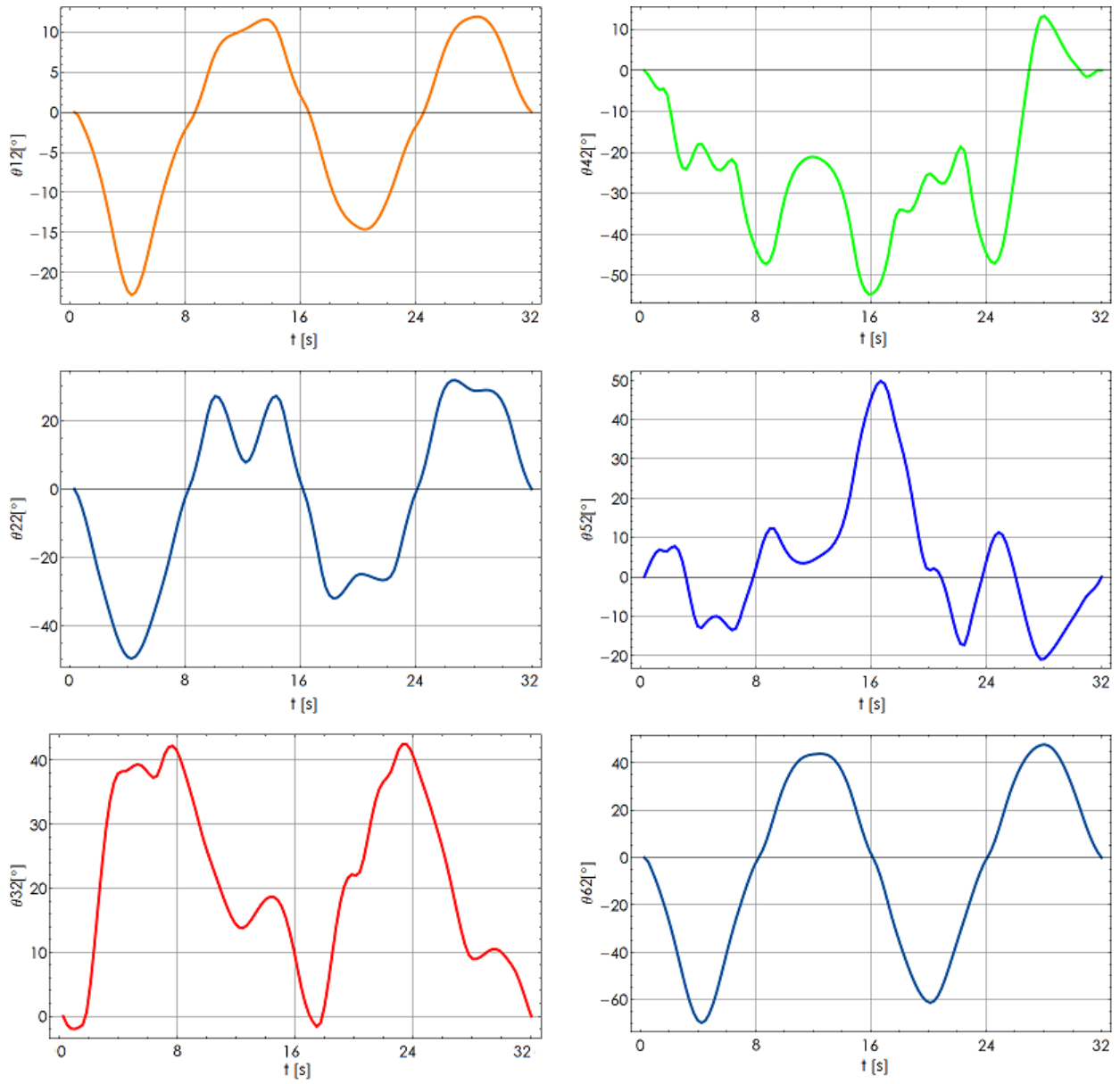


Figura 7.9: Valores adoptados por los ángulos θ_{12} , θ_{22} , θ_{32} , θ_{42} , θ_{52} , θ_{62} durante el ciclo de marcha propuesto.

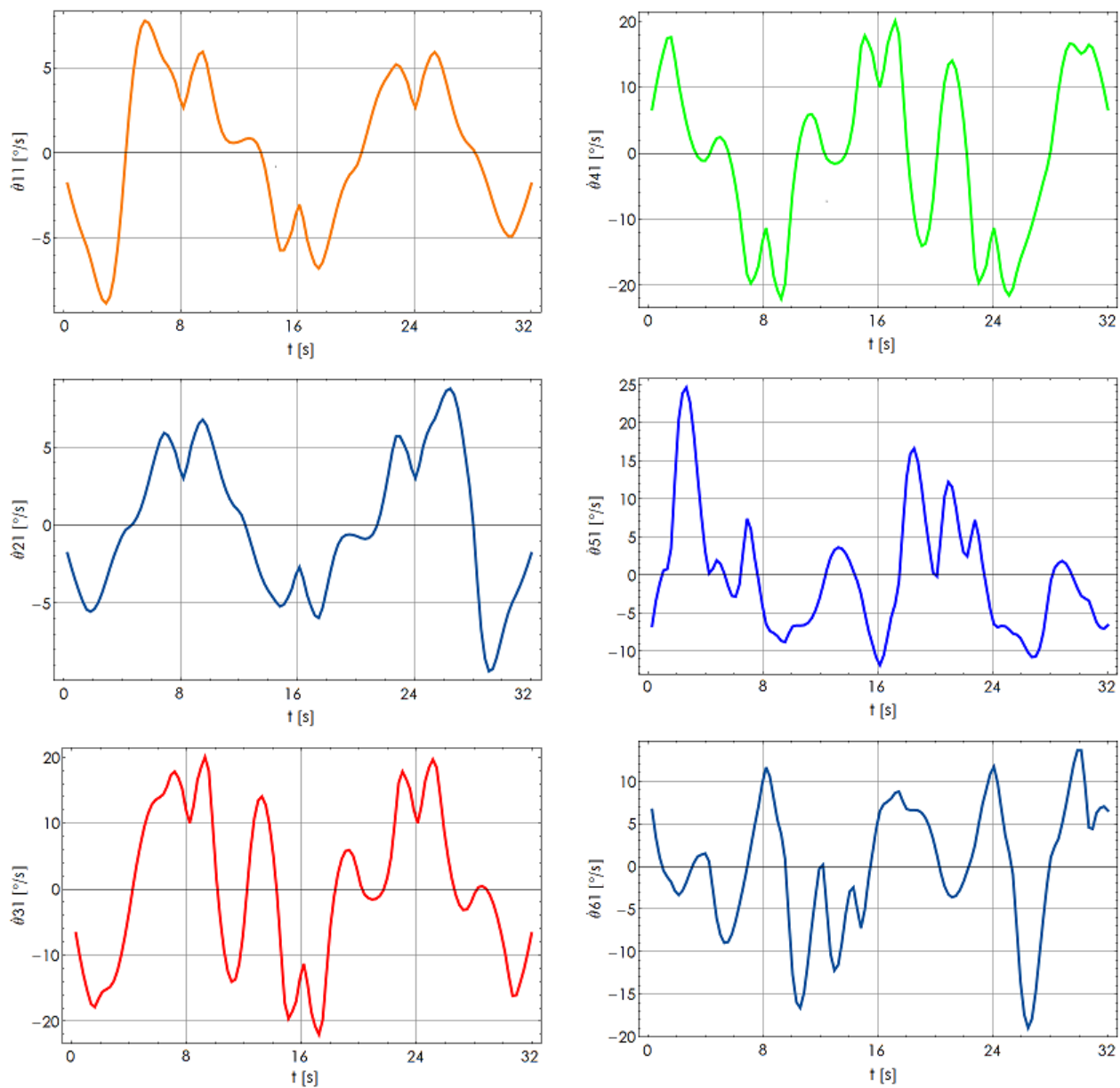


Figura 7.10: Velocidades de las juntas rotacionales θ_{11} , θ_{21} , θ_{31} , θ_{41} , θ_{51} , θ_{61} durante el ciclo de marcha propuesto.

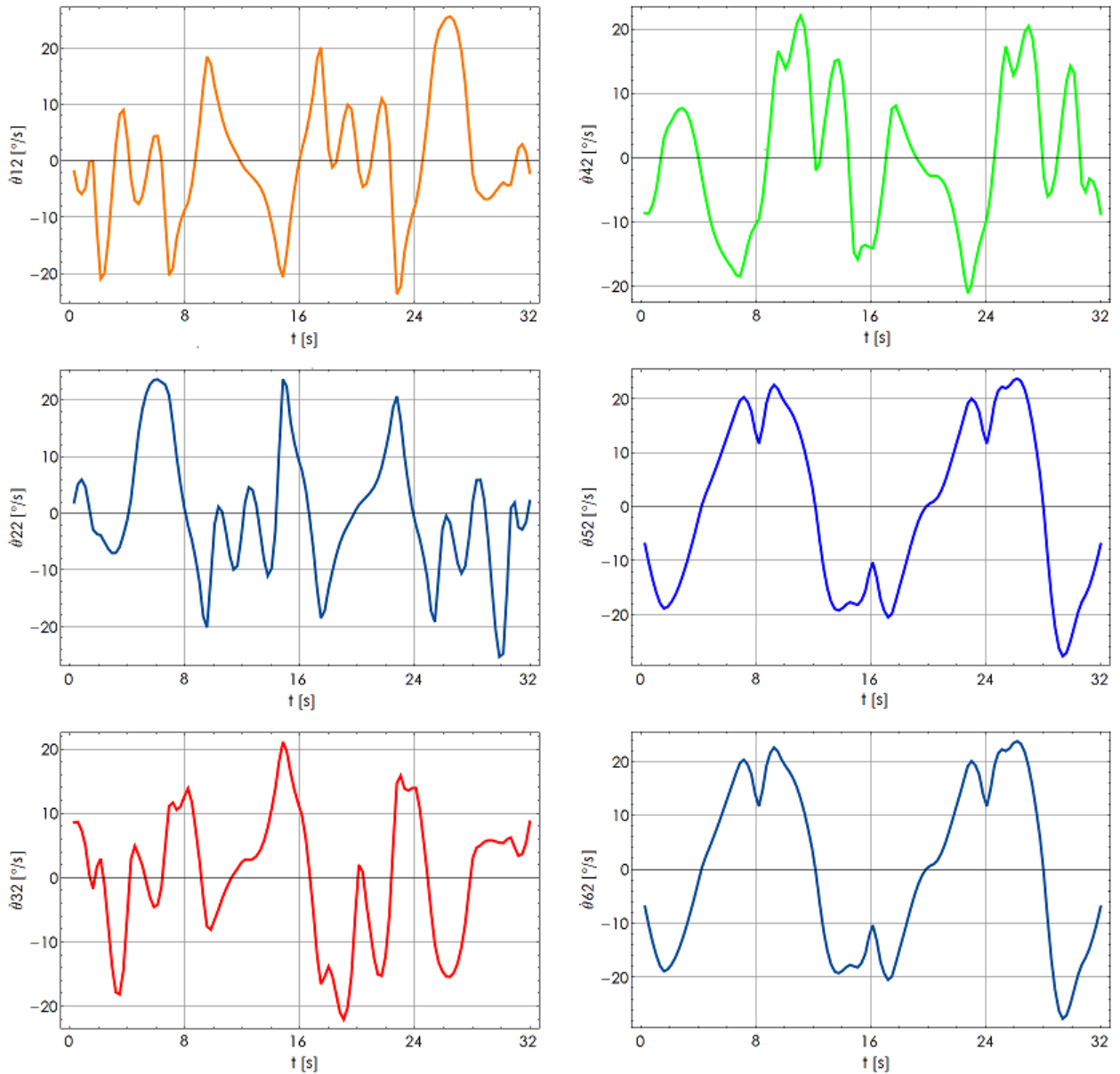


Figura 7.11: Velocidades de las juntas rotacionales θ_{12} , θ_{22} , θ_{32} , θ_{42} , θ_{52} , θ_{62} durante el ciclo de marcha propuesto.

7.3.2. Simulaciones en Mathematica y LabVIEW

Disponiendo de todos los ángulos anteriores se logró animar exitosamente el ciclo de marcha propuesto, tanto en el programa de Mathematica, transcrito en el apéndice D de [43], como en el programa de LabVIEW, descrito en el capítulo 4. En las figuras 7.12 y 7.13 se ofrecen algunos de los cuadros de la animación generada en Mathematica para ilustrar la transición del modelo simplificado del bípedo durante el patrón de caminata. Y en la figura 7.14 se ofrecen algunos de los cuadros de la animación generada en LabVIEW. En ambas animaciones, los cuadros ilustran el desarrollo de la marcha con capturas en los mismos instantes. Mientras la animación de Mathematica permite observar claramente

las trayectorias seguidas por las juntas rotacionales del bípido, la simulación en LabVIEW permite observar si el robot, efectivamente, puede seguir las trayectorias angulares diseñadas o, bien, si los choques debidos al volumen de los eslabones lo impedirán, y, como se observó durante la simulación, no existe en ningún momento algún choque entre los eslabones que dorman al bípido.

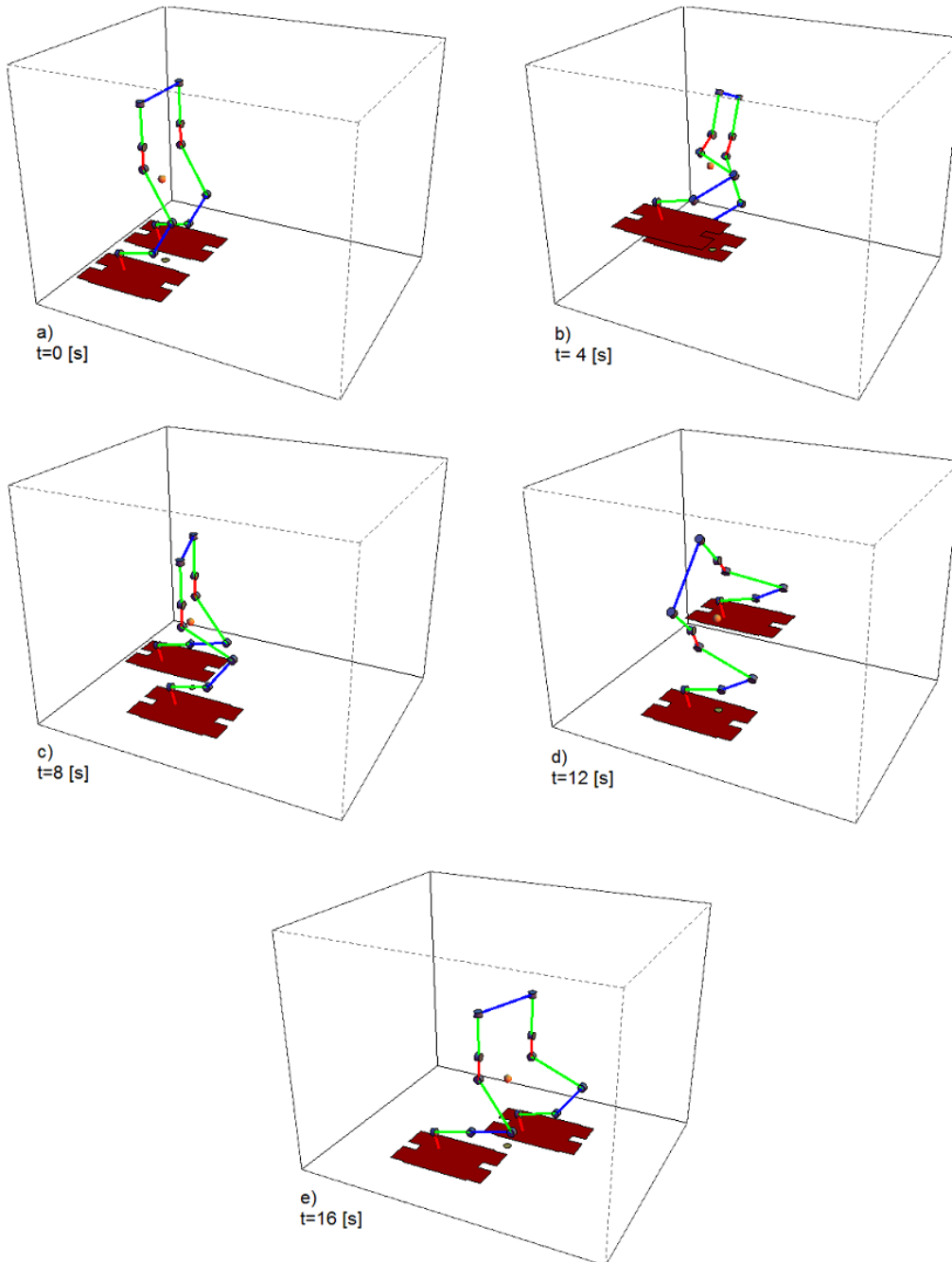


Figura 7.12: Configuraciones adoptadas por el bípido durante la simulación del ciclo de marcha propuesto. a), b) y c) corresponden a la etapa de inicio de la marcha. c), d) y e) corresponden al primer semiciclo de la etapa cíclica de la marcha. La esfera naranja representa la posición del centro de gravedad; el círculo que se traslada sobre la superficie de desplazamiento representa a la posición del ZMP.

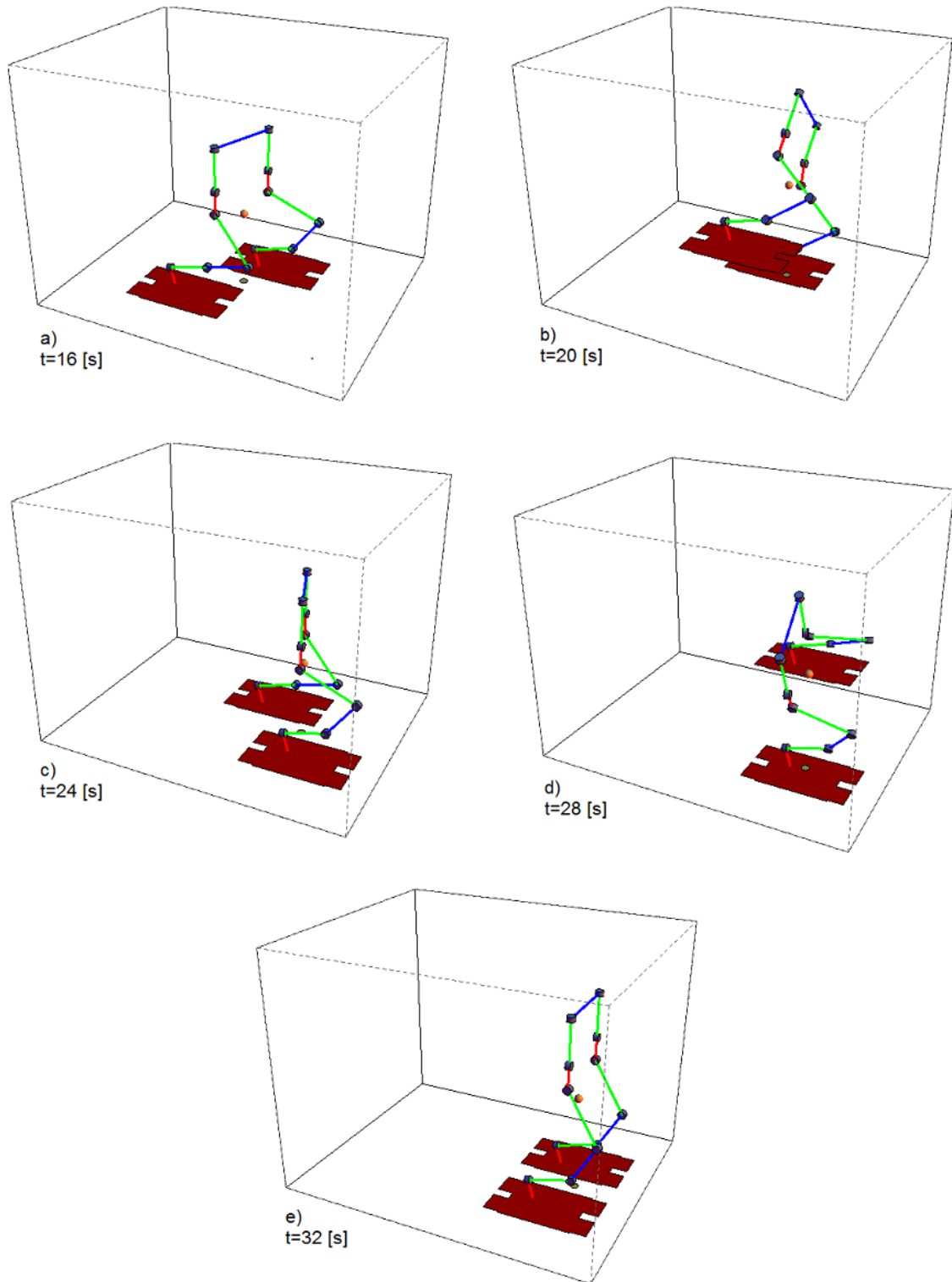


Figura 7.13: Configuraciones adoptadas por el bípido durante la simulación del ciclo de marcha propuesto. a), b) y c) corresponden al segundo semiciclo de la etapa cíclica de la marcha. c), d) y e) corresponden a la etapa de paro de la marcha. La esfera naranja representa la posición del centro de gravedad; el círculo que se traslada sobre la superficie de desplazamiento representa a la posición del ZMP.

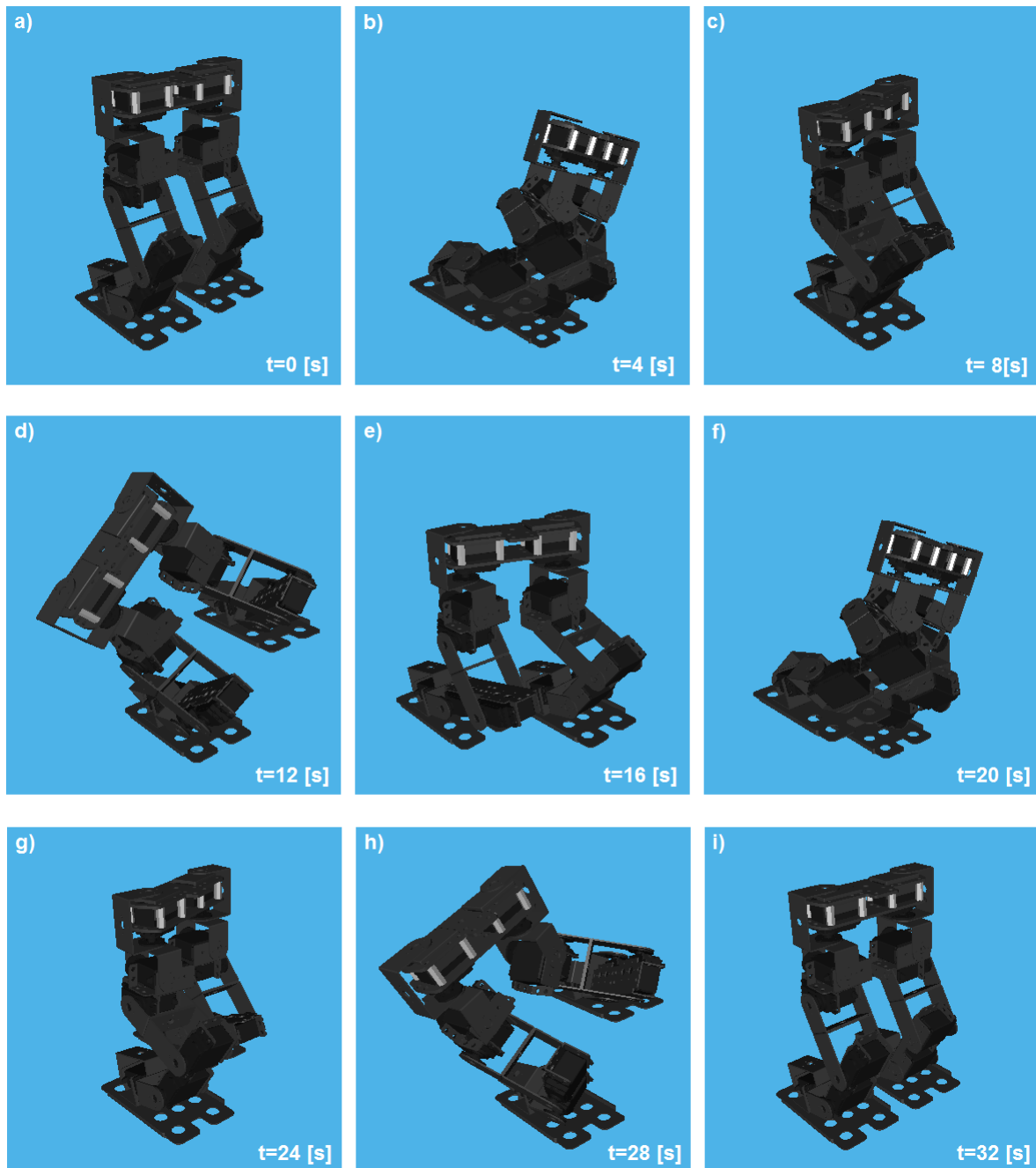


Figura 7.14: Configuraciones adoptadas por el bípido durante la simulación del ciclo de marcha propuesto. a), b) y c) corresponden a la etapa de inicio de la marcha. c), d) y e) corresponden al primer semiciclo de la etapa cíclica de la marcha. e), f) y g) corresponden al segundo semiciclo de la etapa cíclica de la marcha. h), i) y j) corresponden a la etapa de paro de la marcha.

7.3.3. Prueba con el prototipo físico

Al igual que en los modelos virtuales, se logró implementar exitosamente el ciclo de marcha propuesto en el prototipo físico empleando la interfaz descrita en el capítulo 4. En la figura 7.15 se ofrecen

algunas imágenes del ciclo de marcha del robot bípedo. Se emplearon 120 muestras para 32 [s], la señal se actualizaba cada $\Delta t = 270$ [ms]. En estas pruebas se observaron vibraciones en la estructura y problemas de seguimiento de las trayectorias angulares por los servomotores. Durante la marcha, el robot era muy sensible a la orientación de la superficie de apoyo y a cualquier perturbación. Sin embargo, el prototipo físico fue capaz de desplazarse con movimientos similares a los representados en las simulaciones, compartiendo características como la elevación del pie flotante, la inclinación de la cadera con respecto a la superficie de desplazamiento durante el soporte simple y el movimiento frontal de la cadera durante la fase de soporte doble. El mejoramiento de la respuesta de los servomotores y la implementación de un controlador para cambiar las trayectorias angulares de referencia por errores en las idealizaciones o perturbaciones, ayudarían a incrementar el desempeño del robot.

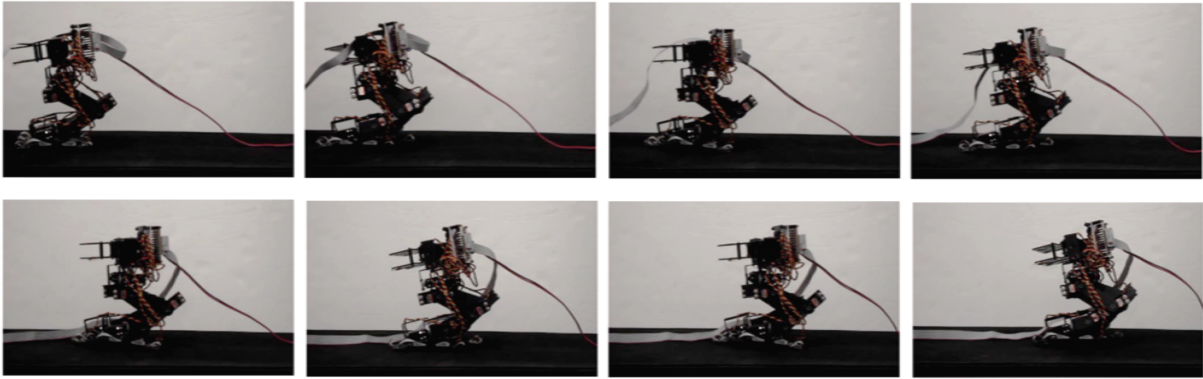


Figura 7.15: Caminata del robot bípedo Scout para el ciclo de marcha propuesto.

Capítulo 8

Conclusiones y trabajo a futuro

8.1. Conclusiones

La robótica bípeda ha sido un punto de partida para el desarrollo de robots humanoides; los avances en su investigación son esenciales para incrementar las funciones de los robots de servicio, y, además, su estudio es útil para comprender el sistema de locomoción humano. Problemas planteados dentro de la robótica bípeda, como la estabilidad, la generación de ciclos de marcha y el diseño de controladores para generar movimientos en lazo cerrado, han favorecido la sinergia entre diversos campos de la ingeniería e, incluso, de otras ciencias. Conocer la historia de la robótica bípeda permite entender su importancia, ayuda a asimilar sus alcances y a esclarecer la viabilidad de sus metas; además, es necesaria para conocer los problemas canónicos que plantea y las maneras en las que se han abordado. El capítulo 2 resultó vital para este trabajo: proporcionó algunos de los fundamentos teóricos empleados, brindó una idea de los posibles aportes y contribuciones de esta investigación a la robótica bípeda y, además, ayudó a plantear los objetivos a futuro.

Al comparar la locomoción de un robot bípedo con la del ser humano, resulta increíble la armonía, naturalidad, coordinación, flexibilidad, versatilidad que el ser humano ha llegado a desarrollar, gracias a un proceso de evolución y adaptación que ha llevado millones de años y a un proceso de aprendizaje y experimentación que se lleva a cabo durante la infancia. En cambio, los robots bípedos surgieron apenas algunas décadas y, por lo general, los algoritmos que utilizan para la generación de movimientos consideran a la marcha como un reflejo innato. Las capacidades motrices de los seres humanos resultan admirables. El diseño de algunos humanoides tiene como objetivo imitar algunas características observadas en los humanos. Algunos métodos para la generación de patrones de marcha emplean demostraciones de ciclos de marcha realizados por humanos, y algunos controladores para la estabilidad del robot están inspirados en el funcionamiento del sistema locomotor humano.

Las investigaciones en la marcha humana han facilitado el avance de la robótica bípeda. Tener una idea de la complejidad de la marcha humana y de las actividades que realiza el sistema locomotor durante la marcha, ayudan a entender y a conocer las razones de las diferencias entre el desempeño de un robot bípedo y un humano, pero también brindan posibles respuestas para reducirlas. En los seres vivos, la locomoción estable y flexible es el resultado de la sincronización entre las actividades rítmicas del sistema nervioso y los movimientos rítmicos del aparato locomotor, incluyendo la interacción con el ambiente. Tanto el sistema nervioso, como el sistema óseo y el sistema muscular desempeñan distintas

funciones para llevar a cabo la marcha; sin embargo, es su conjunto el que hace viable el desarrollo de una marcha estable. En los robots bípedos, intrínseca a la naturaleza de su arquitectura, está su gran movilidad, pero la destreza y maniobrabilidad que pueden llegar a exhibir depende tanto del sistema empleado para determinar los movimientos a producir en base a un objetivo —por ejemplo, la marcha—, como de su capacidad para modificarlos según los cambios en el entorno.

Debido a que ya se contaba con un prototipo físico, no se tocaron los temas de diseño y manufactura. El objetivo de esta tesis se centró en el sistema empleado para determinar los movimientos a producir por el bípedo con la intención de desarrollar patrones de caminata dinámicamente estables sobre una superficie horizontal y en el diseño de una interfaz para facilitar la implementación de dichos patrones. El diseño e implementación de un controlador para modificar los patrones de caminata según los cambios en el entorno, se propone como trabajo a futuro.

Este trabajo surgió con el propósito de darle continuidad a la investigación presentada en [43]; es por ello que se emplearon las ecuaciones cinemáticas de posición y el programa para su cálculo —ambos desarrollados como parte de dicha investigación— dentro de esta tesis. La interfaz gráfica de usuario para el accionamiento y manipulación del prototipo Scout, es el nexo entre los trabajos mencionados, y sirvió para validar los resultados de las ecuaciones de posición. La planificación de la caminata surgió por la necesidad de facilitar la generación de los ciclos de marcha, y la optimización de los patrones de marcha se llevó a cabo porque, con los ciclos de marcha generados con parámetros al azar, el robot caía o arrastraba el pie. Con el algoritmo para la optimización de la marcha se lograron producir patrones de caminata dinámicamente estables. En las pruebas con ciclo de marcha optimizado que se seleccionó, el robot caminaba sin caerse y levantando el pie flotante, con lo que se demostró la validez del método empleado.

La interfaz gráfica de usuario desarrollada facilitó la interacción y comunicación con el robot bípedo Scout. La opción de interacción directa con los servomotores del prototipo Scout ayudó a observar su respuesta según los parámetros con los que se habían programado y a elegir los más adecuados; además, permitió determinar los máximos y mínimos de los parámetros empleados para definir la trayectoria de la cadera. Para la implementación de los ciclos de marcha solo fue necesario crear un archivo con los datos de las posiciones angulares para los servomotores; LabVIEW y el microcontrolador se encargaban de enviarles dichos datos. La incorporación del modelo virtual otorgó la facilidad de simular ciclos de marcha, lo cual permitió observar su movimiento ideal y determinar si existía algún choque entre los eslabones de la estructura. Al contar con una interfaz propia es posible realizar los cambios necesarios para investigaciones futuras: por ejemplo, añadir el servicio de captura de datos para medir el valor de sensores de fuerza.

El método usado para la planificación de la caminata, planificación mediante parámetros, permitió generar las tres etapas de la caminata: puesta en marcha, etapa cíclica y paro de la marcha; cada una con sus fases de soporte doble y soporte simple. Con este método fue muy simple generar patrones de marcha y con características distintas. Considerar parámetros de velocidad y de orientación incrementó el tipo de patrones a producir.

Con la optimización de la caminata fue posible reproducir ciclos de marcha con estabilidad dinámica. Con el método empleado se eligieron las trayectorias deseadas para los pies, mientras que el algoritmo genético determinó la trayectoria óptima para la cadera. Aunque la longitud del paso influyó en los resultados obtenidos, el haber incluido parámetros de velocidad y orientación para definir la trayectoria de la cadera, permitió al algoritmo, en todos los casos, producir elementos con un *fitness* elevado y

conseguir ciclos de marcha cuyo ZMP se mantuviera dentro o cercano al área de seguridad. Se considera que el implementar el mismo algoritmo en robots con tronco o humanoides mejoraría su desempeño.

En las pruebas con el prototipo físico se observó que la estructura vibraba durante la marcha, que los motores tenían problemas para seguir las trayectorias de referencia y que el bípedo era muy sensible a perturbaciones y cambios en el entorno; sin embargo, el patrón de marcha seleccionado se logró implementar exitosamente.

Con la revisión bibliográfica y el trabajo desarrollado en esta tesis fue posible sumergirse en el mundo de la robótica bípeda, entender sus objetivos y metas, conocer los problemas que plantea y estar al tanto de los temas actuales y de sus pretensiones a futuro. En la investigación realizada se utilizaron conceptos y se partió de métodos ampliamente empleados en diversos artículos sobre robótica bípeda. A diferencia de otros trabajos (como [11] o [27]) donde se emplea la planificación mediante parámetros, aquí no solamente se incluyeron trayectorias para definir la posición de los efectores finales, sino también para definir a la orientación de la cadera, lo que dota de mayor movilidad al robot y permite construir patrones de marcha más variados; además, expandió el campo de búsqueda de los algoritmos genéticos, a lo que se debe, en buena medida, sus resultados. El algoritmo para la optimización de la marcha encontró patrones con estabilidad dinámica tan solo modificando la trayectoria de la cadera. Este método puede llegar a ser sumamente útil para dotar al robot de versatilidad y permitirle realizar movimientos más sofisticados, con los que no solamente se logre mejorar la marcha, sino también desarrollar otro tipo de movimientos y habilidades, como patear un balón con cierta dirección y fuerza.

8.2. Trabajo a futuro

Ya que se ha visto que el método para la optimización de la marcha ha dado resultados favorables, el trabajo realizado se puede tomar como base para investigaciones posteriores. Se proponen las siguientes ideas:

- Con los parámetros empleados para la planificación de la caminata solo es posible generar ciclos de marcha con movimiento restringido al plano sagital. Se propone como trabajo a futuro añadir a la planificación de la caminata la posibilidad de generar ciclos de marcha con movimiento en el plano frontal, además de generar patrones para subir o bajar escaleras.
- Dado el tiempo que lleva ejecutar el algoritmo y las ventajas de generar patrones de marcha en modo *online*, es necesario buscar una alternativa para reducir el tiempo de evaluación de los individuos. La solución numérica de las ecuaciones cinemáticas de posición consume mucho tiempo. Se propone como trabajo a futuro diseñar una red neuronal para, dada la posición y orientación de los eslabones L_{61} , L_{62} y L_B del robot, encontrar los valores de los ángulos asociados a las juntas rotacionales que permiten al bípedo asumir la configuración espacial determinada por dichas posiciones y orientaciones.
- La respuesta de los servomotores mermó el desempeño del robot. Para que los servomotores puedan igualar sus movimientos a los de las trayectorias de referencia, es necesario mejorar su control, por lo cual se propone diseñar un controlador propio para los servomotores.
- Los patrones de marcha implementados son ideales: si existen errores en las idealizaciones o se producen perturbaciones, es muy probable que el robot caiga. Por lo cual se propone el

diseño e implementación de un controlador para cambiar las trayectorias de referencia para los servomotores según los cambios en el ambiente o las perturbaciones que se puedan llegar a producir. El control en lazo cerrado requiere conocer el estado real del sistema. Por lo cual también es necesaria su sensorización.

Apéndice A

Código fuente para la síntesis de la marcha

A.1. Programa principal. Archivo SintesisMarcha.m

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%SÍNTESIS DE LA MARCHA%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%POSICIONES INICIALES:
%Pie izquierdo:
x1 = 44.54;
y1 = 0;
z1 = 0;
%Pie derecho:
x2 = 44.54;
y2 = 0;
z2 = 0;
%Cadera:
yB = 38.5;
zBa = 198.37;
paB = 0;
pbBa = 0;
%(pie derecho se mueve primero).
%%
%PARÁMETROS:
%(En milímetros).
%(Primer columna: puesta en marcha; segunda y tercer columna: etapa
%cíclica de la marcha; última columna: paro de la marcha).
%Pie flotante:
pf_x = [5 5 5 5];
pf_y = [40 80 80 40];
pf_z = [30 50 50 30];
pf_zf = [0 0 0 0];%
%Velocidades:
pf_vx = [0 0 0 0];
pf_vy = [0 0 0 0];
pf_vz = [0 0 0 0];

%Cadera:
c_x = [-50 50 -50 50];
```

```

c_y = [pf_y/2 pf_y/2 pf_y/2 pf_y/2];
c_z = [15+pf_zf(1) 15+pf_zf(1) 15+pf_zf(1) 15+pf_zf(1)];
c_zf = [pf_zf(1) pf_zf(2) pf_zf(3) pf_zf(4)];%
c_pa = [15 -30 30 -15]*.5;
c_pb = [5 5 5 5]*1;
c_pbf = [0 0 0 0];%
%Velocidades:
%En x:
c_vx = [20 -20 20 -20]*0;
c_vxi = [0 1 -1 1]*30;%
c_vxf = [1 -1 1 0]*30;%
%En y:
c_vy = [1 1 1 1]*5;
c_vyi = [0 1 1 1]*1;%
c_vyf = [1 1 1 0]*1;%
%En z:
c_vz = [-5 -5 -5 -5];
%
c_vpa = [0 0 0 0];
c_vpb = [0 0 0 0];

%Parámetros de tiempo:
T = 10;
rate = 20;%Número par.
por_tDS = 30;

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%INICIALIZACIÓN ARREGLOS DE TRAYECTORIAS:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Pie izquierdo:
tx1 = zeros(rate*4 +1, 1);
ty1 = zeros(rate*4 +1, 1);
tz1 = zeros(rate*4 +1, 1);
%Pie derecho:
tx2 = zeros(rate*4 +1, 1);
ty2 = zeros(rate*4 +1, 1);
tz2 = zeros(rate*4 +1, 1);
%Torso:
txB = zeros(rate*4 +1, 1);
tyB = zeros(rate*4 +1, 1);
tzB = zeros(rate*4 +1, 1);
tpaB = zeros(rate*4 +1, 1);
tpbB = zeros(rate*4 +1, 1);

tx1(1) = x1;
ty1(1) = y1;
tz1(1) = z1;

tx2(1) = x2;
ty2(1) = y2;
tz2(1) = z2;

tyB(1) = yB;
tzB(1) = zBa;
typaB(1) = paB;

```

```

typbB(1) = pbBa;

%INICIALIZACIÓN BANDERAS DE ERROR:
ePtosItray = zeros(4, 8);
numPI = [1, 0, 1, 1, 0, 1, 1, 1];

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%GENERACIÓN DE LAS TRAYECTORIAS PARA LOS ESLABONES B, L61 Y L62:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j = 1: size(pf_x,2)
    ind = rate*(j-1) +1;

    %VECTORES:
    x = [pf_x(j), c_x(j)];
    y = [pf_y(j), c_y(j)];
    z = [pf_z(j), pf_zf(j), c_z(j), c_zf(j)];
    pa = c_pa(j);
    pb = [c_pb(j), c_pbf(j)];

    vx = [pf_vx(j), c_vx(j), c_vxi(j), c_vxf(j)];
    vy = [pf_vy(j), c_vy(j), c_vyi(j), c_vyf(j)];
    vz = [pf_vz(j), c_vz(j)];
    vp = [c_vpa(j), c_vpb(j)];

    %CELIDAS
    parPos = {x, y, z, pa, pb};
    parVel = {vx, vy, vz, c_vpa, c_vpb};
    parTpo = [T, rate, por_tDS];

    [trayPF, trayCad] = traySplines(parPos, parVel, parTpo);

    if rem(j,2) == 0
        tx1(ind:ind+rate) = trayPF(:,1) + tx1(ind,1);
        ty1(ind:ind+rate) = trayPF(:,2) + ty1(ind,1);
        tz1(ind:ind+rate) = trayPF(:,3) + tz1(ind,1);

        tx2(ind:ind+rate) = tx2(ind,1);
        ty2(ind:ind+rate) = ty2(ind,1);
        tz2(ind:ind+rate) = tz2(ind,1);
    else
        tx1(ind:ind+rate) = tx1(ind,1);
        ty1(ind:ind+rate) = ty1(ind,1);
        tz1(ind:ind+rate) = tz1(ind,1);

        tx2(ind:ind+rate) = trayPF(:,1) + tx2(ind,1);
        ty2(ind:ind+rate) = trayPF(:,2) + ty2(ind,1);
        tz2(ind:ind+rate) = trayPF(:,3) + tz2(ind,1);
    end
    end
    txB(ind:ind+rate) = trayCad(:,1) + txB(ind,1);
    tyB(ind:ind+rate) = trayCad(:,2) + tyB(ind,1);
    tzB(ind:ind+rate) = trayCad(:,3) + tzB(ind,1);
    tpaB(ind:ind+rate) = trayCad(:,4) + tpaB(ind,1);
    tpbB(ind:ind+rate) = trayCad(:,5) + tpbB(ind,1);

```

```

%Error por no cumplir con el número de puntos de inflexión:
for i = 1:size(trayPF,2)
    f = trayPF(:,i);
    indTraySel = i;
    ePtsItray(j,indTraySel) = errorPtsI(f, numPI(indTraySel));
end
for i = 1:size(trayCad,2)
    f = trayCad(:,i);
    indTraySel = i+size(trayPF,2);
    ePtsItray(j,indTraySel) = errorPtsI(f, numPI(indTraySel));
end
end
a=1

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CÁLCULO CINEMÁTICA INVERSA:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(Solo llama al programa de Mathematica).
esc=1/10;
arrTrayMath = [txB, tyB, tzB, tpaB/esc, tx1, ty1, tz1, tx2, ty2,
tz2]*esc;
angs=zeros(rate*4+1, 12);%Inicialización.
ModCin();%Modelo cinemático del robot.
for j=1:rate*4 +1
    [theta_I,theta_D]=CalCinInv(arrTrayMath(j,:), j);
    angs(j,:) = [theta_D(1), theta_I(1) , theta_D(2) , theta_I(2) ,
theta_D(3) , theta_I(3) , theta_D(4) , theta_I(4) , theta_D(5) ,
theta_I(5) , theta_D(6) , theta_I(6)];
end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GENERACIÓN DE ARCHIVOS:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ARCHIVO MATHEMATICA:
%(Genera un archivo para ser emleado en la simulación del robot bípedo
%de Mathematica).
xlswrite('trayectoriasR.xls', arrTrayMath);
% %ARCHIVO LabVIEW:
%(Genera un archivo con los datos de las trayectorias angulares de cada
%junta rotacional para ser usada en el programa de LabVIEW).
% %Offsets y escritura del archivo:
rot=zeros(rate*4 +1, 14);%Inicialización.
offsets = [30, 136, 29, 131, 105, 65, 80, 86, 113, 59, 57, 111];
signAngs= [1, 1, -1, 1, -1, -1, 1, 1, -1, -1, -1, -1];
for j=1:12
    rot(:,j)= angs(:,j)*signAngs(j) + offsets(j);
end
rot(:,13)=[ones(rate+1,1)', ones(rate,1) '*2, ones(rate,1)',
ones(rate,1) '*2];
rot(:,14)=[ones(rate+1,1)', ones(rate,1) '*2, ones(rate,1) '*2,
ones(rate,1) '*3];
rot=ceil(rot);

```



```

%%
%INICIALIZACIÓN:
%Pie izquierdo:
tx1 = zeros((rate+1), 1);
ty1 = zeros((rate+1), 1);
tz1 = zeros((rate+1), 1);
%Torso:
txB = zeros((rate+1), 1);
tyB = zeros((rate+1), 1);
tzB = zeros((rate+1), 1);
tpaB = zeros((rate+1), 1);
tpbB = zeros((rate+1), 1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GENERACIÓN DE TRAYECTORIAS:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Tiempo:
indTmed = round((rate+1)/2);
indTfin = rate+1;
t1 = t(1:indTmed);
t2 = t(indTmed:indTfin) - t(indTmed);

%Trayectorias:
txB(1:indTmed) = spline3(0, c_x, c_vxi, c_vx, t1);
txB(indTmed:indTfin) = spline3(c_x, 0, c_vx, c_vxf, t2);

tyB(1:indTmed) = spline3(0, c_y/2, c_vyi, c_vy, t1);
tyB(indTmed:indTfin) = spline3(c_y/2, c_y, c_vy, c_vyf, t2);

tzB(1:indTmed) = spline3(0, c_z, 0, c_vz, t1);
tzB(indTmed:indTfin) = spline3(c_z, c_zf, c_vz, 0, t2);

%Trayectorias que se realizan solo en fase de soporte simple:
%Tiempo:
i = 1;
while t(i) ~= tss(1)
    i = i + 1;
end
tss = tss - tss(1);
indTmed = round((rateSS+1)/2);
indTfin = rateSS + 1;
t1 = tss(1:indTmed);
t2 = tss(indTmed:indTfin) - tss(indTmed);
indTmed = indTmed + i-1;
indTfin = indTfin + i-1;

%Trayectorias:
tpaB(i:indTfin) = spline3(0, c_pa, 0, 0, tss);

tpbB(i:indTmed) = spline3(0, c_pb, 0, 0, t1);
tpbB(indTmed:indTfin) = spline3(c_pb, c_pbf, 0, 0, t2);

tx1(i:indTmed) = spline3(0, pf_x, 0, pf_vx, t1);
tx1(indTmed:indTfin) = spline3(pf_x, 0, pf_vx, 0, t2);

ty1(i:indTmed) = spline3(0, pf_y/2, 0, pf_vy, t1);

```



```

ty1(indTmed:indTfin) = spline3(pf_y/2, pf_y, pf_vy, 0, t2);

tz1(i:indTmed) = spline3(0, pf_z, 0, pf_vz, t1);
tz1(indTmed:indTfin) = spline3(pf_z, pf_zf, pf_vz, 0, t2);

tpaB(indTfin+1:rate+1) = c_pa;
tpbB(indTfin+1:rate+1) = 0;
tx1(indTfin+1:rate+1) = 0;
ty1(indTfin+1:rate+1) = pf_y;
tz1(indTfin+1:rate+1) = pf_zf;

%Salidas:
trayPF = [tx1, ty1, tz1];
trayCad = [txB, tyB, tzB, tpaB, tpbB];

```

A.3. Programación del polinomio cúbico. Archivo spline3.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SPLINE CÚBICA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function x = spline3(x0, x1, v0, v1, t)
T = t(end) - t(1);
sdoTer = (3*(x1-x0) - T*(2*v0+v1)) * t.^2/T^2;
terTer = (2*(x0-x1) + T*(v0+v1)) * t.^3/ T^3;
x= x0 + v0*t + sdoTer + terTer;
end

```

A.4. Cálculo de la cinemática inversa¹. Archivo CalCinInv.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CÁLCULO DE LA CINEMÁTICA INVERSA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(Solamente llama a programa de Mathematica empleando la función Math).
function [theta_I,theta_D]=CalCinInv(arrTray,i)
% Solucion total
xB = arrTray(:,1);
yB = arrTray(:,2);
zB = arrTray(:,3);
psiB = arrTray(:,4);
x1 = arrTray(:,5);
y1 = arrTray(:,6);
z1 = arrTray(:,7);
x2 = arrTray(:,8);
y2 = arrTray(:,9);
z2 = arrTray(:,10);

```

¹ Solamente llama al programa de Mathematica, empleando la función *math()* de la herramienta *Mathematica Symbolic Toolbox for MATLAB*.

```

math('Clear[theta11, theta21, theta31, theta41, theta51, theta61,
theta12, theta22, theta32, theta42, theta52, theta62];');
math('Clear[xB, yB, zB, thetaB, phiB, psiB, x1, y1, z1, theta1, phi1,
psi1, x2, y2, z2, theta2, phi2, psi2];');

math('theta11i = theta11 /. SolInicialI;');
math('theta21i = theta21 /. SolInicialI;');
math('theta31i = theta31 /. SolInicialI;');
math('theta41i = theta41 /. SolInicialI;');
math('theta51i = theta51 /. SolInicialI;');
math('theta61i = theta61 /. SolInicialI;');

math('theta12i = theta12 /. SolInicialD;');
math('theta22i = theta22 /. SolInicialD;');
math('theta32i = theta32 /. SolInicialD;');
math('theta42i = theta42 /. SolInicialD;');
math('theta52i = theta52 /. SolInicialD;');
math('theta62i = theta62 /. SolInicialD;');

%Valores constantes para las posiciones del torso del bípodo
math('thetaB = 0°;');
math('phiB = 0 °;');
math('psiB = 0 °;');

%Valores constantes para las posiciones del pie izquierdo del bípodo
math('x1 = esc*44.54;');
math('theta1 = 0 °;');
math('phi1 = 0 °;');
math('psi1 = 0 °;');

%Valores constantes para las posiciones del pie derecho del bípodo
math('x2 = esc*44.54;');
math('theta2 = 0 °;');
math('phi2 = 0 °;');
math('psi2 = 0 °;');
% Obtención de las coordenadas articulares solución de las
trayectorias
% definidas
math(['xB=' char(sym(xB)) ';'']);
math(['yB=' char(sym(yB)) ';'']);
math(['zB=' char(sym(zB)) ';'']);
math(['psiB=' char(sym(psiB)) ';'']);
math('psiB= psiB °;');

math(['x1=' char(sym(x1)) ';'']);
math(['y1=' char(sym(y1)) ';'']);
math(['z1=' char(sym(z1)) ';'']);

math(['x2=' char(sym(x2)) ';'']);
math(['y2=' char(sym(y2)) ';'']);
math(['z2=' char(sym(z2)) ';'']);

tmp=['SolPosI[' char(sym(i+1)) ']' = FindRoot[{PosI1[[1, 4]] ==
PosI2[[1, 4]],PosI1[[2, 4]] == PosI2[[2, 4]],PosI1[[3, 4]] == PosI2[[3,

```

```

4]],PosI1[[1, 1]] == PosI2[[1, 1]],PosI1[[2, 2]] == PosI2[[2,
2]],PosI1[[3, 3]] == PosI2[[3, 3]],{theta11, theta11i},{theta21,
theta21i},{theta31, theta31i},{theta41, theta41i},{theta51,
theta51i},{theta61, theta61i},MaxIterations -> 20, AccuracyGoal -> 2,
PrecisionGoal ->2];'];

    math(tmp);
    theta_I=math('math2matlab',[('({theta11, theta21, theta31, theta41,
theta51, theta61} /. SolPosI[' char(sym(i+1)) '])/°']);
    math(['theta11i = theta11 /. SolPosI[' char(sym(i+1)) '];']);
    math(['theta21i = theta21 /. SolPosI[' char(sym(i+1)) '];']);
    math(['theta31i = theta31 /. SolPosI[' char(sym(i+1)) '];']);
    math(['theta41i = theta41 /. SolPosI[' char(sym(i+1)) '];']);
    math(['theta51i = theta51 /. SolPosI[' char(sym(i+1)) '];']);
    math(['theta61i = theta61 /. SolPosI[' char(sym(i+1)) '];']);

    tmp=['SolPosD[' char(sym(i+1)) ' ] = FindRoot[{PosD1[[1, 4]] ==
PosD2[[1, 4]],PosD1[[2, 4]] == PosD2[[2, 4]],PosD1[[3, 4]] == PosD2[[3,
4]],PosD1[[1, 1]] == PosD2[[1, 1]],PosD1[[2, 2]] == PosD2[[2,
2]],PosD1[[3, 3]] == PosD2[[3, 3]],{theta12, theta12i},{theta22,
theta22i},{theta32, theta32i},{theta42, theta42i},{theta52,
theta52i},{theta62, theta62i},MaxIterations -> 20, AccuracyGoal -> 2,
PrecisionGoal ->2];'];

    math(tmp);
    theta_D=math('math2matlab',[('({theta12, theta22, theta32, theta42,
theta52, theta62} /. SolPosD[' char(sym(i+1)) '])/°']);
    math(['theta12i = theta12 /. SolPosD[' char(sym(i+1)) '];']);
    math(['theta22i = theta22 /. SolPosD[' char(sym(i+1)) '];']);
    math(['theta32i = theta32 /. SolPosD[' char(sym(i+1)) '];']);
    math(['theta42i = theta42 /. SolPosD[' char(sym(i+1)) '];']);
    math(['theta52i = theta52 /. SolPosD[' char(sym(i+1)) '];']);
    math(['theta62i = theta62 /. SolPosD[' char(sym(i+1)) '];']);
end

```

Apéndice B

Código fuente para la optimización de la marcha

B.1. Programa principal. Archivo AlgGen.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%ALGORITMO GENÉTICO PARA LA%%%%%%%%%
%%%%%%%%%LA OPTIMIZACIÓN DE LA MARCHA%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PARÁMETROS PARA LA CONFIGURACIÓN DE LA MARCHA:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Valores iniciales:
x1 = 44.54;
x2 = 44.54;

yB = 38.5;
zB = 234.37;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parámetros para el pie flotante:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pf_x = 5;
pf_y = 150;
pf_z = 40;
pf_zf = 0;%
%Velocidades:
pf_vx = 0;
pf_vy = 0;
pf_vz = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parámetros de tiempo:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T = 10;
rate = 20;%Número par.
```

```

por_tDS = 30;
parTpo = [T, rate, por_tDS];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PARÁMETROS PARA EL ALGORITMO GENÉTICO:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tipo = 'c';
tamPob = 30;
maxEpoch = 15;
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GENERACIÓN DE LA POBLACIÓN:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cte = [0, 0, pf_y/2, 0, pf_zf, 0];
[parTxB, parTyB, parTzB, parTpsiB] = GenPob(parTpo, tipo, tamPob, cte);
poblacion = [parTxB, parTyB, parTzB, parTpsiB];
%%
aveFitness = zeros(maxEpoch,1);
for epoch = 1:maxEpoch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%EVALUACIÓN DE LA POBLACIÓN:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Q =zeros(tamPob,1);%Función de costo.
for i = 1:tamPob
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GENERACIÓN DE LAS TRAYECTORIAS PARA LOS EFECTORES FINALES:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parámetros:
%(En milímetros).
parTx1 = [pf_x, pf_vx];
parTy1 = [pf_y, pf_vy];
parTz1 = [pf_z, pf_zf, pf_vz];

parTCad = [parTxB(i,:), parTyB(i,:), cte(3), parTzB(i,:),
parTpsiB(i,:)];
parTPf = [parTx1, parTy1, parTz1];

%Trayectorias:
[trayPF, trayCad] = GenTray(parTCad, parTPf, parTpo);

%Trayectorias del pie izquierdo:
tx1 = trayPF(:,1) + x1;
ty1 = trayPF(:,2);
tz1 = trayPF(:,3);

%Trayectorias del pie derecho:
if tipo == 'c'
tx2 = 0*ones(rate+1,1) + x2;
ty2 = pf_y/2*ones(rate+1,1);
tz2 = pf_zf/2*ones(rate+1,1);
elseif tipo == 'i'
tx2 = 0*ones(rate+1,1) + x2;
ty2 = 0*ones(rate+1,1);
tz2 = 0*ones(rate+1,1);
elseif tipo == 'p'
tx2 = 0*ones(rate+1,1) + x2;

```

```

    ty2 = pf_y*ones(rate+1,1);
    tz2 = pf_zf*ones(rate+1,1);
end

%Posiciones para la cadera:
if tipo == 'c'
    txB = trayCad(:,1);
    tyB = trayCad(:,2) + pf_y/4;
    tzB = trayCad(:,3) + zB;
    tpaB = -trayCad(:,4) + trayCad(end,4)/2;
elseif tipo == 'i'
    txB = trayCad(:,1);
    tyB = trayCad(:,2);
    tzB = trayCad(:,3) + zB;
    tpaB = trayCad(:,4);
elseif tipo == 'p'
    txB = trayCad(:,1);
    tyB = trayCad(:,2) + pf_y/2;
    tyB = tyB - (tyB(end) - 38.5 - pf_y);

    tzB = trayCad(:,3) + zB;
    tpaB = -trayCad(end,4) + trayCad(:,4);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%TRAYECTORIA DEL CENTRO DE GRAVEDAD:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%En centímetros.
esc=1/10;
arrTrayMath = [txB, tyB, tzB, tpaB/esc, tx1, ty1, tz1, tx2, ty2,
tz2]*esc;
if epoch == 1 && i == 1
    ModCin();%Modelo cinemático del robot.
end
angs=zeros(rate+1, 12);
trayCdeG = zeros(rate+1, 3);
for j=1:rate+1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Solución de las coordenadas articulares de ambas piernas:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [theta_I,theta_D]=CalCinInv(arrTrayMath(j,:), j);
    angs(j,:) = [theta_D(1), theta_I(1) , theta_D(2) , theta_I(2) ,
theta_D(3) , theta_I(3) , theta_D(4) , theta_I(4) , theta_D(5) ,
theta_I(5) , theta_D(6) , theta_I(6)];

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Cálculo del centro de gravedad:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    try
        trayCdeG(j, :) = CdeG(angs(j,1:12));%En centímetros.
    catch err
        %'error'
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Cálculo de la aceleración del centro de gravedad:

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
At = T/rate;
t = (0:At:T)';
pp=spline(t, trayCdeG(: , 1));%x'' (en centímetros).
p_der=fnder(pp,2);
x_biprime=ppval(p_der,t);

pp=spline(t, trayCdeG(: , 2));%y'' (en centímetros).
p_der=fnder(pp,2);
y_biprime=ppval(p_der,t);

pp=spline(t, trayCdeG(: , 3));%z'' (en centímetros).
p_der=fnder(pp,2);
z_biprime=ppval(p_der,t);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CÁLCULO DEL ZMP:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%En centímetros.
g=9.8*10^2;%[cm/s^2].
xzmp=(trayCdeG(:,1).*z_biprime - trayCdeG(:,3).*x_biprime)/g +
trayCdeG(:,1);
yzmp=(trayCdeG(:,2).*z_biprime - trayCdeG(:,3).*y_biprime)/g +
trayCdeG(:,2);
%Posición del ZMP con respecto al pie de apoyo:
xzmp = -tx2(1)/10+xzmp;
yzmp = -ty2(1)/10-2+yzmp;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%EVALUACIÓN DEL ZMP:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(Distancia del ZMP con respecto al área de seguridad).

d_ZMP = zeros(rate+1,1);
z1DS = tz1(1);
for j = 1:rate+1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %En fase de soporte doble:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if tz1(j) == z1DS
        %Bandera:
        %band1= 0;
        %Vértices:
        %ver_x1 = 2.5;
        %ver_x2 = 0;
        %ver_x3 = -tx1(j)/10 - tx2(j)/10;
        %ver_x4 = ver_x3 - 2.5;

        %ver_y1 = 0;
        %ver_y2 = ver_y1;
        %ver_y3 = ty1(j)/10 - ty2(j)/10;
        %ver_y4 = ver_y3;

        %Trazo del área de seguridad:
        %trayx = [ver_x1, ver_x2, ver_x3, ver_x4, ver_x4, ver_x3,
ver_x2, ver_x1, ver_x1];

```

```

%trayy = [ver_y1, ver_y2, ver_y3, ver_y4, ver_y4+5, ver_y3+5,
ver_y2+5, ver_y1+5, ver_y1];
%plot(trayx,trayy, xzmp(j),yzmp(j),'rs')

%Regla de correspondencia:
%if xzmp(j) >= ver_x4 && xzmp(j) <= ver_x3
    %yzmp_min = ver_y4;
%elseif xzmp(j) > ver_x3 && xzmp(j) < ver_x2
    %m1 = (ver_y3-ver_y2)/(ver_x3-ver_x2);
    %b1 = ver_y2 - m1*ver_x2;
    %yzmp_min = m1*xzmp(j)+b1;
%elseif xzmp(j) >= ver_x2 && xzmp(j) <= ver_x1
    %yzmp_min = ver_y1;
%else
    %band1 = 1;
%end

%Distancia del ZMP con respecto al área de seguridad:
%if band1 == 0 && yzmp(j) >= yzmp_min && yzmp(j) <= yzmp_min
+ 5
    %d_ZMP(j) = 0;
%elseif xzmp(j)<ver_x4 && yzmp(j)>=ver_y4 &&
yzmp(j)<=ver_y4+5
    %d_ZMP(j) = xzmp(j) - ver_x4;
%elseif xzmp(j)>ver_x1 && yzmp(j)>=ver_y1 &&
yzmp(j)<=ver_y1+5
    %d_ZMP(j) = xzmp(j) - ver_x1;
%else
    %d1 = DistZMP(ver_x1, ver_y1, ver_x2, ver_y2, xzmp(j), yzmp(j));
    %d2 = DistZMP(ver_x2, ver_y2, ver_x3, ver_y3, xzmp(j), yzmp(j));
    %d3 = DistZMP(ver_x3, ver_y3, ver_x4, ver_y4, xzmp(j), yzmp(j));
    %d4 = DistZMP(ver_x1, ver_y1+5, ver_x2, ver_y2+5, xzmp(j), yzmp(j));
    %d5 = DistZMP(ver_x2, ver_y2+5, ver_x3, ver_y3+5, xzmp(j), yzmp(j));
    %d6 = DistZMP(ver_x3, ver_y3+5, ver_x4, ver_y4+5, xzmp(j), yzmp(j));

    %d_ZMP(j) = min([d1, d2, d3, d4, d5, d6]);
    d_ZMP(j) = 0;
%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%En fase de soporte simple:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
else
    z1DS = tz1(end);
    band1= 0;
    %Vértices:
    ver_x1 = -1.318;
    ver_x2 = 3.133;

    ver_y1 = -1.87;
    ver_y2 = -1.87;
    ver_y3 = 6;
    ver_y4 = 6;

    %Trazo del área de seguridad:
    %trayx = [ver_x1, ver_x2, ver_x2, ver_x1, ver_x1];
    %trayy = [ver_y1, ver_y2, ver_y2+5, ver_y1+5, ver_y1];

```



```

plot(trayx,trayy, xzmp(j),yzmp(j),'rs')

%Regla de correspondencia:
if xzmp(j) >= ver_x2 && xzmp(j) <= ver_x1
    yzmp_min = ver_y1;
else
    band1 = 1;
end

%Distancia del ZMP con respecto al área de seguridad:
if band1 == 0 && yzmp(j) >= ver_y1 && yzmp(j) <= ver_y3
    d_ZMP(j) = 0;
elseif xzmp(j)<ver_x2 && yzmp(j)>=ver_y1 && yzmp(j)<=ver_y3
    d_ZMP(j) = xzmp(j) - ver_x2;
elseif xzmp(j)>ver_x1 && yzmp(j)>=ver_y1 && yzmp(j)<=ver_y3
    d_ZMP(j) = xzmp(j) - ver_x1;
else
    d1 = DistZMP(ver_x1, ver_y1, ver_x2, ver_y2, xzmp(j), yzmp(j));
    d2 = DistZMP(ver_x1, ver_y3, ver_x2, ver_y4, xzmp(j), yzmp(j));

    d_ZMP(j) = min([d1, d2]);
end
end
end
%Función de costo:
%   clc
%   i
Q(i) = 1/sum(d_ZMP.^2);
end
% clc
aveFitness(epoch)=sum(Q)/tamPob;
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SELECCIÓN POR TORNEO:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
miemSel=zeros(tamPob*2,1);%Miembros seleccionados en torneo.

probSel=Q/sum(Q)*100;%Asigna una probabilidad a cada elemento.
for i=1:tamPob*2
    probSel2=zeros(tamPob,1);%Probabilidad de selección2.

    sumProb=0;
    numAl=rand*100;
    j=0;
    while sumProb <= numAl && j < tamPob
        sumProb=sumProb+probSel(j+1);
        j=j+1;
    end
    miemSel(i)=j;

%Modificación de la probabilidad:
% np=probSel(j)*.96;%Nueva probabilidad del elemento seleccionado.
% probSel=probSel + ( probSel(j) *.1 / (tamPob-1) );
% probSel(j)=np;
%
%     np=probSel(j)*.15;

```

```

%     probSel(j)=probSel(j)*.85;
%
%     desvProbSel = std(probSel);
%     mediaProbSel = mean(probSel);
%     limInf= mediaProbSel - desvProbSel*2;
%     limSup= mediaProbSel + desvProbSfel*2;
%
%     ind=0;
%     for k=1:tamPob
%         if probSel(k) > limInf && probSel(k) < limSup
%             ind=ind+1;
%             probSel2(ind) = probSel(k);
%         end
%     end
%
%     probSel2 = probSel2/sum(probSel2)*np;
%
%     ind=0;
%     for k=1:tamPob
%         if probSel(k) > limInf && probSel(k) < limSup
%             ind=ind+1;
%             probSel(k) = probSel(k) + probSel2(ind);
%         end
%     end
end
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%CRUZA:%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nvaPoblacion=zeros(tamPob,13);

sep = [1,4; 5,8; 9,12; 13,13];
for i=1:tamPob
    m1=miemSel(i*2 - 1);%Índice miembro 1.
    m2=miemSel(i*2);%Índice miembro 2.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%HAY CRUZA:%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if rand(1) <= .8
        numeros=1:4;%De acuerdo al número de parámetros.
        ptoCruza=ceil(rand(1)*3);
        for j=1:4
            %Acomodando parámetros en orden aleatorio)%
            indElSel = ceil(rand(1)*(5-j));%Índice del elemento
            seleccionado.
            a = sep(numeros(indElSel), 1);
            b = sep(numeros(indElSel), 2);
            if j <= ptoCruza
                nvaPoblacion(i,a:b)=poblacion(m1,a:b);
            else
                nvaPoblacion(i,a:b)=poblacion(m2,a:b);
            end
            numeros(indElSel)=[];%Elimina elemento.
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%NO HAY CRUZA:%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

else
    if rand(1) <= probSel(m1)/(probSel(m1) + probSel(m2))
        nvaPoblacion(i,:)=poblacion(m1,:);
    else
        nvaPoblacion(i,:)=poblacion(m2,:);
    end
end
end
end
end
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%MUTACIÓN:%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:tamPob
    if rand(1) < .1
        individuo = nvaPoblacion(i,:);
        individuoMut = Mutacion(parTpo, tipo, cte, individuo);
        nvaPoblacion(i,:) = individuoMut;
    end
end
end
if epoch < maxEpoch
    poblacion = nvaPoblacion;
    parTxB = poblacion(:,1:4);
    parTyB = poblacion(:,5:8);
    parTzB = poblacion(:,9:12);
    parTpsiB = poblacion(:,13);
end
end

i = 1;
while Q(i) ~= max(Q)
    i = i+1;
end

if tipo == 'c'
    parTCadC = [parTxB(i,:), parTyB(i,:), cte(3), parTzB(i,:), parTpsiB(i,:)];
    parTPfC = [parTx1, parTy1, parTz1];

    c_vxif_c = parTxB(i,3);
    c_vyif_c = parTyB(i,4);
    c_y_c = pf_y/4+parTyB(i,1);
    cte=[c_vxif_c, c_vyif_c, c_y_c, parTzB(i,1), cte(5)/2, parTpsiB(i)/2];

    tipo = 'i';
    pf_y = pf_y /2;
    pf_zf = pf_zf/2;

elseif tipo == 'i'
    parTCadI = [parTxB(i,:), parTyB(i,:), cte(3), parTzB(i,:),
parTpsiB(i,:)];
    parTPfI = [parTx1, parTy1, parTz1];

    c_vxif_c = parTxB(i,3);
    c_vyif_c = parTyB(i,4);
    c_y_c = pf_y - c_y_c;
    cte=[c_vxif_c, c_vyif_c, c_y_c, parTzB(i,2)+pf_zf*2, cte(5), parTpsiB(i)];

```

```

    tipo = 'p';

elseif tipo == 'p'
    parTCadF = [parTxB(i,:), parTyB(i,:), cte(3), parTzB(i,:), parTpsiB(i,:)];
    parTPfF = [parTx1, parTy1, parTz1];
end
%Escribe la secuencia de las trayectorias angulares para realizar el
ciclo de marcha:
xlswrite('trayectorias2.xls', arrTrayMath);

```

B.2. Generación aleatoria de la población. Archivo GenPob.m

```

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GENERADOR ALEATORIO DE POBLACIÓN:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
function [parTxB, parTyB, parTzB, parTpsiB] = GenPob(parTpo, tipo,
tamPob, cte)
%%
%PARÁMETROS Y CONFIGURACIÓN DEL TIEMPO:
%En segundos.
T = parTpo(1);
rate = parTpo(2); %Número par.
At = T/rate;
t = (0:At:T)';
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Configuración del tiempo para la etapa de soporte doble:
indTmed = round((rate+1) * (50/100));
indTfin = rate+1;
t1 = t(1:indTmed);
t2 = t(indTmed:indTfin) - t(indTmed);
%%
% tamPob = 30;
% tipo = 'c';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Trayectoria de la cadera TxB:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Constantes:
c_vxif_c = cte(1);
%Inicialización de variables:
el_c_x = zeros(30,1);
el_c_vx = zeros(30,1);
el_c_vxi = zeros(30,1);
el_c_vxf = zeros(30,1);

```

```

txB = zeros((rate+1), 1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Máximos y mínimos:
c_xm = 15;
c_xM = 80;

c_vxm = -10;
c_vxM = 10;

c_vxifm = 0;
c_vxifM = 35;

i = 1;
while i <= tamPob
    %Selección aleatoria de c_x:
    c_x = round( (c_xM - c_xm)*rand(1,1) + c_xm);

    error = 0;
    j=0;
    while error == 0 && j < 250
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Selección aleatoria de parámetros de velocidad:
        c_vx = round( (c_vxM - c_vxm)*rand(1,1) + c_vxm);
        %Valor de c_vxi y c_vxf de acuerdo al tipo de marcha:
        switch tipo
            case 'i'
                c_vxi = c_vxif_c;
                c_vxf = -c_vxif_c;
            case 'p'
                c_vxi = c_vxif_c;
                c_vxf = -c_vxif_c;
            case 'c'
                c_vxif = round( (c_vxifM - c_vxifm)*rand(1,1) + c_vxifm);
                c_vxi = c_vxif;
                c_vxf = -c_vxif;
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Trazo de la curva con splines cúbicas:
        txB(1:indTmed) = spline3(0, c_x, c_vxi, c_vx, t1);
        txB(indTmed:indTfin) = spline3(c_x, 0, c_vx, c_vxf, t2);
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Evaluación de la curva:
        error = max(txB) < c_xM & ~ errorPtsI(txB, 1);%Cero si hay error.
        j = j + 1;
    end
    if error == 1
        el_c_x(i) = c_x;
        el_c_vx(i) = c_vx;
        el_c_vxi(i) = c_vxi;
        el_c_vxf(i) = c_vxf;
        i = i+1;
    end
end
parTxB = [el_c_x, el_c_vx, el_c_vxi, el_c_vxf];
% i = 1;
%
```

```

% c_x = el_c_x(i);
% c_vx = el_c_vx(i);
% c_vxi = el_c_vxi(i);
% c_vxf = el_c_vxf(i);
%
% txB(1:indTmed) = spline3(0, c_x, c_vxi, c_vx, t1);
% txB(indTmed:indTfin) = spline3(c_x, 0, c_vx, c_vxf, t2);
% plot(1:21, txB);
% i = i+1;
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Trayectoria de la cadera TyB:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Constantes:
c_vyif_c = cte(2);
c_y_c = cte(3);
%Inicialización de variables:
el_yB = zeros(30,1);
el_c_vy = zeros(30,1);
el_c_vyi= zeros(30,1);
el_c_vyf = zeros(30,1);
tyB = zeros((rate+1), 1);
%Máximos y mínimos:
yBm = -25;
yBM = 25;

c_vym = 0;
c_vyM = 15;

c_vyifm = 0;
c_vyifM = 35;

i = 1;
while i <= tamPob
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Selección aleatoria de parámetros:
    c_vy = round( (c_vyM - c_vym)*rand(1,1) + c_vym);
    error = 0;
    j=0;
    while error == 0 && j < 25
        %Valor de yB, c_vyi y c_vyf de acuerdo al tipo de marcha:
        switch tipo
            case 'i'
                yB = 0;

                if c_vyif_c + 8 < c_vyifM
                    c_vyifM = c_vyif_c + 8;
                end

                if c_vyif_c - 8 > c_vyifm
                    c_vyifm = c_vyif_c - 8;
                end
                c_vyif = round( (c_vyifM - c_vyifm)*rand(1,1) + c_vyifm);

                %Velocidades:
                c_vyi = c_vyif;
            end
        end
        j = j + 1;
        error = error + (j < 25);
    end
    i = i + 1;
end

```

```

        c_vyf = c_vyif;
    case 'p'
        yB = 0;

        if c_vyif_c + 8 < c_vyifM
            c_vyifM = c_vyif_c + 8;
        end

        if c_vyif_c - 8 > c_vyifm
            c_vyifm = c_vyif_c - 8;
        end
        c_vyif = round( (c_vyifM - c_vyifm)*rand(1,1) + c_vyifm);

        %Velocidades:
        c_vyi = c_vyif;
        c_vyf = c_vyif;
    case 'c'
        yB = round( (yBM - yBm)*rand(1,1) + yBm);
        %Velocidades:
        c_vyif = round( (c_vyifM - c_vyifm)*rand(1,1) + c_vyifm);
        c_vyi = c_vyif;
        c_vyf = c_vyif;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Trazo de la curva con splines cúbicas:
    tyB(1:indTmed) = spline3(yB, yB+c_y_c/2, c_vyi, c_vy, t1);
    tyB(indTmed:indTfin) = spline3(yB+c_y_c/2, yB+c_y_c, c_vy, c_vyf,
t2);

    %plot(1:21, tyB);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Evaluación de la curva:
    error = ~ errorPtsI(tyB, 1);%Cero si hay error.
    j = j + 1;
end
if error == 1
    el_yB(i) = yB;
    el_c_vy(i) = c_vy;
    el_c_vyi(i) = c_vyi;
    el_c_vyf(i) = c_vyf;
    i = i+1;
end
end
parTyB = [el_yB, el_c_vy, el_c_vyi, el_c_vyf];
% i = 1;
%
% yB = el_yB(i);
% c_vy = el_c_vy(i);
% c_vyi = el_c_vyi(i);
% c_vyf = el_c_vyf(i);
%
% tyB(1:indTmed) = spline3(yB, yB+c_y_c/2, c_vyi, c_vy, t1);
% tyB(indTmed:indTfin) = spline3(yB+c_y_c/2, yB+c_y_c, c_vy, c_vyf, t2);
% plot(1:21, tyB);
% i = i+1;
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%Trayectoria de la cadera TzB:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Constantes:
zB_if = cte(4);
pApo_z = cte(5)/2;
pf_z_f = cte(5);
%Inicialización de variables:
el_zBi = zeros(30,1);
el_zBf = zeros(30,1);
el_c_z = zeros(30,1);
el_c_vz = zeros(30,1);
tzB = zeros((rate+1), 1);

%Máximos y mínimos:
if tipo == 'i' || tipo == 'p'
    zBm = -45;
    zBM = 5;

    c_zm = -45 + pf_z_f;
    c_zM = 5 + pf_z_f;
else
    if pApo_z > 0
        zBm = -35;
        zBM = 10;

        c_zm = -20 + pf_z_f;
        c_zM = 5 + pf_z_f;
    elseif pApo_z == 0
        zBm = -45;
        zBM = -10;

        c_zm = -45 + pf_z_f;
        c_zM = -10 + pf_z_f;
    else
        zBm = -20;
        zBM = 5;

        c_zm = -35 + pf_z_f;
        c_zM = 10 + pf_z_f;
    end
end

c_vzm = -15;
c_vzM = 15;

i = 1;
while i <= tamPob
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Selección aleatoria de c_z:
    c_z = round( (c_zM - c_zm)*rand(1,1) + c_zm );

    error = 0;
    j=0;
    while error == 0 && j < 25

```



```

%Valor de zB de acuerdo al tipo de marcha:
switch tipo
case 'i'
    zBi = 0;
    zBf = zB_if;
case 'p'
    zBi = zB_if;
    zBf = pf_z_f;
case 'c'
    if pApo_z > 0
        zBi = round( (zBM - zBm)*rand(1,1) + zBm);
    else
        zBi = round( (zBM - zBm)*rand(1,1) + zBm + pApo_z);
    end
    zBf = zBi + pf_z_f;

end
c_vz = round( (c_vzM - c_vzm)*rand(1,1) + c_vzm);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Trazo de la curva con splines cúbicas:
tzB(1:indTmed) = spline3(zBi, c_z, 0, c_vz, t1);
tzB(indTmed:indTfin) = spline3(c_z, zBf, c_vz, 0, t2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Evaluación de la curva:
if zBi < zBf
    error = max(tzB) < c_zM & c_z > zBi & ~ errorPtsI(tzB, 1);
else
    error = max(tzB) < c_zM & min(tzB) >= tzB(size(tzB,1)) & ~
errorPtsI(tzB, 1);
end

    j = j + 1;
    %error = c_z > zB;
end
if error == 1
    el_zBi(i) = zBi;
    el_zBf(i) = zBf;
    el_c_z(i) = c_z;
    el_c_vz(i) = c_vz;
    i = i+1;
end
end
parTzB = [el_zBi, el_zBf, el_c_z, el_c_vz];

% i = 1;
%
% zBi = el_zBi(i);
% zBf = el_zBf(i);
% c_z = el_c_z(i);
% c_vz = el_c_vz(i);
%
% tzB(1:indTmed) = spline3(zBi, c_z, 0, c_vz, t1);
% tzB(indTmed:indTfin) = spline3(c_z, zBf, c_vz, 0, t2);
% plot(1:21, tzB);
% i = i+1;
%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Trayectoria de la cadera TpsiB:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

c_pam = 0;
c_paM = 45;
%Generación aleatoria de elementos:
if tipo == 'c'
    parTpsiB = round((c_paM - c_pam)*rand(tamPob,1) + c_pam);
else
    parTpsiB = cte(6)*ones(tamPob,1);
end

```

B.2.1. Evaluación de las curvas. Archivo bandErr.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DETECCIÓN DE ERROR EN LAS CURVAS POR SU NÚMERO DE PUNTOS DE INFLEXIÓN:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function bandErr = errorPtsI(f, numPtsI)
conc = 0;
numCambCon = 0;
bandErr = 0;
ordPtoCon = f(1);
%errPer = abs(max(f)-min(f))/100;
errPer = 0;
for j = 1: size(f,1) -1
    if conc ~= 1 && f(j+1) > f(j)
        if abs(f(j+1) - ordPtoCon) > errPer || conc == 0
            numCambCon = numCambCon + 1;
        else
            numCambCon = numCambCon - 1;
        end
        conc = 1;
        ordPtoCon = f(j+1);
    elseif conc ~= -1 && f(j+1) < f(j)
        if abs(f(j+1) - ordPtoCon) > errPer || conc == 0
            numCambCon = numCambCon + 1;
        else
            numCambCon = numCambCon - 1;
        end
        conc = -1;
        ordPtoCon = f(j+1);
    end
end
if numCambCon > numPtsI+1
    bandErr = 1;
end

```

B.3. Cálculo del centro de masa². Archivo CdeM.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CÁLCULO DEL CENTRO DE MASA:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [pG]=CdeM(angs)
theta11 = angs(2);
theta21 = angs(4);
theta31 = angs(6);
theta41 = angs(8);
theta51 = angs(10);
theta61 = angs(12);

theta12 = angs(1);
theta22 = angs(3);
theta32 = angs(5);
theta42 = angs(7);
theta52 = angs(9);
theta62 = angs(11);

math(['theta11=' char(sym(theta11*pi/180)) ';'']);
math(['theta21=' char(sym(theta21*pi/180)) ';'']);
math(['theta31=' char(sym(theta31*pi/180)) ';'']);
math(['theta41=' char(sym(theta41*pi/180)) ';'']);
math(['theta51=' char(sym(theta51*pi/180)) ';'']);
math(['theta61=' char(sym(theta61*pi/180)) ';'']);

math(['theta12=' char(sym(theta12*pi/180)) ';'']);
math(['theta22=' char(sym(theta22*pi/180)) ';'']);
math(['theta32=' char(sym(theta32*pi/180)) ';'']);
math(['theta42=' char(sym(theta42*pi/180)) ';'']);
math(['theta52=' char(sym(theta52*pi/180)) ';'']);
math(['theta62=' char(sym(theta62*pi/180)) ';'']);

math('SolPosII={theta11 -> theta11, theta21 -> theta21, theta31 ->
theta31, theta41 -> theta41, theta51 -> theta51, theta61 -> theta61};');
math('SolPosDD={theta12 -> theta12, theta22 -> theta22, theta32 ->
theta32, theta42 -> theta42, theta52 -> theta52, theta62 -> theta62};');

%Vectores de posición de los centros de gravedad de los eslabones
%respecto al marco de referencia inercial

%Torso
math('bB = rB + rgB;');

%Pierna izquierda
math('b11 = rB + r01 + rg11 /. SolPosII;');
math('b21 = rB + r01 + r11 + rg21 /. SolPosII;');
```

² El cálculo se realiza en Mathematica, es por ello que gran parte del programa está dentro de la función *math()* de la herramienta *Mathematica Symbolic Toolbox for MATLAB*, que permite ejecutar código de Mathematica desde MATLAB.

```

math('b31 = rB + r01 + r11 + r21 + rg31 /. SolPosII;');
math('b41 = rB + r01 + r11 + r21 + r31 + rg41 /. SolPosII;');
math('b51 = rB + r01 + r11 + r21 + r31 + r41 + rg51 /. SolPosII;');
math('b61 = rB + r01 + r11 + r21 + r31 + r41 + r51 + rg61 /. SolPosII;');

%Pierna derecha
math('b12 = rB + r02 + rg12 /. SolPosDD;');
math('b22 = rB + r02 + r12 + rg22 /. SolPosDD;');
math('b32 = rB + r02 + r12 + r22 + rg32 /. SolPosDD;');
math('b42 = rB + r02 + r12 + r22 + r32 + rg42 /. SolPosDD;');
math('b52 = rB + r02 + r12 + r22 + r32 + r42 + rg52 /. SolPosDD;');
math('b62 = rB + r02 + r12 + r22 + r32 + r42 + r52 + rg62 /. SolPosDD;');

%Cálculo del Centro de Masa
%pG=math('bT = (mB/mT)*bB + (m11/mT)*b11 + (m21/mT)*b21 + (m31/mT)*
b31 + (m41/mT)*b41 + (m51/mT)*b51 + (m61/mT)*b61 + (m12/mT)*
(m22/mT)*b22 + (m32/mT)*b32 + (m42/mT)*b42 + (m52/mT)*
(m62/mT)*b62');
pG=math('math2matlab', 'bT = (mB/mT)*bB + (m11/mT)*b11 + (m21/mT)*b21 +
(m31/mT)*
b31 + (m41/mT)*b41 + (m51/mT)*b51 + (m61/mT)*b61 + (m12/mT)*
b12 + (m22/mT)*b22 + (m32/mT)*b32 + (m42/mT)*b42 + (m52/mT)*
b52 +
(m62/mT)*b62');

```

B.4. Cálculo de la distancia mínima del ZMP al área de seguridad. Archivo distZMP.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CÁLCULO DE LA DISTANCIA MÍNIMA DEL ZMP AL ÁREA DE SEGURIDAD:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function dist = DistZMP(x1, y1, x2, y2, px, py)
%Parámetros de prueba:
% x1 = 5;
% y1 = 0;
%
% x2 = -10;
% y2 = 3;
%
% px = 0;
% py = 9;

% x1 = ver_x3;
% y1 = ver_y3+5;
%
% x2 = ver_x4;
% y2 = ver_y4+5;
%
% px = xzmp;
% py = yzmp;

```

```

%Pendiente y ordenada al origen de la recta1:
m1 = (y2-y1)/(x2-x1);
b1 = y1 - m1 *x1;

%Pendiente y ordenada al origen de la recta2:
%Ésta es perpendicular a la anterior y pasa por el punto "p":
m2 = -1/m1;
b2 = py - m2 *px;

%Punto de intersección entre las rectas 1 y 2:
if m1 ~=0
    xint = (b1-b2) / (-1/m1-m1);
    yint = m1*xint + b1;
else
    xint = px;
    yint = b1;
end

%Abscisa mayor y menor del segmento de recta1:
if x1 > x2
    xM = x1;
    xm = x2;

    yM = y1;
    ym = y2;
else
    xM = x2;
    xm = x1;

    yM = y2;
    ym = y1;
end

%Selección del punto más cercano al segmento de recta1:
if xint < xm
    %x = xint:xM;

    xint = xm;
    yint = ym;
elseif xint > xM
    %x = xm:xint;

    xint = xM;
    yint = yM;
else
    %x = xm:xM;
end

%Dibujo de las rectas:
%plot(x, m1*x+b1, x, m2*x+b2, xint, yint,'rs', x1, y1,'rs', x2,
y2,'rs',px,py,'rs')

%Distanciadel punto "p" al punto más cercano al segmento de recta1:
dist = sqrt((px-xint)^2+(py-yint)^2);

```

B.5. Mutación de un individuo. Archivo MutacionInd.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function individuoMut = MutacionInd(parTpo, tipo, cte, individuo)
%%
%PARÁMETROS Y CONFIGURACIÓN DEL TIEMPO:
%En segundos.
T = parTpo(1);
rate = parTpo(2);%Número par.
At = T/rate;
t = (0:At:T)';
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Configuración del tiempo para la etapa de soporte doble:
indTmed = round((rate+1) * (50/100));
indTfin = rate+1;
t1 = t(1:indTmed);
t2 = t(indTmed:indTfin) - t(indTmed);
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Individuo a mutar:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
indTxB = individuo(1:4);
indTyB = individuo(5:8);
indTzB = individuo(9:12);
indTpsiB = individuo(13);

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Selección de trayectoria a mutar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if tipo == 'c'
    tray_aMut = ceil(rand(1)*4);
else
    tray_aMut = ceil(rand(1)*3);
end
%%
% tamPob = 30;
% tipo = 'c';
if tray_aMut == 1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Trayectoria de la cadera TxB:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Constantes:
    c_vxif_c = cte(1);
    %Inicialización de variables:
    txB = zeros((rate+1), 1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Máximos y mínimos:
    c_xm = 15;
    c_xM = 75;
```

```

c_vxm = -10;
c_vxM = 10;

c_vxifm = 0;
c_vxifM = 35;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Selección de parámetro a mutar:
if tipo == 'c'
    tray_aMut = ceil(rand(1)*3);
else
    tray_aMut = ceil(rand(1)*2);
end

error = 0;
j=0;
while error == 0 && j < 25

    c_x = indTxB(1);
    c_vx = indTxB(2);
    c_vxi = indTxB(3);
    c_vxf = indTxB(4);

    %Mutación de c_x:
    if tray_aMut == 1
        c_x = FlipAlelo(indTxB(1), c_xm, c_xM);
    end

    %Mutación parámetros de velocidad:
    if tray_aMut == 2
        c_vx = FlipAlelo(indTxB(2), c_vxm, c_vxM);
    end
    %Valor de c_vxi y c_vxf de acuerdo al tipo de marcha:
    switch tipo
        case 'i'
            c_vxi = 0;
            c_vxf = c_vxif_c;
        case 'p'
            c_vxi = c_vxif_c;
            c_vxf = 0;
        case 'c'
            if tray_aMut == 3
                c_vxif = FlipAlelo(indTxB(3), c_vxifm, c_vxifM);
                c_vxi = c_vxif;
                c_vxf = -c_vxif;
            end
        end

    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Trazo de la curva con splines cúbicas:
    txB(1:indTmed) = spline3(0, c_x, c_vxi, c_vx, t1);
    txB(indTmed:indTfin) = spline3(c_x, 0, c_vx, c_vxf, t2);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Evaluación de la curva:
    error = max(txB) < c_xM & ~ errorPtsI(txB, 1);%Cero si hay error.
    j = j + 1;
end
if error == 1

```

```

        indTxB = [c_x, c_vx, c_vxi, c_vxf];
    end

    % i = 1;
    %
    % c_x = el_c_x(i);
    % c_vx = el_c_vx(i);
    % c_vxi = el_c_vxi(i);
    % c_vxf = el_c_vxf(i);
    %
    % txB(1:indTmed) = spline3(0, c_x, c_vxi, c_vx, t1);
    % txB(indTmed:indTfin) = spline3(c_x, 0, c_vx, c_vxf, t2);
    % plot(1:21, txB);
    % i = i+1;
end
%%
if tray_aMut == 2
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Trayectoria de la cadera TyB:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Constantes:
    c_vyif_c = cte(2);
    c_y_c = cte(3);
    %Inicialización de variables:
    tyB = zeros((rate+1), 1);
    %Máximos y mínimos:
    yBm = -25;
    yBM = 25;

    c_vym = 0;
    c_vyM = 15;

    c_vyifm = 0;
    c_vyifM = 35;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Selección de parámetro a mutar:
    if tipo == 'c'
        tray_aMut = ceil(rand(1)*3);
    else
        tray_aMut = 2;
    end
end

error = 0;
j=0;
while error == 0 && j < 25
    yB = indTyB(1);
    c_vy = indTyB(2);
    c_vyi = indTyB(3);
    c_vyf = indTyB(4);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Mutación de la velocidad:
    if tray_aMut == 2
        c_vy = FlipAlelo(indTyB(2), c_vym, c_vyM);
    end
end

```



```

%Valor de yB, c_vyi y c_vyf de acuerdo al tipo de marcha:
switch tipo
    case 'i'
        yB = 0;
        %Velocidades:
        c_vyi = 0;
        c_vyf = c_vyif_c;
    case 'p'
        yB = 0;
        %Velocidades:
        c_vyi = c_vyif_c;
        c_vyf = 0;
    case 'c'
        %Mutación de yB y velocidad inicial/final:
        if tray_aMut == 1
            yB = FlipAlelo(indTyB(1), yBm, yBM);
        end
        if tray_aMut == 3
            %Velocidades:
            c_vyif = FlipAlelo(indTyB(3), c_vyifm, c_vyifM);
            c_vyi = c_vyif;
            c_vyf = c_vyif;
        end
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Trazo de la curva con splines cúbicas:
    tyB(1:indTmed) = spline3(yB, yB+c_y_c/2, c_vyi, c_vy, t1);
    tyB(indTmed:indTfin) = spline3(yB+c_y_c/2, yB+c_y_c, c_vy, c_vyf,
    t2);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Evaluación de la curva:
    error = ~ errorPtsI(tyB, 1);%Cero si hay error.
    j = j + 1;
end
if error == 1
    indTyB = [yB, c_vy, c_vyi, c_vyf];
end

% i = 1;
%
% yB = el_yB(i);
% c_vy = el_c_vy(i);
% c_vyi = el_c_vyi(i);
% c_vyf = el_c_vyf(i);
%
% tyB(1:indTmed) = spline3(yB, yB+c_y_c/2, c_vyi, c_vy, t1);
% tyB(indTmed:indTfin) = spline3(yB+c_y_c/2, yB+c_y_c, c_vy, c_vyf,
t2);
% plot(1:21, tyB);
% i = i+1;
end
%%
if tray_aMut == 3
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Trayectoria de la cadera TzB:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Constantes:

```

```

zB_if = cte(4);
pApo_z = cte(5)/2;
pf_z_f = cte(5);
%Inicialización de variables:
tzB = zeros((rate+1), 1);

%Máximos y mínimos:
if pApo_z > 0
    zBm = -35;
    zBM = 10;

    c_zm = -20 + pf_z_f;
    c_zM = 5 + pf_z_f;
elseif pApo_z == 0
    zBm = -35;
    zBM = 10;

    c_zm = -35 + pf_z_f;
    c_zM = 10 + pf_z_f;
else
    zBm = -20;
    zBM = 5;

    c_zm = -35 + pf_z_f;
    c_zM = 10 + pf_z_f;
end

c_vzm = -15;
c_vzM = 15;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Selección de parámetro a mutar:
if tipo == 'c'
    tray_aMut = ceil(rand(1)*3);
else
    tray_aMut = ceil(rand(1)*2) + 1;
end

error = 0;
j=0;
while error == 0 && j < 25
    zBi = indTzB(1);
    zBf = indTzB(2);
    c_z = indTzB(3);
    c_vz = indTzB(4);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Mutación de c_z:
    if tray_aMut == 2
        c_z = FlipAlelo(indTzB(3), c_zm, c_zM);
    end

    if tray_aMut == 3
        c_vz = FlipAlelo(indTzB(4), c_vzm, c_vzM);
    end
end

```

```

%Valor de zB de acuerdo al tipo de marcha:
switch tipo
case 'i'
    zBi = 0;
    zBf = zB_if;
case 'p'
    zBi = zB_if;
    zBf = 0;
case 'c'
    if tray_aMut == 1
        if pApo_z > 0
            zBi = FlipAlelo(indTzB(1), zBm, zBM);
        else
            zBi = FlipAlelo(indTzB(1), zBm, zBM) + pApo_z;
        end
        zBf = zBi + pf_z_f;
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Trazo de la curva con splines cúbicas:
tzB(1:indTmed) = spline3(zBi, c_z, 0, c_vz, t1);
tzB(indTmed:indTfin) = spline3(c_z, zBf, c_vz, 0, t2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Evaluación de la curva:
if zBi < zBf
    error = max(tzB) < c_zM & c_z > zBi & ~ errorPtsI(tzB, 1);
else
    error = max(tzB) < c_zM & min(tzB) >= tzB(size(tzB,1)) & ~
errorPtsI(tzB, 1);
end

    j = j + 1;
    %error = c_z > zB;
end
if error == 1
    indTzB = [zBi, zBf, c_z, c_vz];
end

% i = 1;
%
% zBi = el_zBi(i);
% zBf = el_zBf(i);
% c_z = el_c_z(i);
% c_vz = el_c_vz(i);
%
% tzB(1:indTmed) = spline3(zBi, c_z, 0, c_vz, t1);
% tzB(indTmed:indTfin) = spline3(c_z, zBf, c_vz, 0, t2);
% plot(1:21, tzB);
% i = i+1;
end
%%
if tray_aMut == 4
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Trayectoria de la cadera TpsiB:

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c_pam = 0;
c_paM = 50;
%Generación aleatoria de elementos:
indTpsiB = FlipAlelo(indTpsiB(1), c_pam, c_paM);
end

individuoMut = [indTxB, indTyB, indTzB, indTpsiB];

```

B.5.1 Permutación de un bit de un cromosoma. Archivo FlipAlelo.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%FLIP DE UN ALELO:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function elementoMut = FlipAlelo(elemento, val_min, val_max)
%Valores de prueba:
% val_min = -20;
% val_max = 50;
% elemento = 18;

elementoMut = val_max + 1;
i = 1;
while (elementoMut > val_max || elementoMut < val_min) && i < 10
    %Codificación:
    elemento_ = elemento - val_min;
    elCod = dec2bin(elemento_);

    sizeVal_min = size(dec2bin(val_max),2);
    while(size(elCod,2) < sizeVal_min)
        elCod = strcat('0', elCod);
    end

    %Flip de alelo:
    aleloSel = ceil(rand(1)*size(elCod,2));
    elCod(aleloSel) = dec2bin(~bin2dec(elCod(aleloSel)));

    %Decodificación:
    elemento_ = bin2dec(elCod);
    elementoMut = elemento_ + val_min;

    i = i+1;
end

if (elementoMut > val_max || elementoMut < val_min) && i >= 10
    elementoMut = elemento;
end

```

Bibliografía

- [1] Bardabilla Antonio. La genética de las poblaciones, Noviembre 2011. <http://bioinformatica.uab.es/divulgacio/1a%20genetica%20de%20poblaciones.pdf>.
- [2] The Honda Humanoid Robot ASIMO. History of humanoids, Octubre 2011. <http://world.honda.com/ASIMO/history/history.html>.
- [3] The Honda Humanoid Robot ASIMO. Walking technology, Octubre 2011. http://world.honda.com/ASIMO/technology/walking_02.html.
- [4] P.J. HARLEY ATKINSON, L.V. *Introducción a los métodos numéricos con Pascal*. Addison-Wesley, 1983.
- [5] Díaz Martín Jesús Aznar Acosta Juan. *La célula, unidad de vida*. Instituto de Tecnologías Educativas del Ministerio de Educación, 2011.
- [6] Ben Barrowes. Mathematica symbolic toolbox for matlab-, Noviembre 2011. <http://www.mathworks.com/matlabcentral/fileexchange/6044-mathematica-symbolic-toolbox-for-matlab-version-2-0>.
- [7] P.C. RAIMOND CHAPRA, S.T. *Métodos numéricos para ingenieros*.
- [8] K. Erbaturo and O. Kurt. Natural zmp trajectories for biped robot reference generation. *Industrial Electronics, IEEE Transactions on*, 56(3):835–845, march 2009.
- [9] P. Estévez, J. C. Grieco, and W. Medina-meléndez. Gait synthesis in legged robot locomotion using a cpg-based model.
- [10] Yolis Eugenio. Algoritmos genéticos aplicados a la categorización automática de documentos. Master's thesis, Universidad de Buenos Aires, 2003.
- [11] Dip Goswami. *Biped Locomotion: Stability Analysis, Gait Generation and Control*. PhD thesis, Department of Electrical and Computer Engineering, National University of Singapore, 2009.
- [12] Dip Goswami, Prahlad Vadakkepat, and Phung Duc Kien. Genetic algorithm-based optimal bipedal walking gait synthesis considering tradeoff between stability margin and speed. *Robotica*, pages 355–365, 2009.
- [13] Erico Guizzo. Iran's humanoid robot surena walks, stands on one leg, Septiembre 2011. <http://spectrum.ieee.org/automaton/robotics/humanoids/iran-humanoid-robot-surena-2-walks-stands-on-one-leg>.

- [14] Donald Hearn and Pauline Baker. *Computer Graphic*. Prentice Hall, 1997.
- [15] K Hirai, M Hirose, Y Haikawa, and T Takenaka. The development of honda humanoid robot. *Proceedings 1998 IEEE International Conference on Robotics and Automation Cat No98CH36146*, 2:1321–1326, 1998.
- [16] Masato Hirose and Kenichi Ogawa. Honda humanoid robots development. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):11–19, 2007.
- [17] Hirohisa Hirukawa. Walking biped humanoids that perform manual labor. *Philosophical Transactions of the Royal Society*, pages 65–77, 2006.
- [18] Músculos Cuerpo Humano. Músculos. definición., Noviembre 2011. <http://musculoscuerpohumano.com/definicion-musculos>.
- [19] Waseda University Humanoid. Biped walking robot, Octubre 2011. http://www.humanoid.waseda.ac.jp/booklet/kato_4.html.
- [20] Waseda University Humanoid. Wabot -waseda robot-, Octubre 2011. http://www.humanoid.waseda.ac.jp/booklet/kato_2.html.
- [21] Hironobu Inada and Kazuo Ishii. Bipedal walk using a central pattern generator. *International Congress Series*, 1269:185–188, 2004.
- [22] V.T. Inman, H.J. Ralston, F. Todd, and J.C. Lieberman. *Human walking*. Williams & Wilkins, 1981.
- [23] V. T. Inman y Eberhart H. D. J. B. Saunders. The major determinants in normal and pathological gaits. *JBJS*, 35-A:543–58, 1953.
- [24] Perry Jacquelin. *Gait Analysis: Normal and Pathological Function*. SLACK Incorporated, 1992.
- [25] Sánchez-Lacuesta Javier. *Biomecánica de la marcha humana normal y patológica*. Martín Impresores, 1993.
- [26] Peter Dilworth Jerry Pratt and Gill Pratt. Virtual model control of a bipedal walking robot. *Robotics and Automation*, 1997.
- [27] Seung joon Yi. Proceedings of the twenty-fourth aai conference on artificial intelligence (aaai-10) online learning of uneven terrain for humanoid bipedal walking.
- [28] Hidalgo Pérez José. Distribución equilibrada del esfuerzo de cómputo en algoritmos genéticos paralelos. Master's thesis, Universidad Computense de Madrid, 2006.
- [29] R. Krovetz. *Genetic Programming*. Cambridge.
- [30] Brian Adams y C O'Connor L Vaughan. *Dynamics of human gait*. Kiboho Publishers, 2nd edition, 1992.
- [31] Atsuo Takanashi Laboratory. Biped humanoid robot group. wabian-2r, Octubre 2011. <http://www.takanishi.mech.waseda.ac.jp/top/research/index.htm>.

- [32] MIT Leg Laboratory. 3d one-leg hopper (1983-1984), Octubre 2011. <http://www.ai.mit.edu/projects/leglab/>.
- [33] MIT Leg Laboratory. Planar biped (1985-1990), Octubre 2011. <http://www.ai.mit.edu/projects/leglab/>.
- [34] MIT Leg Laboratory. Planar one-leg hopper (1980-1982), Octubre 2011. <http://www.ai.mit.edu/projects/leglab/>.
- [35] Takanashi Laboratory. Wl (waseda leg) series, Octubre 2011. http://www.takanashi.mech.waseda.ac.jp/top/research/wabian/previous_research/previous_research.htm.
- [36] Hun-Ok Lim and Atsuo Takanashi. Biped walking robots created at waseda university: Wl and wabian family. *Philosophical Transactions of the Royal Society*, pages 49–64, 2006.
- [37] Gestal Pose Marcos. Introducción a los algoritmos genéticos, Noviembre 2011. <http://sabia.tic.udc.es/mgestal/cv/AAGGtutorial/TutorialAlgoritmosGeneticos.pdf>.
- [38] O. Michel. Webotstm: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1:39–42, 2004.
- [39] Luis Diego Murillo Soto. Simulación de un sistema de manufactura flexible con redes de petri coloreadas. *Tecnología en Marcha*, 23:47–62, Enero-Marzo 2010.
- [40] M P MURRAY, A B DROUGHT, and R C KORY. Walking patterns of normal men. *The Journal of Bone and Joint Surgery*, 46(2):335–360, 1964.
- [41] AHRA Network based humanoid MAHRU. Mahru and ahra, Septiembre 2011. <http://humanoid.kist.re.kr/eng/mahruahra>.
- [42] Joon Ng. An anthropomorphic bipedal robo. Master's thesis, The University of Western Australia, 1998.
- [43] Narváez Aroche Octavio. Modelo cinemático y dinámico de un robot bípedo de doce grados de libertad internos. Master's thesis, Universidad Nacional Autónoma de México, 2010.
- [44] Narváez Aroche Octavio. Caminata de un robot bípedo, Octubre 2011. www.youtube.com/watch?v=mpRDsCrwvkk&feature=channel_video_title.
- [45] Arbulú Saavedra M. R. *Stable locomotion of humanoid robots based on mass concentrated model*. PhD thesis, Universidad Carlos III Madrid, 2008.
- [46] Marc H. Raibert. Legged robots. *Communications of the ACM*, 29:499–514, June 1986.
- [47] RoboCup. About robocup, Septiembre 2011. <http://www.robocup.org/about-robocup/>.
- [48] PAL Robotics. Reem-a, Septiembre 2011. <http://www.pal-robotics.com/robots/reem-a>.
- [49] Piedad Tolmos Rodríguez-Piñero. Introducción a los algoritmos genéticos y sus aplicaciones, Noviembre 2011. <http://www.uv.es/asepuma/X/J24C.pdf>.
- [50] Fernández Iglesias Samuel. Locomoción bípeda del robot humanoide nao. Master's thesis, Universidad Politécnica de Cataluña, 2009.

- [51] Philippe Sardain, Mostafa Rostami, and Guy Bessonnet. An anthropomorphic biped robot: dynamic concepts and technological design. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 28(6):823–838, 1998.
- [52] S. Schaal. the new robotics - towards human-centered machines. (2):115–126, 2007.
- [53] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, Berlin, Heidelberg, 2008.
- [54] G Taga, Y Yamaguchi, and H Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65(3):147–159, 1991.
- [55] TeensHealth. Huesos, músculos y articulaciones, Noviembre 2011. http://kidshealth.org/teen/en_espanol/cuerpo/bones_muscles_joints_esp.html.
- [56] Toyota. Partner robot family, Octubre 2011. http://www.toyota-global.com/innovation/partner_robot.
- [57] Yao-Yang Tsai, Wen-Chieh Lin, K.B. Cheng, Jehee Lee, and Tong-Yee Lee. Real-time physics-based 3d biped character animation using an inverted pendulum model. *Visualization and Computer Graphics, IEEE Transactions on*, 16(2):325–337, march-april 2010.
- [58] Clavell S Viladot R, Cohi O. *Órtesis y prótesis del aparato locomotor*. Mason S.A., 1991.
- [59] Branislav Vukobratovic, Mimir y Borovac. Zero-moment point - thirty five years of its life. *I. J. Humanoid Robotics*, 1(1):157–173, 2004.
- [60] Susana Collado Vázquez. *Análisis de la marcha humana con plataformas dinamométricas. Influencia del transporte de carga*. PhD thesis, Universidad Computense de Madrid, 2002.
- [61] Grizzle Jessy Westervelt Eric. *Feedback Control of Dynamic Bipedal Locomoción*. Taylor & Francis Group, LLC, 2007.
- [62] Natalia I. Wiereszen. *Análisis de la actividad muscular en posición bípeda y durante la marcha*. PhD thesis, Universidad del País Vasco, 2005.
- [63] Morales Eduardo y Sucar Succar Enrique. Los robots del futuro y su importancia para México. *Komputer Sapiens*, 2:7–12, 2009.
- [64] Kajita Shuuji y Tani Kazuo. Study of dynamic locomotion on rough terrain. *Robotics and Automation*, pages 1405–1411, 1991.
- [65] Larriva Vásquez José y Vele Guallpa Oscar. Estudio, diseño y construcción de un robot bípedo experimental. Master's thesis, Universidad Politécnica Salesiana, 2006.