



INSTITUTO TECNOLÓGICO
de la laguna



DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“Estudio de movimientos deportivos en humanoides”

POR

Ing. Jesús Eduardo Fierro Proa.

TESIS

**PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL
GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA**

DIRECTOR DE TESIS

Dr. José Alfonso Pámanes García

ISSN: 0188-9060



RIITEC: (19)-TMCIE-2013

Torreón, Coahuila, México,

Diciembre 2013

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
Instituto Tecnológico de la Laguna

"2013, Año de la Lealtad Institucional y Centenario del Ejército Mexicano"

Torreón, Coah., **16/Diciembre/2013**

Dependencia: D:PI

Oficio: DEPI/387/2013

Asunto: Autorización de
Impresión de tesis.

C. JESUS EDUARDO FIERRO PROA
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA.
PRESENTE

Después de haber sometido a revisión su trabajo de tesis titulado:

"Estudios de Movimientos Deportivos en Humanoides"

Habida cuenta

DR. JOSE LUIS MEZA MEDINA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN


Por medio de la presente, hacemos de su conocimiento que después de haber sometido a revisión el trabajo de tesis titulado:


"Estudios de Movimientos Deportivos en Humanoides"


Desarrollado por el **C. JESÚS EDUARDO FIERRO PROA**, con número de control **M06130150** y habiendo cumplido con todas las correcciones que se le indicaron, estamos de acuerdo que se le conceda la autorización de la fecha de examen de grado para que proceda a la impresión de la misma.

ATENTAMENTE
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN


Dr. José A. Pámanes García
Asesor/Director


Dr. Ricardo E. Campa Cocóm
Comité Tutorial


Dr. Enrique Guán Durón
Comité Tutorial


Dr. Víctor A. Santibañez D.
Comité Tutorial



Para mi jefe.

"Realidad no hay más que una"

Haruki Murakami

Estudio de movimientos deportivos en humanoides

Resumen

En este trabajo de tesis se efectúa un estudio de la dinámica del robot humanoide *HRP-4* durante la ejecución de una patada a un balón de fútbol soccer en reposo. Para la realización del estudio aquí presentado se desarrolló una plataforma de software basada en el uso del paquete *OpenHRP-3*, que incluye un programa para la generación de las trayectorias articulares del robot a partir de su modelo inverso de posición, un modelo virtual del robot *HRP-4*, un programa controlador del modelo virtual, y un programa para la generación de las rutas del Punto de Momento Cero en las plantas de los sucesivos pies de apoyo. En esta investigación se propone una serie de movimientos del robot para la ejecución de su tarea, inspirados en los movimientos que realiza un jugador humano de fútbol al ejecutar un *penalty*. El patrón de movimientos propuestos se caracteriza por la incorporación de energía cinética suplementaria al robot antes y durante la ejecución de la patada, con lo que dicho patrón podría superar en eficacia a los que han sido propuestos en la literatura científica al día de hoy. Las simulaciones efectuadas en la plataforma desarrollada permiten probar que el robot se mantiene en equilibrio al realizar los movimientos propuestos.

Study of sports movements in humanoid

Abstract

This thesis work presents a study of the dynamics of the humanoid robot *HRP-4* while performing the task of kicking a soccer ball at rest. To accomplish this study, a software platform was developed based on the package *OpenHRP-3*. Our platform includes a program for the generation of the robot's joint trajectories based on the inverse kinematics equations, a virtual model of the robot *HRP-4*, a program for controlling the virtual model, and a program for the generation of the Zero Moment Point (ZMP) paths inspired in motions of a human player during the accomplishment of the same task, a set of robot motions was proposed in our research. Supplementary kinetic energy is supplied to the robot before and during the kicking by using the proposed motion. The performance obtained using this approach is better than other previously published. Simulations carried out in our platform allow to prove that the robot is kept in balance during the task.

Agradecimientos

Viendo hacia atrás, uno se da cuenta de todo lo que ha cambiado, las experiencias, alegrías, desabores, viejas y nuevas cosas, todo eso forma parte de lo que hace especial cada momento de nuestras vidas, y así como todo, hay un momento para cada cosa, y ahora es tiempo de recordar a todos aquellos que dieron su aportación para la realización de este proyecto.

Primero que nada quisiera expresar mi agradecimiento hacia el Dr. José Alfonso Pámanes por su continuo soporte durante la realización de este proyecto.

En segundo lugar, quisiera agradecer a todo el personal docente del área de posgrado del Instituto Tecnológico de la Laguna por sus aportaciones particulares al tiempo invertido dentro del Instituto.

Durante el desarrollo de cualquier proyecto siempre hay una meta y un objetivo, pero en el camino siempre nos encontramos con nuevas amistades, y esto por sí mismo hace cualquier viaje un éxito, así un agradecimiento especial a todos los camaradas, si ustedes ya saben quiénes son.

Y en especial quiero darte las gracias a ti, ya que sin un lector el presente trabajo no tendría sentido.

Expreso mi reconocimiento al CONACyT por su apoyo económico para el desarrollo de la presente investigación durante los estudios de maestría, así como a la Dirección General de Educación Superior Tecnológica de la S.E.P.

Índice del contenido

Resumen	i
Agradecimientos	ii
Índice del contenido.....	iii
Índice de tablas.....	v
Índice de figuras.....	vi
Abreviaturas y acrónimos.....	viii
1. Introducción.....	1
2. Modelado cinemático del robot HRP-4.....	7
2.1 Descripción del robot humanoide HRP-4.....	7
2.2 Modelado cinemático.....	12
2.3 Problemas directos de velocidad y aceleración.....	23
3. Modelado geométrico del robot HRP-4 en la plataforma <i>OpenHRP3</i>	24
3.1 GrxUI.....	25
3.2 Desarrollo de un modelo virtual.....	26
3.3 Programa controlador del robot virtual.....	41
3.4 Configuración en GrxUI para el manejo de datos.....	45
4. Planificación de movimientos del robot humanoide.....	48
4.1 Biomecánica de un futbolista humano.....	48
4.2 Generación de trayectorias.....	52
5. Generación de las trayectorias articulares y simulación.....	61
5.1 Obtención de trayectorias articulares en Matlab.....	61
5.2 Generación de trayectorias.....	62
5.3 Control del simulador dinámico del paquete <i>OpenHRP3</i>	66
6. Análisis del equilibrio del robot.....	70
6.1 Punto de Momento Cero (ZMP).....	70
6.2 Análisis del Punto de Momento Cero.....	72
6.2.1 Generación del punto de momento cero (ZMP). Programa <i>genzmp</i>	72

6.2.2 Comportamiento del Punto de Momento Cero	74
7. Conclusión	76
REFERENCIAS	79
Apéndice A	83
Apéndice B	108
Apéndice C	111

Índice de tablas

Tabla 2.1 Principales características del robot humanoide HRP-4.....	11
Tabla 2.2 Parámetros Denavit Hartenberg modificados de las piernas del robot HRP-4.....	14
Tabla 2.3 Parámetros Denavit-Hartenberg modificados de los brazos del robot HRP-4.....	15
Tabla 2.4 Valores de los parámetros geométricos de las piernas.....	15
Tabla 2.5 Valores de los parámetros geométricos de los brazos.....	15
Tabla 3.1 Datos de las piezas usadas para la creación del modelo virtual del robot humanoide HRP-4 ...	28
Tabla 4.1 Especificación de parámetros de los movimientos propuestos del humanoide.....	52

Índice de figuras

Figura 1.1 Ejemplos de robots humanoides.....	2
Figura 1.2 Ejemplos de robots humanocides educativos	3
Figura 2.1 Humanoide HRP 4.....	7
Figura 2.2 Modelos P3 y HRP-1 del robot humanoide de Honda.....	9
Figura 2.3 Humanoides serie HRP	9
Figura 2.4 Esquema cinemático del robot HRP-4.....	12
Figura 3.1 OpenHRP3 interfaz principal.....	25
Figura 3.2 Interfaz GrxUI-Eclipse.....	26
Figura 3.3 HRP-4.....	27
Figura 3.4 Creación de un modelo virtual de un robot.....	31
Figura 3.5 Incorporación de un segmento al nuevo modelo en proceso de creación.....	31
Figura 3.6 Incorporación de una nueva pieza al segmento actual.....	32
Figura 3.7 Pieza (cadera) creada.....	32
Figura 3.8 Propiedades físicas del segmento agregado (cadera).....	33
Figura 3.9 Traslación del segmento.....	34
Figura 3.10 Incorporación de un nuevo sensor.....	34
Figura 3.11 Propiedades del sensor añadido.....	35
Figura 3.12 Añadir eslabón.....	36
Figura 3.13 Proceso de integración de un modelo virtual del humanoide HRP-4 en el sistema Open HRP3.....	37
Figura 3.14 Progreso de armado de modelo.....	38
Figura 3.15 Tipo de articulaciones.....	38
Figura 3.16 Identificación de una articulación.....	39
Figura 3.17 Modelado del piso como entorno del robot	40
Figura 3.18 Visualización del piso.....	40
Figura 3.19 Worldstate creado.....	40
Figura 3.20 Configuración de la interacción entre objetos.....	41
Figura 3.21 Creación de proyecto en RTCBuilder.....	42
Figura 3.22 Pestaña Basic.....	43
Figura 3.23 Pestaña Activity.....	43
Figura 3.24 Pestaña Data Ports.....	44
Figura 3.25 Pestaña Language and environment.....	44
Figura 3.26 Generación del código.....	45
Figura 3.27 Pestaña Controller.....	46
Figura 3.28 Selección de controlador.....	46
Figura 3.29 Configuración del puente.....	47
Figura 4.1 Secuencia de posturas de un jugador humano al ejecutar un tiro de castigo.....	49

Figura 4.2 Postura inicial.....	53
Figura 4.3 Postura del robot al finalizar la Fase 1.....	54
Figura 4.4 Postura del robot al finalizar la Fase 2.....	54
Figura 4.5 Postura del robot al terminar la Fase 3.....	55
Figura 4.6 Postura del robot al finalizar la Fase 4.....	56
Figura 4.7 Postura del robot al terminar la Fase 5.....	56
Figura 4.8 Postura del robot al finalizar la Fase 6.....	57
Figura 4.9 Postura del robot al finalizar la Fase 7.....	57
Figura 4.10 Postura del robot al concluir la Fase 8.....	58
Figura 4.11 Fase 9.....	59
Figura 4.12 Rutas de los pies y pelvis en las diferentes fases.....	60
Figura 5.1 Diagrama de flujo del programa para la generación de las trayectorias articulares.....	63
Figura 5.2 Trayectorias articulares correspondientes al movimiento deseado del robot.....	65
Figura 5.3 Estructura de funcionamiento del controlador usado para la simulación dinámica.....	67
Figura 5.4 Secuencia de posturas del modo o virtual del robot obtenidas durante la simulación de los movimientos propuestos.....	69
Figura 6.1 Cargas actuando sobre el pie de apoyo de un bípedo en fase de soporte simple.....	71
Figura 6.2 Estructura de operación del programa genzmp.....	72
Figura 6.3 Sucesivas huellas de los pies de apoyo y rutas del ZMP durante los movimientos del robot. Vista superior. (P.A: pie apoyado).....	74
Figura B.1 Consignas articulares de posición pierna derecha; eje x posición (°) eje y tiempo (s).....	109
Figura B.2 Consignas articulares de posición pierna izquierda; eje x posición (°) eje y tiempo (s).....	109
Figura B.3 Consignas articulares de posición brazos; eje x posición (°) eje y tiempo (s).....	110
Figura B.4 Consignas articulares de posición torso; eje x posición (°) eje y tiempo (s).....	110
Figura C.1 Visualización del ZMP en la planta del pie durante el paso 1.....	112
Figura C.2 Visualización del ZMP en la planta del pie durante el paso 2.....	112
Figura C.3 Visualización del ZMP en la planta del pie durante el paso 3.....	113
Figura C.4 Visualización del ZMP en la planta del pie durante el paso 4.....	113
Figura C.5 Visualización del ZMP en la planta del pie durante el paso 5.....	114
Figura C.6 Visualización del ZMP en la planta del pie durante el paso 6.....	114
Figura C.7 Visualización del ZMP en la planta del pie durante toda la simulación.....	115

Abreviaturas y acrónimos

F_A	Vector de fuerzas en A
M_A	Momentos en A
M_z	Momento de reacción vertical del suelo
m_s	Masa del pie
x_{lf_ini}	Posición inicial del pie izquierdo con respecto al eje x
x_{ptv_ini}	Posición en el eje x_0 de la pelvis
x_{rf_ini}	Posición inicial del pie derecho con respecto al eje x
z_{ptv_ini}	Posición en el eje z de la pelvis
$\delta_{x_{lf}}$	Desplazamiento en el eje x del pie izquierdo
$\delta_{x_{ptv}}$	Desplazamiento en el eje x de la pelvis
$\delta_{x_{rf}}$	Desplazamiento en el eje x del pie derecho
$\delta_{z_{ptv}}$	Desplazamiento en el eje z de la pelvis
δ_t	Incremento del tiempo
δ_μ	Desplazamiento del torso
μ_{ini}	Posición inicial del torso
AIST	Advanced Industrial Science and Technology
d.o.f	Degree of freedom
g.d.l	Grados de libertad

GrxUI	Graphic user interface
HRP	Human-centered Robotics Platform
m.u.a.	Movimiento uniformemente acelerado
METI	Ministry of Economy, Trade and Industry
NEDO	New Energy and Industrial Technology Development Organization
P.A	Pie apoyado
RT	Robot Technology
SYMORO	SYmbolic MOdeling of RObots
ZMP	Punto de momento cero (Zero Moment Point)
G	Centro de la masa del pie
R	Fuerzas de reacción del suelo
$\delta\theta_{femur}$	Desplazamiento angular del fémur
$\delta\theta_{tibia}$	Desplazamiento angular de la tibia.

1. Introducción

La International Federation of Robotics (IFR) [1], ha adoptado recientemente las siguientes definiciones normalizadas (norma ISO 8373: 2012 *Robots and robotic devices - Vocabulary*) relacionadas con los robots:

- *Un robot es un mecanismo actuado programable en dos o más eje,s con un grado de autonomía, moviéndose en su entorno para efectuar tareas intencionales. La autonomía en este contexto significa la capacidad de efectuar tareas intencionales basadas en el estado actual y en la percepción (del entorno), sin intervención humana.*
- *Un robot de servicio es un robot que efectúa tareas útiles para humanos o equipamientos, excluyendo las aplicaciones de automatización industrial.*
- *Un robot personal o un robot para uso personal es un robot de servicio usado para tareas no comerciales, usualmente por personas necesitadas (ancianos, enfermos, minusválidos, etc).*

Teniendo en cuenta las anteriores definiciones de la IFR, se puede definir un robot humanoide como un robot de servicio cuya arquitectura emula la estructura ósea del ser humano. Es decir, que tiene dos piernas y pies, cadera, torso, brazos y cabeza, aunque algunos modelos sólo poseen ciertos elementos, como por ejemplo sólo las piernas, incluso la cara de algunos robots es diseñada con la intención de emular algunas expresiones de los humanos.

Los robots humanoides han sido desarrollados principalmente para auxiliar al ser humano en diversas tareas en los mismos ambientes en que éste se desenvuelve. La ayuda al ser humano podría recuerrirse, por ejemplo, en trabajos peligrosos o tediosos, o en el área de la salud, atendiendo personas enfermas, ancianas o discapacitadas, o incluso se prevé su uso, entre otras actividades, como recepcionistas, músicos, o deportistas.



a) ASIMO



b) Valkyrie

Figura 1.1 Ejemplos de robots humanoides

Entre los robots humanoides que han sido desarrollados se pueden mencionar el robot bípedo humanoide *ASIMO* desarrollado por *HONDA*, con la capacidad de correr; y el robot humanoide *Valkyrie* desarrollado para la exploración espacial por parte de la *NASA* (Figura 1.1)

Existen algunas versiones de robots humanoides para usarse como prototipos para investigación, por ejemplo las series *Nao*, desarrolladas por *Aldebaran Robotics* o las series *Bioloid* desarrolladas por *Robotis*, y las series *Choromet* desarrolladas por la *Advanced Industrial Science and Technology (AIST)* (Figura 1.2).

Dentro de las áreas de estudio más investigadas respecto a robots humanoides se tiene el tema de la generación de patrones de marcha o caminado, bajo diversos criterios, por ejemplo [2], [3]. En el *Instituto Tecnológico de la Laguna* se han desarrollado algunos trabajos relacionados con los patrones de marcha, como lo son [4], [5]. En contraste con esos estudios precursores, en el presente trabajo de tesis, se tiene interés en el análisis de los movimientos de un robot humanoide para efectuar el cobro de un tiro de castigo en futbol soccer.

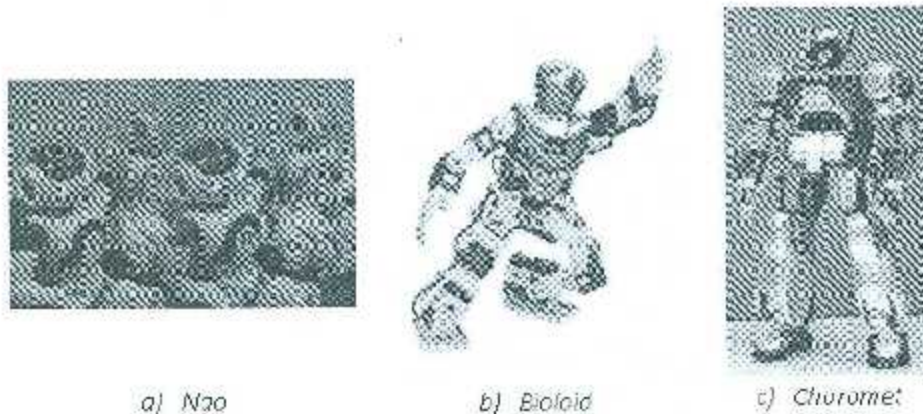


Figura 1.2 Ejemplos de robots humanoides educativos

En relación al estudio de las acciones deportivas de humanoides, en [6] se trató el tema del impacto que soporta un robot al golpear una pelota de minigolf, con el objetivo de capturar datos relativos a la mecánica del golpeo de la pelota (velocidad de golpeo y ángulo de salida de la pelota).

Entre los trabajos previos en el tema de humanoides futbolistas se tiene el presentado en [7], en el cual se estudian las características de un pequeño robot humanoide, llamado *Toni*, creado para participar en la *Robocup*; este trabajo se refiere a la determinación de la posición del balón a partir de los sensores del robot, y al cálculo de las trayectorias articulares del robot para cumplir con la tarea de pateo, controlando los movimientos del pie libre, la rodilla y la cadera en el plano sagital; en este trabajo no se considera la realización de una carrera del robot previa al pateo, que le permita acumular energía cinética. En [8] se trata el comportamiento de humanoides en equipo y el manejo de las trayectorias de éstos en partidos de fútbol de la *Robocup*, en especial se abordó el desarrollo de la plataforma del robot humanoide *Bruno*, el cual fue diseñado para privilegiar la robustez y versatilidad del mismo. La estrategia de pateo que utiliza *Bruno* consiste en la selección de un movimiento de la pierna oscilante, entre un conjunto de movimientos precisados, que mejor se adapte a la situación específica que experimente el robot.

En [9], por otra parte, se propone una solución para obtener una patada efectiva utilizando control adaptativo; la estrategia de control se divide en cuatro etapas: preparación, retracción de la pierna oscilante, ejecución de la patada y conclusión. Los autores diseñan un controlador que se ajusta en cada etapa del pateo con base en la posición actual del balón y el espacio de trabajo disponible, recorriendo la menor distancia posible para cumplir con el objetivo. En otro trabajo [10] se estudia el desarrollo de una patada de un robot humanoide NAO, con base en la estrategia aplicada en [8]; se trata de una patada única, planificada a partir de la percepción del entorno, determinando las consignas a través de la cinemática inversa. A diferencia del enfoque de [8], en [10] se propone el uso de curvas de Bézier para la generación de las trayectorias de la pierna oscilante. Un trabajo similar al de [10] es el observado en [11], el cual aplica el modelo dinámico del humanoide para determinar los movimientos para el pateo de un balón; en este trabajo la planificación de trayectorias utiliza curvas de Bézier y la estabilización del robot se consigue mediante el balanceo del centro de masa por realimentación de un controlador PID.

Los trabajos arriba mencionados están enfocados principalmente en la síntesis de trayectorias óptimas para el pateado, a través de diversos métodos. La principal limitante que podemos observar en estos trabajos consiste en considerar de manera desacopada la carrera del robot y la ejecución de la patada o bien en la realización de la patada sin efectuar una carrera previa. Estas características impiden el máximo aprovechamiento de la energía cinética acumulada del robot para suministrar energía al balón en el momento del impacto. En el marco de esta tesis se propone un enfoque original para la preparación y ejecución del golpeo del balón con el pie, sin desestabilizar al robot, aprovechando la energía cinética acumulada del robot, tanto en la etapa de la carrera como en la del pateado. El objetivo de esta tesis, es el de proponer una secuencia de movimientos estables de un robot humanoide para la realización del pateado de un balón de fútbol soccer en la ejecución de un tiro de castigo, modelar su comportamiento cinemático y dinámico, y verificar la estabilidad mediante simulación.

El enfoque del estudio se basa en la simulación de los movimientos propuestos del humanoide en un mundo virtual aplicando diversas herramientas de software. En este primer trabajo sobre el tema se limitará el estudio de los movimientos del robot a los desplazamientos en el plano sagital.

El presente documento se organiza de la siguiente manera: en el Capítulo 2 se describe el robot humanoide que se seleccionó para este estudio. De la familia de robots humanoides *HRP*, en particular se utilizará el robot humanoide *HRP-4*. Se proporcionan algunos antecedentes de este robot y se describen sus características principales: su arquitectura, las propiedades inerciales de sus eslabones, así como el software utilizado para controlarlo. En el mismo Capítulo 2 se desarrolla el modelo cinemático de posición del humanoide *HRP-4*, describiendo su arquitectura mediante la metodología de *Denavit-Hartenberg* modificada. En el Capítulo 3 se trata la principal herramienta computacional que se utilizó para la simulación dinámica de la acción deportiva estudiada del humanoide; se trata del paquete *OpenHRP3*, del cual se presentan los elementos que lo conforman, así como el funcionamiento de ellos; además se describe el procedimiento para armar un modelo virtual del robot humanoide *HRP-4*, y para la creación de la base del programa controlador del modelo virtual que permite efectuar la simulación. En el Capítulo 4 se planifican los movimientos deseados del humanoide a partir de los movimientos observados de un jugador humano cobrando un tiro de castigo. El Capítulo 5 se centra en la generación de las trayectorias articulares necesarias para la simulación del modelo virtual, y en el desarrollo e integración del software requerido para ese propósito; los programas son creados en *Matlab* para la generación de trayectorias y en *C++* para la edición del programa controlador de la simulación dinámica por parte de *OpenHRP3*; además se da una explicación general acerca de la comunicación entre todos los programas utilizados. En el Capítulo 6 se aborda el estudio del comportamiento del punto de momento cero (*ZMP*) durante los movimientos del humanoide; este estudio se efectúa para validar la estabilidad dinámica del humanoide; adicionalmente se explica el funcionamiento del programa encargado de los cálculos correspondientes. En el Capítulo 7, finalmente, se

recapitulan los resultados obtenidos en este trabajo, se destacan las principales contribuciones del mismo y se propone la realización de posibles trabajos futuros que permitirían avanzar en la investigación de los movimientos óptimos del humanoide HRP-4.

2. Modelado cinemático del robot HRP-4

El robot humanoide seleccionado para nuestro estudio es el *HRP-4*, desarrollado por *Kawada Industries*, debido a sus 34 grados de libertad (*g.d.l.*), de los cuales se tienen 2 *g.d.l.* en el torso, que son esenciales para obtener una movilidad muy parecida al del ser humano.

2.1 Descripción del robot humanoide *HRP-4*

El robot humanoide *HRP-4* es una "Plataforma para la investigación"

El *HRP-4* es la sucesión de los modelos previos de la misma industria; se trata de un diseño optimizado de los modelos *HRP-2 "Promet"* y *HRP-3 "Promet MK-II"*, en el marco del proyecto industrial-académico *HRP-4c*, que tiene como principal objetivo la coexistencia con los humanos, y el reemplazo de éstos en algunas de sus tareas típicas.

Originalmente los proyectos de desarrollo de los robots humanoides *HRP* surgen de la necesidad social de contar con robots que puedan trabajar en un ambiente humano, es decir, en hogares, áreas de trabajo o talleres, en algunas tareas tediosas. Se pensó en robots humanoides teniendo en consideración que dichos ambientes están diseñados para ser habitados por seres humanos; así, son los robots los que se adaptan al entorno humano y no al revés. Humanoides con estas características podrían producirse en forma masiva y crear un mercado para éstos. Tratando de cumplir esta meta se fundó el proyecto por el *Ministry of Economy, Trade and Industry (METI)*, y la *New Energy and Industrial Technology Development Organization (NEDO)*, siendo liderados por *Kawada Industries, Inc* con el apoyo del *National Institute of Advanced Industrial Science and Technology (AIST)* de Japón. El proyecto se inició con la adquisición de los robots *Honda P3* (Figura 2.2), comprados a la compañía *Honda*, a partir de los cuales se desarrolló el robot *HRP-1*. Por otra parte, desde marzo de 2003 *Kawada Industries, General Robotix, Inc.* y *AIST* han distribuido el robot *HRP-2* en diversos centros de investigación y desarrollo (Figura 2.3).

El robot *HRP-4* es un robot humanoide, ligero, compacto, de bajo costo y con una amplia versatilidad para la manipulación de objetos con los brazos; su diseño se caracteriza por la incorporación de diversos elementos de seguridad para el trabajo en un ambiente humano; por ejemplo el uso de motores de baja potencia.

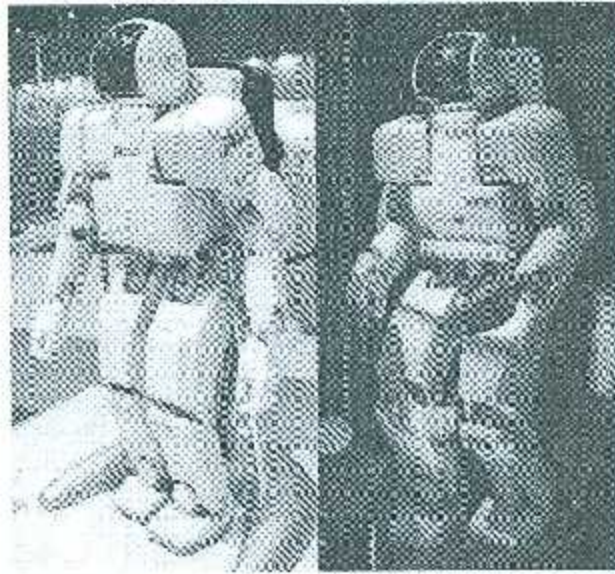


Figura 2.2 Modelos P3 y HRP-1 del robot humanoide de Honda.



Figura 2.3 Humanoides serie HRP

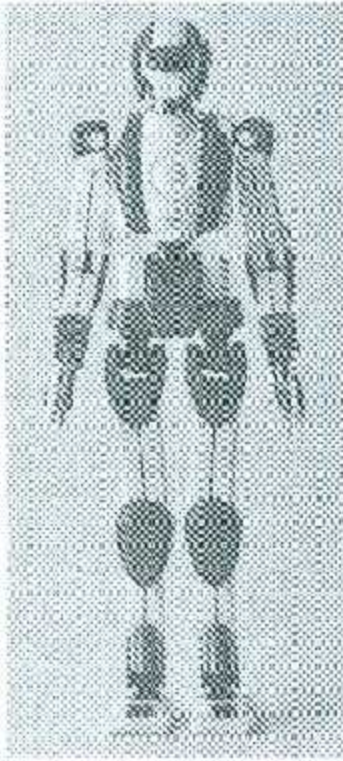
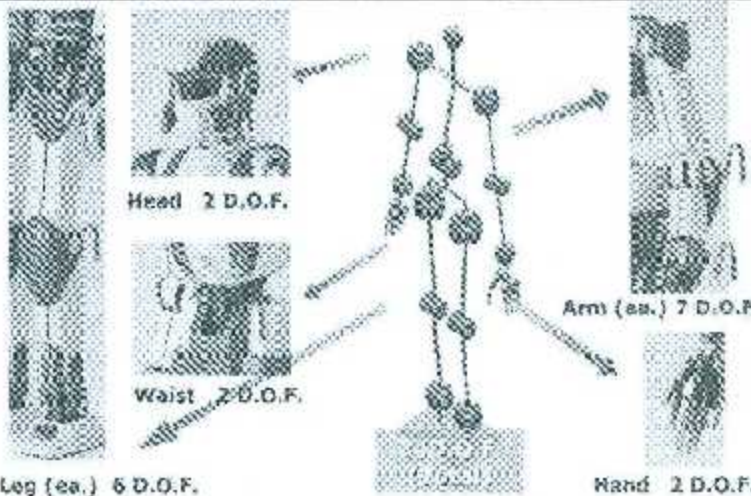
Entre las principales características del humanoide HRP-4 destacan las siguientes:

- 34 grados de libertad (g.d.l.), de los cuales se tienen 7 en cada brazo.
- Altura total de 151 cm y un peso de 39 kg (El modelo HRP-2 tiene 30 grados de libertad, con una altura de 154 cm y un peso de 58 kg)
- Motores de 80 W de potencia o menos en cada articulación.
- Capacidad de carga de cada brazo: 0.5 kg
- Bajo costo como resultado de la optimización de los sistemas montados, y del uso de partes simples y comunes en el diseño de la estructura.
- Capacidad para portar una pequeña PC portátil en la espalda para incrementar de manera opcional la capacidad de procesamiento de información.
- Afinidad física con el ser humano debido al concepto del diseño "*slim athlete*".

Una síntesis de las características de este robot se presenta en la Tabla 2.1 [12].

Para el control, el HRP-4 hace uso del sistema *OpenRTM-aist*, que es una plataforma de software estandarizada para el desarrollo de sistemas robóticos a través de componentes. Dos características importantes de este sistema se refieren, por un lado, a la facilidad de comunicación e intercambio de información entre los mismos componentes del robot así como con el exterior; y, por otro lado, a la facilidad de trabajar en sistemas operativos tales como Windows o Linux. Los programas se pueden desarrollar en C++, Python, Java o prácticamente en cualquier lenguaje de programación que desee. La comunicación entre componentes de *OpenRTM-aist* está basada en los protocolos de información RT (*Robot Technology*).

Tabla 2.1 Principales características del robot humanoide IIRP-4

	Altura	151 cm
	Peso	39 kg.
	Estructura	Cubierta del robot: plástica; esqueleto de aleación de aluminio
	<i>g.d.l</i>	34 <i>g.d.l.</i> (6 en c/ pierna, 2 en cuello, 2 en torso, 7 en c/ brazo y 2 en c/ mano)
		

El *AIST* es el instituto encargado de la realización del sistema de control de movimiento del robot. Esta organización desarrolló el software para el control en tiempo real utilizando procesadores de múltiples núcleos y aplicando el sistema operativo *Linux*, y *OS RT-Preempt*. *RT-Preempt* es un kernel de *Linux* que permite el procesamiento en tiempo real de este sistema operativo. Un kernel es un programa encargado de procesar las entradas y salidas de datos solicitadas por algún software y traduce estas peticiones en datos que puede leer la unidad central de procesamiento (CPU) de una computadora.

2.2 Modelado cinemático

El esquema cinemático del robot humanoide HRP-4 se muestra en la Figura 2.4. La descripción de su cadena cinemática se realiza aplicando la convención de *Denavit-Hartenberg* modificada [13]. Esta metodología parte de la asignación de un marco de referencia ortonormal, Σ_j , a cada eslabón C_j de la cadena cinemática ($j = 0, 1, \dots, n$), siendo C_0 el eslabón fijo y C_n el órgano terminal. Si se considera que el eslabón C_{j-1} se conecta con el C_j mediante la articulación a_j , entonces los ejes del marco Σ_j se definen de acuerdo a las siguientes reglas, enunciadas en los siguientes párrafos:

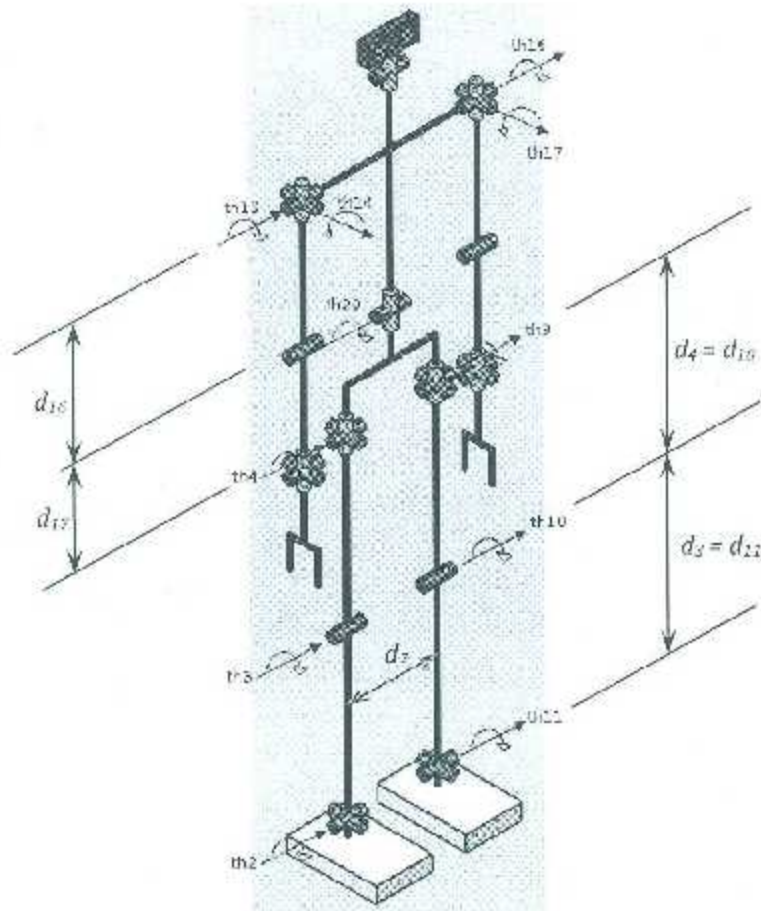


Figura 2.4 Esquema cinemático del robot HRP-4

- El eje z_j se define a lo largo del eje de la articulación α_j . Este eje es el de rotación en el caso de una articulación tipo R (rotacional), o el de deslizamiento en el caso de una articulación tipo P (prismática). En este último caso la posición de z_j se fija arbitrariamente.
- El eje x_j se define a lo largo de la perpendicular común a z_j y a z_{j-1} . Si estos dos ejes son paralelos, x_j no se puede definir de manera única; en tal caso se recomienda seguir un criterio de simetría o de simplicidad para definir la ubicación de este eje. Por supuesto, el punto de intersección de x_j y z_j define el origen o_j del marco.
- El eje y_j se define a partir de los ejes x_j y z_j , de tal manera que se complete un marco de mano derecha.
-

Una vez asignado un marco a cada eslabón, se definen los parámetros geométricos que especifican la posición y orientación de cada marco con respecto al precedente, que son conocidos como parámetros Denavit-Hartenberg modificados (o PDHM):

- α_j Es el ángulo de z_{j-1} a z_j , medido respecto a x_{j-1} conforme a la regla de la mano derecha.
- d_j Es la distancia entre z_{j-1} y z_j , a lo largo de x_{j-1} .
- θ_j Es el ángulo de x_{j-1} a x_j , medido con respecto a z_j , conforme a la regla de la mano derecha.
- r_j Es la distancia entre x_{j-1} y x_j a lo largo de z_j .

Adicionalmente se define el parámetro auxiliar σ_j , el cual toma el valor de 0 si la articulación j es de tipo R, y de 1 si la articulación es de tipo P

Habiendo definidos los PDHM, se pueden conformar las matrices de transformación elementales del robot, las cuales describen la posición y orientación del marco Σ_j con respecto al Σ_{j-1} . Dicha matriz está dada por:

$${}_{j-1}T_j = \begin{bmatrix} c\theta_j & -s\theta_j & 0 & d_j \\ c\alpha_j s\theta_j & c\alpha_j c\theta_j & -s\alpha_j & -r_j s\alpha_j \\ s\alpha_j s\theta_j & s\alpha_j c\theta_j & c\alpha_j & r_j c\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.1)$$

donde c representa la función coseno y s la función seno, es decir $c\alpha_j = \cos(\alpha_j)$ y $s\alpha_j = \sin(\alpha_j)$.

Para el robot HRP-4, cuyo esquema cinemático se muestra en la Figura 2.4 se tienen los PDHM de la Tabla 2.2 para las piernas, y de la Tabla 2.3 para cada brazo. Los valores de los parámetros constantes no nulos se dan en la Tabla 2.4 y en la Tabla 2.5.

Tabla 2.2 Parámetros Denavit-Hartenberg modificados de las piernas del robot HRP-4

Articulaciones	a	α	d	θ	r
1	0	0°	0	θ_1	0
2	0	-90°	0	θ_2	0
3	0	0°	d_3	θ_3	0
4	0	0°	d_4	θ_4	0
5	0	-90°	0	θ_5	r_5
6	0	-90°	0	θ_6	r_6
7	0	0°	d_7	θ_7	0
8	0	-90°	0	θ_8	r_8
9	0	-90°	0	θ_9	0
10	0	0°	d_{10}	θ_{10}	0
11	0	0°	d_{11}	θ_{11}	0
12	0	90°	0	θ_{12}	0

Tabla 2.3 Parámetros Denavit-Hartenberg modificados de los brazos del robot HRP-4

Orientaciones	a_{i-1}	α_i	d_i	θ_i	r_i
13	0	90°	0	θ_{13}	r_{13}
14	0	-90°	0	θ_{14}	0
15	0	0°	d_{15}	θ_{15}	0
16	0	-90°	0	θ_{16}	r_{16}
17	0	90°	0	θ_{17}	0
18	0	0°	d_{18}	θ_{18}	0

Tabla 2.4 Valores de los parámetros geométricos de las piernas

Dimensiones	d_1	d_2	d_3	d_{10}	d_{11}	r_7	r_8	r_9	r_{10}
mm	340	358	175	358	340	10	197	0	10

Tabla 2.5 Valores de los parámetros geométricos de los brazos

Dimensiones	d_{14}	d_{15}	r_{13}	r_{14}
mm	234	234	283	283

Modelo inverso de posición de las piernas del robot

Sustituyendo los parámetros de Denavit-Hartenberg modificados de las tablas 2.2 y 2.3 en la matriz de la ecuación (2.1) se obtienen las matrices elementales de las piernas del robot

$${}^0_1T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & d_3 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & d_4 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4_5T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & r_5 \\ -s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^5_6T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^6_pT = \begin{bmatrix} 0 & 1 & 0 & d_{7^*} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -r_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^p_7T = \begin{bmatrix} s\theta_7 & c\theta_7 & 0 & -d_{7^*}c\theta_7 \\ c\theta_7 & -s\theta_7 & 0 & d_{7^*}s\theta_7 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^7_8T = \begin{bmatrix} c\theta_8 & 0 & -s\theta_8 & 0 \\ -s\theta_8 & 0 & -c\theta_8 & 0 \\ 0 & 1 & 0 & -r_8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^8T_9 = \begin{bmatrix} c\theta_9 & -s\theta_9 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_9 & -c\theta_9 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^9T_{10} = \begin{bmatrix} c\theta_{10} & -s\theta_{10} & 0 & d_{10} \\ s\theta_{10} & c\theta_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{10}T_{11} = \begin{bmatrix} c\theta_{11} & -s\theta_{11} & 0 & d_{11} \\ s\theta_{11} & c\theta_{11} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{11}T_{12} = \begin{bmatrix} c\theta_{12} & -s\theta_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_{12} & c\theta_{12} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En las matrices 7T_p y 6T se tiene que d_7^* es igual a $d_7/2$.

A partir de las matrices elementales de las piernas del robot, se resuelve el modelo inverso de posición de éstas, con el cual se calculan los valores de las variables articulares que permiten obtener la posición y la orientación deseadas de la pelvis, y el pie oscilante del robot. La derivación de las ecuaciones del mencionado modelo se basó en el enfoque propuesto en [5] que, a su vez, se inspira en el principio del que parte el método de Paul [14].

La matriz de transformación homogénea que se utiliza para especificar la posición y orientación deseadas de la pelvis del robot, con respecto al marco del mundo Σ_M está dada por:

$${}^M T_p = \begin{bmatrix} t_{11} & t_{12} & t_{13} & x_p \\ t_{21} & t_{22} & t_{23} & y_p \\ t_{31} & t_{32} & t_{33} & z_p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

A partir de esta matriz, y de las matrices elementales de las piernas del robot, se obtienen las siguientes ecuaciones del modelo inverso de posición de la pierna de apoyo de robot:

$$a_6 = (-t_{23}r_6 + y_p)t_{31} + (-z_p + t_{33}r_6)t_{21} + (t_{21}t_{32} - t_{22}t_{31} + t_{31})d_{7^*}$$

$$b_6 = (-t_{23}r_6 + y_p + d_{7^*})t_{32} + (-z_p + t_{33}r_6)t_{22}$$

$$c_6 = 1$$

$$\theta_6 = \text{atan2}(c_6 a_6, -c_6 b_6)$$

$$a_1 = t_{31}r_5 c_6 + t_{32}r_5 s_6 - t_{32}d_{7^*} - t_{33}r_6 + z_p$$

$$b_1 = -t_{22}r_5 s_6 - t_{21}r_5 c_6 + t_{22}d_{7^*} + t_{23}r_6 - d_{7^*}y_p$$

$$c_1 = 1$$

$$\theta_1 = \text{atan2}(c_1 a_1, -c_1 b_1)$$

$$a_5 = (-t_{21}s_1 + t_{31}c_1)s_6 - (-t_{22}s_1 + t_{32}c_1)c_6$$

$$b_5 = -t_{23}s_1 + t_{33}c_1$$

$$\theta_5 = \text{atan2}(a_5, b_5)$$

$$x_{14/l} = (t_{22}c_1 s_6 + t_{32}s_1 s_6 + t_{21}c_1 c_6 + t_{31}s_1 r_6)r_5 - (t_{22}c_1 + t_{32}s_1)d_{7^*} + (d_{7^*} - t_{23}r_6 + y_p)c_1 + (-t_{33}r_6 + z_p)s_1$$

$$z_{14/l} = (t_{12}s_6 + t_{21}c_6)r_5 - t_{12}d_{7^*} - t_{13}r_6 + x_p$$

$$r_{14} = \sqrt{z_{14/l}^2 + x_{14/l}^2}$$

$$z_{14/lu} = \frac{z_{14/l}}{r_{14}}$$

$$x_{14/lu} = \frac{x_{14/l}}{r_{14}}$$

$$g = \frac{1}{2r_{14}}(d_3^2 - d_4^2 - r_{14}^2)$$

$$m = \frac{1}{2r_{14}}(d_4^2 - d_3^2 + r_{14}^2)$$

$$h = \sqrt{d_3^2 - g^2}$$

$$\epsilon_2 = 1$$

$$r_{tax} = \epsilon_2 h x_{14/10} + g z_{14/10}$$

$$r_{tay} = g x_{14/10} - \epsilon_2 h z_{14/10}$$

$$r_{fax} = z_{14/l} - r_{tax}$$

$$r_{fay} = x_{14/l} - r_{tay}$$

$$\theta_{tach} = \text{atan2}(r_{tay}, r_{tax})$$

$$\theta_{fach} = \text{atan2}(r_{fay}, r_{fax})$$

$$\theta_2 = \theta_{tach} - 90^\circ$$

$$\theta_3 = \theta_{fach} - \theta_{tach}$$

$$s_{234} = (t_{11}s_6 - t_{12}c_6)c_5 + t_{13}s_5$$

$$c_{234} = t_{11}c_6 + t_{12}s_6$$

$$\theta_{234} = \text{atan2}(s_{234}, c_{234})$$

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3$$

La matriz de transformación homogénea que se utiliza para especificar la posición y orientación deseadas del pie oscilante del robot, con respecto al marco de la pelvis Σ_p , se calcula mediante:

$${}_{12}^p T = {}_M^p T {}_{12}^M T \quad (2.3)$$

Donde ${}_M^p T$ es la matriz conocida que determina la posición y la orientación deseadas del marco del mundo con respecto a la pelvis, mientras que ${}_{12}^M T$ especifica la posición y orientación deseadas del pie oscilante con respecto al marco del mundo Σ_M . Los elementos de la matriz ${}_{12}^p T$, calculada mediante la ecuación (2.3), se designan de la siguiente manera:

$${}_{12}^p T = \begin{bmatrix} s_{fpx} & n_{fpx} & a_{fpx} & p_{fpx} \\ s_{fpy} & n_{fpy} & a_{fpy} & p_{fpy} \\ s_{fpz} & n_{fpz} & a_{fpz} & p_{fpz} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Siguiendo la misma metodología utilizada para la pierna de apoyo se obtienen las siguientes ecuaciones del modelo inverso de posición de la pierna oscilante:

$$a_7 = a_{fpy} p_{fpz} - a_{fpz} d_7 - a_{fpz} p_{fpy}$$

$$h_7 = a_{fpx} p_{fpz} - a_{fpz} p_{fpx}$$

$$\epsilon_7 = 1$$

$$\theta_7 = \text{atan2}(\epsilon_7 a_7 - \epsilon_7 b_7)$$

$$a_8 = p_{fpz}$$

$$h_8 = d_7 c_7 - s_7 p_{fpx} - c_7 p_{fpy}$$

$$\epsilon_8 = 1$$

$$\theta_8 = \text{atan2}(\epsilon_8 a_8, -\epsilon_8 b_8)$$

$$r_{812x} = s_7 c_8 p_{fpx} + c_7 c_8 p_{fpy} + s_8 p_{fpx} - c_7 c_8 d_7$$

$$r_{812z} = c_7 p_{fpx} - s_7 p_{fpy} + s_7 d_7 - r_8$$

$$r_{812} = \sqrt{r_{812x}^2 + r_{812z}^2}$$

$$r_{812ux} = \frac{r_{812x}}{r_{812}}$$

$$r_{812uz} = \frac{r_{812z}}{r_{812}}$$

$$g_9 = \frac{d_{10}^2 - d_{12}^2 + r_{812}^2}{2r_{812}}$$

$$h_9 = \sqrt{d_{10}^2 - g_9^2}$$

$$\epsilon_9 = 1$$

$$r_{f1x} = \epsilon_9 h_9 r_{812ux} + g_9 r_{812uz}$$

$$r_{f1y} = g_9 r_{812ux} - \epsilon_9 h_9 r_{812uz}$$

$$r_{t1x} = r_{812x} - r_{f1x}$$

$$r_{t1y} = r_{812x} - r_{f1y}$$

$$\theta_{f1ch} = \text{atan2}(r_{f1y}, r_{f1x})$$

$$\theta_{t1ch} = \text{atan2}(r_{t1y}, r_{t1x})$$

$$\theta_9 = \theta_{f1ch} - 90^\circ$$

$$\theta_{16} = \theta_{t1ch} - \theta_{f1ch}$$

$$s_{12} = s_{fpz}c_8 - (s_7s_{fpx} + c_7s_{fpy})s_8$$

$$c_{12} = n_{fpz}c_8 - (s_7n_{fpx} + c_7n_{fpy})s_8$$

$$\theta_{12} = \text{atan2}(s_{12}, c_{12})$$

$$a_{11} = c_{\theta_{10}} = \cos(\theta_9 + \theta_{10})$$

$$b_{11} = -s_{\theta_{10}} = -\sin(\theta_9 + \theta_{10})$$

$$d_{11} = c_7a_{fpx} - s_7a_{fpy}$$

$$e_{11} = 1$$

$$\theta_{11} = \text{atan2}(s'_{11}, c'_{11})$$

Modelo directo de posición de los brazos del robot

A diferencia de las piernas, los brazos del robot HRP-4 no requieren del modelo inverso de posición toda vez que el movimiento de éstos no se especifica en coordenadas operacionales. Las trayectorias articulares asociadas a las variables articulares activas de los brazos se especifican de manera explícita mediante leyes de movimiento de tipo cicloidal en función del tiempo. Las variables articulares que se determinan de esta manera son θ_{13} , θ_{14} , θ_{15} , θ_{16} , θ_{17} y θ_{18} . Las matrices elementales correspondientes a los brazos, en función de sus PDHM (Tabla 2.3) son las siguientes:

$${}_{13}^{00}T = \begin{bmatrix} c_{13} & -s_{13} & 0 & 0 \\ 0 & 0 & -1 & -r_{13} \\ s_{13} & c_{13} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{14}^{13}T = \begin{bmatrix} c_{14} & -s_{14} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_{14} & -c_{14} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{15}^{14}T = \begin{bmatrix} c_{15} & -s_{15} & 0 & d_{15} \\ s_{15} & c_{15} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{16}^{15}T = \begin{bmatrix} c_{16} & -s_{16} & 0 & 0 \\ 0 & 0 & 1 & r_{16} \\ -s_{16} & -c_{16} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{17}^{16}T = \begin{bmatrix} c_{17} & -s_{17} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{17} & c_{17} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{18}^{17}T = \begin{bmatrix} c_{18} & -s_{18} & 0 & d_{18} \\ s_{18} & c_{18} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los productos de las matrices elementales de cada brazo permiten resolver los modelos directos de posición de éstos.

2.3 Problemas directos de velocidad y aceleración

Los problemas directos de velocidad y aceleración se resolverán derivando numéricamente las trayectorias articulares, una vez que éstas se hayan obtenido, para determinar así las velocidades y aceleraciones articulares del robot.

3. Modelado geométrico del robot HRP-4 en la plataforma *OpenHRP3*

Durante el diseño y la planificación de movimientos de robots se recurre comúnmente a la realización de procesos de simulación a fin de refinar y optimizar su funcionamiento. Con esta intención se desarrolló el simulador de robots llamado *OpenHRP3* (*Open-architecture Human-centered Robotics Platform*). Se trata de un paquete de software que permite efectuar la simulación física de varios tipos de robots, incluyendo manipuladores de tipo serial, móviles con ruedas y humanoides (Figura 3.1). *OpenHRP3* incluye funciones para la simulación de imágenes obtenidas de cámaras virtuales y la generación de información a partir de sensores virtuales. Este paquete es distribuido a través de una licencia de software libre [15], y se utilizó en los estudios sobre el humanoide HRP-4 en este trabajo de tesis. En este capítulo se hace una descripción del paquete *OpenHRP3* y del proceso de modelado del robot HRP-4 en dicho software.

OpenHRP3 es una plataforma común para el desarrollo de componentes de software, de forma unificada, para diversos robots, basados en *RT-Middleware* (software estándar en tecnologías de la robótica) [16]. *OpenHRP3* está diseñado como un sistema distribuidor de objetos, compuesto por un programa cliente que maneja a los servidores como a un grupo de programas servidores que ofrecen varias funciones. Esta organización del paquete permite mejorar el mantenimiento y portabilidad en desarrollos de sistemas de simulación a gran escala.

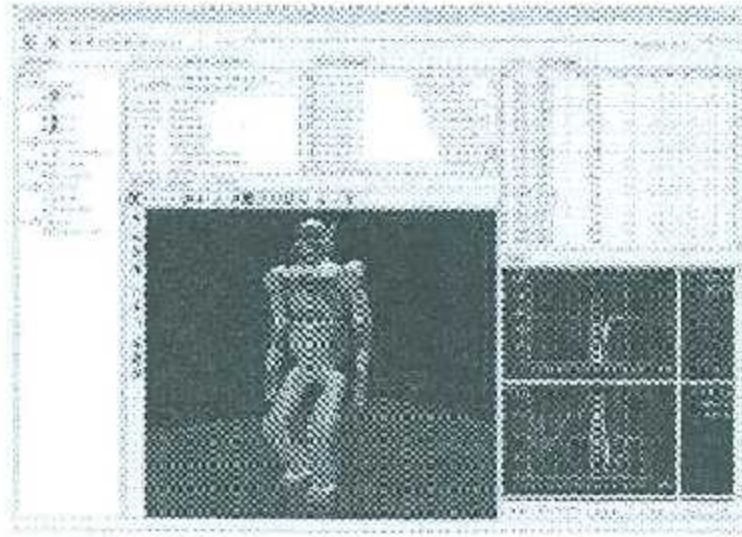


Figura 3.1 OpenHRP3 interfaz principal

3.1 GrxUI

GrxUI, es la interfaz gráfica de la plataforma *OpenHRP3*, desarrollada por *General Robotix, Inc*, subsidiaria de *Kawada industries*; funciona sobre el paquete Eclipse [17].

OpenHRP3 funciona de manera modular. Los diferentes elementos que participan en la simulación dinámica de un robot, ya sea humanoide, serial o móvil, están siendo calculados en módulos especializados para cada tarea, por ejemplo, el simulador dinámico, el detector de colisiones, el lector de modelos, todos ellos funcionan como entidades separadas, pero la interfaz de *GrxUI* es la encargada de mostrar los resultados de esta simulación en un solo punto(Figura 3.2).

La idea principal del funcionamiento de *GrxUI* consiste en la creación de proyectos que contienen un modelo virtual del robot a trabajar, así como un modelo virtual del mundo con el que interactúa. Asimismo, la interfaz permite configurar la *colisión de par*, que determina la manera en que interactúan dos objetos en contacto del robot. Por otra parte, se especifican los métodos de integración que se usarán en una simulación, así como el tiempo que durará ésta y, por último, se especifica un controlador externo encargado de verificar las consignas de movimiento y de procesar

la información de los sensores asignados específicamente a cada robot.

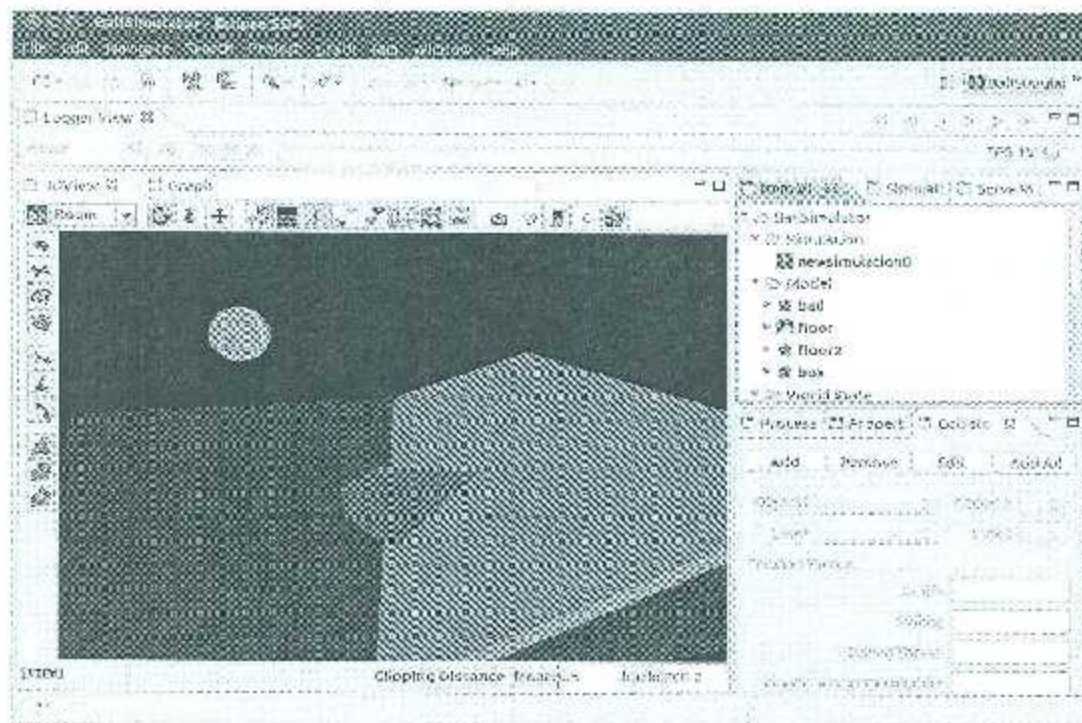


Figura 3.2 Interfaz GxUI-Eclipse

La interfaz permite crear modelos virtuales de los robots y de su entorno, así como monitorear el proceso de simulación; despliega la información de cada articulación de los robots simulados y de los sensores en general, admite la grabación de videos y la toma de fotografías de la simulación. Permite, asimismo, verificar el estado de los servidores y reiniciar los servidores.

3.2 Desarrollo de un modelo virtual

En esta sección se presenta el proceso de creación de un modelo virtual del robot humanoide *HRP-4* utilizando la interfaz *GxUI*. Las geometrías de los eslabones así como la arquitectura del robot que se usó para la creación del modelo virtual se basaron en la información del robot publicada por Kawada, Inc. (Figura 3.3).



Figura 3.3 HRP-4

1. **Preparación de datos.** Se prepara por anticipado la información necesaria acerca de las piezas que se utilizarán para el armado del modelo virtual. La especificación de la geometría de una pieza se efectúa a través de un archivo *.vrl*, el cual se crea mediante programas de desarrollo de ambientes virtuales, tales como *3D World Editor* (editor de ambiente *VRML* de *Simulink*) o *SolidWorks*. También es posible crear las piezas dentro de *GrxUI* a partir de geometrías primitivas.

Por otra parte, los datos físicos que se requieren de cada pieza son los siguientes:

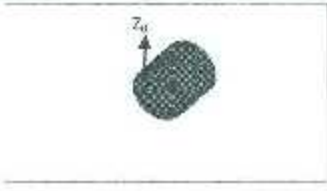




- o Masa
- o Centro de masa
- o Momentos de inercia

En la Tabla 3.1 se presentan los datos suministrados a *GrxUI* para la creación del modelo virtual del robot humano de HRP-4.

Tabla 3.1 Datos de las piezas usadas para la creación del modelo virtual del robot humanoide HRP-4

Pieza	Figura (DPL)	Masa (kg)	Centros de masa (mm)	Ejes de inercia	Momentos de inercia (kg/mm ²)
A1		570.90	$x = 86.50$ $y = 55.00$ $z = 15.00$	$I_x = (1,0,0)$ $I_y = (0,1,0)$ $I_z = (0,0,1)$	$P_x = 618475$ $P_y = 1466689.67$ $P_z = 1999529.67$
A2		342.90	$x = 0.00$ $y = 55.00$ $z = 0.00$	$I_x = (0,1,0)$ $I_y = (0,0,1)$ $I_z = (1,0,0)$	$P_x = 170119.77$ $P_y = 430814.34$ $P_z = 430814.34$
A3		342.90	$x = 55.00$ $y = 0.00$ $z = 0.00$	$I_x = (1,0,0)$ $I_y = (0,0,-1)$ $I_z = (0,1,0)$	$P_x = 170119.77$ $P_y = 430814.34$ $P_z = 430814.34$
B1		2066.12	$x = 0.00$ $y = 40.00$ $z = -136.91$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 2150248.94$ $P_y = 19942575.64$ $P_z = 19996192.16$
C1		2181.32	$x = 0.00$ $y = 40.00$ $z = -146.30$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 2273128.94$ $P_y = 23238467.81$ $P_z = 23292084.32$
C2		402.12	$x = 40.00$ $y = 0.00$ $z = 0.00$	$I_x = (1,0,0)$ $I_y = (0,0,-1)$ $I_z = (0,1,0)$	$P_x = 321699.09$ $P_y = 375315.60$ $P_z = 375315.60$

D1		1571.45	$x = 0.00$ $y = 0.00$ $z = 140.80$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 1542177.19$ $P_y = 9766889.30$ $P_z = 9766889.30$
E1		5908.65	$x = 0.00$ $y = 0.00$ $z = 216.97$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 32579086.36$ $P_y = 50456831.53$ $P_z = 76867394.03$
E2		477.52	$x = 0.00$ $y = 47.50$ $z = 0.00$	$I_x = (0,1,0)$ $I_y = (0,0,1)$ $I_z = (-1,0,0)$	$P_x = 302017.67$ $P_y = 550145.23$ $P_z = 550145.23$
F1		3832.42	$x = 0.00$ $y = 0.00$ $z = 151.62$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 13022182.71$ $P_y = 19003936.96$ $P_z = 19003936.96$
F2		402.12	$x = 0.00$ $y = 40.00$ $z = 0.00$	$I_x = (0,1,0)$ $I_y = (0,0,1)$ $I_z = (-1,0,0)$	$P_x = 321699.09$ $P_y = 375315.60$ $P_z = 375315.60$
G1		1687.59	$x = 0.00$ $y = 0.00$ $z = -148.57$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 1679458.14$ $P_y = 11608984.26$ $P_z = 11608984.26$
G2		510.19	$x = 0.00$ $y = 50.75$ $z = 0.00$	$I_x = (0,1,0)$ $I_y = (0,0,1)$ $I_z = (-1,0,0)$	$P_x = 408155.72$ $P_y = 642090.59$ $P_z = 642090.59$

G3		402.12	$x = 40.00$ $y = 0.00$ $z = 0.00$	$I_x = (1,0,0)$ $I_y = (0,0,-1)$ $I_z = (0,1,0)$	$P_x = 321699.09$ $P_y = 375315.60$ $P_z = 375315.60$
H1		402.12	$x = 0.00$ $y = 40.00$ $z = 0.00$	$I_x = (0,1,0)$ $I_y = (0,0,1)$ $I_z = (1,0,0)$	$P_x = 321699.09$ $P_y = 375315.60$ $P_z = 375315.60$
I1		875.72	$x = 0.00$ $y = 0.00$ $z = 31.64$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 826872.42$ $P_y = 2133416.21$ $P_z = 2133416.21$
J1		402.12	$x = 0.00$ $y = 40.00$ $z = 0.00$	$I_x = (0,1,0)$ $I_y = (0,0,1)$ $I_z = (1,0,0)$	$P_x = 321699.09$ $P_y = 375315.60$ $P_z = 375315.60$
K1		1195.20	$x = 0$ $y = 0$ $z = -107.45$	$I_x = (0,0,1)$ $I_y = (0,-1,0)$ $I_z = (1,0,0)$	$P_x = 1288060.71$ $P_y = 4269981.53$ $P_z = 4269997.31$

Cabe mencionar que los centros de masa de las piezas están en relación a los marcos de origen de las piezas mostrados en las figuras.



Figura 3.4 Creación de un modelo virtual de un robot.

2. **Inicio de la creación de un robot.** En la pestaña "Item View", colocando el apuntador sobre la carpeta *Model* (Figura 3.4), oprimir el botón derecho del mouse y en el menú desplegado elegir la opción "create" para crear un nuevo modelo virtual de un robot. Después de esto, debajo de la carpeta *Model*, aparecerá un nuevo directorio con el nombre del robot asignado por el usuario (*dummyhrp45* en la Figura 3.4).

3. **Incorporación de piezas al modelo.** Colocando el apuntador sobre la opción "root", se oprime el botón derecho del mouse, y en el menú desplegado elegir "add segment" para añadir un segmento (Figura 3.5). Después, se coloca el apuntador sobre el segmento generado (Figura 3.6) y seleccionar "add shape from VRML97" en el menú desplegado, para después seleccionar el archivo .wrl que contiene la geometría de la pieza que se desea agregar al modelo; por ejemplo, en la Figura 3.7 se selecciona la pieza E2.



Figura 3.5 Incorporación de un segmento al nuevo modelo en proceso de creación.

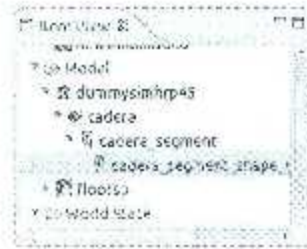


Figura 3.6 Incorporación de una nueva pieza al segmento actual.

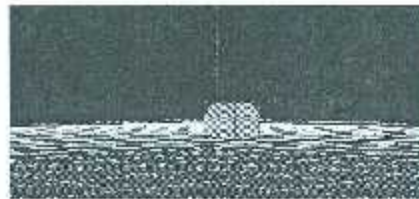


Figura 3.7 Pieza (cadera) creada.

Para especificar las características físicas deseadas de la pieza correspondiente, se selecciona la pestaña "Property View" (Figura 3.8). Este paso se repetirá tantas veces como sea necesario a lo largo del ensamble del modelo. Cada vez que se agrega una pieza se requiere especificar su masa, centro de masa y momentos de inercia así como una traslación de la posición de la pieza (Figura 3.9). Por lo general, al insertar una nueva pieza ésta se inserta en las coordenadas de la pieza anterior, ya que la forma de agregar las piezas lleva una lógica anidada.

4. **Incorporación de sensores.** Se coloca el apuntador sobre el nombre del eslabón "cadera" (antes "root") y se oprime el botón derecho de ratón; en el menú desplegado seleccionar la opción "add sensor" introduciendo el nombre que se desee (*sgyroscad* en la Figura 3.10). Por omisión, el sensor que se inserta es del tipo "force sensor"; pero es posible cambiar después este sensor seleccionándolo entre varios tipos (sensor de fuerza, acelerómetro, cámara, etc.); en este caso se selecciona un giroscopio ("*RateGyro*"). Se deben especificar las

características deseadas del sensor añadido, usando la pestaña "Property View" (Figura 3.11).



Name	Value
centerOfMass	0.0387500
name	cadena_agregada
mass	0.01750
radius	0.100
momentOfInertia	1.0749046650363320000e+000000000
translation	0.00

Figura 3.8 Propiedades físicas del segmento agregado (cadena).

Param	Value
torqueConst	1.0
BIAS	0.0000
centerOffset	0.00047500
motorResistor	1.0
offset	0.000
encoderVelocity	0.00000
encoderPulse	1.0
velocity	0.00000
rotationOffset	0.2014564600000000532807564000000000
rotationLimit	1.0
mode	torque
position	0.00000
mass	0.047772
rotationOffset	rotationOffset
jointType	revolute
rotation	0.000
rotationVelocity	0.0
rotation	0.0

Figura 3.9 Traslación del segmento

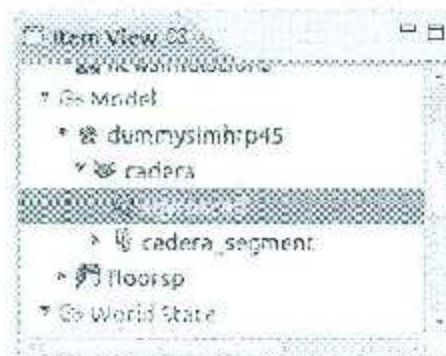


Figura 3.10 Incorporación de un nuevo sensor



Figura 3.11 Propiedades del sensor añadido.

5. **Conexión de eslabones.** Para conectar un eslabón nuevo al eslabón 'cadera' se coloca el apuntador sobre el eslabón "cadera", se oprime el botón derecho del ratón, y sobre el menú desplegado se elige la opción "add joint" (Figura 3.12) lo cual añadirá un eslabón dependiente del eslabón sobre el que fue creado (el "cadera").
6. **Integración del robot.** El robot humanoide se divide en 3 partes principales partiendo de la cadera, estas son: pierna izquierda, pierna derecha y torso. Las piezas se conectan de manera anidada; es decir, cada pieza depende de la pieza a la cual se conectó, de forma descendente. Se generan familias de eslabones para cada una de las extremidades. Así, añadiendo eslabones a partir de la cadera se integran las extremidades y el robot completo como se indica en la Figura 3.13. Después de esto, la pestaña 'Item View' despliega el árbol de la jerarquía de objetos del robot que se muestra en la Figura 3.14

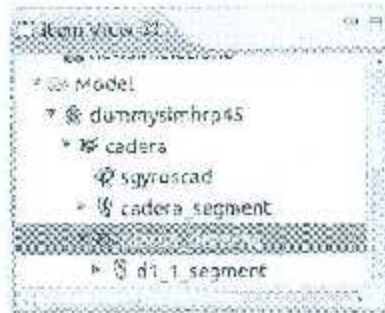


Figura 3.12 Añadir eslabón

7. **Identificación de articulaciones.** Después de haber completado la integración del modelo, y de haber añadido las características físicas de los diferentes objetos que lo conforman, se especifican las características de las articulaciones del robot seleccionando la pestaña "property view". Las articulaciones que utilizan los robots humanoides generalmente son actuadas, y pueden ser del tipo rotacional o del tipo prismático, a las cuales corresponden las opciones "rotate" y "slide" respectivamente. La opción "fixed", se refiere a una conexión que no permite movimiento relativo entre las piezas que conecta. Y la opción "free", finalmente, se refiere a una articulación pasiva. La especificación del tipo de movimiento relativo que se tiene en una articulación, se efectúa mediante la opción "jointAxis" del menú "Property View", la cual describe el movimiento que tendrá la articulación con respecto a los ejes x, y o z (Figura 3.15).

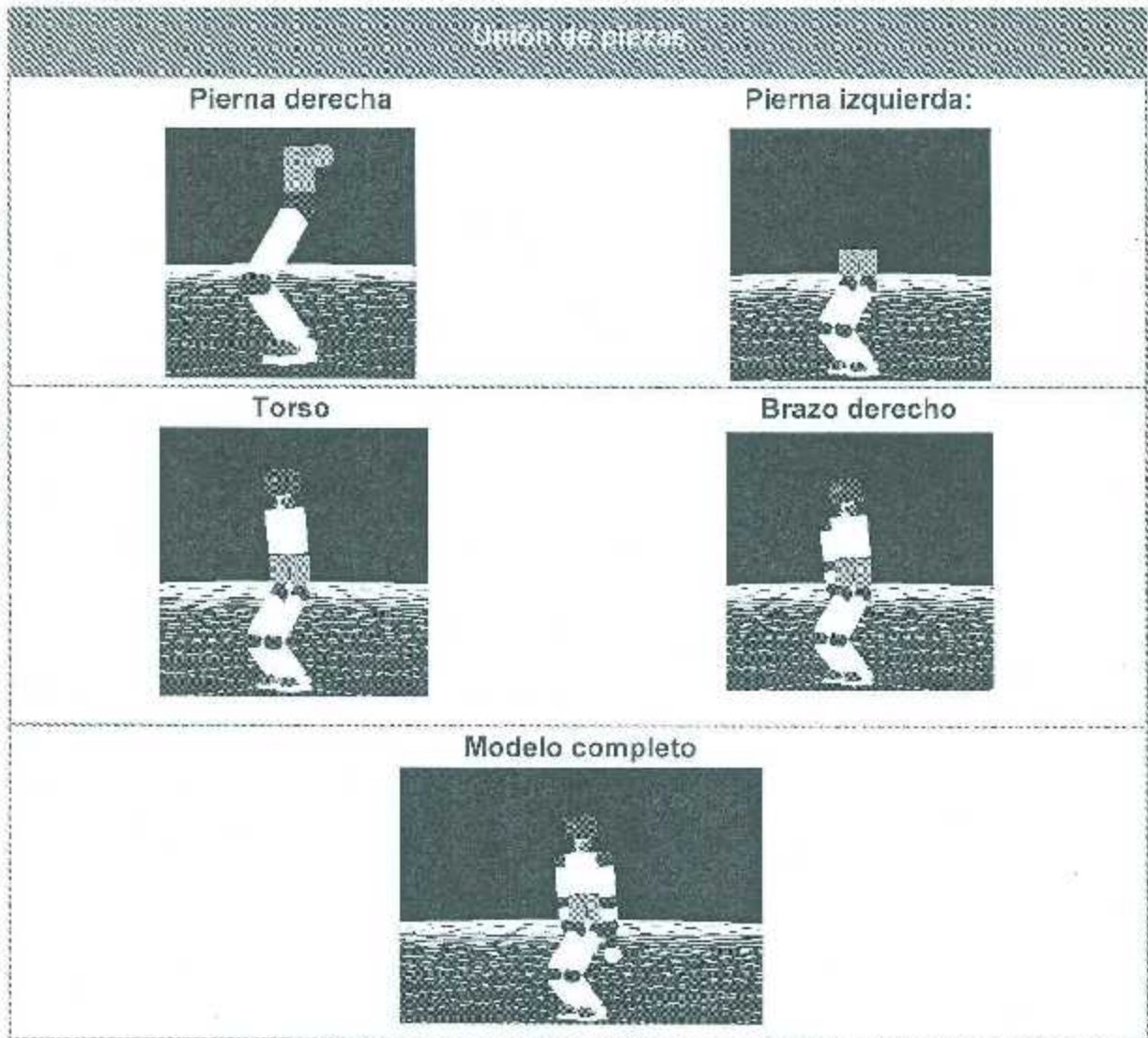


Figura 3.13 Proceso de integración de un modelo virtual del humanoide IIRP-4 en el sistema Open HRP3

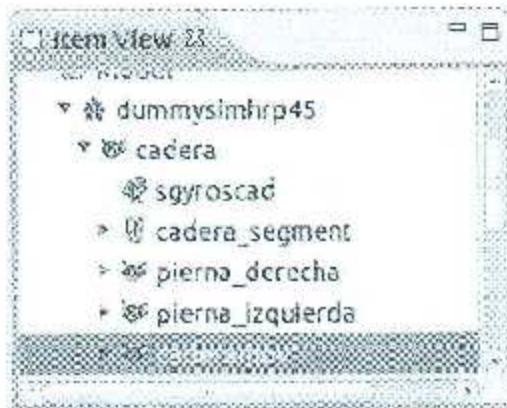


Figura 3.14 Progreso de armado del modelo

Name	Value
torqueConst	1.0
limit	0.0 0.0
centerOfMass	0.0 0.0475 0.0
colorResistor	1.0
ulimit	0.0 0.0
angularVelocity	0.0 0.0 0.0
encoderPulse	1.0
translation	0.063 0.095 0.76
velocity	0.0 0.0 0.0
momentsOfInertia	5.50195E-4 0.0 0.0 0.0 3.82018E-4 0.0 0.0 0.0
rotorInertia	1.0
mode	Torque
jointAxis	—
mass	0.547752
NumOfAAD	original data
jointType	free
rotation	fixed
jointVelocity	rotator
tolerance	slide

Figura 3.15 Tipo de articulaciones

Ingresando 0 1 0 en " *jointAxis*" para una articulación del tipo rotacional, por ejemplo, se produce una rotación relativa con respecto al eje y de la articulación actual (Figura 3.16). Cada articulación del robot se identifica usando la opción " *jointId*", aplicando números consecutivos a partir del 0. De igual manera se numeran los sensores por separado a partir del 0. Estos números permiten al usuario interactuar con las articulaciones y sensores desde el programa.

8. **Definición del piso.** El procedimiento para agregar un piso es el mismo que se sigue para añadir un nuevo robot, oprimiendo el botón derecho del ratón al estar el apuntador sobre la carpeta " *Model*" de la pestaña " *Item View*". En este caso se inserta el piso como si fuera un nuevo robot. Una vez insertado el piso, y especificadas sus propiedades físicas y su apariencia, se requiere cambiar sus atributos declarándolo como un *modelo de ambiente* el menú que incluye esta opción se despliega al oprimir el botón derecho del mouse al estar el apuntador sobre el *modelo del piso* de la pestaña " *Item View*" (Figura 3.17 y Figura 3.18).

Name	Value
torqueControl	1.0
limit	0.0 0.0
linkerOfMass	0.0 0.0 0.0
linkResistor	1.0
limit	0.0 0.0
enable Pulse	1.0
translation	0.0 0.0
momentsOfInertia	1.507145E+00 0.0 0.0 0.0 3.33019E+00 0.0 0.0 0.0 0.0
linkInertia	1.0
mode	HighGain
mass	0.47752
NumOfAxis	3
jointType	rotata
rotation	0 0 1 0
jointVelocity	0.0
tolerance	0.0
jointId	-2
exLimit	0.0 0.0
gearRatio	1.0

Figura 3.16 Identificación de una articulación.



Figura 3.17 Modelado del piso como entorno del robot.

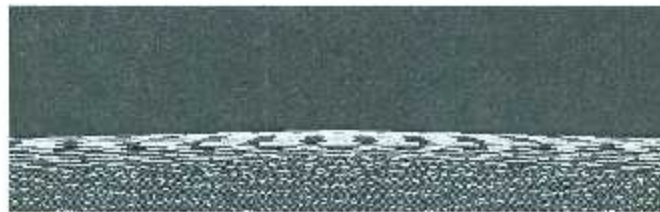


Figura 3.18 Visualización del piso.

9. **Habilitación del proyecto para simulación.** Esta habilitación se consigue al crear un nuevo "worldstate", oprimiendo el botón derecho del ratón al colocar el apuntador sobre la carpeta "World State" de la pestaña "Item View" y agregando un nuevo estado al seleccionar la opción "create" (Figura 3.19).



Figura 3.19 Worldstate creado

10. **Configuración de la interacción entre objetos.** Para especificar las características de la interacción entre el robot y su entorno, se selecciona la pestaña "Collision Pair" y se eligen los objetos que interactúan (el piso y el robot en el caso mostrado), y se especifican las propiedades físicas correspondientes, tales como los coeficientes de fricción y el coeficiente de restitución, y se selecciona el botón "Ok" para terminar. Por otra parte, la opción "Link" en la pestaña "Collision Pair" permite configurar la interacción entre eslabones (Figura 3.20).
11. **Guardado del modelo.** El modelo creado del robot se guarda en formato VRML97 oprimiendo el botón derecho del ratón estando el apuntador sobre la pestaña "ItemView" y seleccionando "save as".

3.3 Programa controlador del robot virtual

En esta Sección se describe el proceso de creación de un programa controlador y de su vinculación en el proyecto principal para la simulación del movimiento del robot humanoide HRP-4. El principio de funcionamiento de este programa será descrito en el Capítulo 5, en la sección 5.3 de este.



Figura 3.20 Configuración de la interacción entre objetos.



Figura 3.21 Creación de proyecto en RTCBuilder

De la misma manera que se preparan los datos de propiedades físicas de las piezas que integran un robot antes de crear su modelo, es necesario preparar por anticipado los datos necesarios para las trayectorias que asignará el controlador al modelo del robot. Estos datos, que contienen todas las consignas articulares para cada uno de los g.d.l. del humanoide, se prepararán mediante los programas descritos en el Capítulo 5. Por otra parte, se debe generar un *esqueleto* del controlador [18], con base en *Open-RTM*. Para ello se utiliza la herramienta *RTCBuilder* del paquete *OpenHRP3*, y se sigue el procedimiento que se enuncia en los siguientes párrafos:

1. **Iniciación de un nuevo proyecto mediante RTCBuilder.** Se tiene acceso a *RTCBuilder* desde la interfaz de *Eclipse*, de manera similar a *GrxUI*. La función principal del programa consiste en generar una estructura predefinida de programación para un controlador (Figura 3.21).
2. **Configuración de la pestaña Basic del nuevo proyecto de RTCBuilder.** Esta pestaña contiene información básica para el controlador (por ejemplo el nombre del módulo y el tiempo de ejecución del controlador). Se configura con los datos que se indican en la ventana de la Figura 3.22



4. **Configuración de los puertos de datos (Data Ports).** Los puertos de datos de *RTCbuilder* constituyen los canales de comunicación y transferencia de datos entre el simulador y el controlador. Estos puertos de datos están vinculados con los puertos de datos de *GrxUI*. En la configuración de los puertos de datos en *RTCBuilder* (Figura 3.24) se establece cuales canales serán de entrada y cuales de salida así como el tipo de dato con el que se trabajará. Los datos de entrada se especifican en la caja del lado izquierdo de la sección "Data Ports", mientras que los de salida se definen en la caja del lado derecho.
5. **Selección del lenguaje de programación.** El lenguaje de programación que prefiera el usuario se selecciona en la pestaña "Lenguaje and Environment" (Figura 3.25)



Figura 3.24 Pestaña Data Ports



Figura 3.25 Pestaña Language and environment

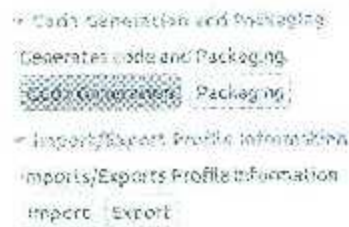


Figura 3.26 Generación del código

6. **Compilación y creación del esqueleto.** En esta última etapa en *RTCToolbox* se genera el código base del controlador Figura 3.26 En el esqueleto se realiza un programa para la comunicación entre el controlador y los servidores del paquete *OpenHRP3*, bajo los estándares de comunicación *RT* [19]

3.4 Configuración en *GrxUI* para el manejo de datos

La fase final de preparación para la simulación se refiere a la configuración de entradas y salidas de datos desde *GrxUI*. En este proceso se ejecutan los siguientes pasos:

1. **Selección del robot.** En la pestaña "*Controller*" se selecciona la opción "*Edit*" (Figura 3.27).
2. **Especificación del controlador.** En la pestaña "*Controller*" se especifica la ruta en que se encuentran el programa del controlador con el que se va a trabajar y el archivo de configuración del controlador. Asimismo, se especifica la ruta correspondiente al archivo *.sh* que contiene la configuración de los puertos; se trata del puente (*Controller Bridge*) de entradas y salidas del simulador, y de manejo de señales de los sensores con los que se trabaja (Figura 3.28 y Figura 3.29). Para iniciar una simulación sólo se selecciona el icono de "*Start simulation*"

simulation 

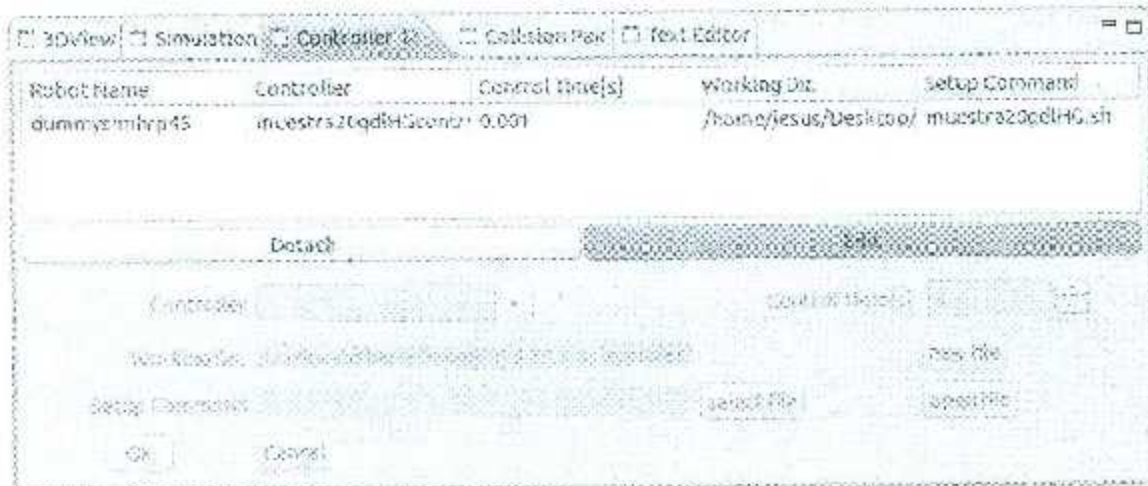


Figura 3.27 Pestaña Controller



Figura 3.28 Selección de controlador

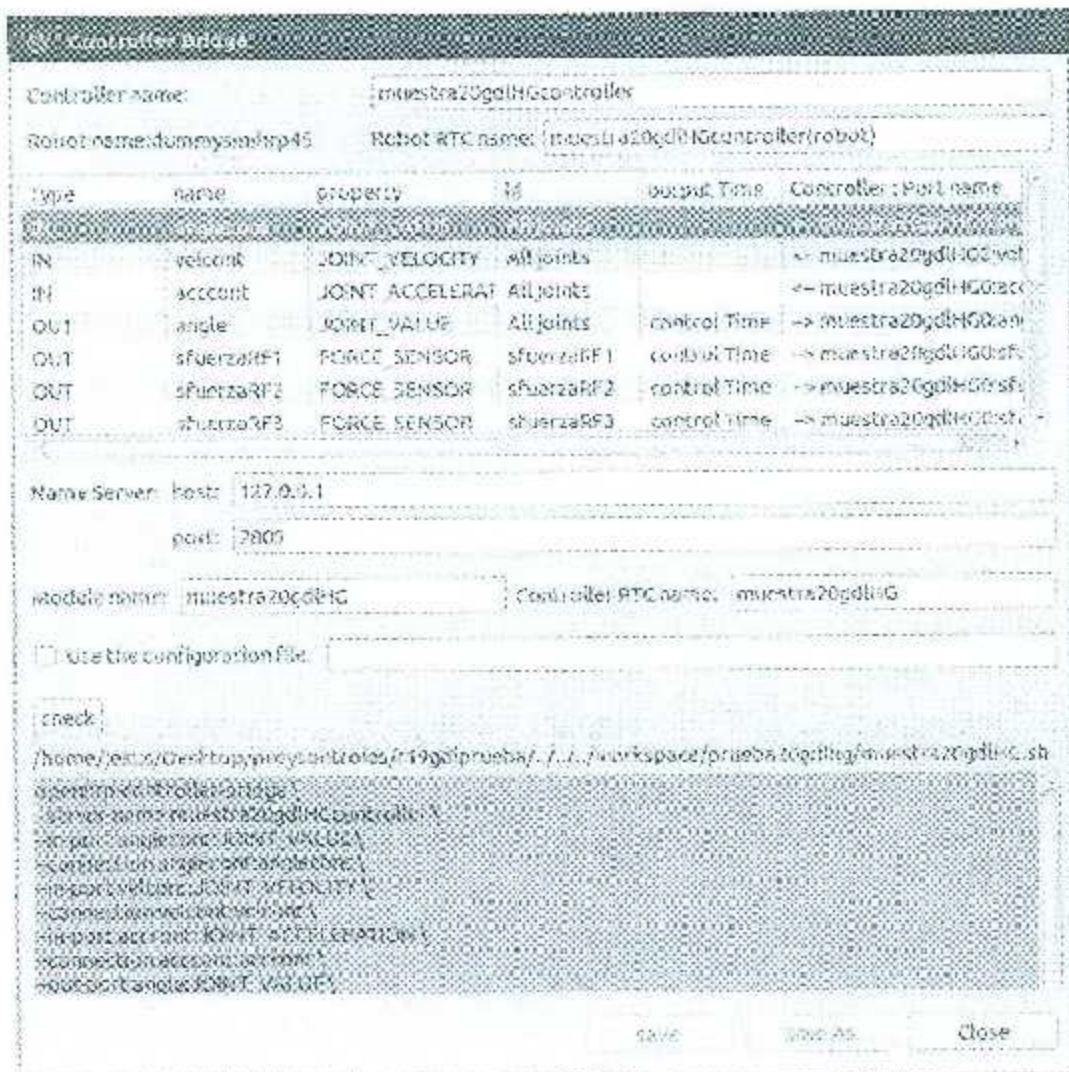


Figura 3.29 Configuración del puente

4. Planificación de movimientos del robot humanoide

En el presente capítulo se aborda el problema de la planificación de movimientos del robot humanoide HRP-4 para el cumplimiento de la acción deportiva propuesta, que consiste en el pateado de un balón de fútbol soccer en reposo, como si se tratara de la ejecución de un tiro de castigo. Para este propósito se toma como referencia la biomecánica de los movimientos de un futbolista humano al ejecutar la misma acción deportiva.

4.1 Biomecánica de un futbolista humano

La biomecánica del ser humano es una disciplina de gran extensión; comprende el estudio del comportamiento del cuerpo humano en diferentes acciones; no deportivas y deportivas. En el marco de esta tesis interesa el comportamiento cinemático del cuerpo humano durante una acción típica de fútbol soccer.

En la literatura científica existen múltiples trabajos relacionados con esta parte de la biomecánica. En [20] se estudian los movimientos de un grupo de jugadores de fútbol; se tomó una muestra de 14 jugadores en la ejecución de un tiro de castigo, de los cuales se analizó la cinemática de sus tiros, analizando la velocidad del tobillo, rodilla y cadera, así como el ángulo de entrada de estos al momento de impactar el balón y haciendo una comparación entre las patadas que fueron efectivas y las que no lo fueron. En otro estudio [21] se aborda el tema del tiro de castigo pero esta vez haciendo énfasis en el ángulo de aproximación para el cobro del tiro y la eficacia de los diferentes resultados obtenidos; se estudia la cinemática con diferentes ángulos de ataque hacia el balón (se analizaron ángulos de 30° , 45° y 60°). Por otra parte, se han efectuado diversos trabajos relativos al desempeño de futbolistas profesionales [22] durante la ejecución de un tiro libre; entre las variables relevantes del estudio destacan la posición del centro de masa del jugador durante la realización del tiro, la velocidad de

aproximación del jugador al balón, la velocidad y la longitud del último paso, así como los ángulos de aproximación, y la velocidad del pie. En otro documento [23] se hace un extenso estudio sobre la cinemática de los movimientos necesarios para el cobro de un tiro utilizando la parte interna del pie; se analiza la influencia del ángulo de aproximación del jugador al balón, así como el contacto entre el pie y el balón.

Tomando en consideración los trabajos previos en la biomecánica de la patada, finalmente se decidió analizar los movimientos de un futbolista humano que ejecuta exitosamente un tiro de castigo con una técnica simple y eficiente [24]. En la Figura 4.1 se aprecia una secuencia de posturas de este futbolista correspondientes a diferentes instantes durante la ejecución del disparo.

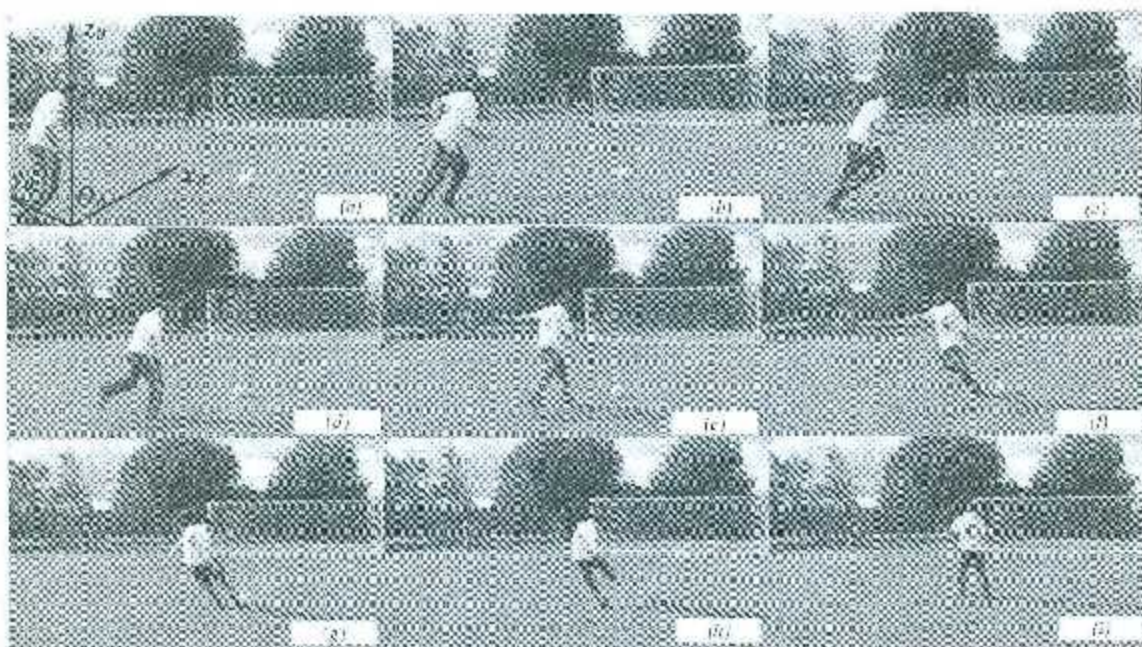


Figura 4.1 Secuencia de posturas de un jugador humano al ejecutar un tiro de castigo.

Si bien, los desplazamientos del futbolista tienen componentes en el espacio 3D, para los propósitos de esta tesis se consideran solamente las componentes del movimiento plano de mayor amplitud, que son las que se proyectan en el plano sagital (plano x_0z_0 del marco de referencia mostrado en la Figura 4.1 se aprecian las siguientes etapas de movimiento:

- a) **Inicio.** El futbolista se encuentra en estado de reposo con los pies juntos, y las rodillas ligeramente flexionadas, así como una ligera inclinación del torso hacia adelante.
- b) **Paso de preparación para el impulso inicial.** A partir de la postura inicial, con el pie izquierdo apoyado, el jugador desplaza el pie derecho hacia atrás hasta alcanzar la postura de impulso inicial (Figura 4.1a), incrementando ligeramente la inclinación del torso hacia adelante (*pitch*). Ubica con la mirada la zona de la portería en la que desea colocar el balón.
- c) **Primer paso.** El jugador se apoya en el pie derecho para impulsarse y acelerar el cuerpo en dirección x_0 en el plano sagital (Figura 4.1b); gracias a este impulso, la cadera inicia un movimiento uniformemente acelerado (*m.u.a.*) en esa dirección, el cual se prolongará durante los siguientes pasos. El torso aumenta su inclinación hacia el frente (*pitch*) con un movimiento uniformemente acelerado, contribuyendo a incrementar la inercia del jugador. El pie izquierdo se desplaza hacia el frente hasta tomar su nueva posición al concluir el paso.
- d) **Segundo paso.** El jugador se apoya en el pie izquierdo y la cadera continúa con *m.u.a.*, incrementando su velocidad en dirección x_0 (Figura 4.1c). El torso conserva su inclinación mientras que el pie derecho avanza a su siguiente posición.
- e) **Tercer paso y preparación para la patada.** El pie de apoyo es el derecho, y la cadera continúa ganando velocidad en dirección x_0 con *m.u.a.* (Figura 4.1d). El torso conserva su inclinación y el pie izquierdo avanza. Se observa que en cada nuevo paso el desplazamiento del pie es mayor en comparación con el desplazamiento del pie en el paso precedente. También se constata que en este

paso, el pie apoyado se despega del suelo unas fracciones de segundo antes de que aterrice el pie oscilante, efectuándose una fase muy corta de vuelo (Figura 4.1e). El pie izquierdo concluye su desplazamiento ocupando una posición a la izquierda del balón, ligeramente atrás de éste (Figura 4.1f). En la fase de vuelo, la extremidad derecha ejecuta rápidamente una doble oscilación (*swing*) hacia atrás, con el muslo y la pierna, que concluye al aterrizar el pie izquierdo (Figura 4.1f). En esta pequeña fase de vuelo el jugador también endereza el torso girándolo ligeramente hacia atrás.

- f) **Ejecución de la patada.** Al llegar el pie izquierdo al piso al concluir el tercer paso, el pie derecho se encuentra plegado hacia atrás (Figura 4.1f); se observa que al descender rápidamente el pie derecho desde esa postura éste golpeará el balón con la máxima velocidad relativa con respecto a la cadera, justo en el momento en que la cadera alcanza la máxima velocidad (Figura 4.1g) por lo tanto el choque con el balón se producirá con la máxima energía cinética acumulada en el pie, suministrando así la máxima velocidad al balón. Al iniciar la extremidad derecha del jugador el *swing* hacia el abajo para impactar el balón, el torso vuelve a oscilar hacia el frente, lo que le ayudará al jugador a mantenerse en equilibrio después de golpear el balón.

- g) **Movimiento post-patada.** Después de impactar el balón, la extremidad derecha continúa su oscilación elevándose hacia el frente hasta detener el *swing* mientras que el pie izquierdo se mantiene apoyado en el piso (Figura 4.1h). Al mismo tiempo la cadera desacelera uniformemente hasta detenerse. Inmediatamente después de suspender el *swing* el pie derecho desciende lentamente hasta hacer contacto con el piso (Figura 4.1i) mientras que la cadera realiza algunos desplazamientos buscando el equilibrio, estando los dos pies apoyados en el piso.

A partir de los movimientos observados en los párrafos anteriores, se propondrá una secuencia de movimientos del robot humanoide, con base en trayectorias específicas de puntos del robot en la cadera y en el pie oscilante.

4.2 Generación de trayectorias.

Teniendo en cuenta la biomecánica del jugador descrita en la sección anterior, se definen las características deseadas de los movimientos de las piernas, la cadera y el torso del robot HRP-4.

Tabla 4.1 Especificación de parámetros de los movimientos propuestos del humanoide

Paso ($t_{imp, paso}$)	Tiempo (s)	Cadera		Pierna				Pie		Pie (oscilante)		$\delta\theta_{torso}$ ($^{\circ}$)	$\delta\theta_{cabeza}$ ($^{\circ}$)	
		$\delta\theta_1$ ($^{\circ}$)	θ_{max} ($^{\circ}$)	$\delta\theta_2$ ($^{\circ}$)	$x_{pie, osc}$ (m)	δx_{pie} (m)	$z_{pie, osc}$ (m)	δz_{pie} (m)	$x_{pie, osc}$ (m)	δx_{pie} (m)	$x_{pie, osc}$ (m)			δx_{pie} (m)
1	0-1	0.80	7 $^{\circ}$	10 $^{\circ}$	0	0	0.89	0	0	-0.15	0	0	-	-
2	1-2	0.80	17 $^{\circ}$	10 $^{\circ}$	0	0.255	0.82	-0.025	-0.15	0	0	0.11	-	-
3	2-3	0.86	27 $^{\circ}$	0 $^{\circ}$	0.055	0.1650	0.755	-0.025	-0.15	0.46	0.11	0	-	-
4	3-4	0.80	27 $^{\circ}$	10 $^{\circ}$	0.22	0.2750	0.77	0	0.31	0	0.11	0.5072	-	-
5	4-5	0.14	37 $^{\circ}$	-8.5 $^{\circ}$	0.495	0.0594	0.77	0.025	-	-	0.6172	0	2	25
6	5-6	0.14	23.5 $^{\circ}$	-8.5 $^{\circ}$	0.5544	0.0629	0.795	0	-	-	-	-	-16.60	34.83
7	6-7	0.14	20 $^{\circ}$	-30 $^{\circ}$	0.6172	0.276	0.795	0	-	-	-	-	-16.80	34.83
8	7-8	0.30	-10 $^{\circ}$	0 $^{\circ}$	0.0446	0	0.795	0.025	0.33	0.0	0.6172	0	-	-
9	8-9	1.00	-10 $^{\circ}$	55 $^{\circ}$	0.6442	0	0.82	-0.25	0.33	0	0.6172	0	-	-

Inspirados en las etapas identificadas en la sección 4.1, se propone una serie de fases del movimiento del robot, cuyos parámetros se dan en la Tabla 4.1. A fin de simplificar el análisis, en este primer estudio no se consideran desplazamientos de los brazos, excepto en la última fase. Asimismo, se supone que no existen fases de vuelo. Las orientaciones de la pelvis y los pies se mantienen paralelas al plano horizontal (el $x_0 - y_0$) en todo momento. La aceleración constante de la pelvis en dirección x_0 es de una magnitud de 0.1719 m/s^2 . Los movimientos cicloidales considerados se basan en los aplicados en [4], [25] y [26]. Las características específicas de las 10 fases se detallan en los siguientes párrafos.

Fase 0. Postura inicial del robot en reposo. En la Figura 4.2 se muestra esta postura del robot. Sus características son las siguientes:

- $t = 0 \text{ s}$.
- Torso: Rotado con respecto al eje y_0 un ángulo $\mu = 7^\circ$ (*pitch*).
($\mu = 0^\circ$ corresponde a una orientación del torso perpendicular al plano $x_0 - y_0$)
- Pelvis: El centro geométrico se encuentra en el eje z_0 con una altura de 0.82 m
- Pies: Ambos pies juntos, apoyados en el suelo.

Fase 1. Preparación para el impulso inicial. En la Figura 4.3 se aprecia la postura del robot al final de esta fase

- Periodo: De $t = 0$ a $t = 0.8 \text{ seg}$.
- Torso: Rotación del torso con respecto al eje y_0 , de $\mu = 7^\circ$ a $\mu = 17^\circ$, con una ley semi-cicloidal de arranque.
- Pelvis: Sin desplazamiento.
- Pies: El pie derecho se desplaza en dirección de x_0 desde $x = 0$ hasta $x = -0.15 \text{ m}$ con un movimiento cicloidal completo. El pie izquierdo permanece sin desplazamiento.



Figura 4.2 Postura inicial.

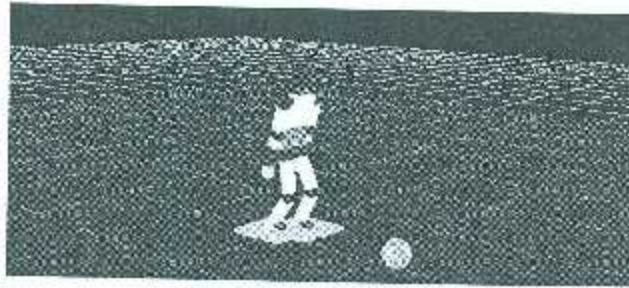


Figura 4.3 Postura del robot al finalizar la Fase 1

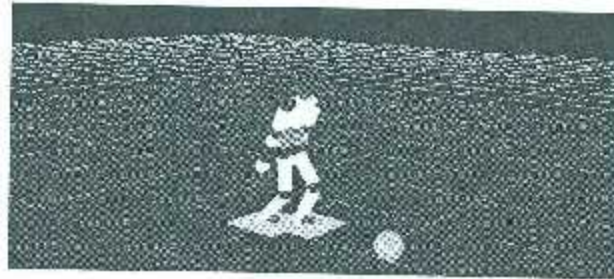


Figura 4.4 Postura del robot al finalizar la Fase 2.

Fase 2. Paso 1 de carrera. En la Figura 4.4 se observa la postura del robot al término de esta fase.

- **Periodo:** De $t = 0.8$ s a $t = 1.6$ s.
- **Torso:** Rotación del torso con respecto al eje y_0 , de $\mu = 17^\circ$ a $\mu = 27^\circ$, con una ley semi-cicloidal de detención.
- **Pelvis:** Inicia un desplazamiento en el plano sagital (plano $x_0 - z_0$), con un movimiento uniformemente acelerado en dirección x_0 , desde $x=0$ hasta $x=0.055$ m, y un movimiento cicloidal en dirección z_0 desde $z=0.82$ m hasta $z=0.795$ m.
- **Pies:** El pie derecho es el de apoyo, y el pie izquierdo se mueve en el plano sagital con una ley cicloidal en dirección x_0 , desde $x=0$ hasta $x=0.11$ m mientras que en dirección z_0 lo hace con dos leyes cicloidales, una para subir el pie y otra para bajarlo, cada una de ellas en un tiempo de 0.1 s.

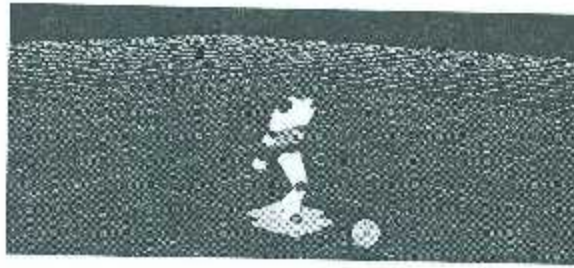


Figura 4.5 Postura del robot al terminar la Fase 3.

Fase 3. Paso 2 de carrera. En la Figura 4.5 se aprecia la postura del robot al finalizar esta fase.

- Periodo: De $t = 1.6$ s a $t = 2.4$ s.
- Torso: Conserva la orientación alcanzada en la fase anterior.
- Pelvis: Continúa su desplazamiento en el plano sagital, con movimiento uniformemente acelerado en dirección x_0 , desde $x = -0.055$ m hasta $x = 0.22$ m, y cicloidal en dirección z_0 de $z = 0.795$ m a $z = 0.77$ m.
- Pies: El pie izquierdo es el de apoyo, y el pie derecho se mueve en el plano sagital con una ley cicloidal en dirección x_0 , desde $x = -0.15$ m hasta $x = 0.48$ m, mientras que en dirección z_0 lo hace con dos leyes cicloidales: una para subir el pie y otra para bajarlo, cada una de ellas en un tiempo de 0.4 seg.

Fase 4. Paso 3 de carrera. En la Figura 4.6 se aprecia la postura del robot al terminar esta fase.

- Periodo: De $t = 2.4$ s a $t = 3.2$ s.
- Torso: Rotación del torso con respecto al eje y_0 , de $\mu = 27^\circ$ a $\mu = 37^\circ$, siguiendo una ley cicloidal.
- Pelvis: Desplazamiento en el plano sagital, con m.u.a. en x_0 , de $x = 0.22$ m a $x = 0.495$ m. No hay componente de desplazamiento en z_0 .
- Pies: Pie derecho apoyado, y el izquierdo se mueve en el plano sagital con una ley cicloidal en dirección x_0 , de $x = 0.11$ m a $x = 0.5072$ m, mientras que en dirección z_0 lo hace con dos leyes cicloidales: una

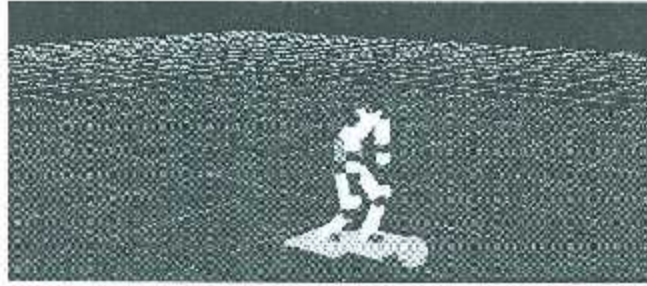


Figura 4.6 Postura del robot al finalizar la Fase 4.

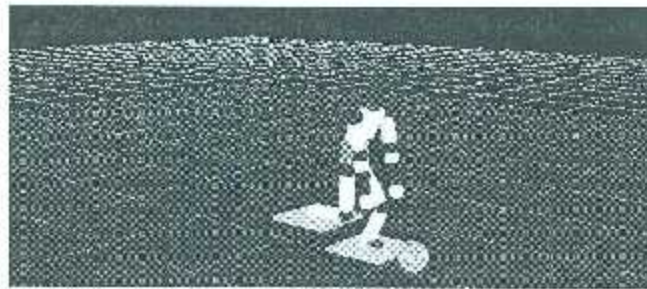


Figura 4.7 Postura del robot al terminar la Fase 5.

- para subir el pie y otra para bajarlo, cada una de ellas en $0,4$ s.

Fase 5. Preparación de patada. En la Figura 4.7 se aprecia la postura del robot al terminar esta fase.

- Periodo: De $t = 3,2$ s a $t = 3,34$ s.
- Torso: Rotación del torso con respecto al eje y_0 , de $\mu=37^\circ$ a $\mu=28,5^\circ$, siguiendo una ley semi-cicloidal de arranque.
- Pelvis: Desplazamiento en el plano sagital, con m.u.a. en dirección x_0 , de $x=0,495$ m a $x = 0,554$ m, y movimiento cicloidal en dirección z_0 desde $z = 0,77$ m hasta $z = 0,795$ m.
- Pies: El pie izquierdo es el de apoyo, y la extremidad derecha oscila hacia atrás a fin de incrementar la energía potencia del pie derecho. En esta oscilación, partiendo de la última postura de la extremidad derecha obtenida en la Fase 4, la articulación del fémur gira un ángulo $\delta\theta_{fémur} = 2^\circ$ mientras que la articulación de la tibia gira un ángulo $\delta\theta_{tibia} = 85^\circ$. Ambas rotaciones se producen con movimientos cicloidales.

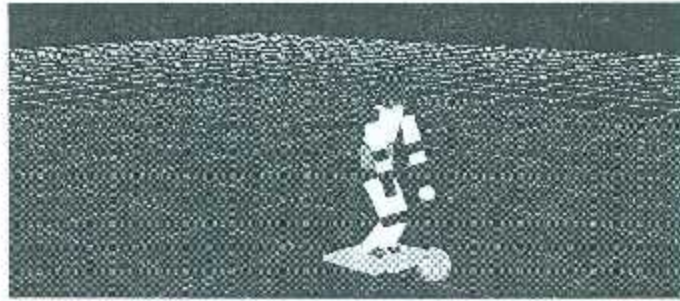


Figura 4.8 Postura del robot al finalizar la Fase 6

Fase 6. Patada. En la Figura 4.8 se observa la postura del robot al terminar esta fase.

- **Periodo:** De $t = 3.34$ s a $t = 3.48$ s.
- **Torso:** Rotación con respecto al eje y_0 , de $\mu = 28.5^\circ$ a $\mu = 20^\circ$, siguiendo una ley semi-cicloidal de detención.
- **Pelvis:** Desplazamiento en el plano sagital, con m.u.a. en dirección x_0 de $x = 0.554$ m a $x = 0.617$ m. En dirección z_0 no hay desplazamiento.
- **Pies:** El pie izquierdo es el de apoyo y la extremidad derecha oscila hacia abajo a fin de impactar el balón con el pie derecho. En esta oscilación, (*swing*) partiendo de la última postura de la extremidad derecha obtenida en la Fase 5, la articulación de fémur gira un ángulo $\delta\theta_{fémur} = -16.6^\circ$ mientras que la articulación de la tibia gira un ángulo $\delta\theta_{tibia} = -34.83^\circ$. Ambas rotaciones se producen con movimientos semi-cicloidales de arranque.

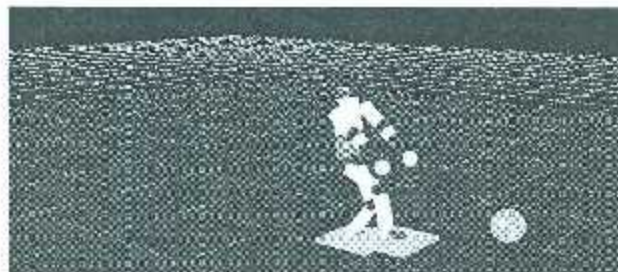


Figura 4.9 Postura del robot al finalizar la Fase 7.

Fase 7. Swing. En la Figura 4.9 se presenta la postura del robot al terminar esta fase.

- Periodo: De $t = 3.48$ s a $t = 3.62$ s.
- Torso: Rotación con respecto al eje y_0 de $\mu = 20^\circ$ a $\mu = -10^\circ$, siguiendo una ley cicloidal.
- Pelvis: Desplazamiento en el plano sagital, con m.u.a. (desaceleración) en dirección x_0 , desde $x = 0.617$ m hasta llegar al reposo en $x = 0.6448$ m. En dirección z_0 no hay desplazamiento.
- Pies: El pie izquierdo es el de apoyo, y la extremidad derecha continúa el *swing* hacia adelante después de impactar el balón con el pie (Fase 6) En esta oscilación, la articulación del fémur gira un ángulo $\delta\theta_{fémur} = -16.6^\circ$ y la articulación de la tibia gira un ángulo $\delta\theta_{tibia} = -34.83^\circ$. Ambas rotaciones se producen con movimientos semi-cicloidales de detención.

Fase 8. Fin de patada. En la Figura 4.10 se muestra la postura del robot al terminar esta fase.

- Periodo: De $t = 3.62$ s a $t = 3.92$ s.
- Torso: Conserva la última orientación de la fase anterior.
- Pelvis: No ocurre desplazamiento en dirección del eje x_0 , mientras que en dirección de z_0 se produce un desplazamiento positivo de 0.025 m.
- Pies: El pie de apoyo es el izquierdo. El pie derecho se desplaza de la última posición de la fase anterior a la definida por $x = 0.93$ m y z al nivel del piso. Tanto la componente de desplazamiento en x como la de z son de tipo cicloidal.

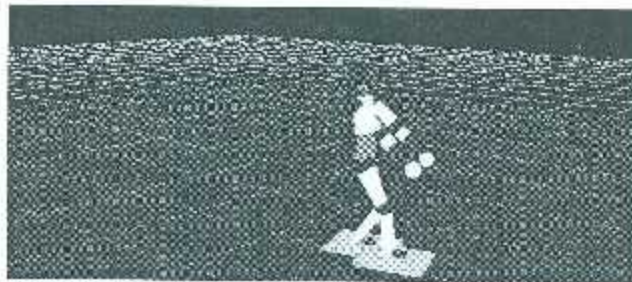


Figura 4.10 Postura del robot al concluir la Fase 8.

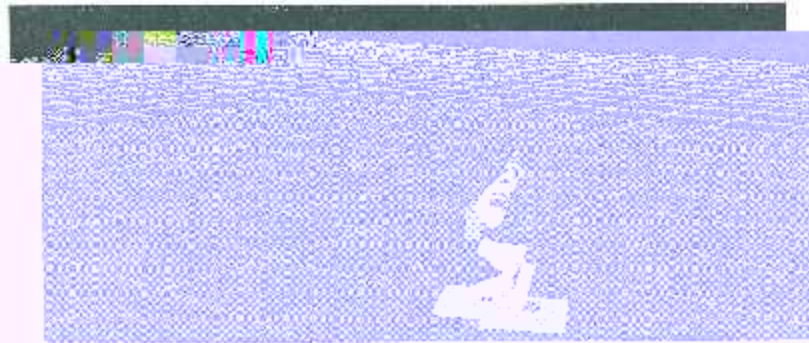


Figura 4.11 Fase 9

Fase 9. Conclusión del movimiento Aún cuando los desplazamientos asignados al robot se realizan con cambios graduales de aceleración y , en consecuencia, el comportamiento de las fuerzas de inercia no presenta perturbaciones relevantes. Después de ejecutar la patada el robot se presenta un leve proceso de inestabilidad en un periodo breve. Teniendo esto en consideración, en la última fase se asignan al robot movimientos suplementarios que contribuyan a solventar el mencionado proceso y permitan al robot preservar el equilibrio bajando la posición de su centro de masa con respecto al piso. En esta fase se aplican desplazamientos laterales a los brazos del robot hacia arriba a fin de evitar colisiones de éstos con el piso. En la Figura 4.11 se conserva la postura del robot al terminar esta fase.

- Período: De $t = 3.92$ s a $t = 4.92$ s.
- Torso: Rotación con respecto al eje y_0 , de $\mu = -10^\circ$ a $\mu = 35^\circ$, siguiendo una ley cicloidal.
- Pelvis: Se le aplica un desplazamiento negativo en el eje x_0 de 0.25 m. No se asigna desplazamiento en dirección x_0 .
- Pies: No se asignan desplazamientos a los pies.

En la Figura 4.12 se muestran de manera esquemática las rutas que describen los pies y la pelvis del HRP-4 durante las fases 1 a 9.

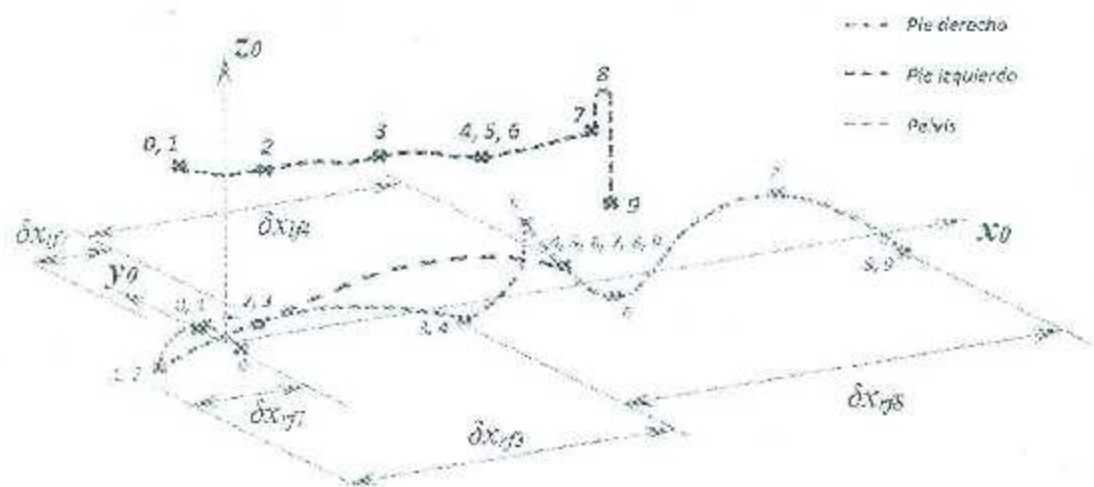


Figura 4.12 Rutas de los pies y pelvis en las diferentes fases.

5. Generación de las trayectorias articulares y simulación

Tras la especificación en el capítulo anterior del comportamiento ceseado del robot humanoide *HRP-4* inspirado en los movimientos de un jugador humano, y haciendo una adaptación de estos movimientos a las dimensiones del humanoide, en el presente capítulo se obtienen las trayectorias articulares de éste, aplicando el modelo inverso de posición estudiado en el Capítulo 2. Las trayectorias articulares así obtenidas se suministran como consignas de movimiento al modelo virtual del robot de *OpenHRP3* para validar, mediante la simulación dinámica, el equilibrio del robot.

5.1 Obtención de trayectorias articulares en *Matlab*.

En el marco de esta tesis se usa el paquete *Matlab* con la finalidad de obtener los valores numéricos correspondientes al comportamiento de las variables articulares del robot *HRP-4* en función del tiempo, los cuales serán utilizados como datos para el simulador dinámico del paquete *OpenHRP3*.

El modelo cinemático inverso de posición del robot se presentó en el Capítulo 2. El procedimiento de obtención de dicho modelo se basó en el que se aplicó al robot humanoide *Bioid* en [27]. Este procedimiento se adaptó a la arquitectura del robot *HRP-4*. Por otra parte, se obtuvo también el modelo cinemático directo de posición del humanoide para utilizarlo en la validación preliminar de los resultados de modelo inverso. En efecto, el modelo directo permite construir una representación geométrica *de alambre* del robot para observar los movimientos de éste antes de efectuar las simulaciones en el paquete *OpenHRP-3*.

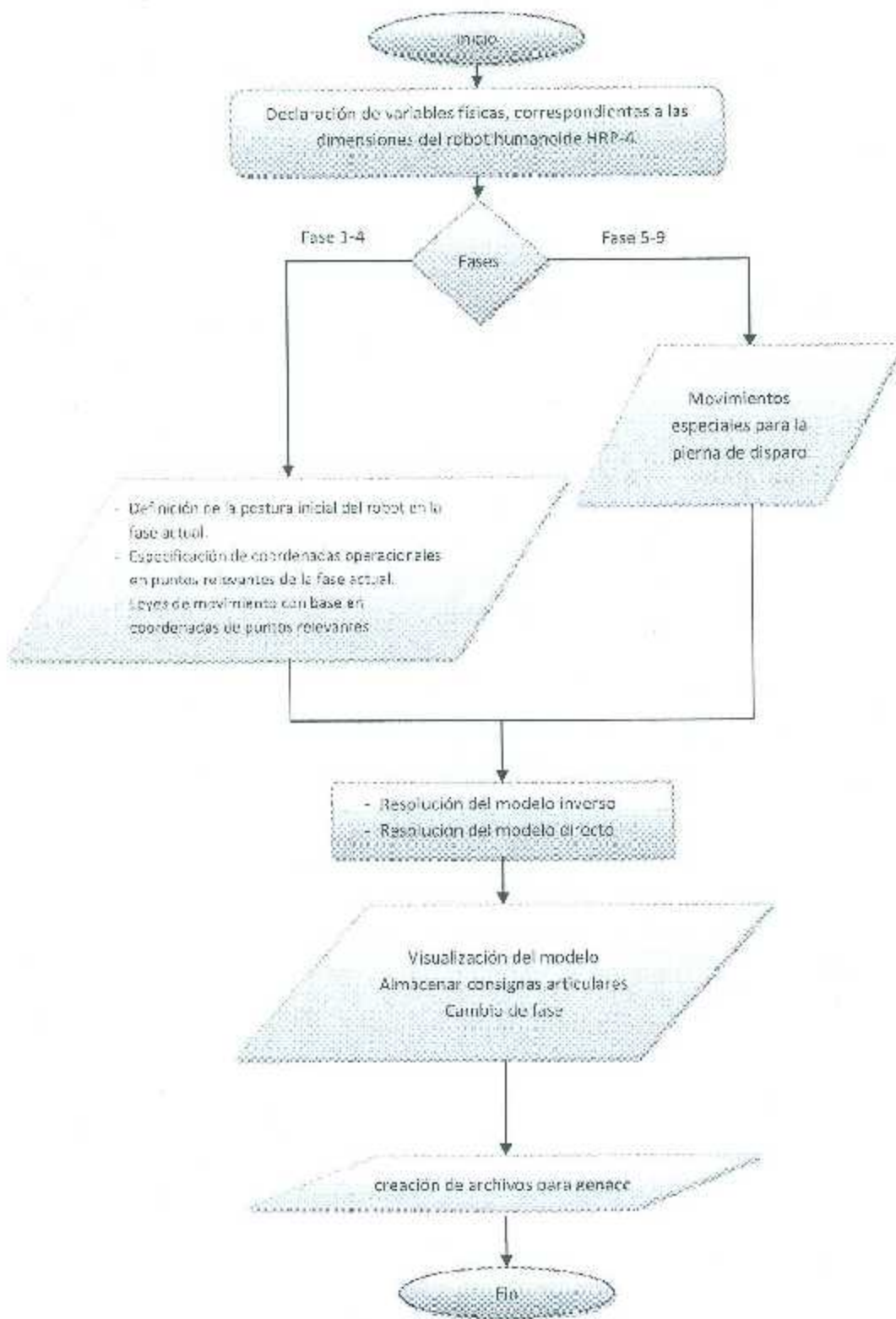


Figura 5.1 Diagrama de flujo del programa para la generación de las trayectorias articulares.

- **Definición de la postura inicial del robot en la fase actual**

Al inicio de cada fase es necesario conocer la ubicación del robot. En la fase 1 estos datos se especifican directamente, mientras que en las fases posteriores la postura inicial se define con base en la postura final de la fase anterior.

- **Coordenadas operacionales**

A partir de la posición inicial del humanoide, en las fases 1 a 4, que corresponden al avance del robot, se determinan las posiciones cartesianas y orientaciones, en una muestra de valores del tiempo, para el pie oscilante, la cadera, el torso y los brazos con respecto al marco del mundo, teniendo en cuenta las correspondientes leyes de movimiento.

- **Trayectorias específicas**

Las fases 5 a 8 se refieren a los movimientos que efectúa finalmente el robot para ejecutar la patada. En estas fases se determinan las posiciones cartesianas y orientaciones, en una muestra de valores del tiempo, solamente para la cadera, y el torso con respecto al marco del mundo, teniendo en cuenta las correspondientes leyes de movimiento. Para la pierna que efectúa el *swing* y para los brazos se determinan directamente valores específicos de las variables articulares que conforman dichas extremidades, así como el torso. En la fase 9 sólo se determinan las coordenadas operacionales de la pelvis.

- **Modelo cinemático inverso y directo.**

En la siguiente etapa del programa, aplicando el modelo inverso de posición del robot, se calculan los valores numéricos de las variables articulares que determinan las posturas deseadas del robot. En las fases 1 a 4 se obtienen las magnitudes de las variables correspondientes a las articulaciones de la pierna de apoyo y de la pierna oscilante. En las fases 5 a 8 el modelo inverso de posición proporciona los valores numéricos de las variables articulares asociadas solamente a la pierna de apoyo, mientras que las variables de las articulaciones de la pierna oscilante se determinan directamente, sin necesidad del modelo inverso, en la muestra de tiempo especificada por el usuario.

Conociendo los valores numéricos de de las variables articulares de todo el robot, se resuelve el modelo directo de posición con el fin de calcular las coordenadas de posición de los extremos de cada eslabón del robot con respecto al marco del mundo.

- **Visualización del modelo de alambre.**

Las coordenadas de posición de los extremos de cada eslabón del robot que se obtuvieron en la etapa anterior permiten efectuar una representación geométrica del robot en diagrama de alambre, para cada postura de la muestra de tiempo especificada, con lo cual se tiene una animación del humanoide que permite validar visualmente en *Matlab* los movimientos especificados.

Por otra parte las trayectorias articulares, también obtenidas en la etapa precedente, se almacenan numéricamente en un archivo de texto que servirá como archivo de datos de entrada para efectuar la simulación física y dinámica del robot en el paquete *Open HRP-3*. En la Figura 5.2 se aprecia gráficamente el comportamiento que tienen finalmente las 11 variables articulares que determinan los movimientos deseados del robot en el tiempo total de 0 a 4.92 s.

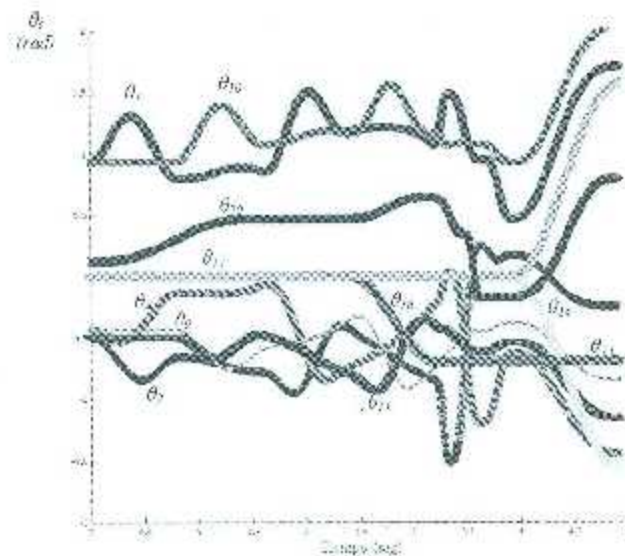


Figura 5.2 Trayectorias articulares correspondientes al movimiento deseado del robot

5.3 Control del simulador dinámico del paquete OpenHRP3.

Las trayectorias articulares generadas en la Sección 5.1 son leídas por un controlador, que trabaja en lenguaje C++ y que es configurado previamente en el programa RTCBuilder del paquete *OpenHRP3*, para crear y ejecutar en tiempo real las consignas de movimiento del humanoide virtual durante su simulación. La estructura de funcionamiento del controlador se ilustra en el diagrama de la Figura 5.3, y se describe brevemente en los siguientes párrafos.

- **Definición de entradas, salidas y variables**

En este sector se declaran todas las variables y herramientas con los cuales trabajará el programa encargado de controlar los movimientos del humanoide en el simulador dinámico. Las variables que almacenan datos deben de ser del mismo tipo de las variables que leen los datos de los archivos de referencia. Por otra parte, deberán de especificarse los canales de entrada y salida de datos, así como el tipo de datos que fluirán a través de éstos. El protocolo de comunicación entre los archivos de referencia, el controlador y el simulador dinámico está desarrollado con base en *RT-Middleware* [28].

- **Administrador**

El sector del controlador encargado de alternar las etapas por las que pasa la simulación del humanoide es el *Administrador (Manager)*. Éste alterna los ciclos del programa entre el encendido del robot, inicialización, estado activo, ejecución, finalización, error y apagado. Para realizar una simulación se recurre a los estados de inicialización, activación y ejecución.

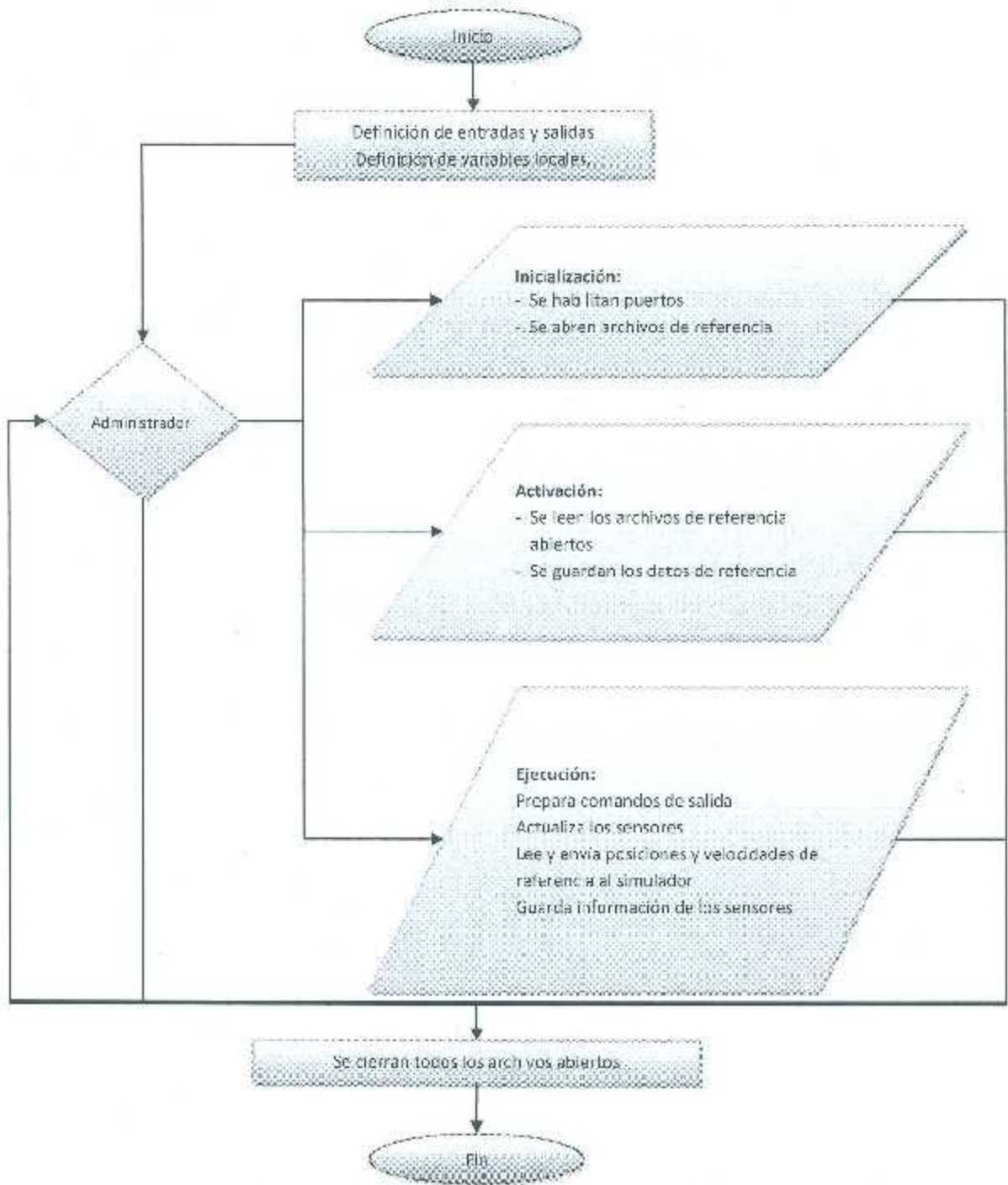


Figura 5.3 Estructura de funcionamiento del controlador usado para la simulación dinámica

- **Inicialización (*OnInit*)**

En esta etapa se preparan los elementos con los que trabaja el controlador. Se trata de la habilitación de puertos de comunicación, tiempos y variables a trabajar, así como de la apertura de los archivos de referencia para los movimientos del humanoide.

- **Activación (*OnInit*)**

Esta sección es invocada cada vez que es llamado el controlador. Se especializa en la recolección de datos, por lo que lee los archivos de referencia creados en *Matlab*, y guarda la información en variables locales para su uso, además de recopilar la información que se genera en los sensores en el humanoide.

- **Ejecución (*OnInit*)**

Esta parte del programa se activa cada vez que el controlador está en funcionamiento. En esta parte se ejecutan las instrucciones de movimiento del humanoide, por lo que se realizan funciones de lectura y escritura de los movimientos a realizar a partir de los archivos de referencia (la escritura se refiere al envío de valores numéricos específicos a cada articulación del humanoide). Asimismo, se recopila la información concerniente al estado del humanoide; es decir, guarda los datos capturados por los sensores del robot y los guarda en un archivo (a diferencia de la sección de *Activación* que guarda estos datos en una variable local).

- **Destructor**

A finalizar el programa, se cierran todos los archivos que se encuentran abiertos.

La ejecución de los programas aquí descritos permite efectuar la simulación dinámica de los movimientos del robot humanoide HRP-4; en la Figura 5.4 se muestra una secuencia de posturas del robot correspondientes a los movimientos de éste, obtenidos finalmente en un proceso de simulación con las trayectorias articulares generadas en la Sección 6.1.

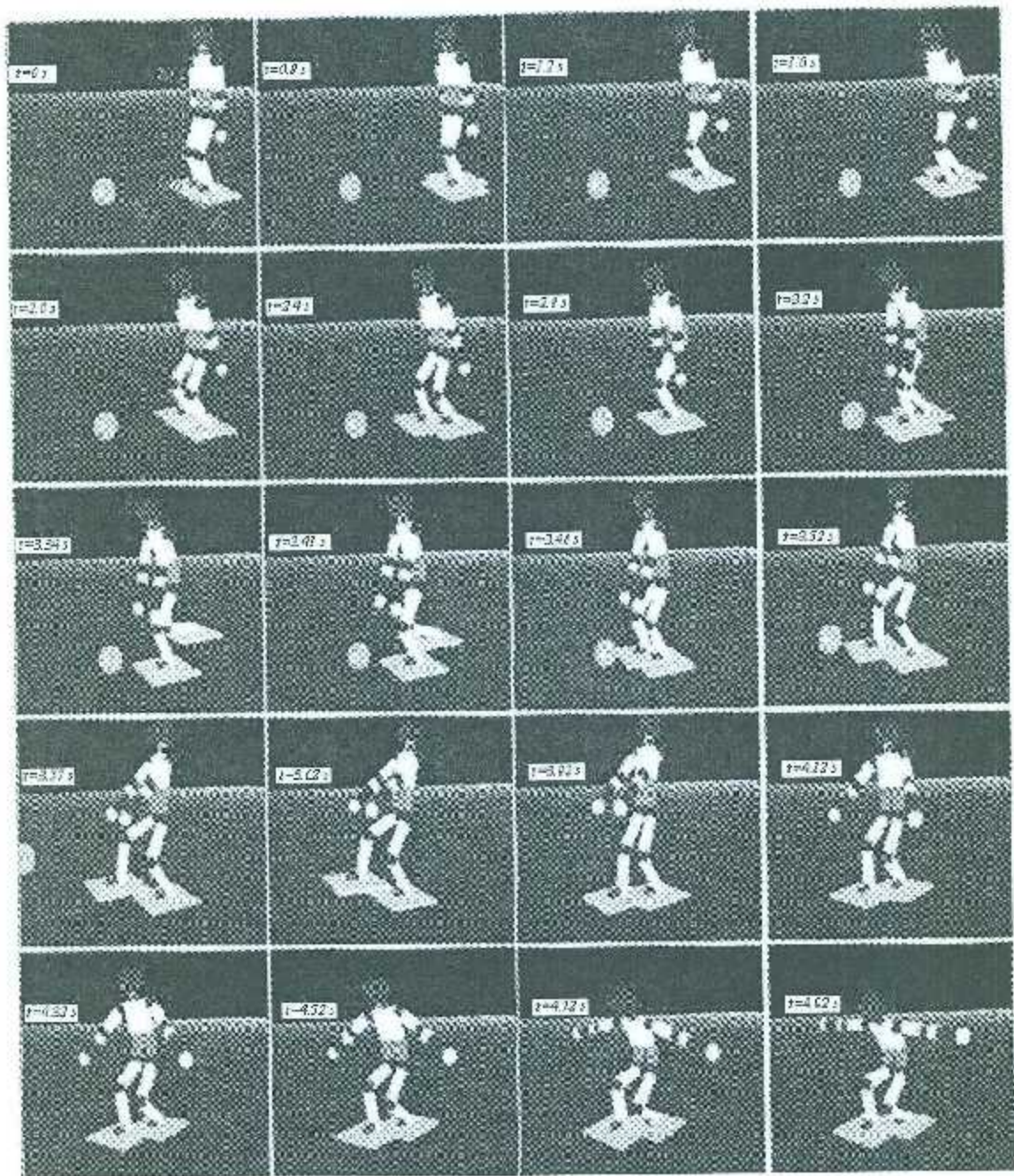


Figura 5.4 Secuencia de posturas del modelo virtual del robot obtenidas durante la simulación de los movimientos propuestos.

6. Análisis del equilibrio del robot

Los movimientos propuestos para la ejecución de la patada del robot humanoide HRP-4, y los parámetros que especifican dichos movimientos son de utilidad sólo si se prueba que el robot permanece en equilibrio al ejecutar la mencionada acción deportiva. Si bien el equilibrio del robot durante la simulación de los movimientos del robot virtual en OpenHRP-3 constituye una prueba de la validez de los movimientos propuestos, toda vez que se trata de una simulación física, se consideró de interés en nuestros estudios efectuar un análisis del equilibrio del robot utilizando el concepto del punto de momento cero (*ZMP* por las letras iniciales del término en inglés: **Z**ero **M**oment **P**oint). En este capítulo se presenta dicho análisis.

6.1 Punto de Momento Cero (*ZMP*).

El Punto de Momento Cero (*ZMP*) ha sido utilizado en diversos estudios como criterio para probar el equilibrio de un robot bípedo durante la marcha, v.gr. [30], [3]. El concepto del *ZMP* fue propuesto por Vukobratovic en 1969 [32], y ha sido abordado en otros trabajos posteriores en [29], [31]. El Punto de Momento Cero es un punto sobre la superficie de sustentación del pie de apoyo del caminante, en el cual la suma de las componentes horizontales de los momentos debidos a las fuerzas inerciales y la acción de la gravedad sobre los elementos físicos del cuerpo, se anulan. Bajo esta condición, en ese mismo punto debe actuar la resultante concentrada equivalente R , de la fuerza reactiva del suelo sobre el pie durante la fase de soporte simple (Figura 6.1). En consecuencia, la existencia del *ZMP* dentro de la superficie de sustentación garantiza la presencia de la fuerza que mantiene en equilibrio al bípedo.

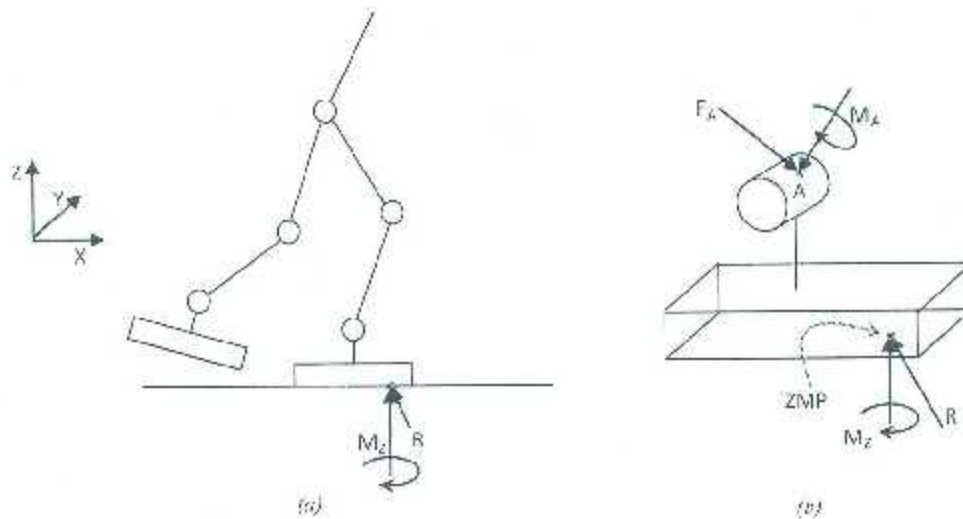


Figura 6.1 Cargas actuando sobre el pie de apoyo de un bípedo en fase de soporte simple.

En la Figura 6.1 (a), se muestra una representación simplificada de un bípedo en fase de soporte simple. La fuerza R es la resultante equivalente de las fuerzas de reacción del suelo sobre el pie y M_z es la componente vertical del momento de reacción vertical del suelo (bajo la condición de equilibrio no existen componentes horizontales de este momento). En la Figura 6.1(b) se muestra un diagrama de cuerpo libre del pie de apoyo, en el que se aprecia cómo actúa la fuerza R justamente en el ZMP, la fuerza F_A es el vector de fuerza resultante sobre el tobillo y M_A es el vector de momento resultante sobre el tobillo; ambas cargas son equivalentes a la acción de los eslabones superiores del bípedo en el punto A del tobillo. La componente M_z del par que ejerce el piso sobre la suela es ocasionado por las fuerzas de fricción entre suela y piso. La localización del ZMP sobre la suela cambia en la medida en que el bípedo se mueve; sin embargo, el ZMP nunca debe abandonar la superficie de la suela mientras el bípedo se mantenga en equilibrio; ésta es la condición que debe satisfacer el bípedo durante la marcha para mantenerse en equilibrio.

6.2 Análisis del Punto de Momento Cero.

En la simulación hecha en el paquete *OpenHRP3*, se instalaron sensores virtuales en las plantas de los pies del humanoide, con la finalidad de calcular las coordenadas donde se encuentra el punto de momento cero (ZMP) a partir de las señales obtenidas de estos sensores.

6.2.1 Generación del punto de momento cero (ZMP). Programa *genzmp*.

Para el procesamiento de las señales capturadas por los sensores, se desarrolló el programa *genzmp* en *Matlab*, cuya estructura lógica se muestra en la Figura 6.2, y se describe brevemente en los siguientes párrafos.

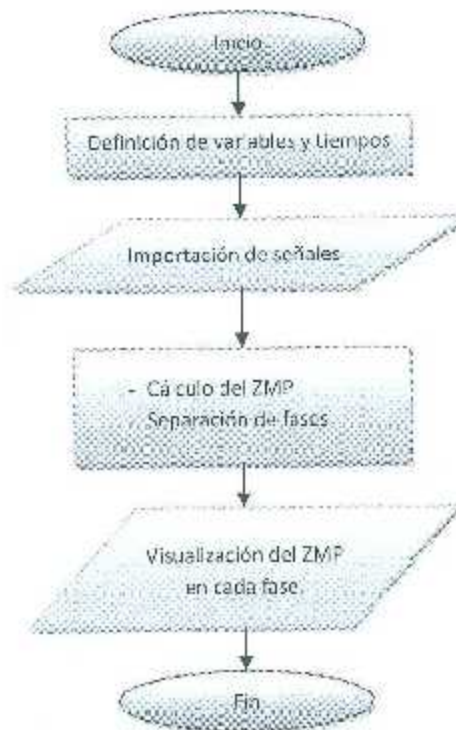


Figura 6.2 Estructura de operación del programa *genzmp*

- **Definición de variables y tiempos**

En esta etapa se prepara el espacio en el que se importarán las señales proporcionadas por los sensores; para ello es necesario sincronizar los tiempos entre los diversos programas, así como el tipo de variables con el que se trabajará.

- **Importación de señales.**

Las señales generadas por los sensores del modelo virtual del robot se almacenan en un archivo de texto durante la simulación en *OpenHRP3*; dichas señales son importadas por el programa *genzmp* y almacenadas en arreglos para su posterior procesamiento.

- **Cálculo del ZMP**

En esta sección se procesan las señales de los sensores de fuerzas y momentos importados previamente para calcular el *ZMP* de cada pie de apoyo en las muestras de tiempo definidas en la simulación. El procedimiento que se siguió para el cálculo del *ZMP* a partir de las fuerzas es el aplicado en [33].

- **Separación de fases**

Debido a las características del proceso de captura de señales en línea mediante el controlador, dichas señales se deben separar y asociar al pie de apoyo que corresponda según el instante de tiempo de la simulación.

- **Visualización del ZMP**

Por último se grafican las posiciones del *ZMP* correspondientes a los tiempos de muestreo para cada paso. En la Figura 6.3 se aprecian las rutas graficadas para el *ZMP* en cada pie de apoyo durante el proceso de simulación.

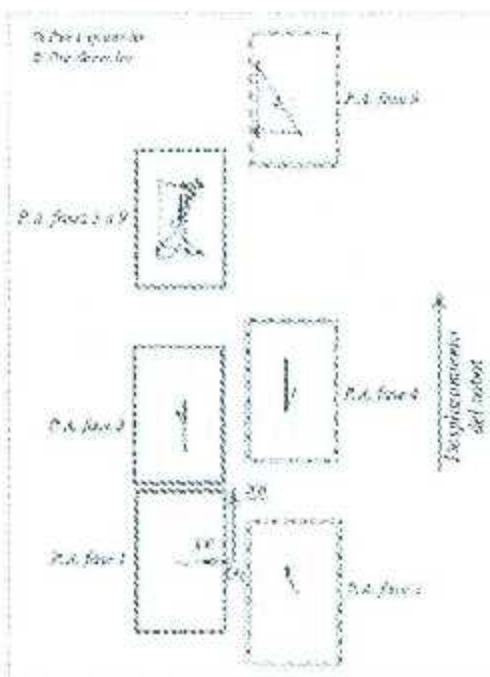


Figura 6.3 Sucesivas huellas de los pies de apoyo y rutas del ZMP durante los movimientos del robot. Vista superior. (P.A: pie apoyado)

6.2.2 Comportamiento del Punto de Momento Cero

Si bien, las rutas del ZMP se vuelven más largas en la medida en que los pasos del pie libre son más grandes, y la velocidad de la pelvis es mayor, en las fases de movimiento 1 a 4 que corresponden a la etapa de carrera, como se aprecia en la Figura 6.3. Se observa que el ZMP no abandona la planta del pie de apoyo.

Es necesario advertir que las dimensiones de las plantas de los pies del robot *HRP-4* modelado en *OpenHRP3* son mayores que las reales. Se hizo esta consideración con el propósito de facilitar la permanencia del ZMP en la planta del pie y, consecuentemente, garantizar el equilibrio dinámico de robot en cada fase de los movimientos propuestos, toda vez que este análisis es de carácter preliminar y, en consecuencia, los parámetros del movimiento aún no han sido optimizados.

En las fases 5 a 9, la dinámica del robot es más compleja ya que en dichas fases se produce el swing de la perna que impacta el balón, el pateo del balón, el reequilibrio y la detención del robot después de la patada. En consecuencia, las rutas del *ZMP* son más irregulares. No obstante, el *ZMP* permanece dentro de la huella del pie de apoyo, que es el izquierdo. En la fase 9 y última, el área del polígono de sustentación se incrementa dado que en esta fase el robot es soportado por los dos pies; el polígono de sustentación es la base sobre la cuál descansa el cuerpo representada por una área geométrica, en este caso las suelas de los pies del humanoide. La complejidad de las rutas del *ZMP* en estas últimas fases justifica la decisión de aumentar las dimensiones de los pies del robot.

Los resultados que se obtuvieron del análisis del comportamiento del *ZMP* durante la ejecución de los movimientos propuestos para el robot confirman la observación efectuada en la simulación física de los movimientos del robot virtual en el paquete OpenHRP-3, en el sentido de que el humanoide conserva el equilibrio al realizar la mencionada tarea. Esta observación valida los movimientos propuestos. Esta validación está condicionada, ciertamente, a la utilización de pies del humanoide con dimensiones mayores a las que posee el robot real. Ello no le resta utilidad a los resultados de nuestro estudio toda vez que éste es de carácter preliminar, en el cual no se consideraron las componentes de las fuerzas a las que está expuesto el robot fuera del plano sagital, ni se ha realizado ningún esfuerzo de optimización de los parámetros que determinan los movimientos del HRP-4. La consideración de estos dos efectos ayudará a reducir las dimensiones de los pies del robot. En estudios futuros deberán incorporarse tales efectos.

7. Conclusión

En este trabajo de tesis se efectuó el modelado cinemático, la simulación dinámica y el análisis del equilibrio de un robot humanoide durante la realización de una acción deportiva que consiste en la ejecución de una patada a un balón de fútbol soccer en reposo. Los movimientos que se propusieron para el robot fueron inspirados en los que realiza un jugador humano de fútbol al ejecutar un *penalty*.

La metodología considerada para la realización de este trabajo incluyó la obtención de los modelos cinemáticos del robot, la creación y animación de un modelo geométrico de alambre en el paquete de software *Matlab*, la creación de un modelo físico virtual en el paquete de software *OpenHRP-3*, la edición de un programa controlador del modelo virtual, la simulación de los movimientos del modelo virtual en *OpenHRP-3*, la generación de las rutas del *Punto de Momento Cero* en las plantas de los pies de apoyo y la validación del equilibrio del robot.

En el marco de la metodología mencionada se evaluaron las características de diferentes robots humanoides candidatos, existentes en la actualidad, y se seleccionó el modelo *HRP-4*, de la compañía Kawada. Este robot posee la arquitectura y la movilidad requerida para la ejecución de la acción deportiva de nuestro interés. Otra de las características interesantes de este robot se refiere a la tecnología *RT (Robot Technology)* que utiliza, y la disponibilidad de dicha tecnología de manera gratuita, en particular del paquete *OpenHRP3* y de sus protocolos de comunicación.

El modelado cinemático del robot *HRP-4* implicó la resolución de los problemas directo e inverso de posición. El modelo cinemático directo de posición permite determinar las posiciones de todos los eslabones del robot en cada instante durante su movimiento, partiendo de un comportamiento definido de las variables articulares en función del tiempo. Por otro lado, mediante el modelo cinemático inverso de posición se determinan las trayectorias articulares del robot a partir de las posturas especificadas de la pelvis y el pie oscilante en cada paso. Ambos modelos cinemáticos fueron fundamentales en el estudio realizado; el modelo inverso para la generación de las consignas articulares del robot, y el modelo directo para la visualización preliminar de los movimientos del robot mediante un modelo de alambre en el paquete *Matlab*.

Por otra parte se preparó el ambiente de simulación virtual en el paquete *OpenHRP3*. Se creó un modelo virtual con las características físicas del robot humanoide *HRP-4*; para ello se desarrollaron las piezas del modelo virtual en un paquete capaz de crear objetos del tipo *.vrml* (en este caso se utilizó *SolidWorks*), para posteriormente ensamblarlas virtualmente en la interfaz gráfica del paquete *OpenHRP3* llamada *GrxUI*. A cada pieza del ensamblaje se le asignaron sus propiedades físicas (pesos, centro de masa, coeficientes de fricción, etc.), y se configuraron los canales de comunicación entre el simulador y el programa controlador del modelo virtual.

Los movimientos que se propusieron para el humanoide se inspiraron en los de un jugador humano. Un análisis de estos movimientos permitió especificar las sucesivas posturas deseadas de la pelvis, los pies, y el torso en las diferentes etapas de movimiento del robot, tanto en la carrera de éste como en la ejecución del *swing* final para el pateo del balón. Las leyes de movimiento propuestas en la mayor parte de los eslabones del robot son de tipo cicloidal (movimientos cicloidales completos y semi-cicloidales). Soamente el movimiento de la pelvis en dirección del eje *x* fue uniformemente acelerado. Las características de estas leyes de movimiento contribuyeron de manera significativa al equilibrio del robot, antes y después de la patada.

A partir de los movimientos especificados de esta manera, aplicando el modelo inverso de posición, se generaron los archivos de trayectorias articulares necesarios para la simulación del modelo virtual.

Las simulaciones en el modelo virtual, finalmente, permitieron observar los movimientos del robot, y validar tanto visualmente (gracias a la simulación física en OpenHRP-3) como analíticamente (gracias a las trayectorias del ZMP) la ejecución exitosa de la tarea del robot: sin que éste pierda el equilibrio.

Las principales contribuciones de este trabajo de tesis son las siguientes:

- La especificación de movimientos estables de un humanoide, en el plano sagital, que facilitan la acumulación de energía cinética en el robot, con lo que se dota a éste de la capacidad de disparar el balón con una mayor potencia y de aumentar, por lo tanto, la probabilidad de anotar un gol. Al conocimiento del autor de esta tesis, en la literatura científica no existe ningún trabajo que proponga movimientos con estas características para el pateado de un balón por un robot humanoide. Uno de los atributos importantes de los movimientos propuestos se refiere al hecho de que se caracterizan mediante ecuaciones cerradas en función de un número reducido de parámetros. Gracias a esto, se facilitará la formulación a futuro de un problema de optimización orientado a la determinación de los valores de dichos parámetros que permitan minimizar el tamaño de los pies. Esta formulación deberá considerar movimientos del robot en el espacio 3D.
- La integración de una plataforma experimental de software para el análisis de los movimientos del robot HRP-4, que es un humanoide cuyo estudio se ha generalizado en múltiples centros de investigación en todo el mundo y que, en consecuencia, se ha convertido en un estándar para la investigación en planificación de movimientos y en control de humanoides. Gracias a las características de esta plataforma en tecnología RT, ésta facilitará la realización de experimentos con el robot HRP-4.

REFERENCIAS

- [1]. International Federation of Robotics. IFR International Federation of Robotics.
<http://www.ifr.org/>(Consultado el 3 de Noviembre de 2013)
- [2]. Hobbelen, D. G., & Wisse, M. (2007). Limit Cycle Walking. En M. Hackel, Humanoid Robots, Human-like Machines (pp. 277-294). Itach.
- [3]. Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., y otros. (2001). Planning Walking Patterns for a Biped Robot. IEEE Transactions on robotics and automation, vol. 17, no. 3 ,pp. 280-289.
- [4]. Nuñez, J., Olvera, L., & Pámanes, J. (2011). Simulation and Experimentation on Walking of the Bioloid Humanoid Robot. Proceedings of the 11th World Congress on Theory of Machines and Mechanisms. Guanajuato, México.
- [5]. Arias, L., Olvera, L., Pámanes, J., & Nuñez, J. (2012). Patrón de Marcha 3D de Tipo Cicloidal para Humanoides y su Aplicación al Robot Bioloid.
- [6]. Kronander, K., Khansari-Zadeh, M. S., & Billard, A. (2011). Learning to Control Planar Hitting Motions in a Minigolf-like Task. Intelligent Robots and Systems (IROS), pp.710-717. IEEE
- [7]. Behnke, S., Müller, J., & Schreibe, M. (2006). Toni: A Soccer Playing Humanoid Robot. RoboCup-2005: Robot Soccer World Cup IX ,pp. 59- 70, Lecture Notes in Artificial Intelligence. LNAI 4020, Springer.
- [8]. Friedmann, M., Kissner, J., Petters, S., Thomas, D., von Stryk, O., & Sakamoto, H. (2006). Versatile, high-quality motions and behavior control. Proceedings of the Workshop on Humanoid Soccer Robots of the 2006 IEEE-RAS International Conference on Humanoid Robots, (pp. 9-16). Genoa, Italy.

- [9]. Xu, Y., & Mellmann, H. (2010). Adaptive Motion Control: Dynamic Kick for a Humanoid Robot. Berlin, Germany. KI 2010: Advances in Artificial Intelligence Lecture Notes in Computer Science Volume 6359, 2010, pp 392-399
- [10]. Ferreira, R., Reis, L. P., Moreira, A. P., & Lau, N. Development of an Omnidirectional Kick for a NAO Humanoid Robot. Advances in Artificial Intelligence – IBERAMIA 2012 Lecture Notes in Computer Science Volume 7637, 2012, pp 571-580
- [11]. Müller, J., Laue, T., & Röfer, T. (2011). Kicking a Ball – Modeling Complex Dynamic Motions for Humanoid Robots. RoboCup 2010, pp 109-120.
- [12]. Kawada Industries, Inc. (2010). sitio web de Kawada Industries, Inc. <http://global.kawada.jp/mechatronics/hrp4.html> (Consultado el 11 de Noviembre de 2013, de)
- [13]. Khalil, W., & Kleifinger, M. (1986). A new geometric notation for open and closed-loop robots. Proceedings of the IEEE International Conference on Robotics and Automation pp 1174-1180.
- [14]. Paul, R. (1981). Robot Manipulators: Mathematics, Programming, and Control. Massachusetts Institute of Technology, 1981.
- [15]. OpenHRP Team, H. R. Sitio oficial de OpenHRP3, <http://www.openhrp.jp/openhrp3/en/about.html> (Consultado el 22 de Noviembre de 2013)
- [16]. Nakaoka, S. AIST., http://www.aist.go.jp/aist_e/aist_today/2008_30/pdf/2008_30_013.pdf (Consultado el 23 de Noviembre de 2013)
- [17]. OpenHRP Team, H. R. OpenHRP3 Official Site., UserInterface: <http://www.openhrp.jp/openhrp3/en/overview.html> (Consultado el 23 de Noviembre de 2013)
- [18]. OpenHRP Team, H. R. OpenHRP3 Official site., Controller Building Guide : http://www.openhrp.jp/openhrp3/en/controller_pd.html (Consultado el 23 de Noviembre de 2013)

- [19]. National Institute of Advanced Industrial Science and Technology, Japan. (2010). OpenRTM-aist.. OpenRTM-aist official website
<http://openrtm.org/openrtm/en/node/629> (Consultado el 24 de Noviembre de 2013)
- [20]. Gheid, N. (2010). Kinematic comparison of successful and unsuccessful instep kick in indoor soccer. American Journal of Applied Sciences vol. 7 .
- [21]. Scurr, J., & Hall, B. (2009). The effects of approach angle on penalty kicking accuracy and kick kinematics with recreational soccer players. Journal of Sports Science and Medicine (2009) vol. 8 , pp. 230-234.
- [22]. Acero, J., & Albarracín, J. (2005). Biomecánica bidimensional (2d) del tiro libre preferencial en fútbol: un modelo integral experimental.
- [23]. Basumatary, S. (1998). Biomechanical analysis of instep kick in soccer. Tesis M.A.S Victoria university of technology p.155
- [24]. Youtube. (2010). Slow motion top corner penalty kick,
<http://www.youtube.com/watch?v=-qADnRqbS2Q> (Consultado el 27 de Octubre de 2012)
- [25]. Reyes, V., Pámanes, J., & Nuñez, J. (2011). Análisis de la marcha del robot humanoide bioloid en una ruta circular con parámetros óptimos. Memorias del XIII Congreso Mexicano de Robótica (COMRob 2011). Universidad Autónoma de San Luis Potosí. Matehuala, San Luis Potosí.
- [26]. Zúñiga, C., Pámanes, J., Nuñez, V., & Arias, L. (2010). Optimización de la marcha de robots bípedos: Aplicación al robot Bioloid. Memorias del XII Congreso Mexicano de Robótica (COMRob 2010). Universidad Autónoma de Sinaloa y Universidad Politécnica de Sinaloa. Mazatlán, Sinaloa.
- [27]. Pámanes, J., & Arias, L. (2009). Modelado cinemático del robot humanoide Bioloid. Reporte Interno. Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Coahuila. Torreón, Coahuila.
- [28]. National Institute of Advanced Industrial Science and Technology, Japan. (2010). OpenRTM-aist., What is RT-Middleware?:

<http://www.openrtm.org/openrtm/en/content/what-rt-m-iddleware-0> (Consultado el 3 de Diciembre de 2013)

- [29]. Vukobratovic, M., & Borovac, B. (2004). zero-moment point — thirty five years of its life. *International Journal of Humanoid Robotics* ,pp 157-173.
- [30]. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa H.. (2003). Biped walking pattern generation by using preview control of zero-moment point. *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*. Taipei, Taiwan.
- [31]. Vukobratovic, M., Borovac, B., & Surcilovic, D. (2001). Zero-Moment Point - Proper interpretation and new applications. *Proceedings of the IEEE RAS International Conference on Humanoid Robots* ,pp 237-244.
- [32]. Vukobratovic, M., & Juricic, D. (1969). Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*, Vol. 16, No.1 , pp. 1-6.
- [33]. Reyes, V. G. (2012). Optimización de la Marcha Omnidireccional del Robot Humanoide Bioloid. Tesis de maestría, Instituto Tecnológico de la Laguna, Torreón, Coahuila.

Código para el programa generación de trayectorias

En esta sección añadiremos el código del programa generación de trayectorias, este está desarrollado en un script de *Matlab* y genera las consignas articulares para el humanoide.

Breve descripción del programa.

1. Declaración de variables. Se declaran dimensiones físicas y tiempos a utilizar.
2. Movimientos y coordenadas del robot. Ubicación espacial de el robot en sus fases
3. Modelo directo e inverso. A partir de una posición deseada del órgano terminal se obtienen las consignas articulares.
4. Visualización del modelo. Representación grafica para el usuario.
5. Movimientos especiales. En el movimiento oscilatorio, las articulaciones correspondientes a la tibia y fémur tienen movimientos cicloidales especiales.
6. Creación de archivos con las consignas articulares. Se guardan y exportan los datos obtenidos

Generación de trayectorias

demostración de modelo cinemático

1---Modicas---1

clear

3---Modicas Plus---

```
m1 =21.743:828;
d1 =-23:878;
r5 =-0.69437:077:813;
r6 =-11.7558:813:5;
d7 =-3.220376331:820;
s(mid)=7.77057951:2:810;
r7 =0;
r8 =-5.850324077:810;
d10 =25:827;
d11 =13.743:828;
k10x =-40;
```

3---Modicas praxis---

```
r10=7:0:3;
r10=0;
d10=13;
k10c=40;
T13=d(mid)+3;
T17=0;
d18=13;
m=0;
C=0;
d19=25;
d20=11;
d26=pi*(10*T+9*T^2);
ang5fma=0.3*(8, 8);
```

meetroscotales=7:10;

1---1100:1:40:1/2) 0:1:1000

```
T0=100:8/5:1.10sec;
T1=100:8/5:42.275: 30.1sec;
T4=100:8/5:821.75: 80.1sec;
T2a=0;
T2c=0;
T3a=0;
T3c=0;
```

```
T2a=T2a+T2a+T2a;
a1=atan(tan(0.1));
a2=atan(tan(0.1));
a3=atan(tan(0.1));
a4=atan(tan(0.1));
a5=atan(tan(0.1));
a6=atan(tan(0.1));
a7=atan(tan(0.1));
a8=atan(tan(0.1));
a9=atan(tan(0.1));
a10=atan(tan(0.1));
a11=atan(tan(0.1));
a12=atan(tan(0.1));
a13=atan(tan(0.1));
a14=atan(tan(0.1));
a15=atan(tan(0.1));
a16=atan(tan(0.1));
a17=atan(tan(0.1));
a18=atan(tan(0.1));
a19=atan(tan(0.1));
a20=atan(tan(0.1));
a21=atan(tan(0.1));
a22=atan(tan(0.1));
a23=atan(tan(0.1));
a24=atan(tan(0.1));
a25=atan(tan(0.1));
a26=atan(tan(0.1));
a27=atan(tan(0.1));
a28=atan(tan(0.1));
a29=atan(tan(0.1));
a30=atan(tan(0.1));
a31=atan(tan(0.1));
a32=atan(tan(0.1));
a33=atan(tan(0.1));
a34=atan(tan(0.1));
a35=atan(tan(0.1));
a36=atan(tan(0.1));
a37=atan(tan(0.1));
a38=atan(tan(0.1));
a39=atan(tan(0.1));
a40=atan(tan(0.1));
a41=atan(tan(0.1));
a42=atan(tan(0.1));
a43=atan(tan(0.1));
a44=atan(tan(0.1));
a45=atan(tan(0.1));
a46=atan(tan(0.1));
a47=atan(tan(0.1));
a48=atan(tan(0.1));
a49=atan(tan(0.1));
a50=atan(tan(0.1));
a51=atan(tan(0.1));
a52=atan(tan(0.1));
a53=atan(tan(0.1));
a54=atan(tan(0.1));
a55=atan(tan(0.1));
a56=atan(tan(0.1));
a57=atan(tan(0.1));
a58=atan(tan(0.1));
a59=atan(tan(0.1));
a60=atan(tan(0.1));
a61=atan(tan(0.1));
a62=atan(tan(0.1));
a63=atan(tan(0.1));
a64=atan(tan(0.1));
a65=atan(tan(0.1));
a66=atan(tan(0.1));
a67=atan(tan(0.1));
a68=atan(tan(0.1));
a69=atan(tan(0.1));
a70=atan(tan(0.1));
a71=atan(tan(0.1));
a72=atan(tan(0.1));
a73=atan(tan(0.1));
a74=atan(tan(0.1));
a75=atan(tan(0.1));
a76=atan(tan(0.1));
a77=atan(tan(0.1));
a78=atan(tan(0.1));
a79=atan(tan(0.1));
a80=atan(tan(0.1));
a81=atan(tan(0.1));
a82=atan(tan(0.1));
a83=atan(tan(0.1));
a84=atan(tan(0.1));
a85=atan(tan(0.1));
a86=atan(tan(0.1));
a87=atan(tan(0.1));
a88=atan(tan(0.1));
a89=atan(tan(0.1));
a90=atan(tan(0.1));
a91=atan(tan(0.1));
a92=atan(tan(0.1));
a93=atan(tan(0.1));
a94=atan(tan(0.1));
a95=atan(tan(0.1));
a96=atan(tan(0.1));
a97=atan(tan(0.1));
a98=atan(tan(0.1));
a99=atan(tan(0.1));
a100=atan(tan(0.1));
```

h=atan(tan(0.1));

```
L1x=0.150*1000*(62.5/895);
L2x=0.11*1000*(62.5/895);
L3x=0.48*1000*(62.5/895);
L4x=0.608*1000*(62.5/895);
L5x=0.7*1000*(62.5/895);
```

ang1=atan(pi/2+0.145);

ang2=atan(pi/3+0.102);

ang3=atan(pi/6+0);

L1x=0.301*1000*(62.5/895);

L2x=0.473*1000*(62.5/895);

L3x=0.475*1000*(62.5/895);

L4x=0.475*1000*(62.5/895);

L5x=0*1000*(62.5/895);

L1c=0;

L2c=0;

L3c=0;

```

i2xpp=(L2x/(1100*(57.5/895)))/(T2*ajustetiempo2/Tajustetiempo);
v12=L2xpp*(Tajustetiempo)/v1;
L2x=0;
L3xpp=L2xpp*(0.33-vf1*(T3*ajustetiempo3)/(T2*ajustetiempo2+(T3*ajustetiempo3)/2))&v;
&v1x=(1+L3xpp*(T3*ajustetiempo3)/(Tajustetiempo2/2)-vax2p(T2)*Tajustetiempo2)/
v13=L3xpp*(T3*ajustetiempo3)/v13;
L4x=0;
L4xpp=L2xpp*(0.68-vf1*(T4*ajustetiempo4)/(T3*ajustetiempo3+(T4*ajustetiempo4)/2))&v&v;
&v14=(1+L4xpp*(T4*ajustetiempo4)/(Tajustetiempo3/2)-vax2p(T3)*Tajustetiempo3)/
v14=L4xpp*(T4*ajustetiempo4)/v14;
L5x=0;
L5xpp=L3xpp*(10.075-vf1*(T5*ajustetiempo5)/(T4*ajustetiempo4+(T5*ajustetiempo5)/2))&v;
&v15=L5xpp*(T5*ajustetiempo5)/(Tajustetiempo4/2)-vax2p(T4)*Tajustetiempo4);
L5x=L5xpp*(T5*ajustetiempo5)/v15;
L6x=0;
L6xpp=L5xpp*(0.1-vf1*(T6*ajustetiempo6)/(T5*ajustetiempo5+(T6*ajustetiempo6)/2))&v;
&v16=L6xpp*(T6*ajustetiempo6)/v16;
d6=rafor(v16/(1+vax1*(T6*ajustetiempo6)))/(T6*ajustetiempo6);
v7=(0.85&vaxpp*(T6*ajustetiempo6)+v16);
L6x=L6xpp;
L5d=ppa10-vf1*(T5*ajustetiempo5)/(T5*ajustetiempo4+(T5d*ajustetiempo5)/2);
L1x=0.448-277;
L2x=-0.025*1000*(42.5/895);
L3x=-0.025*1000*(42.5/895);
L4x=0;
L5x=0;
L6x=0.025*1000*(42.5/895);
L7x=0;
L8x=0;
L9x=-0.025*1000*(42.5/895);
T1x=T1;
T2x=T2;
T3x=T3;
T4x=T4;
T1x1=1/3;
T1x2=2*T1/3;
T1x3=T2/2;
T1x4=T2/2;
T2x1=T3/2;
T2x2=T3/2;
T3x1=T4/4;
T4x1=3*T4/4;
T1x1=T1/3;
T1x2=2*T1/3;
T1x3=T2;
T1x4=T2/2;
T2x1=T2/2;
T2x2=T2;
T3x1=T3/2;
T3x2=3*T3/4;
T4x1=3*T4/4;
T4x2=3*T4/4;
&v1x1=L1x/(T1/3);
v1x1=-0;
&vax=0x/(T1/3);
rad1x=rafor(45.5423); % Este radio es igual a la altura especificada entre 2*pi
&vax1x=v1x1;
rad1x=rafor(45.5423); % Este radio es igual a la altura especificada entre 2*pi
&vax2x=2x/2;
rad1x=rafor(45.5423); % Este radio es igual a la altura especificada entre 2*pi
&vax3x=3x/(2*pi);
rad1x=rafor(45.5423); % Este radio es igual a la altura especificada entre 2*pi
&vax4x=4x/2;
rad1x=rafor(45.5423); % Este radio es igual a la altura especificada entre 2*pi
&vax5x=5x/(2*pi);

```

```

radio3=coses(49.5493); % Este radio es igual a la altura especificada entre P1 y
hes3=124/2;
radio2=hes3; % Este radio es igual a la altura especificada del pie entre 2

suno4=14/(2*pi);
radio4=coses(43.5493); % Este radio es igual a la altura especificada entre P1 y
hes4=142/2;
radio4=hes4; % Este radio es igual a la altura especificada del pie entre 2

suno5=15/(2*pi);
radio5=coses(43.5493); % Este radio es igual a la altura especificada entre P1 y
hes5=152/2;
radio5=hes5; % Este radio es igual a la altura especificada del pie entre 2

Delta1100=0;
Delta1120=10;
Delta1113=0; %Brazo derecho
Delta1114=-90; %Brazo izquierdo
Delta1115=0; %Codo
Delta1116=0; %Codo izquierdo
Delta1117=0; %Codo
Delta1118=0; %Codo

Delta2113=0;
Delta2114=0; %Brazo derecho
Delta2115=0; %Codo
Delta2116=0; %Codo izquierdo
Delta2117=0; %Codo
Delta2118=0; %Codo

Delta3113=0;
Delta3114=0; %Brazo derecho
Delta3115=0; %Codo
Delta3116=0; %Codo izquierdo
Delta3117=0; %Codo
Delta3118=0; %Codo

Delta4113=0;
Delta4114=0; %Brazo derecho
Delta4115=0; %Codo
Delta4116=0; %Codo izquierdo
Delta4117=0; %Codo
Delta4118=0; %Codo

Delta5113=0;
Delta5114=0; %Brazo derecho
Delta5115=0; %Codo
Delta5116=0; %Codo izquierdo
Delta5117=0; %Codo
Delta5118=0; %Codo

Delta6113=0;
Delta6114=0; %Brazo derecho
Delta6115=0; %Codo
Delta6116=0; %Codo izquierdo
Delta6117=0; %Codo
Delta6118=0; %Codo

Delta7113=0;
Delta7114=0; %Brazo derecho
Delta7115=0; %Codo
Delta7116=0; %Codo izquierdo
Delta7117=0; %Codo
Delta7118=0; %Codo

Delta8113=0;
Delta8114=0; %Brazo derecho
Delta8115=0; %Codo
Delta8116=0; %Codo izquierdo
Delta8117=0; %Codo
Delta8118=0; %Codo

```

```

Defina5b: h18=3:8H2
c:aae= 30:30:
c:bbb= 20:5:85:

Delta5b: h20=30:
Delta5b: h19=0: #branco
Delta5b: h18=0: #branco_direito
Delta5b: h16=0: #93a
Delta5b: h15=0: #H0a
Delta5b: h14=0: #branco_esquerdo
Delta5b: h17=0: #93a
Delta5b: h13=0: #H2
c:Dee= 40:20:
c:Ude=0:4-65:

Delta5b: h20=0:
Delta5b: h19=0: #branco
Delta5b: h13=0: #branco_direito
Delta5b: h14=0: #93a
Delta5b: h15=0: #H0a
Delta5b: h16=0: #branco_esquerdo
Delta5b: h17=0: #93a
Delta5b: h18=0: #H2
c:Dee=0:20:
c:Ude=0:
xemp=13, 866:0:
yemp=20:
zemp=0:
xpp=0:
ypp=0:
zpp=0:
xpa=0.87: #H0a: #U: #096: #500: #53:0: # 12
xpb=0.4-47.886: #4: #5: #13: #66:
yfa=0: #0: #11:
yfb=0: #13, 7568: #12: #5:

roq: ra=aa+1:1:
switch: faces

case 1
  np=1:
  # pedra de caras litor
  for i=1:np
Temp:aa= (1:0 0 0 yemp;
0 1 0 zemp;
0 0 1 xemp;
0 0 0 1);

Temp:aa= (1:0 0 0 xemp;
0 1 0 yemp;
0 0 1 xemp+90a;
0 0 0 1);
# ----pedra de 12 faces-----
ang:pi/3:
if i%2==0:
var:aa=ang*pi-(i/ang):
else:
var:aa=90a+1-deepi*(i/ang):
end
z= -r*pp
z1=0:
z2=0:
z3=0:
z4=0:
z5=0:
z6=0:
z7=0:
z8=0:
z9=0:
z10=0:
z11=0:
z12=0:
z13=0:
ang:aa=deq:aa-(de:ta10)*6+6e:ts2:h*0:1:
q:aa=ad=8g*rad: Delta5b: h18: mag:aa=ad*(17-T1:U2): # 7/(2*pi)*faces(12*pi/3)/(2L+2D):1:

```

```

rmln(1)=1;
hans=1+sqrt(5)*rmln(1)/T1;
knois=cos(1) sin(1)Forme
k xcal(1)=L10ax;

Equivalente cilindro

sax(1)=110ax*(110axp/2)+((10*a*(a*settempono)+3)/(3*(T1*a*(a*settempono)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
+*(s)/(2*p)+1)*ajustetiempono)/T1*a*(a*settempono))-((11*a*(a*settempono)/(4*(p)*3)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
sax(1)=110axp/2)*1;
(T1*a*(a*settempono)/(2*(p)*3)+1+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)+(T1*a*(a*settempono)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
11)*cos(10*pi*(1+(T1*a*(a*settempono)/(T1*a*(a*settempono)+1)*sin(10*pi*(1+(T1*a*(a*settempono)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
dxypp(1)=110axp*(110ax*(a*settempono)/(11*a*(a*settempono))+
T1/(2*(p)*3)*sin(2*pi*(1+(T1*a*(a*settempono)/(T1*a*(a*settempono)+1)*sin(10*pi*(1+(T1*a*(a*settempono)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
sax(1)=sqrt(11*(100*(100)*2)*2*55);
sax(1)=sqrt(110000)*sin(10*pi*(1+(T1*a*(a*settempono)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
xp=exp(-sax(1));
yp=yp0;
x=exp(sax(1));

desp=exp(0)*100*(cos(10*pi*(1+(T1*a*(a*settempono)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
despady=0;
despax=0)*100*(sin(10*pi*(1+(T1*a*(a*settempono)+(1+(T1*a*(a*settempono)-2)/(4*(p)*3)
xp=exp(-despax);
yp=yp+despady;
xp=exp(-despax);

sh20=-ydespax;
T1210=cos(T120) 0 sin(T120) 0;
0 1 0 0;
sin(T120) 0 cos(T120) 0;
0 0 0 1;
T121ax=[ 0 0 0+despax;
0 0 0+despady;
0 0 1 0+despax;
0 0 0 1];
T121ax2=exp(sax)*T121ax2;
T12=[t11 t12 t13 xp;
t21 t22 t23 yp;
t31 t32 t33 ax;
0 0 0 1];

tax(1)=t;

mta=2*pi*(a*(1)/T1;
sax(1)=radius*(cos(mta)+i*sin(mta));
sax(1)=radius*(1-cos(mta));
xi=xi+tax(1);
yf=yf+;
zf=zf+tax(1);

T1=[ 0 0 0 mta;
0 0 0 -yf;
0 0 0 mta;
0 0 0 1];

axp=exp(2*cos(0/2)*t-2045/2);
T120=exp(sax)*T120*cos(2*pi*(1)/T1);

xm=-t11*xp-t21*yp-t31*ap;
ym=-t12*xp-t22*yp-t32*ap;
zm=-t13*xp-t23*yp-t33*ap;

T12=[t11 t21 t31 xm;
t12 t22 t32 ym;
t13 t23 t33 zm;
0 0 0 1];

VP=exp(T12);
T12=exp(T12);
T12=T12*cos(0/2)*T12;
xpx=T12(1,1);
xpy=T12(1,2);
xpz=T12(1,3);
npx=T12(1,4);

```

```

r1px=TP12(2,2);
r1py=TP12(2,3);

r1px=TP12(3,2);
r1py=TP12(3,3);
r1pz=TP12(3,3);

r1px=TP12(4,2);
r1py=TP12(4,3);
r1pz=TP12(4,3);

!----- f1a
L0=1;
r0=(-t21*x0+yp*(t21 - (-xp+t23*x6)*t21+(t21*t22+t23*t21+t21)*(-d*mid))/
L0+(-t21*x0+yp*(-d*mid)+t22+(-xp+t23*x6)+t22;
th0=atan2(L0*a6,-B6*b5);

r1=r31*t2*cos(th6)+t22*t2*sin(th6)+t32*(d7*id)+t33*t2*ep;
h1=(t2*t2*sin(th6)-t21*t2*cos(th6)+t22*d7*mid+t23*t2*d7*mid*yp;
R1=1;
th1=atan2(h1*a1,-B1*b1);

r2=(-t21*x0+yp*(t21 - (-xp+t23*x6)*t21+(t21*t22+t23*t21+t21)*(-d*mid))/
L0+(-t21*x0+yp*(-d*mid)+t22+(-xp+t23*x6)+t22;
th0=atan2(L0*a6,-B6*b5);

x14=(-t22*t2*cos(th1)+t23*t2*sin(th1)+t21*cos(th6)+t22*cos(th1)*cos(th6)+t23*sin(th1)*t26*t25;
x12=t22*cos(th1)+t23*t2*sin(th1)+(-d7*mid+xp)*cos(th6)+t23*t26*ep*sin(th1);
x14=(x12*sin(th6)+t21*cos(th6)+t25+t22*(-d7*mid)+t23*t26*ep;
r14=x14/r11;
x14c=x14/r11;
g=(1/2)*t21*(1+(d7*2)-d4*2)-t21*t21;
m=(1/2)*t21*(1-(d7*2)+d4*2)-(1/2)*t21;
b=sgn(t21*(2)-t21*(2));
r1=1;
r1ax=t21*x14c-g*t14c;
r1ay=t21*y14c-g*t14c;
r1ax=t14c-r1ax;
r1ay=t14c-r1ay;
th1ch=atan2(r1ay,r1ax);
th1ch=atan2(r1ay,r1ax);
r1c=t14c-r1ax;
r1s=t14c-r1ay;
x13=(t14c*cos(th6)+t12*cos(th6)+t12*sin(th6);
x13=(t14c*cos(th6)+t12*cos(th6);
L1234=atan2(x13,r14);
th0=th234-th0+pi*0.5;

!----- f1b
a=-d7*yp*t2*cos(d7*mid)-d7*yp*t2*ep;
b=-d7*yp*t2*cos(d7*mid)-d7*yp*t2*ep;
F=1;
th2=atan2(b/a, a/t2);
r6=ep*t2;
r6=(-d7*mid)*cos(th7)-sin(th7)*d7*ep*cos(th7)*ep;
L0=1;
L0=atan2(L0*a6,-B6*b5);
r12x=sin(th7)*cos(th6)*ep*cos(d7*mid)+t21*cos(th6)+t22*ep*cos(th6)+t23*ep*cos(th6)*(-d7*mid);
r12y=cos(th7)*d7*ep*cos(th6)+t21*cos(th6)+t22*ep*cos(th6)+t23*ep*cos(th6)*(-d7*mid)-r0;
r12=sqrt((r12x^2)+(r12y^2));
r12x=r12x/r12;
r12y=r12y/r12;
g=(d7*mid)-d7*mid*(r12x^2)/(2*B6);
h1=sgn((d7*mid)-g*t2);
r9=1;
r11x=d5*th0*(r12x-g*t2);
r11y=g*(r12x-g*t2)+r12y;
r11x=r12x-r11x;
r11y=r12y-r11y;
th1ch=atan2(r11y,r11x);
th1ch=atan2(r11y,r11x);
L12=atan2(L12*a6,-B6*b5);
th12=atan2(L12*a6,-B6*b5);
h12=atan2(h12*a1,-B1*b1);
x11=atan2(x11+th12);

```

```

x11=sin(h19-h11);
g11=cos(h17)+ipermio(h17)*oCpys;
s11=1;
a11=a11/g11+1+e11/gpqr/(a11**2)+(b11**2)-(g11**2);
f11=s11/g11+e11+g11/gqr/(a11**2)+(b11**2)-(g11**2);
th11=atan2(real(f11),real(g11))+gi/epic/(q1*mead-gi/epic);

l=deg2rad(Delta1th11);
ch19=(L/2)/(1-cos(pi*f17/pi));

f1h19=-deg2rad(10*ccmoy);
C19=cos(f1h19);
s19=sin(f1h19);

%trans_avevno

l=deg2rad(Delta1th13);
f1h13=(L/2)/(1-cos(pi*f17/pi));
th13=L;
f13=cos(th13);
s13=sin(th13);
l=deg2rad(Delta1th14);
th14=(L/2)/(1-cos(pi*f17/pi));
C14=cos(th14);
s14=sin(th14);

l=deg2rad(Delta1th15);
f1h15=(L/2)/(1-cos(pi*f17/pi));
C15=cos(th15);
s15=sin(th15);

%trans_avevno

l=deg2rad(Delta1th16);
f1h16=(L/2)/(1-cos(pi*f17/pi));
th16=L;
C16=cos(th16);
s16=sin(th16);

l=deg2rad(Delta1th17);
f1h17=(L/2)/(1-cos(pi*f17/pi));
C17=cos(th17);
s17=sin(th17);

l=deg2rad(Delta1th18);
f1h18=(L/2)/(1-cos(pi*f17/pi));
C18=cos(th18);
s18=sin(th18);

%trans_avevno

Te13=[1 0 0 0;
      0 1 0 -0.7*ico;
      0 0 1 -0.4*o;
      0 0 0 1];

%Te10=100*(100 - sin(th18) 0 0;
      a11(h19) cos(h13) 0 0;
      0 0 1 -1*o;
      0 0 0 1];

%Te10=103*1020*1000;
Te10=Te10*Te13;

%ponto medio de la cara
TeCs=[1 0 0 0;
      0 1 0 0;
      0 0 1 0;
      0 0 0 1];

%centro cara
Te1=Te10*TeCs;
xCa=Te1(1,1);
yCa=Te1(2,1);
zCa=Te1(3,1);

```

-----Fin muestra inversa-----
cc=1/3)

```
§  
§  
§ (Coordenadas de los puntos del pie derecho (x10) en el grupo E)  
§
```

```
xpd0=0;  
ypd0=1;  
zpd0=0;  
pied0=(xpd0 ypd0 zpd0 1)';  
pied0ca=Temp10ca*pied0;  
xpd0=pied0ca(1);  
ypd0=pied0ca(2);  
zpd0=pied0ca(3);  
  
xpd1=1;  
ypd1=1;  
zpd1=0;  
pied1=(xpd1 ypd1 zpd1 1)';  
pied1ca=Temp11ca*pied1;  
xpd1=pied1ca(1);  
ypd1=pied1ca(2);  
zpd1=pied1ca(3);  
  
xpd2=-1;  
ypd2=1;  
zpd2=0;  
pied2=(xpd2 ypd2 zpd2 1)';  
pied2ca=Temp12ca*pied2;  
xpd2=pied2ca(1);  
ypd2=pied2ca(2);  
zpd2=pied2ca(3);  
  
xpd3=-1;  
ypd3=1;  
zpd3=-2;  
pied3=(xpd3 ypd3 zpd3 1)';  
pied3ca=Temp13ca*pied3;  
xpd3=pied3ca(1);  
ypd3=pied3ca(2);  
zpd3=pied3ca(3);  
  
xpd4=-1;  
ypd4=-1;  
zpd4=-2;  
pied4=(xpd4 ypd4 zpd4 1)';  
pied4ca=Temp14ca*pied4;  
xpd4=pied4ca(1);  
ypd4=pied4ca(2);  
zpd4=pied4ca(3);  
  
xpi10=0;  
ypi10=1;  
zpi10=0;  
  
xpi11=1;  
ypi11=1;  
zpi11=0;  
  
xpi12=-1;  
ypi12=1;  
zpi12=0;  
  
xpi13=-1;  
ypi13=1;  
zpi13=-2;  
  
xpi14=-1;  
ypi14=-1;  
zpi14=-2;
```



```

3
*****
*
* Definición de los rotores que delimitan el bob derecho
*
*****

```

```

r1pa=xap1 apd1);
r1py=xap1 apd1);
r1pz=xap1 apd1);
r1pa=xpd1 apd1);
r1pb=xpd1 apd1);
r1pc=xpd1 apd1);
r1pd=xpd1 apd1);
r1pe=xpd1 apd1);
r1pf=xpd1 apd1);
r1pg=xpd1 apd1);
r1ph=xpd1 apd1);
r1pi=xpd1 apd1);
r1pj=xpd1 apd1);
r1pk=xpd1 apd1);
r1pl=xpd1 apd1);
r1pm=xpd1 apd1);
r1pn=xpd1 apd1);
r1po=xpd1 apd1);

```

```

4 Fin pla derecho
*****

```

```

5
*
* Definición de las bobinas
*

```

```

t1=2*as*(ap-ep1+1)-t*(1+as*(ap-ep1+1))/Tfact;
traidos de trochod
tr13 = -tr13+tr1;
tr14 = -tr14+tr1;
tr15 = -tr15+tr1;
tr16 = -tr16+tr1;
tr17 = -tr17+tr1;
tr18 = -tr18+tr1;
tr19 = -tr19+tr1;

```

```

d1=dn(tr1);
d1=cos(tr1);
d1=sin(tr1);
d1=cos(tr2);
d1=sin(tr2);
d1=cos(tr3);
d1=sin(tr3);
d1=cos(tr4);
d1=sin(tr4);
d1=cos(tr5);
d1=sin(tr5);
d1=cos(tr6);
d1=sin(tr6);
d1=cos(tr7);
d1=sin(tr7);
d1=cos(tr8);
d1=sin(tr8);
d1=cos(tr9);
d1=sin(tr9);
d1=cos(tr10);
d1=sin(tr10);
d1=cos(tr11);
d1=sin(tr11);
d1=cos(tr12);
d1=sin(tr12);
d1=cos(tr13);
d1=sin(tr13);
d1=cos(tr14);
d1=sin(tr14);
d1=cos(tr15);
d1=sin(tr15);
d1=cos(tr16);
d1=sin(tr16);
d1=cos(tr17);
d1=sin(tr17);
d1=cos(tr18);
d1=sin(tr18);
d1=cos(tr19);
d1=sin(tr19);
d1=cos(tr20);
d1=sin(tr20);

```

```

000  -x00(t000);
001  -y00(t000);
002  -z00(t000);
003  -x01(t001);
004  -y01(t001);
005  -z01(t001);
006  -x02(t002);
007  -y02(t002);
008  -z02(t002);
009  -x03(t003);
010  -y03(t003);
011  -z03(t003);

```

```

%
%
%  Etiqueta de coordenadas de la perna derecha: "D1"
%

```

```

T000(14, 14, 30)

```

```

x00=01*01*01;
y00=01*00*01;
z00=00*00;
T000=[x00 y00 z00]';
T000=Templaza*T000;
x00=T000(1);
y00=T000(2);
z00=T000(3);

```

```

x01=01*01*01;
y01=01*01*01;
z01=00*00*00;
T001=[x01 y01 z01]';
T001=Templaza*T001;
x01=T001(1);
y01=T001(2);
z01=T001(3);

```

```

x02=01*00*00;
y02=01*00*00;
z02=00*00*00;
T002=[x02 y02 z02]';
T002=Templaza*T002;
x02=T002(1);
y02=T002(2);
z02=T002(3);

```

```

x03=01*00*00;
y03=01*00*00;
z03=00*00*00;
T003=[x03 y03 z03]';
T003=Templaza*T003;
x03=T003(1);
y03=T003(2);
z03=T003(3);

```

```

%
%
%  Etiqueta para la perna izquierda:
%

```

```

%  Punto 06' en T :
%
%  x06p7 = 0;
%  y06p7 = 0;
%  z06p7 = 0;
%  Etiqueta 07' en T :
%  x077 = 0;
%  y077 = 0;
%  z077 = 0;
%
%  Etiqueta 08' en T :
%
%  x0807 = 0;
%  y0807 = 0;
%  z0807 = 0;

```

```

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```


TOT1232 = (-1) * (234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12 - (C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011 * C12
 TOT1233 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1234 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1235 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1236 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1237 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1238 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1239 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1240 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1241 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1242 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1243 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1244 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1245 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1246 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1247 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1248 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1249 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12
 TOT1250 = (-1) * ((C234 * C5 * C67 * C234 * C67) * C8 * C234 * C5 * C67 * C91011) * C12 - (C234 * C5 * C67 * C234 * C67) * C81011 * C12

8 Transformación de los componentes de los vectores de posición de los puntos
 8 de la planta del pie izquierdo, del punto 12 al 51

TOT12 = (TOT1231 TOT1232 TOT1233 TOT1234;
 TOT1231 TOT1232 TOT1233 TOT1234;
 TOT1231 TOT1232 TOT1233 TOT1234;
 0 0 0 0);

TOT1231 = C12 * C1 * C91011 * C12 * C67;
 TOT1232 = -C12 * C91011;
 TOT1233 = -C8 * C12 * C91011 * C12;
 TOT1234 = -C78 * C91011 * C12 * C12 * C67;
 TOT1235 = C12 * C91011;
 TOT1236 = -C12 * C67 * C91011 * C12 * C67;
 TOT1237 = C8 * C91011;
 TOT1238 = C91011;
 TOT1239 = -C8 * C91011;
 TOT1240 = -C12 * C8 * C91011 * C12 * C91011;
 TOT1241 = C8 * C12 * C91011 * C12 * C91011;
 TOT1242 = -C12 * C91011 * C12 * C91011 * C8;
 TOT1243 = -C12 * C91011 * C12 * C91011 * C8;

TOT12 = (TOT1231 TOT1232 TOT1233 TOT1234;
 TOT1231 TOT1232 TOT1233 TOT1234;
 TOT1231 TOT1232 TOT1233 TOT1234;
 0 0 0 0);

TOT1231 = 87;
 TOT1232 = 87;
 TOT1233 = 0;
 TOT1234 = -87;
 TOT1235 = 0;
 TOT1236 = 87;
 TOT1237 = 87;
 TOT1238 = 87;
 TOT1239 = -87;
 TOT1240 = 0;
 TOT1241 = 87;
 TOT1242 = 87;
 TOT1243 = 87;
 TOT1244 = 87;
 TOT1245 = 87;
 TOT1246 = 87;
 TOT1247 = 87;
 TOT1248 = 87;
 TOT1249 = 87;
 TOT1250 = 87;
 TOT1251 = 87;
 TOT1252 = 87;
 TOT1253 = 87;
 TOT1254 = 87;
 TOT1255 = 87;
 TOT1256 = 87;
 TOT1257 = 87;
 TOT1258 = 87;
 TOT1259 = 87;
 TOT1260 = 87;
 TOT1261 = 87;
 TOT1262 = 87;
 TOT1263 = 87;
 TOT1264 = 87;
 TOT1265 = 87;
 TOT1266 = 87;
 TOT1267 = 87;
 TOT1268 = 87;
 TOT1269 = 87;
 TOT1270 = 87;
 TOT1271 = 87;
 TOT1272 = 87;
 TOT1273 = 87;
 TOT1274 = 87;
 TOT1275 = 87;
 TOT1276 = 87;
 TOT1277 = 87;
 TOT1278 = 87;
 TOT1279 = 87;
 TOT1280 = 87;
 TOT1281 = 87;
 TOT1282 = 87;
 TOT1283 = 87;
 TOT1284 = 87;
 TOT1285 = 87;
 TOT1286 = 87;
 TOT1287 = 87;
 TOT1288 = 87;
 TOT1289 = 87;
 TOT1290 = 87;
 TOT1291 = 87;
 TOT1292 = 87;
 TOT1293 = 87;
 TOT1294 = 87;
 TOT1295 = 87;
 TOT1296 = 87;
 TOT1297 = 87;
 TOT1298 = 87;
 TOT1299 = 87;
 TOT1300 = 87;

```

T01721 T01722 T01723 T01724;
T01731 T01732 T01733 T01734;
0 0 0 10;

```

```

4 1012 -T06*T07*T712;

```

```

p00p01 = Template*T012* xp:012 yp:010 zp:012 111;
p00p02 = Template*T012* xp: 12 yp:112 zp:112 111;
p00p03 = Template*T012* xp:212 yp:212 zp:212 111;
p00p04 = Template*T012* xp:312 yp:312 zp:312 111;
p00p05 = Template*T012* xp:412 yp:412 zp:412 111;

```

```

xp10en0 =p00p01(1);
yp10en0 =p00p01(2);
zp10en0 =p00p01(3);

```

```

xp11en0 =p00p11(1);
yp11en0 =p00p11(2);
zp11en0 =p00p11(3);

```

```

xp12en0 =p00p21(1);
yp12en0 =p00p21(2);
zp12en0 =p00p21(3);

```

```

xp13en0 =p00p31(1);
yp13en0 =p00p31(2);
zp13en0 =p00p31(3);

```

```

xp14en0 =p00p41(1);
yp14en0 =p00p41(2);
zp14en0 =p00p41(3);

```

```

4-----Escribo-----5

```

```

4-----Dibajo Izquierdo-----5

```

```

#####

```

```

5 Definir tipo de las rectas que definen el pie Izquierdo:
5
#####

```

```

r0px=(x011 xp10en0);
r0py=(y011 xp10en0);
r0px2=(y011 yp10en0);
r1px=(xp11en0 xp12en0);
r1py=(yp11en0 xp12en0);
r2px=(xp12en0 xp13en0);
r2py=(yp12en0 xp13en0);
r3px=(xp13en0 xp14en0);
r3py=(yp13en0 xp14en0);
r4px=(xp14en0 xp15en0);
r4py=(yp14en0 xp15en0);
r5px=(xp15en0 xp16en0);
r5py=(yp15en0 xp16en0);

```

```

5 Definir tipo de las rectas que definen la piedad Derecha:
5

```

```

r1x=(x04 x01);
r1y=(y04 x01);
r2x=(x04 y01);
r2y=(y04 x01);
r3x=(x04 y01);
r3y=(x04 y01);
r4x=(x04 y01);
r4y=(x04 y01);
r5x=(x04 y01);
r5y=(x04 y01);
r6x=(x04 y01);
r6y=(x04 y01);

```

```

x3x=[x06 x06];
x4x=[x06 x06];
y3y=[y06 y06];
y4y=[y06 y06];
x5x=[x06 x06];
x6x=[x06 x06];
y5y=[y06 y06];
y6y=[y06 y06];
x7x=[x07 x07];
x8x=[x07 x07];
y7y=[y07 y07];
y8y=[y07 y07];
x9x=[x08 x08];
x10x=[x08 x08];
y9y=[y08 y08];
y10y=[y08 y08];
x11x=[x09 x09];
x12x=[x09 x09];
y11y=[y09 y09];
y12y=[y09 y09];
x13x=[x09 x09];
x14x=[x09 x09];
y13y=[y09 y09];
y14y=[y09 y09];
x15x=[x10 x10];
x16x=[x10 x10];
y15y=[y10 y10];
y16y=[y10 y10];
x17x=[x10 x10];
x18x=[x10 x10];
y17y=[y10 y10];
y18y=[y10 y10];
x19x=[x10 x10];
x20x=[x10 x10];
y19y=[y10 y10];
y20y=[y10 y10];
x21x=[x10 x10];
x22x=[x10 x10];
y21y=[y10 y10];
y22y=[y10 y10];
x23x=[x10 x10];
x24x=[x10 x10];
y23y=[y10 y10];
y24y=[y10 y10];
x25x=[x10 x10];
x26x=[x10 x10];
y25y=[y10 y10];
y26y=[y10 y10];
x27x=[x10 x10];
x28x=[x10 x10];
y27y=[y10 y10];
y28y=[y10 y10];
x29x=[x10 x10];
x30x=[x10 x10];
y29y=[y10 y10];
y30y=[y10 y10];
x31x=[x10 x10];
x32x=[x10 x10];
y31y=[y10 y10];
y32y=[y10 y10];
x33x=[x10 x10];
x34x=[x10 x10];
y33y=[y10 y10];
y34y=[y10 y10];
x35x=[x10 x10];
x36x=[x10 x10];
y35y=[y10 y10];
y36y=[y10 y10];
x37x=[x10 x10];
x38x=[x10 x10];
y37y=[y10 y10];
y38y=[y10 y10];
x39x=[x10 x10];
x40x=[x10 x10];
y39y=[y10 y10];
y40y=[y10 y10];
x41x=[x10 x10];
x42x=[x10 x10];
y41y=[y10 y10];
y42y=[y10 y10];
x43x=[x10 x10];
x44x=[x10 x10];
y43y=[y10 y10];
y44y=[y10 y10];
x45x=[x10 x10];
x46x=[x10 x10];
y45y=[y10 y10];
y46y=[y10 y10];
x47x=[x10 x10];
x48x=[x10 x10];
y47y=[y10 y10];
y48y=[y10 y10];
x49x=[x10 x10];
x50x=[x10 x10];
y49y=[y10 y10];
y50y=[y10 y10];
x51x=[x10 x10];
x52x=[x10 x10];
y51y=[y10 y10];
y52y=[y10 y10];
x53x=[x10 x10];
x54x=[x10 x10];
y53y=[y10 y10];
y54y=[y10 y10];
x55x=[x10 x10];
x56x=[x10 x10];
y55y=[y10 y10];
y56y=[y10 y10];
x57x=[x10 x10];
x58x=[x10 x10];
y57y=[y10 y10];
y58y=[y10 y10];
x59x=[x10 x10];
x60x=[x10 x10];
y59y=[y10 y10];
y60y=[y10 y10];
x61x=[x10 x10];
x62x=[x10 x10];
y61y=[y10 y10];
y62y=[y10 y10];
x63x=[x10 x10];
x64x=[x10 x10];
y63y=[y10 y10];
y64y=[y10 y10];
x65x=[x10 x10];
x66x=[x10 x10];
y65y=[y10 y10];
y66y=[y10 y10];
x67x=[x10 x10];
x68x=[x10 x10];
y67y=[y10 y10];
y68y=[y10 y10];
x69x=[x10 x10];
x70x=[x10 x10];
y69y=[y10 y10];
y70y=[y10 y10];
x71x=[x10 x10];
x72x=[x10 x10];
y71y=[y10 y10];
y72y=[y10 y10];
x73x=[x10 x10];
x74x=[x10 x10];
y73y=[y10 y10];
y74y=[y10 y10];
x75x=[x10 x10];
x76x=[x10 x10];
y75y=[y10 y10];
y76y=[y10 y10];
x77x=[x10 x10];
x78x=[x10 x10];
y77y=[y10 y10];
y78y=[y10 y10];
x79x=[x10 x10];
x80x=[x10 x10];
y79y=[y10 y10];
y80y=[y10 y10];
x81x=[x10 x10];
x82x=[x10 x10];
y81y=[y10 y10];
y82y=[y10 y10];
x83x=[x10 x10];
x84x=[x10 x10];
y83y=[y10 y10];
y84y=[y10 y10];
x85x=[x10 x10];
x86x=[x10 x10];
y85y=[y10 y10];
y86y=[y10 y10];
x87x=[x10 x10];
x88x=[x10 x10];
y87y=[y10 y10];
y88y=[y10 y10];
x89x=[x10 x10];
x90x=[x10 x10];
y89y=[y10 y10];
y90y=[y10 y10];
x91x=[x10 x10];
x92x=[x10 x10];
y91y=[y10 y10];
y92y=[y10 y10];
x93x=[x10 x10];
x94x=[x10 x10];
y93y=[y10 y10];
y94y=[y10 y10];
x95x=[x10 x10];
x96x=[x10 x10];
y95y=[y10 y10];
y96y=[y10 y10];
x97x=[x10 x10];
x98x=[x10 x10];
y97y=[y10 y10];
y98y=[y10 y10];
x99x=[x10 x10];
x100x=[x10 x10];
y99y=[y10 y10];
y100y=[y10 y10];

```

```

x=[x00 x00];
y=[y00 y00];
x1=[x01 x01];
y1=[y01 y01];
x2=[x02 x02];
y2=[y02 y02];
x3=[x03 x03];
y3=[y03 y03];
x4=[x04 x04];
y4=[y04 y04];
x5=[x05 x05];
y5=[y05 y05];
x6=[x06 x06];
y6=[y06 y06];
x7=[x07 x07];
y7=[y07 y07];
x8=[x08 x08];
y8=[y08 y08];
x9=[x09 x09];
y9=[y09 y09];
x10=[x10 x10];
y10=[y10 y10];
x11=[x11 x11];
y11=[y11 y11];
x12=[x12 x12];
y12=[y12 y12];
x13=[x13 x13];
y13=[y13 y13];
x14=[x14 x14];
y14=[y14 y14];
x15=[x15 x15];
y15=[y15 y15];
x16=[x16 x16];
y16=[y16 y16];
x17=[x17 x17];
y17=[y17 y17];
x18=[x18 x18];
y18=[y18 y18];
x19=[x19 x19];
y19=[y19 y19];
x20=[x20 x20];
y20=[y20 y20];
x21=[x21 x21];
y21=[y21 y21];
x22=[x22 x22];
y22=[y22 y22];
x23=[x23 x23];
y23=[y23 y23];
x24=[x24 x24];
y24=[y24 y24];
x25=[x25 x25];
y25=[y25 y25];
x26=[x26 x26];
y26=[y26 y26];
x27=[x27 x27];
y27=[y27 y27];
x28=[x28 x28];
y28=[y28 y28];
x29=[x29 x29];
y29=[y29 y29];
x30=[x30 x30];
y30=[y30 y30];
x31=[x31 x31];
y31=[y31 y31];
x32=[x32 x32];
y32=[y32 y32];
x33=[x33 x33];
y33=[y33 y33];
x34=[x34 x34];
y34=[y34 y34];
x35=[x35 x35];
y35=[y35 y35];
x36=[x36 x36];
y36=[y36 y36];
x37=[x37 x37];
y37=[y37 y37];
x38=[x38 x38];
y38=[y38 y38];
x39=[x39 x39];
y39=[y39 y39];
x40=[x40 x40];
y40=[y40 y40];
x41=[x41 x41];
y41=[y41 y41];
x42=[x42 x42];
y42=[y42 y42];
x43=[x43 x43];
y43=[y43 y43];
x44=[x44 x44];
y44=[y44 y44];
x45=[x45 x45];
y45=[y45 y45];
x46=[x46 x46];
y46=[y46 y46];
x47=[x47 x47];
y47=[y47 y47];
x48=[x48 x48];
y48=[y48 y48];
x49=[x49 x49];
y49=[y49 y49];
x50=[x50 x50];
y50=[y50 y50];
x51=[x51 x51];
y51=[y51 y51];
x52=[x52 x52];
y52=[y52 y52];
x53=[x53 x53];
y53=[y53 y53];
x54=[x54 x54];
y54=[y54 y54];
x55=[x55 x55];
y55=[y55 y55];
x56=[x56 x56];
y56=[y56 y56];
x57=[x57 x57];
y57=[y57 y57];
x58=[x58 x58];
y58=[y58 y58];
x59=[x59 x59];
y59=[y59 y59];
x60=[x60 x60];
y60=[y60 y60];
x61=[x61 x61];
y61=[y61 y61];
x62=[x62 x62];
y62=[y62 y62];
x63=[x63 x63];
y63=[y63 y63];
x64=[x64 x64];
y64=[y64 y64];
x65=[x65 x65];
y65=[y65 y65];
x66=[x66 x66];
y66=[y66 y66];
x67=[x67 x67];
y67=[y67 y67];
x68=[x68 x68];
y68=[y68 y68];
x69=[x69 x69];
y69=[y69 y69];
x70=[x70 x70];
y70=[y70 y70];
x71=[x71 x71];
y71=[y71 y71];
x72=[x72 x72];
y72=[y72 y72];
x73=[x73 x73];
y73=[y73 y73];
x74=[x74 x74];
y74=[y74 y74];
x75=[x75 x75];
y75=[y75 y75];
x76=[x76 x76];
y76=[y76 y76];
x77=[x77 x77];
y77=[y77 y77];
x78=[x78 x78];
y78=[y78 y78];
x79=[x79 x79];
y79=[y79 y79];
x80=[x80 x80];
y80=[y80 y80];
x81=[x81 x81];
y81=[y81 y81];
x82=[x82 x82];
y82=[y82 y82];
x83=[x83 x83];
y83=[y83 y83];
x84=[x84 x84];
y84=[y84 y84];
x85=[x85 x85];
y85=[y85 y85];
x86=[x86 x86];
y86=[y86 y86];
x87=[x87 x87];
y87=[y87 y87];
x88=[x88 x88];
y88=[y88 y88];
x89=[x89 x89];
y89=[y89 y89];
x90=[x90 x90];
y90=[y90 y90];
x91=[x91 x91];
y91=[y91 y91];
x92=[x92 x92];
y92=[y92 y92];
x93=[x93 x93];
y93=[y93 y93];
x94=[x94 x94];
y94=[y94 y94];
x95=[x95 x95];
y95=[y95 y95];
x96=[x96 x96];
y96=[y96 y96];
x97=[x97 x97];
y97=[y97 y97];
x98=[x98 x98];
y98=[y98 y98];
x99=[x99 x99];
y99=[y99 y99];
x100=[x100 x100];
y100=[y100 y100];

```



```

brayppr16 = d18*d17;
braxp = d18*d17*d14;
vec6=Templasa2*Te6*(braxx0 brayx0 braxp 0);
braxxp=vec6(1);
brayyp=vec6(2);
braxzpp=vec6(3);
ppr1(1)=braxxp;
ppr1(2)=brayyp;
ppr1(3)=braxzpp;
xTo16=0;
yTo16=r16;
zTo16=0;
vec6=Templasa2*Te6*(xTo16 yTo16 zTo16 1);
xTo16=vec6(1);
yTo16=vec6(2);
zTo16=vec6(3);

xTo17=0;
yTo17=r16;
zTo17=0;
vec7=Templasa2*Te7*(xTo17 yTo17 zTo17 0);
xTo17=vec7(1);
yTo17=vec7(2);
zTo17=vec7(3);
xTo18=d18*d18*d18*d17;
yTo18=r16 * d18*d17;
zTo18=d18*d17*d16;
vec1=Templasa2*Te8*(xTo18 yTo18 zTo18 1);
xTo18=vec8(1);
yTo18=vec8(2);
zTo18=vec8(3);

Templasa3
brax1x=[xTo16 xTo17];
brax1y=[yTo16 yTo17];
brax1z=[zTo16 zTo17];
brax2x=[xTo17 xTo18];
brax2y=[yTo17 yTo18];
brax2z=[zTo17 zTo18];
TTol11=d16*d17*d18 - d18*d17*d18;
TTol12 = d17*d18 - d18*d17;
TTol13 = d16*d17*d18 - d17*d18*d16;
TTol14 = -d16*d17*d18 - d16*d18*d17;
TTol15 = d17*d18 - d17*d18;
TTol16 = 0;
TTol17 = d16;
TTol18 = d16;
TTol19 = d16*d17*d17;
TTol20 = d16 - d18*d17;
TTol21 = -d18*d17*d16;
Tol1 = [TTol11 TTol12 TTol13 TTol14];
TTol22 TTol23 TTol24 TTol25 TTol26;
TTol27 TTol28 TTol29 TTol30;
0 0 0 0;
Te16 = Te6*Te7;

%Las dimensiones para definir la pinta, sean, parte de teoria matricial
%por la coordenada ultima, modificar el se queda
brapp1 = [d18 0 0 1];
brax1= Templasa2*Te18*brapp1;

brapp2 = [d18 0 0 1];
brax2= Templasa2*Te18*brapp2;

brapp3 = [d18-1 0 0 1];
brax3= Templasa2*Te18*brapp3;

brapp4 = [d18 -1 0 1];
brax4= Templasa2*Te18*brapp4;

brapp5 = [d18-1 -1 0 1];
brax5= Templasa2*Te18*brapp5;

%algo aun k giro la pinta
brad1x=[brax1(1) brax2(1)];
brad1y=[brax1(2) brax2(2)];

```

```

bradd12x=(bradd1(2) bradd2(3))
bradd13x=(bradd1(1) bradd3(1))
bradd13y=(bradd1(2) bradd3(2))
bradd13z=(bradd1(3) bradd3(3))
bradd14x=(bradd1(1) bradd4(1))
bradd14y=(bradd1(2) bradd4(2))
bradd14z=(bradd1(3) bradd4(3))
bradd15x=(bradd1(1) bradd5(1))
bradd15y=(bradd1(2) bradd5(2))
bradd15z=(bradd1(3) bradd5(3))
bradd16x=(bradd1(1) bradd6(1))
bradd16y=(bradd1(2) bradd6(2))
bradd16z=(bradd1(3) bradd6(3))

```

Alas dimensiones para definir la matriz

```

Capp1 = [4 4 0 1]';
Capp1= Temp1aa2*Tea1*Capp1;

```

```

Capp2 = [4 -4 0 1]';
Capp2= Temp1aa2*Tea1*Capp2;

```

```

Capp3 = [-1 -1 0 1]';
Capp3= Temp1aa2*Tea1*Capp3;

```

```

Capp4 = [-1 4 0 1]';
Capp4= Temp1aa2*Tea1*Capp4;

```

```

Capp5 = [4 4 0 1]';
Capp5= Temp1aa2*Tea1*Capp5;

```

```

Capp6 = [4 -4 0 1]';
Capp6= Temp1aa2*Tea1*Capp6;

```

```

Capp7 = [0 0 0 1]';
Capp7= Temp1aa2*Tea1*Capp7;

```

```

Capp8 = [4 4 0 1]';
Capp8= Temp1aa2*Tea1*Capp8;

```

Algo para el giro de la matriz

```

Cabr17x=(Cabr1(1) Cabr2(1))
Cabr17y=(Cabr1(2) Cabr2(2))
Cabr17z=(Cabr1(3) Cabr2(3))
Cabr18x=(Cabr1(1) Cabr3(1))
Cabr18y=(Cabr1(2) Cabr3(2))
Cabr18z=(Cabr1(3) Cabr3(3))
Cabr19x=(Cabr1(1) Cabr4(1))
Cabr19y=(Cabr1(2) Cabr4(2))
Cabr19z=(Cabr1(3) Cabr4(3))
Cabr20x=(Cabr1(1) Cabr5(1))
Cabr20y=(Cabr1(2) Cabr5(2))
Cabr20z=(Cabr1(3) Cabr5(3))
Cabr21x=(Cabr1(1) Cabr6(1))
Cabr21y=(Cabr1(2) Cabr6(2))
Cabr21z=(Cabr1(3) Cabr6(3))
Cabr22x=(Cabr1(1) Cabr7(1))
Cabr22y=(Cabr1(2) Cabr7(2))
Cabr22z=(Cabr1(3) Cabr7(3))
Cabr23x=(Cabr1(1) Cabr8(1))
Cabr23y=(Cabr1(2) Cabr8(2))
Cabr23z=(Cabr1(3) Cabr8(3))
Cabr24x=(Cabr1(1) Cabr9(1))
Cabr24y=(Cabr1(2) Cabr9(2))
Cabr24z=(Cabr1(3) Cabr9(3))
Cabr25x=(Cabr1(1) Cabr10(1))
Cabr25y=(Cabr1(2) Cabr10(2))
Cabr25z=(Cabr1(3) Cabr10(3))
Cabr26x=(Cabr1(1) Cabr11(1))
Cabr26y=(Cabr1(2) Cabr11(2))
Cabr26z=(Cabr1(3) Cabr11(3))
Cabr27x=(Cabr1(1) Cabr12(1))
Cabr27y=(Cabr1(2) Cabr12(2))
Cabr27z=(Cabr1(3) Cabr12(3))
Cabr28x=(Cabr1(1) Cabr13(1))
Cabr28y=(Cabr1(2) Cabr13(2))
Cabr28z=(Cabr1(3) Cabr13(3))
Cabr29x=(Cabr1(1) Cabr14(1))
Cabr29y=(Cabr1(2) Cabr14(2))
Cabr29z=(Cabr1(3) Cabr14(3))
Cabr30x=(Cabr1(1) Cabr15(1))
Cabr30y=(Cabr1(2) Cabr15(2))
Cabr30z=(Cabr1(3) Cabr15(3))
Cabr31x=(Cabr1(1) Cabr16(1))
Cabr31y=(Cabr1(2) Cabr16(2))
Cabr31z=(Cabr1(3) Cabr16(3))

```

```

and Templasa2*TeP;

xTeTercel1,41;
yTeTercel1,41;
xTeTercel3,41;

TedTel3x=(xTeTe xTe13);
TedTel3y=(xTeTe yTe13);
TedTel3z=(xTeTe zTe13);
TedTel6x=(xTeTe xTe6);
TedTel6y=(yTeTe yTe6);
TedTel6z=(xTeTe zTe6);

Cabr*Templasa2*Jasa;
ceckx=(xTeTe Cab(1,41));
cecky=(yTeTe Cab(2,41));
ceckz=(xTeTe Cab(3,41));

TMIS=Templasa2*TM*TELSa;
x'=(xTe19(1,41));
y'=(yTe19(2,41));
z'=(zTe19(1,41));

na=(1 0 0);
n1=(0 0 -dTe19);
n2=(0 0 1 0);
n3=(0 0 0 1);
TMPCa=Templasa2*TM*TePa;

xTP=TMPCa(1,4);
yTP=TMPCa(2,4);
zTP=TMPCa(3,4);
!-lines a la cadere--
dP1x=(xTP x19);
dP1y=(yTP y19);
dP1z=(zTP z19);

!-lines Teasa--
TePa=(0 dMid(3 4));
TePz=Templasa2*TeP*TePa;

TePb=(0 dMid(3 4));
TePz=Templasa2*TeP*TePb;

TePc=(0 dMid(3 4));
TePz=Templasa2*TeP*TePc;

TePd1x=(TePd1(1) TePd1(1));
TePd1y=(TePd1(2) TePd1(2));
TePd1z=(TePd1(3) TePd1(3));

TePd2x=(TePd2(1) TePd2(1));
TePd2y=(TePd2(2) TePd2(2));
TePd2z=(TePd2(3) TePd2(3));

TePd3x=(TePd3(1) TePd3(1));
TePd3y=(TePd3(2) TePd3(2));
TePd3z=(TePd3(3) TePd3(3));

TePd4x=(TePd4(1) TePd4(1));
TePd4y=(TePd4(2) TePd4(2));
TePd4z=(TePd4(3) TePd4(3));
if banderablocad==1

figura 02

all 3 data incrementale limpa el crechea de arboriment,de
solo on
! Graficad,plot della ruta del OE:
for j=0:4
    plot3(ppa1(j)-1,ppa1(j)+1,ppa2(j)-1,ppa2(j)+1,ppa3(j)-1,ppa3(j)+1);
end

```

```

plot3(ppon1(1-1),ppon1(2),lyon2(1-1),ppon2(1),lppon1(1-1),ppon1(2))
plot3(ppon1(1-1),ppon1(2),lppon2(1-1),lppon2(1),lppon2(1-1),ppon2(2))

hold on
% Esta línea funciona para poder tener la traza de la ruta
% en cada iteración, la ruta se muestra desde el primer
% punto hasta el actual

end
axis on
plot3(brad1x,brad1y,brad1z,'b-o',brad2x,brad2y,brad2z,'k-o','LineWidth',1.5) % r1x,r1y,r1z,'k-
o',r2x,r2y,r2z,'k','LineWidth',1.5)
plot3(brad1x,brad1y,brad1z,'r','LineWidth',1)
plot3(brad1x,brad1y,brad1z,'r','LineWidth',1)
plot3(brad21x,brad21y,brad21z,'b','LineWidth',1)

plot3(brad21x,brad21y,brad21z,'b-o',brad22x,brad22y,brad22z,'k-o','LineWidth',1) % r2x,r2y,r2z,'k-
o',r2x,r2y,r2z,'k','LineWidth',1.5)
plot3(brad22x,brad22y,brad22z,'r',brad23x,brad23y,brad23z,'r','LineWidth',1)
plot3(brad23x,brad23y,brad23z,'r','LineWidth',1.5)
plot3(brad24x,brad24y,brad24z,'b-o')

plot3(Tod101x,Tod101y,Tod101z,'k-o',Tod102x,Tod102y,Tod102z,'k-o','LineWidth',1.5) % r3x,r3y,r3z,'k-
o',r3x,r3y,r3z,'k','LineWidth',1.5)

plot3(Cabd1x,Cabd1y,Cabd1z,'r',Cabd2x,Cabd2y,Cabd2z,'r',Cabd3x,Cabd3y,Cabd3z,'r',Cabd4x,Cabd4y,Cabd4
z,'r','LineWidth',1.5)

plot3(Cabd2x,Cabd2y,Cabd2z,'k',Cabd3x,Cabd3y,Cabd3z,'k',Cabd4x,Cabd4y,Cabd4z,'k',Cabd5x,Cabd5y,Cabd5
z,'k',Cabd6x,Cabd6y,Cabd6z,'k',Cabd7x,Cabd7y,Cabd7z,'k',Cabd8x,Cabd8y,Cabd8z,'k',Cabd9x,Cabd9y,Cabd9
z,'k','LineWidth',1.5)

plot3(Tord1x,Tord1y,Tord1z,'b',Tord2x,Tord2y,Tord2z,'k',Tord3x,Tord3y,Tord3z,'k',Tord4x,Tord4y,Tord4
z,'k')
plot3(r1px,r1py,r1pz,r2px,r2py,r2pz,r3px,r3py,r3pz,r4px,r4py,r4pz,r5px,r5py,r5pz,r6px,r6py,r6pz,r7px,r7
py,r7pz,r8px,r8py,r8pz,r9px,r9py,r9pz,r10px,r10py,r10pz,r11px,r11py,r11pz,r12px,r12py,r12pz,r13px,r13py,r13
pz,r14px,r14py,r14pz,r15px,r15py,r15pz,r16px,r16py,r16pz,r17px,r17py,r17pz,r18px,r18py,r18pz,r19px,r19py,r19
pz,r20px,r20py,r20pz,r21px,r21py,r21pz,r22px,r22py,r22pz,r23px,r23py,r23pz,r24px,r24py,r24pz,r25px,r25py,r25
pz,r26px,r26py,r26pz,r27px,r27py,r27pz,r28px,r28py,r28pz,r29px,r29py,r29pz,r30px,r30py,r30pz)

%ondax, r1py, r1pz, r2px, r2py, r2pz, r3px, r3py, r3pz, r4px, r4py, r4pz, r5px, r5py, r5pz, r6px, r6py, r6pz, r7
px, r7py, r7pz, r8px, r8py, r8pz, r9px, r9py, r9pz, r10px, r10py, r10pz, r11px, r11py, r11pz, r12px, r12py, r12
pz, r13px, r13py, r13pz, r14px, r14py, r14pz, r15px, r15py, r15pz, r16px, r16py, r16pz, r17px, r17py, r17pz, r18px,
r18py, r18pz, r19px, r19py, r19pz, r20px, r20py, r20pz, r21px, r21py, r21pz, r22px, r22py, r22pz, r23px, r23py, r23
pz, r24px, r24py, r24pz, r25px, r25py, r25pz, r26px, r26py, r26pz, r27px, r27py, r27pz, r28px, r28py, r28pz, r29px,
r29py, r29pz, r30px, r30py, r30pz)

axis([-40,40,-50,50,-20,130])

view(10,-1,0) % Vista lateral

end

tort1(1)=t1;
tort2(1)=t2;
tort3(1)=t3;
tort4(1)=t4;
tort5(1)=t5;
tort6(1)=t6;
tort7(1)=t7;
tort8(1)=t8;
tort9(1)=t9;
tort10(1)=t10;
tort11(1)=t11;
tort12(1)=t12;
tort13(1)=t13;
tort14(1)=t14;
tort15(1)=t15;
tort16(1)=t16;
tort17(1)=t17;
tort18(1)=t18;
tort19(1)=t19;
tort20(1)=t20;
corrigasf(1,1)=t1;
corrigasf(1,2)=t2;
corrigasf(1,3)=t3;
corrigasf(1,4)=t4;
corrigasf(1,5)=t5;
corrigasf(1,6)=t6;
corrigasf(1,7)=t7;

```

```

cos11=cosf(1,10)+t10;
cos12=cosf(1,11)+t11;
cos13=cosf(1,12)+t12;
cos14=cosf(1,13)+t13;
cos15=cosf(1,14)+t14;
cos16=cosf(1,15)+t15;
cos17=cosf(1,16)+t16;
cos18=cosf(1,17)+t17;
cos19=cosf(1,18)+t18;
cos100=cosf(1,20)+t20;

c0ptet1(1)=-t10(1)-pi/31;
c0ptet2(1)=-t11(1);
c0ptet3(1)=-t12(1);
c0ptet4(1)=-t13(1);
c0ptet5(1)=-t14(1)-pi/21;
c0ptet6(1)=-t15(1);
c0ptet7(1)=-t16(1);
c0ptet8(1)=-t17(1)-pi/21;
c0ptet9(1)=-t18(1);
c0ptet10(1)=-t19(1);
c0ptet11(1)=-t20(1);
c0tet12(1)=-t10(1)-pi/21;
c0ptet13(1)=-t11(1)-pi/21;
c0ptet14(1)=-t12(1);
c0ptet15(1)=-t13(1);
c0ptet16(1)=-t14(1)-pi/2;
c0ptet17(1)=-t15(1)-t16(1);
c0ptet18(1)=-t17(1)-t18(1);
c0ptet19(1)=-t19(1);
c0ptet20(1)=-t20(1);

if i==np/2
    i=i+1;
end

end

case 2
    i=i+1;
end

x0=10*(kemp-c011)+abs(x0); % x0-degprod*xemp;
y0=yp-degprod; % y0-degprod;
z0=zp-degprod;
x1=x0+1;
y1=y0+1;
z1=z0+1;
x2=x1;
y2=y1;
z2=z1;
x3=x2;
y3=y2;
z3=z2;

case 3
    i=i+1;
end

x0=kemp-xp+1; % x0=degprod+abs(x0); % x0-degprod+remor;
y0=yp-degprod;
xp=yp-degprod;
x1=x0+1;
y1=y0+1;
x2=x1;
y2=y1;
z2=z1;
y3=y2;
z3=z2;

```

```

zfa(zf);

      case 4
      ...
      end

xpr=xmp*xp-xc11;
ypr=yp-despcady;typ=despcady;
zpr=zp-despcadz;
xmp=xc11;
ypm=yc11;
zpm=zc11;
xia=x2;
yfa=yf;
zfa=zf;

case 3
...
      end

xpr=xmp*xp-xc11/8*(xmp-zc11)-zfa*xp/zxp/8*xpr/xmpr;
ypr=yp-despcady;
zpr=zp-despcadz;
xmp=xc11;
ypm=yc11;
zpm=zc11;
xia=x2;
yfa=yf;
zfa=zf;
(Mppp,Mcbox)=size(cpptet1);
cppte1=0;
cppte121=deg(rad1)-501;
cppte122=deg(rad1)-501;
for veex=1:13
    for vedy=1:Mppx
        switch veex
            case 1
                Kangu1ocpp(vedy,veex)=fady/Mcbox*(tiomporc11)/81ump;
            case 2
                Kangu1ocpp(vedy,veex)=cppte10; %caden
            case 3
                Kangu1ocpp(vedy,veex)=cppte0; %pierna_derecha
            case 4
                Kangu1ocpp(vedy,veex)=cppte16(vedy); %d1_7
            case 5
                Kangu1ocpp(vedy,veex)=cppte1(veex); %c1
            case 6
                Kangu1ocpp(vedy,veex)=cppte4(vedy); %c
            case 7
                Kangu1ocpp(vedy,veex)=cppte1(veex); %c
            case 8
                Kangu1ocpp(vedy,veex)=cppte11(vedy); %a2
            case 9
                Kangu1ocpp(vedy,veex)=cppte1(vedy); %a3
            case 10
                Kangu1ocpp(vedy,veex)=cppte0y; %a1
            case 11
                Kangu1ocpp(vedy,veex)=cppte00; %viena_izquierda
            case 12
                Kangu1ocpp(vedy,veex)=cppte10(vedy); %d1_7
            case 13
                Kangu1ocpp(vedy,veex)=cppte16(vedy); %d2
            case 14
                Kangu1ocpp(vedy,veex)=cppte9(vedy); %c11
            case 15
                Kangu1ocpp(vedy,veex)=cppte17(vedy); %a1
            case 16
                Kangu1ocpp(vedy,veex)=cppte11(vedy); %a2
            case 17
                Kangu1ocpp(vedy,veex)=cppte12(vedy); %a3
            case 18
                Kangu1ocpp(vedy,veex)=cppte0; %a1
            case 19
                Kangu1ocpp(vedy,veex)=cppte10(vedy); %c1caco
            case 20
                Kangu1ocpp(vedy,veex)=cppte11; %c1caxa
            case 21
                Kangu1ocpp(vedy,veex)=cppte10; %c1

```

```

case 23
    Manguloecpp(vecy,vecx)=oppot13(vecy); $orazo_derechd;
case 23
    Manguloecpp(vecy,vecx)=oppot17(vecy); $A3;
case 24
    Manguloecpp(vecy,vecx)=oppot23; $B1;
case 25
    Manguloecpp(vecy,vecx)=oppot13(vecy); $M1;
case 26
    Manguloecpp(vecy,vecx)=oppot10; $H1_1;
case 27
    Manguloecpp(vecy,vecx)=oppot0; $H1_2;
case 28
    Manguloecpp(vecy,vecx)=oppot0; $J1;
case 29
    Manguloecpp(vecy,vecx)=oppot0; $K1;
case 30
    Manguloecpp(vecy,vecx)=oppot13(vecy); $orazo_derechd;
case 31
    Manguloecpp(vecy,vecx)=oppot14(vecy); $B3;
case 32
    Manguloecpp(vecy,vecx)=oppot21; $G1;
case 33
    Manguloecpp(vecy,vecx)=oppot10(vecy); $H1;
case 34
    Manguloecpp(vecy,vecx)=oppot0; $T1_1;
case 35
    Manguloecpp(vecy,vecx)=oppot0; $T1_2;
case 36
    Manguloecpp(vecy,vecx)=oppot0; $C1;
case 37
    Manguloecpp(vecy,vecx)=oppot0; $K1;
case 38
    Manguloecpp(vecy,vecx)=oppot20(vecy); $Cadsramo;
otherwise;
end
end
end

save('gnt\ayacarias\anguloecpp.dat','manguloecpp','-ascii');
comp=cat;
year=cat;
comp=year;

otherwise;
$error 6 $chdir;
end

end

temp1=rest(rest(ol1));
figure(5);
subplot(2,3,1);
plot(temp, (cat1)*100/pi);

title('cat1');
subplot(2,3,2);
plot(temp, (cat2)*100/pi);
title('cat2');
subplot(2,3,3);
plot(temp, (cat3)*100/pi);
title('cat3');
subplot(2,3,4);
plot(temp, (cat4)*100/pi);

title('cat4');
subplot(2,3,5);
plot(temp, (cat5)*100/pi);

title('cat5');
subplot(2,3,6);
plot(temp, (cat6)*100/pi);
title('cat6');

figure(7);
subplot(2,3,1);
plot(temp, (cat12)*100/pi);
title('cat12');
subplot(2,3,2);

```

```

plot(temp, (lat12)*180/pi)
title('ch11')
subplot(2,3,3)
plot(temp,(tet10)*180/pi)
title('ch12')
subplot(2,3,4)
plot(temp,(tet4)*180/pi)
title('ch5')
subplot(2,3,5)
plot(temp,(tet5)*180/pi)
title('ch6')
subplot(2,3,6)
plot(temp,(tet7)*180/pi)
title('ch7')

Figure8)
subplot(2,3,1)
plot(temp, (lat13)*180/pi)
title('ch13')
subplot(2,3,2)
plot(temp,(tet14)*180/pi)
title('ch14')
subplot(2,3,3)
plot(temp,(tet15)*180/pi)
title('ch15')
subplot(2,3,4)
plot(temp,(tet16)*180/pi)
title('ch16')
subplot(2,3,5)
plot(temp,(tet17)*180/pi)
title('ch17')
subplot(2,3,6)
plot(temp,(tet18)*180/pi)
title('ch18')

Figure9)
subplot(2,3,3)
plot(temp,(tet19)*180/pi)
title('ch19')
subplot(2,3,2)
plot(temp,(tet20)*180/pi)
title('ch20')

save('condicionesI1.dat','condicionesI1','-ASCII')
save('condicionesI2.dat','condicionesI2','-ASCII')
save('condicionesI3.dat','condicionesI3','-ASCII')
save('condicionesI4.dat','condicionesI4','-ASCII')

```


Visualización de consignas articulares

En esta sección añadiremos las graficas que representan las consignas articulares para cada eslabón desarrolladas en *Matlab*.

Pierna derecha

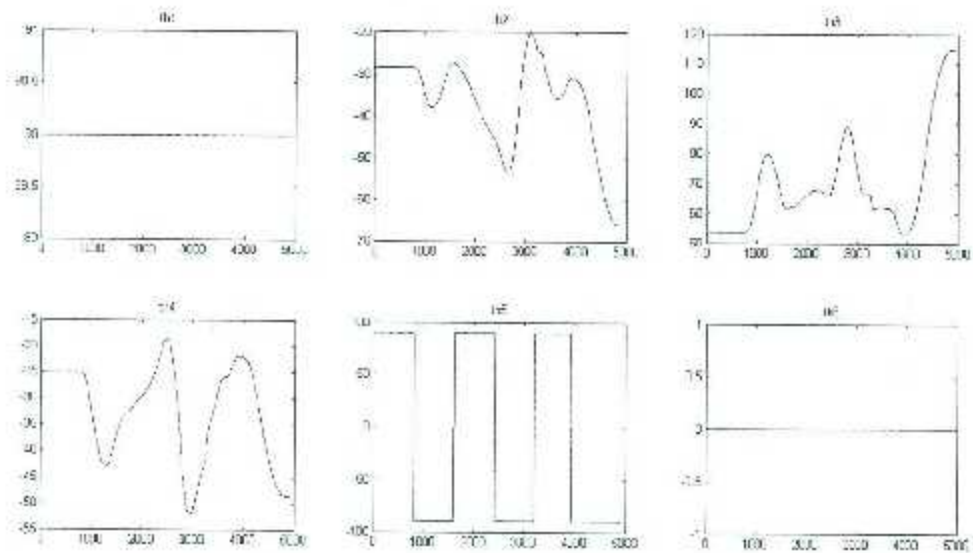


Figura B.1 Consignas articulares de posición pierna derecha; eje x posición (°) eje y tiempo (s)

Pierna izquierda

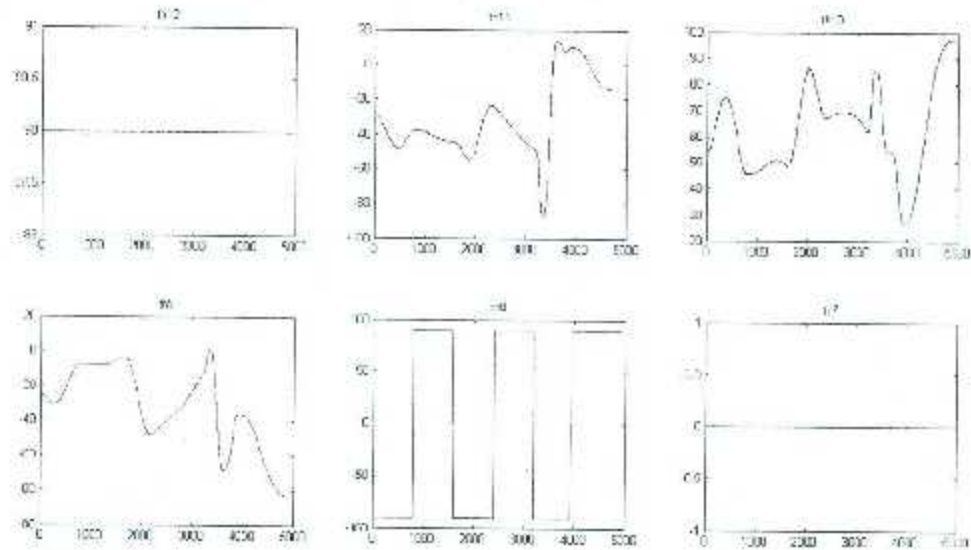


Figura B.2 Consignas articulares de posición pierna izquierda; eje x posición (°) eje y tiempo (s)

Brazos

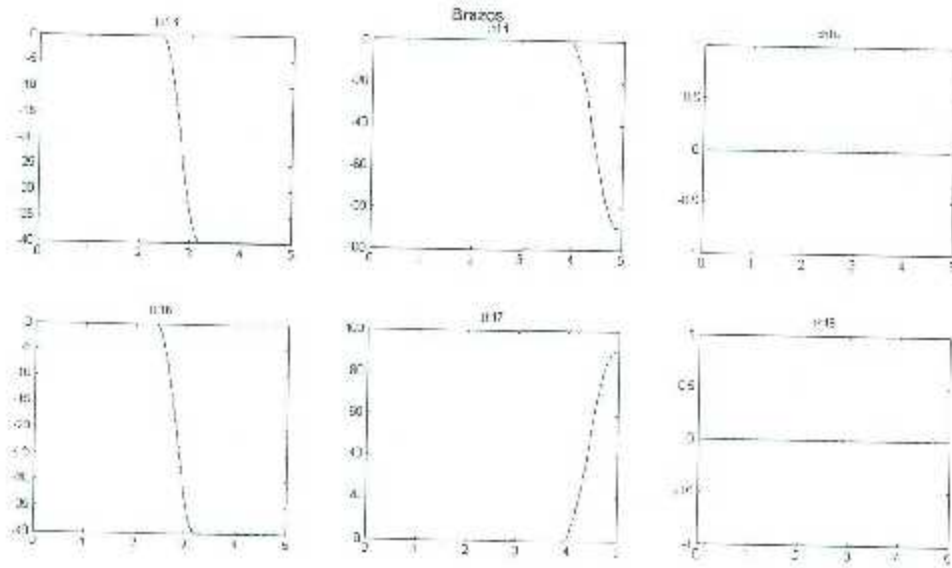


Figura B.3 Consignas articulares de posición brazos; eje x posición (°) eje y tiempo (s)

Cadera

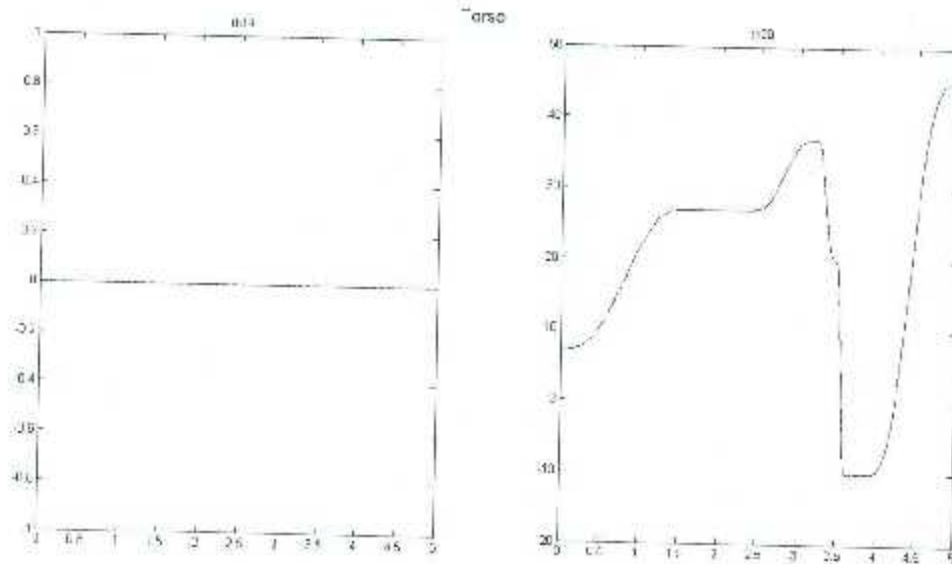


Figura B.4 Consignas articulares de posición torso; eje x posición (°) eje y tiempo (s)

Visualización del punto de momento cero (ZMP)

En esta sección añadiremos una visualización del ZMP en cada uno de los pasos que da el humanoide. Los pasos que representan la etapa de carrera son los pasos del 1 al 4; el paso que representan la patada es el paso 5 y el paso que representa la etapa de recuperación es el paso 6.

ZMP en paso 1

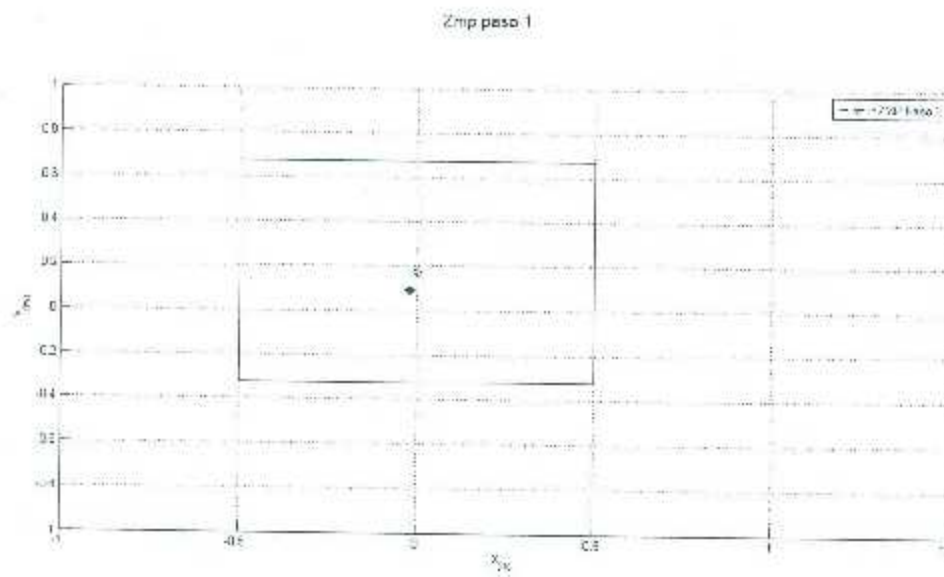


Figura C.1 Visualización del ZMP en la planta del pie durante el paso 1

ZMP en paso 2

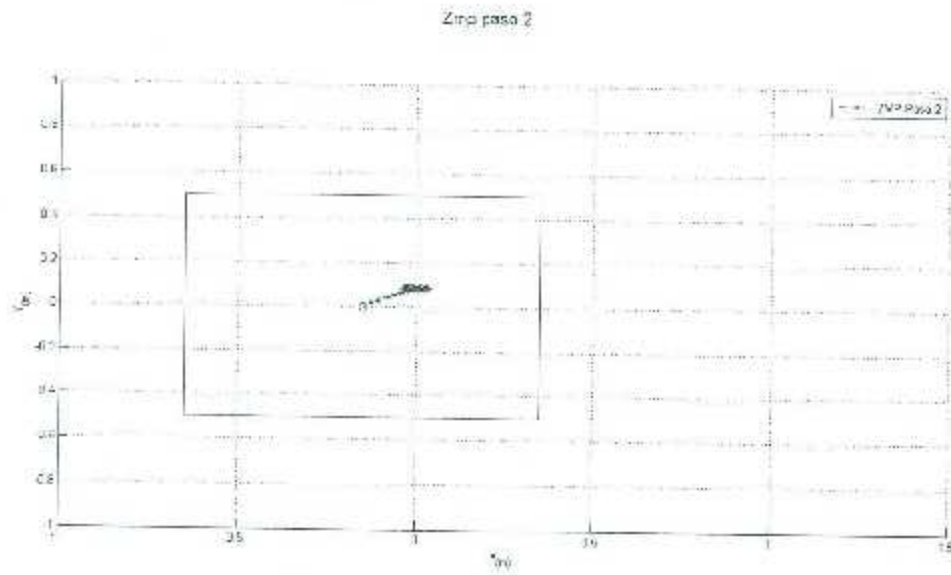


Figura C.2 Visualización del ZMP en la planta del pie durante el paso 2

ZMP en paso 3

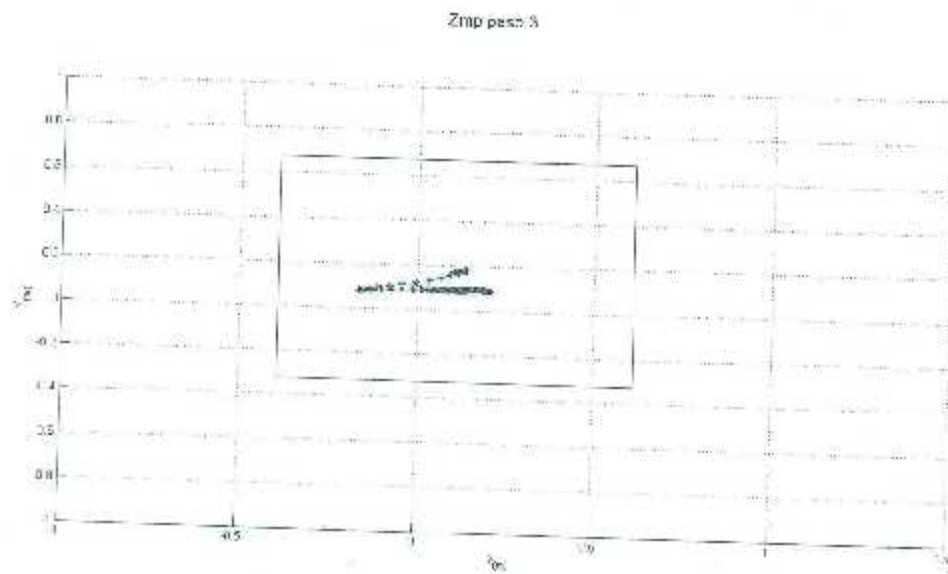


Figura C.3 Visualización del ZMP en la planta del pie durante el paso 3

ZMP en paso 4

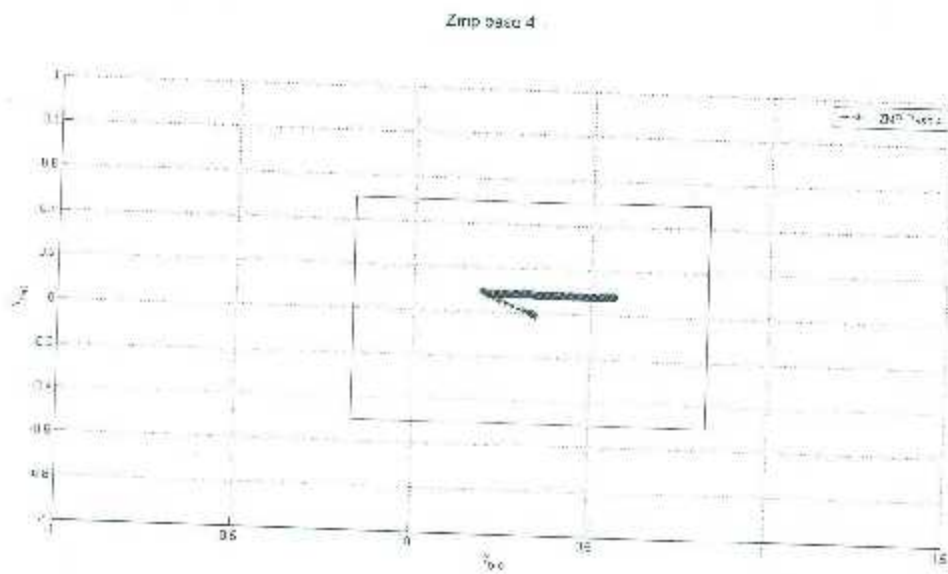


Figura C.4 Visualización del ZMP en la planta del pie durante el paso 4

ZMP en Paso 5

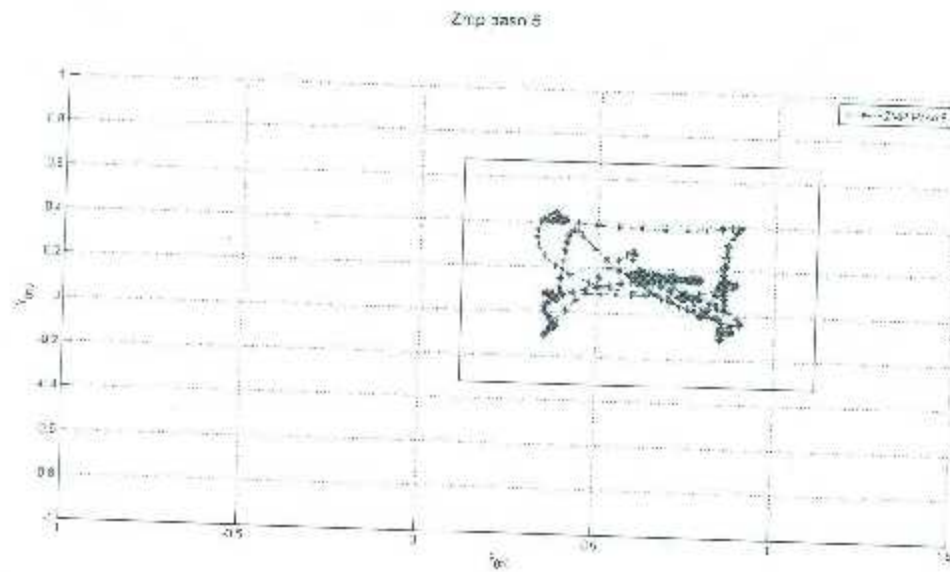


Figura C.5 Visualización del ZMP en la planta del pie durante el paso 5

ZMP en paso 6

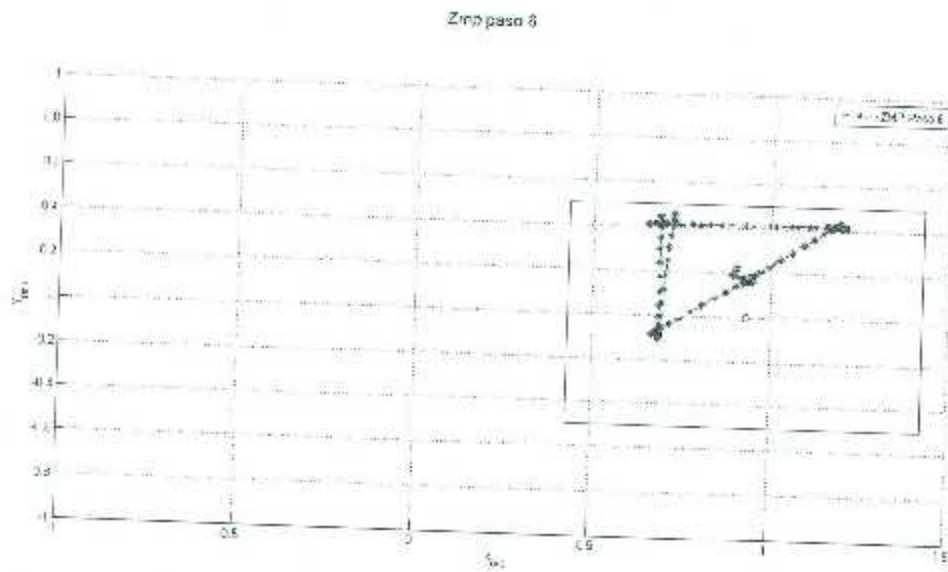


Figura C.6 Visualización del ZMP en la planta del pie durante el paso 6

ZMP Global

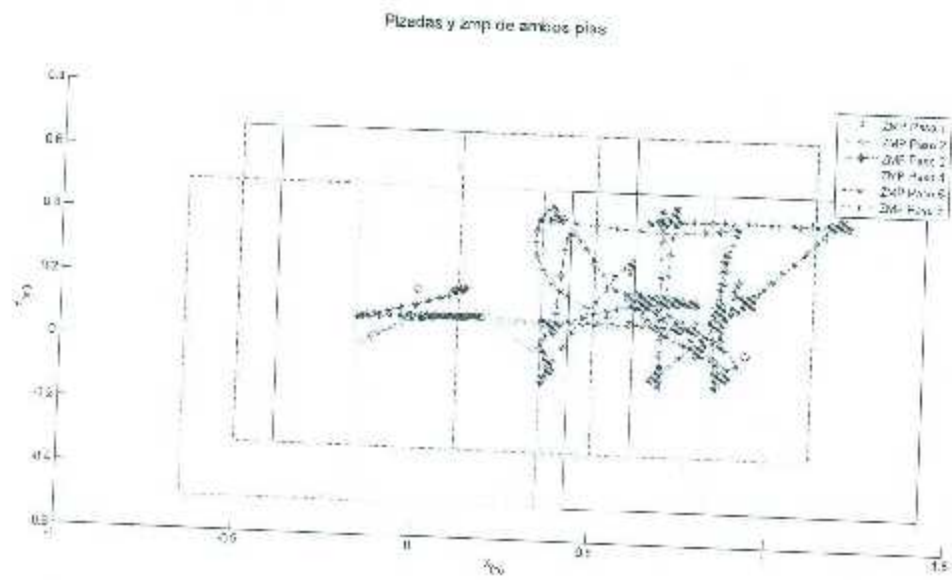


Figura C.7 Visualización del ZMP en la planta del pie durante toda la simulación

