

UNIVERSIDAD CARLOS III DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR

Dpto. de INGENIERÍA DE SISTEMAS Y AUTOMÁTICA



Universidad  
Carlos III de Madrid

INGENIERÍA TÉCNICA INDUSTRIAL  
ESPECIALIDAD ELECTRÓNICA INDUSTRIAL

PROYECTO FIN DE CARRERA

**INTERFAZ PARA LA GENERACIÓN  
DE TRAYECTORIAS PARA LAS  
PLATAFORMAS ROBÓTICAS  
HOAP-3 Y RH-2**

**AUTOR:** FERNANDO BORJA SABIO GALLEGO

**TUTOR:** CONCEPCIÓN ALICIA MONJE MICHARET



Oxen that rattle the yoke and chain or halt in the leafy shade,  
what is that you express in your eyes?  
It seems to me more than all the print I have read in my life.

Walt Whitman, *Song of Myself*



# Agradecimientos

---

En primer lugar tengo que agradecer a mi tutora Concha el apoyo y la paciencia que me ha mostrado en todo momento, sin los que me habría resultado imposible superar las dificultades que he encontrado en este proyecto. Si volviera al día en que empecé este trabajo, sin duda la volvería a elegir como tutora.

Gracias también a Paolo, por no dudar en dedicar parte de su escaso tiempo a ayudarme, y por ser exigente con las trayectorias. Poder ver al robot moverse ha compensado con creces todo el esfuerzo con las interpolaciones.

Gracias a mi compañera Tamara y su OpenHRP. A ella tengo que agradecer todas las simulaciones de este proyecto. Juntos hemos comprendido un poquito mejor cómo funciona nuestro pequeño amigo HOAP. Gracias también al resto de compañeros por hacer más agradables las horas que he pasado en el laboratorio.

Gracias a mis padres por ser para mí un modelo de lo que algún día aspiro a ser. Gracias por creer siempre en mí y saber enseñarme qué cosas son importantes en la vida. Gracias a mis hermanos por ser los mejores, que sea el hermano mayor no impide que cada día aprenda algo de ellos.

Gracias a toda mi familia, soy consciente de la suerte que tengo al tener una familia tan grande y a la vez tan cercana. Gracias a Irene por compartir su vida conmigo, sin ella no sería el mismo. Estoy seguro de que ningún logro en mi vida superará el de haber encontrado la mejor compañera.

Me considero muy afortunado por estar siempre rodeado de gente noble y extraordinaria, que cree en mí. Espero algún día poder hacer por ellos todo lo que ellos han hecho por mí, son la mejor motivación para seguir creciendo y buscando mi camino.

BORJA SABIO GALLEGO

21 de abril de 2011





## CONTENIDOS

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>9</b>
1.1	INTRODUCCIÓN.....	9
1.2	OBJETIVOS.....	10
1.3	ESTRUCTURA DEL DOCUMENTO.....	11
<b>2</b>	<b>EL ROBOT HOAP-3 .....</b>	<b>13</b>
2.1	LA SERIE DE HUMANOIDES HOAP .....	13
2.2	HOAP-3.....	14
2.2.1	<i>Aspectos generales</i> .....	14
2.2.2	<i>Control del robot</i> .....	15
2.2.3	<i>Modelo VRML del robot</i> .....	21
2.2.4	<i>Criterios para la interfaz</i> .....	24
2.2.5	<i>Archivos de trayectorias</i> .....	26
<b>3</b>	<b>EL ROBOT RH-2.....</b>	<b>31</b>
3.1	LA SERIE DE HUMANOIDES RH .....	31
3.2	RH-2 .....	33
3.2.1	<i>Aspectos generales</i> .....	33
3.2.2	<i>Control del robot</i> .....	33
3.2.3	<i>Modelo VRML del robot</i> .....	38
3.2.4	<i>Archivos de trayectorias</i> .....	40
<b>4</b>	<b>FUNCIONAMIENTO DE LA INTERFAZ.....</b>	<b>43</b>
4.1	INTRODUCCIÓN.....	43
4.2	ARCHIVOS NECESARIOS .....	44
4.2.1	<i>Equipos con el programa MATLAB instalado</i> .....	44
4.2.2	<i>Equipos que no tienen instalado el programa MATLAB</i> .....	44
4.2.3	<i>Plataformas compatibles</i> .....	45
4.3	GENERACIÓN DE TRAYECTORIAS .....	45
4.3.1	<i>Aspectos generales</i> .....	45
4.3.2	<i>Selección de parámetros de la trayectoria</i> .....	46
4.3.3	<i>Definir los movimientos de las articulaciones</i> .....	48
4.3.4	<i>Crear la trayectoria</i> .....	51
4.3.5	<i>Guardar la trayectoria</i> .....	52
4.4	CONCATENACIÓN DE TRAYECTORIAS.....	54
4.4.1	<i>Aspectos generales</i> .....	54
4.4.2	<i>Selección de parámetros</i> .....	55



4.4.3	<i>Carga de las trayectorias</i> .....	56
4.4.4	<i>Creación de la trayectoria</i> .....	57
4.5	CONVERSIÓN DE TRAYECTORIAS .....	58
4.5.1	<i>Aspectos generales</i> .....	58
4.5.2	<i>Selección de parámetros</i> .....	58
4.6	SIMULACIÓN DE TRAYECTORIAS.....	60
4.6.1	<i>Aspectos generales</i> .....	60
4.6.2	<i>Selección de parámetros</i> .....	60
4.6.3	<i>Representación gráfica</i> .....	61
4.7	GUARDAR GRÁFICAS .....	62
4.8	DOCUMENTO DE AYUDA “HELP” .....	63
<b>5</b>	<b>DISEÑO DE LA INTERFAZ</b> .....	<b>65</b>
5.1	MATLAB.....	65
5.1.1	<i>Introducción</i> .....	65
5.1.2	<i>Variables en MATLAB</i> .....	66
5.1.3	<i>Tipos de datos en MATLAB</i> .....	68
5.1.4	<i>Programación con MATLAB</i> .....	71
5.1.5	<i>Creación de interfaces gráficas con MATLAB</i> .....	73
5.2	DISEÑO DE LA INTERFAZ.....	77
5.2.1	<i>Estructura del programa</i> .....	77
5.2.2	<i>Funciones empleadas</i> .....	80
<b>6</b>	<b>CREACIÓN DE UNA TRAYECTORIA</b> .....	<b>101</b>
6.1	INTRODUCCIÓN.....	101
6.2	TRAYECTORIA 1: “SUBE Y BAJA” .....	102
6.3	TRAYECTORIA 2: “SALUDO DERECHA” .....	105
6.4	TRAYECTORIA 3: “SALUDO IZQUIERDA” .....	108
6.5	TRAYECTORIA 4: “GIMNASIA” .....	110
6.6	TRAYECTORIA 5: “FLEXIÓN BRAZOS” .....	112
6.7	TRAYECTORIA 6: “BAILE” .....	115
<b>7</b>	<b>CONCLUSIONES</b> .....	<b>119</b>
<b>8</b>	<b>TRABAJOS FUTUROS</b> .....	<b>121</b>
<b>9</b>	<b>ANEXO - DOCUMENTO DE AYUDA</b> .....	<b>123</b>
<b>10</b>	<b>REFERENCIAS</b> .....	<b>163</b>



## ÍNDICE DE FIGURAS

Figura 2.1. De izquierda a derecha, HOAP-1, HOAP-2 y HOAP-3.....	14
Figura 2.2. Grados de libertad del robot HOAP-3 [1].....	16
Figura 2.3. Movimientos Roll, Pitch y Yaw en el espacio.....	16
Figura 2.4. Definición de parámetros de DH en el HOAP-3.....	18
Figura 2.5 Articulaciones en el modelo VRML del HOAP-3 [11].....	21
Figura 2.6.Vista parcial de un archivo de trayectorias angle.dat para el HOAP-3.....	26
Figura 2.7. Vista parcial de un archivo de trayectorias con extensión csv.....	29
Figura 3.1. De izquierda a derecha, RH-0, RH-1 y RH-2.....	32
Figura 3.2. Distribución de GDL en el RH-2.....	35
Figura 3.3. Articulaciones de las piernas del RH-2.....	35
Figura 3.4. Modelo VRML del robot RH-2 [13].....	38
Figura 3.5. Vista parcial de un archivo de trayectorias angle.dat para el RH-2 [12].....	40
Figura 4.1. Pantalla principal de la interfaz.....	44
Figura 4.2. Pantalla de creación de nuevas trayectorias para el HOAP-3.....	46
Figura 4.3. Parámetros de la trayectoria.....	46
Figura 4.4. Selección de paso 3 ms.....	47
Figura 4.5. Selección de paso distinta de 3 ms.....	47
Figura 4.6. Definir movimientos.....	48
Figura 4.7. Menú Right Leg.....	49
Figura 4.8. Mensaje falta el valor del paso.....	49
Figura 4.9. Mensaje para ángulo fuera de rango.....	50
Figura 4.10. Mensaje valor de tiempo erróneo para paso 3 ms.....	50
Figura 4.11. Mensaje valores eliminados.....	51
Figura 4.12. Mensaje con la duración de la trayectoria.....	51
Figura 4.13. Cuadro de diálogo para guardar archivo con extensión .dat.....	52
Figura 4.14. Mensaje archivo con extensión .dat guardado.....	52
Figura 4.15. Cuadro de diálogo para guardar archivo con extensión .csv.....	53
Figura 4.16. Barra de espera al guardar archivo con extensión .csv.....	53
Figura 4.17. Mensaje archivo con extensión .csv guardado.....	53
Figura 4.18. Archivo de puntos de la trayectoria.....	54
Figura 4.19. Pantalla de concatenación de trayectorias para HOAP-3.....	54
Figura 4.20. Parámetros para la concatenación de trayectorias para el HOAP-3.....	55
Figura 4.21. Mensaje de error al cargar la trayectoria.....	56
Figura 4.22. Mensaje de trayectoria cargada correctamente.....	56
Figura 4.23. Mensaje con las articulaciones fuera del rango permitido.....	57
Figura 4.24. Vista general de la pantalla: "Convertir trayectorias HOAP-3".....	58
Figura 4.25. Parámetros para convertir una trayectoria del RH-2.....	59
Figura 4.26. Parámetros para convertir una trayectoria del HOAP-3.....	59
Figura 4.27. Vista general de la pantalla "Simular trayectorias HOAP-3".....	60

Figura 4.28. Parámetros de la simulación, para el RH-2.....	60
Figura 4.29. Parámetros de la simulación, para el HOAP-3.....	61
Figura 4.30. Detalle de la pantalla “Simular trayectoria HOAP-3”.....	61
Figura 4.31. Panel "Guardar gráficas" .....	62
Figura 4.32: Ejemplo de gráficos tras crear la trayectoria .....	63
Figura 4.33. Botón Help en pantalla "Convertir trayectoria "RH-2".....	63
Figura 5.1. Estructura del programa (I).....	78
Figura 5.2. Estructura del programa (II).....	79
Figura 6.1. Puntos definidos para la trayectoria “Sube y baja” .....	102
Figura 6.2. Posición inicial en la trayectoria "Sube y baja".....	103
Figura 6.3. Trayectoria "Sube y baja": rodillas flexionadas .....	103
Figura 6.4. Gráficas de posición en la trayectoria "Sube y baja" .....	104
Figura 6.5. Gráficas de velocidad en la trayectoria "Sube y baja" .....	104
Figura 6.6. Gráficas de aceleración en la trayectoria "Sube y baja" .....	105
Figura 6.7. Trayectoria "Saludo derecha".....	106
Figura 6.8. Gráficas de posición de los brazos en la trayectoria "Saludo derecha" .....	106
Figura 6.9. Gráficas de velocidad de los brazos en la trayectoria "Saludo derecha" .....	107
Figura 6.10. Gráficas de aceleración de los brazos en la trayectoria "Saludo derecha".....	107
Figura 6.11. Trayectoria "Saludo izquierda".....	108
Figura 6.12. Gráficas de posición en la trayectoria "Saludo izquierda" .....	108
Figura 6.13. Gráficas de velocidad en la trayectoria "Saludo izquierda" .....	109
Figura 6.14. Gráficas de aceleración en la trayectoria "Saludo izquierda" .....	109
Figura 6.15. Comienzo de la trayectoria "Gimnasia".....	110
Figura 6.16. Brazos arriba en la trayectoria "Gimnasia" .....	110
Figura 6.17. Gráficas de posición en la trayectoria “Gimnasia” .....	111
Figura 6.18. Gráficas de velocidad en la trayectoria “Gimnasia” .....	111
Figura 6.19. Gráficas de aceleración en la trayectoria “Gimnasia” .....	112
Figura 6.20. Brazo izquierdo extendido en la trayectoria "Flexión brazos" .....	113
Figura 6.21. Brazo derecho extendido en la trayectoria "Flexión brazos".....	113
Figura 6.22. Brazos extendidos en la trayectoria "Flexión brazos" .....	114
Figura 6.23. Gráficas de posición para los brazos en la trayectoria “Flexión brazos”.....	114
Figura 6.24. Gráficas de velocidad para los brazos en la trayectoria “Flexión brazos”.....	115
Figura 6.25. Gráficas de aceleración para los brazos en la trayectoria “Flexión brazos” .....	115
Figura 6.26. Secuencia de movimientos en la trayectoria "Baile" (I).....	116
Figura 6.27. Secuencia de movimientos en la trayectoria "Baile" (II).....	117
Figura 9.1. Parámetros de la trayectoria.....	127
Figura 9.2. Selección de paso 3 ms.....	127
Figura 9.3. Definir movimientos .....	129
Figura 9.4. Menú Right Leg del HOAP-3 .....	129
Figura 9.5. Mensaje de paso no indicado .....	130

Figura 9.6. Mensaje de ángulo fuera de rango .....	130
Figura 9.7. Mensaje de valor de tiempo erróneo para paso 3 ms .....	131
Figura 9.8. Mensaje de valores eliminados.....	131
Figura 9.9. Mensaje de falta de valores .....	132
Figura 9.10. Mensaje con duración de la trayectoria .....	132
Figura 9.11. Cuadro de diálogo para guardar archivo con extensión .dat .....	133
Figura 9.12. Mensaje de archivo con extensión .dat guardado .....	133
Figura 9.13. Cuadro de diálogo para guardar archivo con extensión .csv .....	134
Figura 9.14. Barra de espera guardar archivo con extensión .csv .....	134
Figura 9.15. Mensaje de archivo con extensión .csv guardado .....	134
Figura 9.16. Archivo de puntos de la trayectoria .....	135
Figura 9.17. Parámetros para la concatenación de trayectorias para el HOAP-3 .....	135
Figura 9.18. Mensaje de error al cargar trayectoria.....	137
Figura 9.19. Mensaje de trayectoria cargada correctamente .....	137
Figura 9.20. Mensaje que muestra articulaciones fuera de rango permitido .....	137
Figura 9.21. Cuadro de diálogo para guardar archivo con extensión .dat .....	138
Figura 9.22. Mensaje de archivo con extensión .dat guardado .....	139
Figura 9.23. Cuadro de diálogo para guardar archivo con extensión .csv .....	139
Figura 9.24. Barra de espera para guardar archivo con extensión .csv.....	139
Figura 9.25. Mensaje de archivo con extensión .csv guardado .....	140
Figura 9.26. Parámetros para convertir trayectoria del RH-2.....	140
Figura 9.27. Parámetros para convertir trayectoria del HOAP-3.....	141
Figura 9.28. Cuadro de diálogo para seleccionar archivo .....	142
Figura 9.29. Mensaje que muestra articulaciones fuera de rango permitido .....	142
Figura 9.30. Cuadro de diálogo para guardar archivo .dat .....	143
Figura 9.31. Mensaje de archivo con extensión .dat guardado .....	143
Figura 9.32. Cuadro de diálogo para guardar archivo con extensión .csv .....	144
Figura 9.33. Barra de espera para guardar archivo con extensión .csv.....	144
Figura 9.34. Mensaje de archivo con extensión .csv guardado .....	144
Figura 9.35. Parámetros para simulación del RH-2.....	145
Figura 9.36. Parámetros para simulación del HOAP-3 .....	145
Figura 9.37. Mensaje que muestra articulaciones fuera de rango permitido .....	146
Figura 9.38. Detalle del panel de simulaciones .....	146
Figura 9.39. Panel "Guardar gráficas" .....	146
Figura 9.40. Ejemplo de gráficos tras crear la trayectoria .....	147
Figura 9.41. Grados de libertad del HOAP-3.....	148
Figura 9.42. Definición de parámetros de DH en el HOAP-3 .....	150
Figura 9.43. Articulaciones en el modelo VRML del HOAP-3 .....	151
Figura 9.44. Vista parcial de un archivo de trayectorias angle.dat.....	153
Figura 9.45. Vista parcial de un archivo de trayectorias con extensión csv.....	156



*Figura 9.46. Distribución de GDL en el RH-2..... 157*

*Figura 9.47. Modelo VRML del robot RH-2 ..... 160*

*Figura 9.48. Archivo de trayectorias angle.dat para el RH-2..... 161*



## ÍNDICE DE TABLAS

<i>Tabla 2.1. Movimiento asociado a cada articulación en el HOAP-3 [1].....</i>	<i>17</i>
<i>Tabla 2.2. Tabla de equivalencias de los motores 22 y 23 [1].....</i>	<i>19</i>
<i>Tabla 2.3. Rango de movimiento permitido para cada articulación en el HOAP-3 .....</i>	<i>20</i>
<i>Tabla 2.4. Correspondencia entre articulaciones del robot HOAP-3 y el modelo VRML.....</i>	<i>22</i>
<i>Tabla 2.5. Relación de articulaciones con convenio.....</i>	<i>23</i>
<i>Tabla 2.6. Rango de movimiento de las articulaciones del HOAP-3 para el usuario.....</i>	<i>25</i>
<i>Tabla 2.7. Articulaciones en archivos de trayectorias para el HOAP-3.....</i>	<i>27</i>
<i>Tabla 3.1. Movimiento asociado a cada articulación en el RH-2.....</i>	<i>34</i>
<i>Tabla 3.2. Rango de movimiento permitido para cada articulación.....</i>	<i>36</i>
<i>Tabla 3.3. Correspondencia entre articulaciones del robot y el modelo .....</i>	<i>39</i>
<i>Tabla 3.4. Articulaciones en archivos de trayectorias del RH-2.....</i>	<i>41</i>
<i>Tabla 9.1. Movimiento asociado a cada articulación en el HOAP-3.....</i>	<i>149</i>
<i>Tabla 9.2. Rango de movimiento de las articulaciones del HOAP-3 para el usuario.....</i>	<i>152</i>
<i>Tabla 9.3. Articulaciones en archivos de trayectorias para el HOAP-3.....</i>	<i>154</i>
<i>Tabla 9.4. Movimiento asociado a cada articulación en el RH-2.....</i>	<i>158</i>
<i>Tabla 9.5. Rango de movimiento permitido para cada articulación del RH-2.....</i>	<i>159</i>
<i>Tabla 9.6. Articulaciones en archivos de trayectorias del RH-2.....</i>	<i>162</i>



# **1 Introducción**

## **1.1 Introducción**

La razón principal para el empleo de simuladores en investigación robótica es prevenir posibles accidentes que podrían dañar los costosos prototipos o incluso llegar a poner en peligro a las personas que trabajan con ellos al mostrar comportamientos inesperados. Por ello, la simulación de plataformas robóticas constituye una manera segura de poner a prueba las trayectorias y controles diseñados antes de implementarlos en el robot real.

Estas plataformas constituyen una potente herramienta de estudio que permite validar el software empleado y el control ejercido sobre un robot. Además de permitir visualizar de una manera muy intuitiva los movimientos del robot a través de un modelo en tres dimensiones que interacciona con un entorno de realidad virtual, hacen posible evaluar los valores de las fuerzas y velocidades que presentan las articulaciones del robot al ejecutar los movimientos. En algunos casos incluso permiten al usuario interactuar con el propio robot en tiempo real.

Para que la información obtenida pueda ser fiable, es importante construir modelos que se asemejen en todo lo posible al robot real. En la medida en que el modelo se asemeje a la realidad, será fiable la validación realizada por medio de la simulación.

Las simulaciones a las que se hace referencia en este documento se han realizado con el programa OpenHRP3. Este programa trabaja con archivos de datos con extensión *.dat*. Los datos de estos archivos representan posición, velocidad y

aceleración de las articulaciones a lo largo de la trayectoria, y se expresan en *radianes*, *radianes/segundo* y *radianes/segundo<sup>2</sup>*, respectivamente.

Sin embargo, los robots emplean otro sistema para definir las trayectorias. En el caso del HOAP-3, maneja archivos con extensión .csv que reflejan las posiciones de los encoder de las articulaciones. Estos datos se expresan en pulsos decimales, existiendo un factor de conversión entre grados y pulsos específico para dichos sensores.

Por tanto estos formatos, el del simulador y el de los robots reales, son incompatibles entre sí. De ahí surge la necesidad de crear una herramienta para poder transformar los archivos entre uno y otro formato. Esta es la motivación principal de esta interfaz.

Además de esta conversión entre formatos de trayectorias ya existentes, se quiere generar trayectorias para dichos robots. La generación de trayectorias implica un conocimiento exhaustivo de las características dinámicas y cinemáticas del robot. La cinemática directa, consiste en definir unos movimientos para las articulaciones y calcular la posición que alcanzará la herramienta o extremo del robot. El proceso inverso, conocido como cinemática inversa, permite conocer el movimiento que se ha de dar en cada articulación para alcanzar una posición y orientación concretas del efector final. El algoritmo de Denavit Hartenberg se emplea normalmente para el cálculo de una y otra cinemática.

Con este trabajo, se pretende proponer una solución para de manera sencilla generar trayectorias articulares que definan un movimiento concreto del robot en el tiempo, permitiendo la posterior conversión de las mismas a los formatos correspondientes para su ejecución en las plataformas robóticas HOAP-3 y RH-2.

## **1.2 Objetivos**

El objetivo principal de este proyecto es el desarrollo de una interfaz gráfica que facilite al usuario la generación y el estudio de trayectorias asociadas a los robots HOAP-3 y RH-2.

La herramienta pretende constituir una plataforma que permita:



- generar de manera intuitiva nuevas trayectorias para el robot y para el simulador
- generar nuevas trayectorias uniendo una secuencia de trayectorias existentes
- obtener los archivos de velocidades y aceleraciones de una determinada trayectoria.
- observar de manera gráfica el comportamiento de las articulaciones durante la ejecución de la trayectoria. En adelante nos referiremos a esta función de la interfaz como “Simulación de trayectorias”, aunque no se trata de simulaciones propiamente dichas
- convertir archivos entre los formatos empleados por el simulador y el robot

Una vez alcanzado este primer objetivo, se pondrá a prueba la herramienta diseñada creando una trayectoria y haciendo las simulaciones correspondientes en OpenHRP3. Después se probarán las trayectorias creadas para el HOAP-3 en el propio robot. No será posible en el caso del RH-2 ya que el prototipo no está terminado aún.

## 1.3 Estructura del documento

En el capítulo 1, “**Introducción**”, se presentan los objetivos del proyecto y se describe brevemente la estructura del presente documento.

En el capítulo 2, “**El robot HOAP-3**”, se describen en profundidad las características de este robot y el modo de controlarlo, así como el modelo del mismo y los archivos de trayectorias necesarios para realizar las simulaciones.

En el capítulo 3, “**El robot RH-2**”, se estudia dicho robot del mismo modo que en el capítulo 3 se hace para el HOAP-3.

En el capítulo 4, “**Funcionamiento de la interfaz**”, se dan las instrucciones necesarias para poner la aplicación en funcionamiento. Después se explican detalladamente las funcionalidades de la misma. Se analiza el proceso de creación de trayectorias, de conversión entre distintos formatos y de obtención de archivos de velocidad y aceleración.

En el capítulo 5, “**Diseño de la interfaz**”, se introduce brevemente al diseño de interfaces gráficas por medio de MATLAB, y se explica en detalle cómo se ha diseñado la interfaz y el funcionamiento del programa.

En el capítulo 6, “**Creación de una trayectoria**”, se muestran cinco trayectorias que han sido creadas por medio de la interfaz, y que unidas dan lugar a la trayectoria sexta que se ha probado en el robot HOAP-3 real. Se aportan gráficas e imágenes de dichas pruebas.

En el capítulo 7, “**Conclusiones**”, se reflexiona sobre el trabajo realizado y el rendimiento que se puede obtener con la herramienta diseñada.

En el capítulo 8, “**Trabajos futuros**”, se exploran las posibilidades abiertas en torno al tema que trata el proyecto. Se comentan posibles mejoras para nuevas versiones de la interfaz.

En el capítulo 9 se muestra el **anexo**, en el que se puede encontrar el documento de ayuda de que dispone la interfaz.

Por último, en el capítulo 10 se citan las **referencias** a las fuentes consultadas en la elaboración del proyecto.

En la documentación que se presenta en el CD adjunto a esta memoria se encuentra la aplicación diseñada, así como los archivos relativos a las trayectorias diseñadas mediante la misma.

En la carpeta “**Generador de trayectorias**” se encuentra el archivo ejecutable de la aplicación y los programas necesarios para ejecutarlo en un ordenador que no tiene instalado MATLAB.

En la carpeta “**Trayectorias generadas**” están los archivos de las trayectorias generadas, un archivo con los puntos definidos al crear la trayectoria, y los videos de las simulaciones realizadas en OpenHRP3 y de la prueba en el robot real.

En la carpeta “**Programas en MATLAB**” se puede encontrar todo el código en ficheros .m así como las imágenes y documentos de texto necesarios para hacer funcionar la aplicación a través de MATLAB.

## **2 El robot HOAP-3**

### **2.1 La serie de humanoides HOAP**

En septiembre de 2001, Fujitsu Automation Ltd. presenta en colaboración con Fujitsu Laboratories Ltd. el robot humanoide en miniatura **HOAP-1**. '*HOAP*' es acrónimo del inglés '*Humanoid for Open Architecture Platform*'.

Con un peso de 6 kg y una altura de 48 cm, el **HOAP-1** (Figura 2.1, izquierda) está dotado de 20 grados de libertad. Se concibe para un amplio abanico de aplicaciones en investigación y desarrollo de tecnologías robóticas. La arquitectura interna del HOAP-1 es abierta, con el objeto de permitir a los usuarios desarrollar sus propios programas. De este modo, constituye una plataforma de estudio que facilita actividades como el desarrollo de algoritmos de control de movimiento, el desarrollo de interfaces de comunicación hombre-robot, etc.

Dos años después de la aparición de este primer modelo, en 2003, se presenta el **HOAP-2** (Figura 2.1, centro), de 7 kg y 50 cm de altura. Esta vez presenta 25 grados de libertad y es capaz de mover cuello, cintura y manos. Además tiene dedos articulados que le permiten agarrar objetos.

Finalmente, en 2005 aparece el **HOAP-3**, con un peso de 8.8 kg y 60 cm de altura. Esta tercera versión presenta 28 grados de libertad (Figura 2.1, derecha), y entre sus nuevas características destacan las capacidades avanzadas de reconocimiento de imagen y de audio [1].

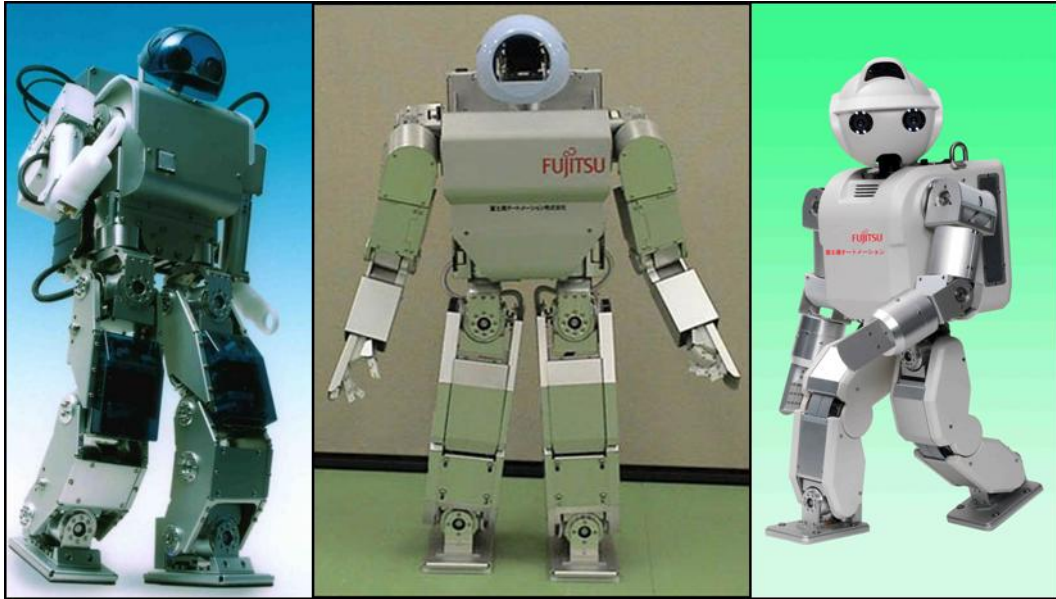


Figura 2.1. De izquierda a derecha, HOAP-1, HOAP-2 y HOAP-3.

## 2.2 HOAP-3

### 2.2.1 Aspectos generales

Los 28 grados de libertad (GDL<sup>1</sup>) del HOAP-3 se distribuyen del siguiente modo: seis GDL en cada pierna, cinco GDL en cada brazo, un GDL en la cintura, un GDL en cada mano y tres GDL en el cuello. Esto se comenta en detalle en el siguiente apartado, “Control del robot”.

El HOAP-3 presenta distintas mejoras sobre sus predecesores, orientadas a su empleo en la investigación de la comunicación con robots. A tal efecto, éste incorpora una cámara, un micrófono, un altavoz y dos LED imitando los ojos humanos para expresar emociones. Así mismo trae implementadas funciones como reconocimiento de voz, reconocimiento de imagen y síntesis vocal.

Como en los anteriores modelos del HOAP, la interfaz interna de hardware y software es de código abierto, lo que deja al usuario el control total de la programación en C/C++ [21].

El robot lleva incorporado un PC-104 embebido en su interior. El sistema operativo es RTLinux, Linux en tiempo real (sin el parche de tiempo real no se podrían controlar los

---

<sup>1</sup> GDL es el acrónimo de *grados de libertad*. También es común el empleo de DOF, acrónimo del inglés *degrees of freedom*.

servos del robot). Además, junto con el robot, el fabricante proporciona un ordenador idéntico al que lleva en su interior el HOAP-3, con el mismo hardware y software instalado.

El robot viene equipado con una serie de sensores: encoder ópticos incrementales en las articulaciones, acelerómetros y giróscopos en los tres ejes, sensores de fuerza en los pies y manos y sensores infrarrojos de distancia [22].

El robot se puede controlar en dos modos: online y offline. En el modo online, el robot recibe las instrucciones a través del PC externo al que está conectado por medio de un cable USB. En el modo offline, se conecta a través de un router.

Como se verá más adelante, nosotros trabajaremos con el robot en modo offline, diseñando trayectorias que guardaremos en archivos de datos con extensión csv, para ser cargados al robot a través del router.

### **2.2.2 Control del robot**

Desde un punto de vista mecánico, el robot mueve las articulaciones por medio de 23 motores. Los primeros 21 controlan las articulaciones de las piernas y los brazos, mientras que el motor 22 controla los 3 grados de libertad de la cabeza y el motor 23 permite la rotación y el agarre de las manos.

Los primeros 21 motores llevan acoplados encoder relativos, y pueden ser controlados en posición y velocidad. El propósito de la aplicación diseñada es generar trayectorias para controlar el robot en posición.

En la Figura 2.2 se presenta la estructura mecánica del robot, indicándose la localización de los grados de libertad en el mismo.

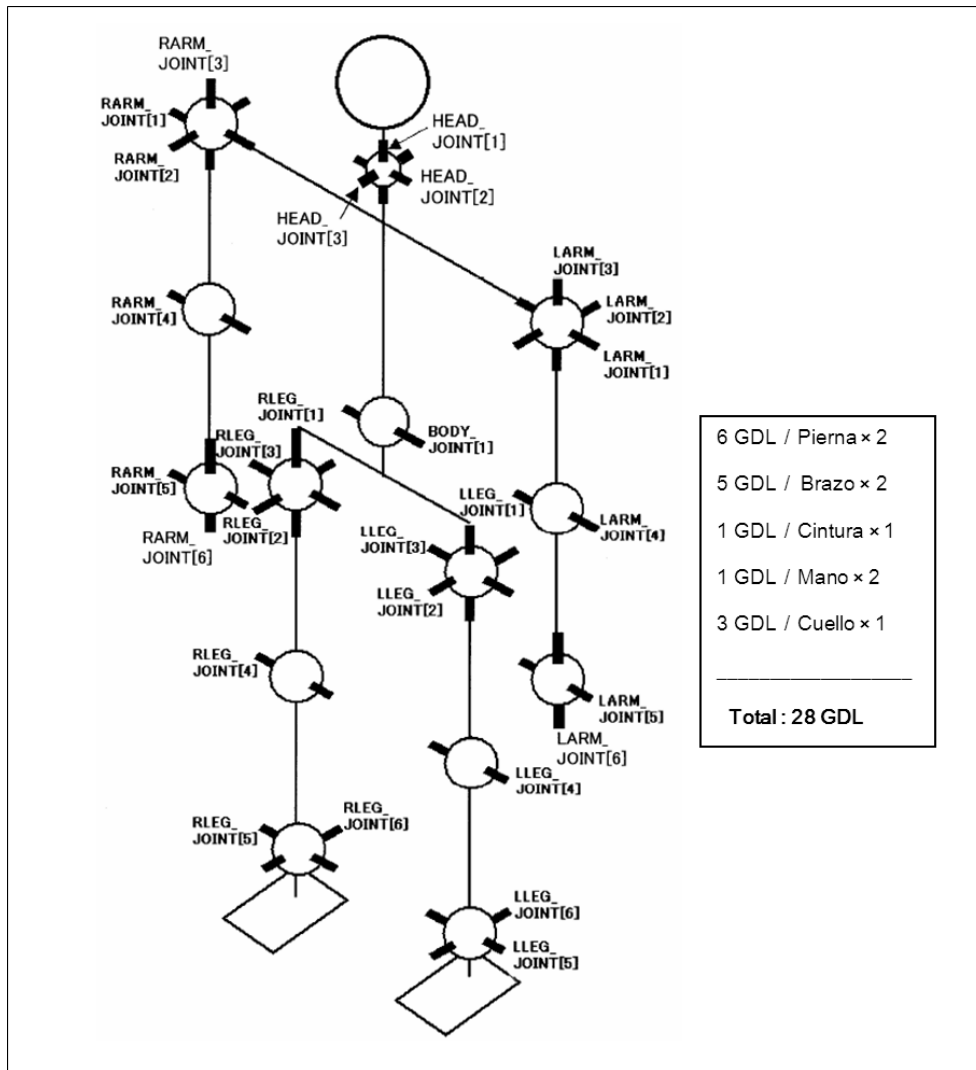


Figura 2.2. Grados de libertad del robot HOAP-3 [1]

En la Tabla 2.1 se indica el movimiento asociado a cada articulación en función de los ejes *Roll*, *Pitch* y *Yaw* que aparecen referenciados en la Figura 2.3.

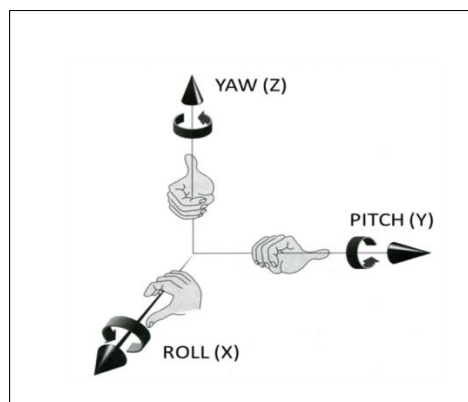


Figura 2.3. Movimientos *Roll*, *Pitch* y *Yaw* en el espacio

Articulación	Identificador	Movimiento
RLEG_JOINT1	1	Right hip joint torsion
RLEG_JOINT2	2	Right hip joint roll
RLEG_JOINT3	3	Right hip joint pitch
RLEG_JOINT4	4	Right knee
RLEG_JOINT5	5	Right ankle pitch
RLEG_JOINT6	6	Right ankle roll
RARM_JOINT1	7	Right shoulder pitch
RARM_JOINT2	8	Right shoulder roll
RARM_JOINT3	9	Right shoulder torsion
RARM_JOINT4	10	Right elbow
LLEG_JOINT1	11	Left hip joint torsion
LLEG_JOINT2	12	Left hip joint roll
LLEG_JOINT3	13	Left hip joint pitch
LLEG_JOINT4	14	Left knee
LLEG_JOINT5	15	Left ankle pitch
LLEG_JOINT6	16	Left ankle roll
LARM_JOINT1	17	Left shoulder pitch
LARM_JOINT2	18	Left shoulder roll
LARM_JOINT3	19	Left shoulder torsion
LARM_JOINT4	20	Left elbow
BODY_JOINT1	21	Waist pitch
HEAD_JOINT1	22	Head torsion
HEAD_JOINT2	22	Head pitch
HEAD_JOINT3	22	Head roll
LARM_JOINT5	23	Left fingers open/close
RARM_JOINT5	23	Right fingers open/close
LARM_JOINT6	23	Left hand torsion
RARM_JOINT6	23	Right hand torsion

Tabla 2.1. Movimiento asociado a cada articulación en el HOAP-3 [1]

### 2.2.2.1 Denavit-Hartenberg:

El método de Denavit-Hartenberg hace posible describir y representar de una manera sistemática la geometría espacial de los elementos de un robot con respecto a un sistema de referencia fijo.

Así, el problema cinemático directo queda reducido a encontrar una matriz de transformación homogénea 4x4 que relacione la localización espacial del extremo del robot con respecto al sistema de coordenadas de su base. Para el caso del robot HOAP-3 se emplean los ejes de coordenadas de la Figura 2.4

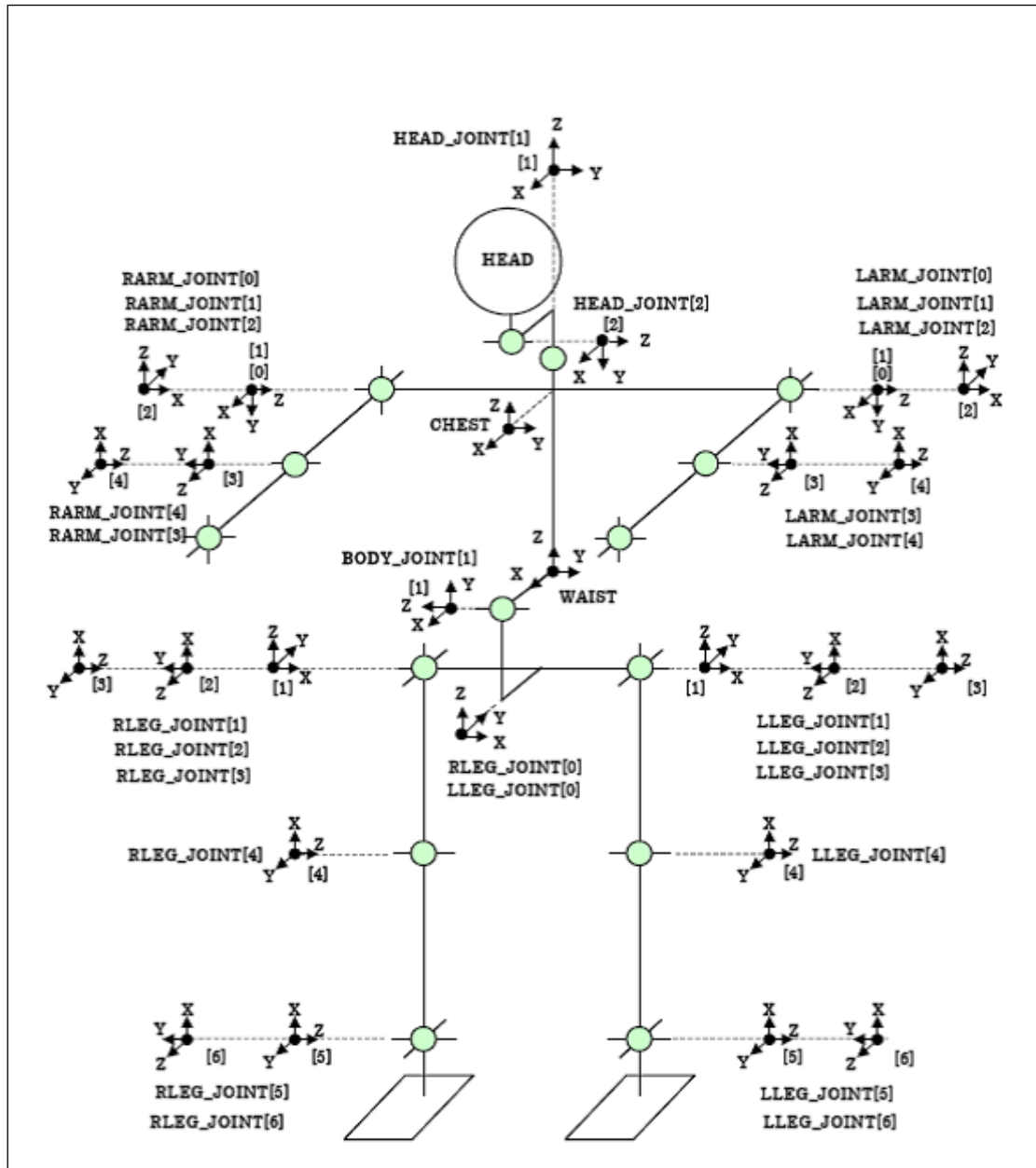


Figura 2.4. Definición de parámetros de DH en el HOAP-3

Cada articulación gira en torno a su eje Z, quedando definidos la dirección y el sentido de los ejes por la regla de la mano derecha.



Como se ha comentado antes, los motores llevan acoplados un encoder. Para controlar dichos motores se dan órdenes al robot de girar un determinado número de pulsos del encoder. El factor de conversión de pulsos a grados es 209 [1].

De este modo, si queremos que el motor gire un ángulo  $\alpha$  ( $^\circ$ ), tenemos que darle el valor equivalente  $\theta$  en pulsos del encoder:

$$\theta \text{ (pulsos)} = \alpha \text{ (}^\circ\text{)} \times 209$$

Esta ecuación se aplica para los primeros 21 motores. Para controlar el motor 22 y el motor 23, que corresponden a las articulaciones del cuello y las manos, se emplea la Tabla 2.2

	Transmitting data (※)														Drive part	Angle							
	Binary							Decimal								angle	direction	note					
M22	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15	(= 0+ 15)	Head tilt	-45°	down	min	
	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	60	(= 0+ 60)		0°	front	org	
	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	120	(= 0+ 75)		15°	up	max	
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	256	(= 256+ 0)	Head pan	-60°	right	min	
	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	316	(= 256+ 60)		0°	front	org	
	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	376	(= 256+ 120)		60°	left	max	
	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	562	(= 512+ 45)	Head roll	-15°	left	min
	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	572	(= 512+ 60)	0°		center	org	
0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	582	(= 512+ 75)	15°	right		max		
M23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(= 0+ 0)	Right hand Rotate (※※)	-60°		min	
	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	60	(= 0+ 60)	0°		center	org		
	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	120	(= 0+ 120)	60°			max		
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	256	(= 256+ 0)	Left hand Rotate (※※)	-60°		min	
	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	316	(= 256+ 60)		0°	center	org	
	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	376	(= 256+ 120)		60°		max	
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	512	(= 512+ 0)	Right hand Open/close	-60°	open	min	
	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	572	(= 512+ 60)		0°	center	org	
	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	632	(= 512+ 120)		60°	close	max	
	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	768	(= 768+ 0)	Left hand Open/close	-60°	close	min	
	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	828	(= 768+ 60)		0°	center	org	
	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	888	(= 768+ 120)		60°	open	max	

Tabla 2.2. Tabla de equivalencias de los motores 22 y 23 [1]

Articulación	Id	Movimiento asociado	Parámetro mínimo DH		Parámetro máximo DH	
			Grados	Pulsos	Grados	Pulsos
RLEG_JOINT1	1	Right hip joint torsion	-91	-19019	31	6479
RLEG_JOINT2	2	Right hip joint roll	-31	-6479	21	4389
RLEG_JOINT3	3	Right hip joint pitch	-71	-14839	82	17138
RLEG_JOINT4	4	Right knee	-1	-209	130	27170
RLEG_JOINT5	5	Right ankle pitch	-61	-12749	61	12749
RLEG_JOINT6	6	Right ankle roll	-25	-5225	25	5225
RARM_JOINT1	7	Right shoulder pitch	-91	-19019	151	31559
RARM_JOINT2	8	Right shoulder roll	-96	-20064	1	209
RARM_JOINT3	9	Right shoulder torsion	-91	-19019	91	19019
RARM_JOINT4	10	Right elbow	-1	-209	115	24035
LLEG_JOINT1	11	Left hip joint torsion	-31	-6479	91	19019
LLEG_JOINT2	12	Left hip joint roll	-21	-4389	31	6479
LLEG_JOINT3	13	Left hip joint pitch	-82	-17138	71	14839
LLEG_JOINT4	14	Left knee	-130	-27170	1	209
LLEG_JOINT5	15	Left ankle pitch	-61	-12749	61	12749
LLEG_JOINT6	16	Left ankle roll	-25	-5225	25	5225
LARM_JOINT1	17	Left shoulder pitch	-151	-31559	91	19019
LARM_JOINT2	18	Left shoulder roll	-1	-209	96	20064
LARM_JOINT3	19	Left shoulder torsion	-91	-19019	91	19019
LARM_JOINT4	20	Left elbow	-115	-24035	1	209
BODY_JOINT1	21	Waist pitch	-1	-209	90	18810
HEAD_JOINT1	22	Head torsion	-60	*	60	*
HEAD_JOINT2	22	Head pitch	-45	*	15	*
HEAD_JOINT3	22	Head roll	-15	*	15	*
LARM_JOINT5	23	Left fingers open/close	-60	*	60	*
RARM_JOINT5	23	Right fingers open/close	-60	*	60	*
LARM_JOINT6	23	Left hand torsion	-60	*	60	*
RARM_JOINT6	23	Right hand torsion	-60	*	60	*

**Tabla 2.3. Rango de movimiento permitido para cada articulación en el HOAP-3**

\* Consultar en la Tabla 2.2.

En la Tabla 2.3 se puede consultar el rango de movimiento permitido para las articulaciones del robot. Es importante señalar que en la práctica, por seguridad, deber reducirse este rango de movimientos en  $1^\circ$  por encima del mínimo y  $1^\circ$  por debajo del máximo. Siguiendo esta directriz, y a modo de ejemplo, el rango real de movimiento para la articulación RLEG\_JOINT1 quedará reducido a  $(-90^\circ, 30^\circ)$ .

### 2.2.3 Modelo VRML del robot

El modelo que empleamos en las simulaciones del robot en OpenHRP3 está programado en lenguaje VRML 2.0.

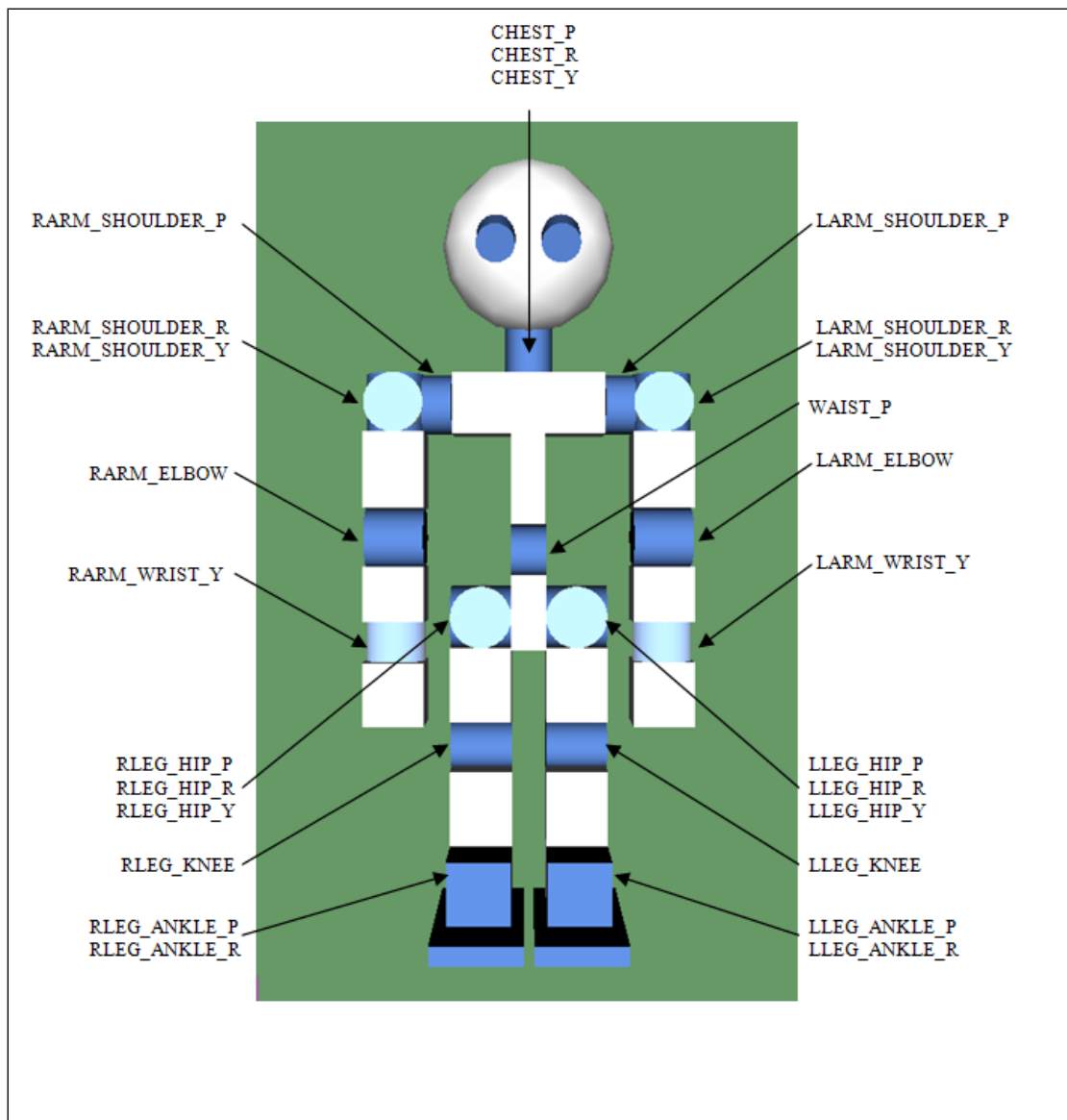


Figura 2.5 Articulaciones en el modelo VRML del HOAP-3 [11]

Articulación HOAP-3	Identificador HOAP-3	Articulación modelo VRML	Identificador modelo VRML
RLEG_JOINT1	1	RLEG_HIP_Y	0
RLEG_JOINT2	2	RLEG_HIP_R	1
RLEG_JOINT3	3	RLEG_HIP_P	2
RLEG_JOINT4	4	RLEG_KNEE	3
RLEG_JOINT5	5	RLEG_ANKLE_P	4
RLEG_JOINT6	6	RLEG_ANKLE_R	5
RARM_JOINT1	7	RARM_SHOULDER_P	6
RARM_JOINT2	8	RARM_SHOULDER_R	7
RARM_JOINT3	9	RARM_SHOULDER_Y	8
RARM_JOINT4	10	RARM_ELBOW	9
LLEG_JOINT1	11	LLEG_HIP_Y	10
LLEG_JOINT2	12	LLEG_HIP_R	11
LLEG_JOINT3	13	LLEG_HIP_P	12
LLEG_JOINT4	14	LLEG_KNEE	13
LLEG_JOINT5	15	LLEG_ANKLE_P	14
LLEG_JOINT6	16	LLEG_ANKLE_R	15
LARM_JOINT1	17	LARM_SHOULDER_P	16
LARM_JOINT2	18	LARM_SHOULDER_R	17
LARM_JOINT3	19	LARM_SHOULDER_Y	18
LARM_JOINT4	20	LARM_ELBOW	19
BODY_JOINT1	21	WAIST_P	20
HEAD_JOINT1	22	CHEST_Y	21
HEAD_JOINT2	22	CHEST_P	22
HEAD_JOINT3	22	CHEST_R	23
RARM_JOINT6	23	RARM_WRIST_Y	24
LARM_JOINT6	23	LARM_WRIST_Y	25

**Tabla 2.4. Correspondencia entre articulaciones del robot HOAP-3 y el modelo VRML**

Este modelo respeta los parámetros del robot, conservando las medidas y masas de cada una de las piezas originales [11]. Pero hay una diferencia importante que cabe señalar entre el modelo VRML y el robot. En el modelo no se han incluido los grados de libertad correspondientes a las manos. Éstos son las articulaciones LARM\_JOINT5 y RARM\_JOINT5, que sirven para abrir y cerrar los dedos permitiendo así agarrar objetos.

Por tanto, el modelo VRML presenta 26 GDL en lugar de los 28 que tiene el robot real. En la Figura 2.5 se puede observar el modelo con las articulaciones indicadas, y en la Tabla 2.4 la correspondencia entre las articulaciones del robot y las del modelo.

También hay que observar que mientras que la posición de reposo para el HOAP (es decir, donde todos los encoder de los motores marcan 0°) corresponde a la posición en que el robot tiene los brazos extendidos en perpendicular al tronco, en el modelo VRML se define la posición de reposo cuando el robot tiene los brazos extendidos paralelos al tronco. Estas diferencias pueden apreciarse examinando la Figura 2.4 y la Figura 2.5. De este modo, hay una diferencia de 90° entre ambos modelos en las articulaciones RARM\_SHOULDER\_P y LARM\_SHOULDER\_P.

En el caso de algunas articulaciones, el sentido positivo del movimiento del motor correspondiente no coincide con el sentido positivo del sistema coordinado de la Figura 2.4. En la Tabla 2.5 se ofrece la relación de articulaciones que presentan dicha circunstancia.

<b>HOAP-3</b>	<b>Modelo VRML</b>
RLEG_JOINT3	RLEG_HIP_P
RLEG_JOINT6	RLEG_ANKLE_R
RARM_JOINT4	RARM_ELBOW
LLEG_JOINT4	LLEG_KNEE
LLEG_JOINT5	LLEG_ANKLE_P
LLEG_JOINT6	LLEG_ANKLE_R
LARM_JOINT1	LARM_SHOULDER_P
BODY_JOINT1	WAIST_P

**Tabla 2.5. Relación de articulaciones con convenio de signos contrario en el HOAP-3**

## 2.2.4 Criterios para la interfaz

Debido a la naturaleza de los motores 22 y 23 con los que cuenta el HOAP-3, no es posible controlar el tiempo que tardarán las articulaciones asociadas a ellos en alcanzar una determinada posición. Su funcionamiento es distinto al de los primeros 21 motores. Para moverlos se asignan valores según la Tabla 2.2, y el robot mueve estas articulaciones a una velocidad constante hasta que alcanza la posición requerida. Después ejecuta la siguiente petición.

Este modo de funcionamiento de los motores no encaja con las características de la interfaz que aquí se presenta. Con la interfaz se quiere crear trayectorias por medio de la definición de coordenadas articulares en el tiempo, y esto no es posible para las articulaciones de la cabeza y las manos. Por tanto, no se podrá asignar valores a las articulaciones CHEST\_Y, CHEST\_P, CHEST\_R, RARM\_WRIST\_Y y LARM\_WRIST\_Y. La solución por la que se ha optado es por mantener estas articulaciones en reposo, en la posición 0°. Por este motivo, no han sido incluidas en la Tabla 2.6.

Para facilitar el diseño de trayectorias, se ha adaptado el programa de modo que el usuario sólo tiene que seguir el criterio positivo de los parámetros de Denavit-Hartenberg de la Figura 2.4. Aunque como ya se ha comentado en la Tabla 2.5 hay motores cuyo sentido de giro positivo es contrario a este criterio, el programa hace las modificaciones necesarias a los valores introducidos por el usuario para compensar esta diferencia.

Así mismo, como se comentó más arriba, el usuario debe considerar como posición inicial la posición inicial según el modelo VRML, reflejada en la Figura 2.5.

De este modo, el rango de movimientos que puede asignar el usuario a las articulaciones queda definido en la Tabla 2.6. No obstante, por seguridad, hay que respetar un margen de 1° tanto al definir el ángulo máximo como el mínimo. Al introducir los valores el usuario para diseñar la trayectoria, el programa comprobará que los valores respetan el rango teniendo en cuenta este margen.

Nótese también la diferencia de 90° con el robot real, en las articulaciones RARM\_SHOULDER\_P y LARM\_SHOULDER\_P, como se comentó en el apartado anterior.

Articulación	Id	Movimiento asociado	Valor mínimo	Valor máximo
RLEG_JOINT1	1	Right hip joint torsion	-91	31
RLEG_JOINT2	2	Right hip joint roll	-31	21
RLEG_JOINT3	3	Right hip joint pitch	-82	71
RLEG_JOINT4	4	Right knee	-1	130
RLEG_JOINT5	5	Right ankle pitch	-61	61
RLEG_JOINT6	6	Right ankle roll	-25	25
RARM_JOINT1	7	Right shoulder pitch	-181	61
RARM_JOINT2	8	Right shoulder roll	-96	1
RARM_JOINT3	9	Right shoulder torsion	-91	91
RARM_JOINT4	10	Right elbow	-115	1
LLEG_JOINT1	11	Left hip joint torsion	-31	91
LLEG_JOINT2	12	Left hip joint roll	-21	31
LLEG_JOINT3	13	Left hip joint pitch	-82	71
LLEG_JOINT4	14	Left knee	-1	130
LLEG_JOINT5	15	Left ankle pitch	-61	61
LLEG_JOINT6	16	Left ankle roll	-25	25
LARM_JOINT1	17	Left shoulder pitch	-181	61
LARM_JOINT2	18	Left shoulder roll	-1	96
LARM_JOINT3	19	Left shoulder torsion	-91	91
LARM_JOINT4	20	Left elbow	-115	1
BODY_JOINT1	21	Waist pitch	-90	1

**Tabla 2.6. Rango de movimiento de las articulaciones del HOAP-3 para el usuario**

## 2.2.5 Archivos de trayectorias

### 2.2.5.1 Archivos .dat:

Para el simulador, la trayectoria del robot viene definida por tres archivos que forman parte del controlador. Éstos indican los ángulos en radianes (*angle.dat*), velocidades en radianes/segundo (*vel.dat*) y aceleraciones en radianes/segundo<sup>2</sup> (*acc.dat*) que debe tomar cada una de las articulaciones a cada paso de simulación de la trayectoria completa [11].

El archivo *angle.dat* está compuesto por 27 columnas. La primera, es una secuencia de valores de tiempo que parte de cero y en cada fila se ve aumentada en el valor del paso de la simulación en milisegundos. Ésta será la base temporal de la simulación. Las otras 26 columnas corresponden a las 26 articulaciones del robot. En éstas, los valores de cada fila representan la posición que deberá marcar el encoder del motor correspondiente en el instante de tiempo que indica el valor de la primera columna para esa misma fila. El orden de las articulaciones es el indicado en la Tabla 2.7. En la Figura 2.6 se muestra el comienzo de un archivo *angle.dat*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	0	0.003340343	-0.308564191	0	0.796421296	-0.484934305	0.027891865		0	-0.167017153	8,35E+00	-0.734958982	0	0.003340343	-0.311486991
2	0.002	0.003262688	-0.308690561	-7,80E+00	0.796583023	-0.484986705	0.02789176		0	-0.167017153	8,35E+00	-0.734958982	0	0.003340343	-0.311486991
3	0.004	0.003189373	-0.308816904	-0.000155512	0.796745215	-0.485039505	0.027887312		0	-0.167017153	8,35E+00	-0.734958982	0	0.003344693	-0.311486956
4	0.006	0.003124228	-0.308943462	-0.000232656	0.796908375	-0.485092964	0.027874692		0	-0.167017153	8,35E+00	-0.734958982	0	0.003357223	-0.311486862
5	0.008	0.003066428	-0.309070182	-0.000309393	0.797072394	-0.485147015	0.027854726		0	-0.167017153	8,35E+00	-0.734958982	0	0.003377105	-0.311486711
6	0.01	0.003015468	-0.309197035	-0.000385725	0.797237202	-0.485201635	0.02782792		0	-0.167017153	8,35E+00	-0.734958982	0	0.003403833	-0.311486505
7	0.012	0.002971036	-0.309323997	-0.000461654	0.797402753	-0.485256791	0.027794587		0	-0.167017153	8,35E+00	-0.734958982	0	0.003437092	-0.311486244
8	0.014	0.002932936	-0.309451054	-0.000537182	0.797569017	-0.485312465	0.027754924		0	-0.167017153	8,35E+00	-0.734958982	0	0.003476685	-0.311485936
9	0.016	0.002901041	-0.309578197	-0.000612312	0.797735973	-0.485368658	0.027709059		0	-0.167017153	8,35E+00	-0.734958982	0	0.003522482	-0.311485572
10	0.018	0.002875265	-0.309705415	-0.000687044	0.797903602	-0.485425348	0.02765708		0	-0.167017153	8,35E+00	-0.734958982	0	0.003574396	-0.311485157
11	0.02	0.002855548	-0.309832714	-0.00076138	0.798071891	-0.485482531	0.027599046		0	-0.167017153	8,35E+00	-0.734958982	0	0.003632366	-0.311484684
12	0.022	0.002841842	-0.309960076	-0.000835321	0.798240826	-0.485540202	0.027535007		0	-0.167017153	8,35E+00	-0.734958982	0	0.003696342	-0.311484151
13	0.024	0.002834113	-0.310087495	-0.000908865	0.798410395	-0.485598352	0.027464998		0	-0.167017153	8,35E+00	-0.734958982	0	0.003766288	-0.311483567
14	0.026	0.002832328	-0.31021498	-0.000982024	0.798580587	-0.485656975	0.027389053		0	-0.167017153	8,35E+00	-0.734958982	0	0.00384217	-0.311482908
15	0.028	0.002836459	-0.310342513	-0.001054785	0.798751388	-0.485716064	0.0273072		0	-0.167017153	8,35E+00	-0.734958982	0	0.003923958	-0.311482188
16	0.03	0.002846448	-0.310470094	-0.001127164	0.798922788	-0.485775614	0.027219466		0	-0.167017153	8,35E+00	-0.734958982	0	0.004011625	-0.311481414
17	0.032	0.002862367	-0.310597716	-0.001199151	0.799094772	-0.485835616	0.027125878		0	-0.167017153	8,35E+00	-0.734958982	0	0.004105143	-0.311480654
18	0.034	0.002884094	-0.310725375	-0.00127075	0.799267329	-0.485896065	0.027026461		0	-0.167017153	8,35E+00	-0.734958982	0	0.004204488	-0.311479905
19	0.036	0.002911638	-0.310853065	-0.001341964	0.799440446	-0.485956952	0.02692124		0	-0.167017153	8,35E+00	-0.734958982	0	0.004309633	-0.311478591
20	0.038	0.002944975	-0.31098078	-0.001412794	0.799614108	-0.48601827	0.026810239		0	-0.167017153	8,35E+00	-0.734958982	0	0.004420553	-0.311477493
21	0.04	0.002984081	-0.311108515	-0.001483241	0.799788301	-0.486080013	0.026693483		0	-0.167017153	8,35E+00	-0.734958982	0	0.004537222	-0.311476305
22	0.042	0.003028934	-0.311236263	-0.001553305	0.799963012	-0.486142171	0.026570996		0	-0.167017153	8,35E+00	-0.734958982	0	0.004659617	-0.311475033

Figura 2.6. Vista parcial de un archivo de trayectorias *angle.dat* para el HOAP-3

De manera análoga, el archivo *vel.dat* y *acc.dat* mostrarán los valores de velocidad y aceleración respectivamente, que deben tomar las articulaciones a cada valor de tiempo de la primera columna.



<b>Articulación HOAP-3</b>	<b>Articulación modelo VRML</b>	<b>Columnas archivo .dat</b>	<b>Columnas archivo .csv</b>
RLEG_JOINT1	RLEG_HIP_Y	2	3
RLEG_JOINT2	RLEG_HIP_R	3	4
RLEG_JOINT3	RLEG_HIP_P	4	5
RLEG_JOINT4	RLEG_KNEE	5	6
RLEG_JOINT5	RLEG_ANKLE_P	6	7
RLEG_JOINT6	RLEG_ANKLE_R	7	8
RARM_JOINT1	RARM_SHOULDER_P	8	9
RARM_JOINT2	RARM_SHOULDER_R	9	10
RARM_JOINT3	RARM_SHOULDER_Y	10	11
RARM_JOINT4	RARM_ELBOW	11	12
LLEG_JOINT1	LLEG_HIP_Y	12	13
LLEG_JOINT2	LLEG_HIP_R	13	14
LLEG_JOINT3	LLEG_HIP_P	14	15
LLEG_JOINT4	LLEG_KNEE	15	16
LLEG_JOINT5	LLEG_ANKLE_P	16	17
LLEG_JOINT6	LLEG_ANKLE_R	17	18
LARM_JOINT1	LARM_SHOULDER_P	18	19
LARM_JOINT2	LARM_SHOULDER_R	19	20
LARM_JOINT3	LARM_SHOULDER_Y	20	21
LARM_JOINT4	LARM_ELBOW	21	22
BODY_JOINT1	WAIST_P	22	23
HEAD_JOINT1	CHEST_Y	23	24
HEAD_JOINT2	CHEST_P	24	24
HEAD_JOINT3	CHEST_R	25	24
RARM_JOINT6	RARM_WRIST_Y	26	25
LARM_JOINT6	LARM_WRIST_Y	27	25

**Tabla 2.7. Articulaciones en archivos de trayectorias para el HOAP-3**

### 2.2.5.2 Archivos .csv:

El robot puede funcionar en dos modos, online y offline. En el modo offline le cargaremos una secuencia de comandos que el robot irá ejecutando paso a paso. Estos archivos deben estar en formato *csv*, acrónimo del inglés '*comma separated value*', o archivo de valores separados por comas. Cada fila del archivo corresponde a un comando que se le manda al robot, especificándole el valor que debe tomar cada articulación. La frecuencia con la que el robot ejecuta cada comando es el paso de la simulación.

Cada comando está compuesto por 29 valores separados por comas. La estructura es la siguiente:

Modo de funcionamiento, paso de la simulación, posición motor 1, posición motor 2, posición motor 3, y así hasta indicar la posición de los 23 motores.

Como se puede comprobar observando la Tabla 2.7, los primeros 21 motores corresponden a las primeras 21 articulaciones (posiciones 3 a 23 en cada fila del archivo). Los motores 22 y 23 se controlan por medio de los valores asignados en las posiciones 24 y 25, respectivamente.

De este modo, atendiendo a la Tabla 2.5, según el valor asignado al motor 22 (posición 24 en cada fila del archivo) se estará moviendo una de las tres articulaciones de la cabeza y según el valor asignado al motor 23 (posición 25 del archivo) se estará moviendo una de las dos muñecas.

Como se comentaba en la sección anterior "Criterios para la interfaz", no podemos definir posiciones para algunas de las articulaciones mediante esta interfaz, pero en el archivo de trayectorias hay que asignarle valores a todas. Por tanto, al crear trayectorias por medio de la interfaz, el programa asignará automáticamente el valor de 0° a las articulaciones que no puede controlar el usuario, permaneciendo constante este valor durante toda la trayectoria.

Como estamos generando trayectorias para su uso en el modo offline, el valor del modo de funcionamiento debe ser "2" para todos los comandos de la trayectoria.

El valor del paso de la simulación que se puede elegir para crear trayectorias con esta interfaz puede tomar los valores 1 ms, 2 ms, 3 ms, 4 ms ó 5 ms, pero debe ser igual para todos los comandos de la trayectoria.

Hay que recordar que el valor de los ángulos debe estar en pulsos del encoder, como se explicó en el apartado anterior donde se detallan los procedimientos de conversión entre grados y pulsos.

Después de los 25 valores descritos más arriba, hay 4 valores que corresponden a los sensores. En el modo de funcionamiento offline, tomarán el valor R.

En la Figura 2.7 se puede observar la estructura que se ha indicado. En este ejemplo, el paso de la simulación elegido para la trayectoria es 2. A continuación se reflejan los valores de la primera línea.

2, 2, 0, 42, 3762, 9530, -5810, -334, 18810, -2006, 1, 8778, 0, 42, -3762,  
-9530, 5810, 424, -18810, 1998, 0, -8778, 418, 60, 60, R, R, R, R

Estos valores constituyen el primer comando que recibirá el robot. Éste se emplea para llevarlo a la posición inicial, y es igual para cualquier trayectoria generada para el HOAP-3 a través de esta interfaz.

	A	B	C	D	E	F	G	H	I
1	2,2,0,42,3762	9530,-5810,-334,18810,-2006,1,8778,0,42,-3762,-9530,5810,424,-18810,1998,0,-8778,418,60,60,R,R,R,R							
2	2,2,0,42,3762	9530,-5810,-334,18810,-2006,1,8778,0,42,-3762,-9530,5810,424,-18810,1998,0,-8778,418,60,60,R,R,R,R							
3	2,2,0,42,3762	9530,-5810,-334,18810,-2006,1,8778,0,42,-3762,-9530,5810,424,-18810,1998,0,-8778,418,60,60,R,R,R,R							
4	2,2,0,42,3762	9530,-5810,-334,18809,-2006,1,8778,0,42,-3762,-9530,5810,424,-18809,1998,0,-8778,418,60,60,R,R,R,R							
5	2,2,0,42,3762	9530,-5810,-334,18809,-2006,1,8778,0,42,-3762,-9530,5810,424,-18809,1998,0,-8778,418,60,60,R,R,R,R							
6	2,2,0,42,3762	9530,-5810,-334,18809,-2006,1,8778,0,42,-3762,-9530,5810,424,-18809,1998,0,-8778,418,60,60,R,R,R,R							
7	2,2,0,42,3762	9530,-5810,-334,18808,-2006,1,8778,0,42,-3762,-9530,5810,424,-18808,1998,0,-8778,418,60,60,R,R,R,R							
8	2,2,0,42,3762	9530,-5810,-334,18807,-2006,1,8778,0,42,-3762,-9530,5810,424,-18807,1998,0,-8778,418,60,60,R,R,R,R							
9	2,2,0,42,3762	9530,-5810,-334,18806,-2006,1,8778,0,42,-3762,-9530,5810,424,-18806,1998,0,-8778,418,60,60,R,R,R,R							
10	2,2,0,42,3762	9530,-5810,-334,18805,-2006,1,8778,0,42,-3762,-9530,5810,424,-18805,1998,0,-8778,418,60,60,R,R,R,R							
11	2,2,0,42,3762	9530,-5810,-334,18804,-2006,1,8778,0,42,-3762,-9530,5810,424,-18804,1998,0,-8778,418,60,60,R,R,R,R							
12	2,2,0,42,3762	9530,-5810,-334,18803,-2006,1,8778,0,42,-3762,-9530,5810,424,-18803,1998,0,-8778,418,60,60,R,R,R,R							
13	2,2,0,42,3762	9530,-5810,-334,18802,-2006,1,8778,0,42,-3762,-9530,5810,424,-18802,1998,0,-8778,418,60,60,R,R,R,R							
14	2,2,0,42,3762	9530,-5810,-334,18801,-2006,1,8778,0,42,-3762,-9530,5810,424,-18801,1998,0,-8778,418,60,60,R,R,R,R							
15	2,2,0,42,3762	9530,-5810,-334,18799,-2006,1,8778,0,42,-3762,-9530,5810,424,-18799,1998,0,-8778,418,60,60,R,R,R,R							
16	2,2,0,42,3762	9530,-5810,-334,18797,-2006,1,8778,0,42,-3762,-9530,5810,424,-18797,1998,0,-8778,418,60,60,R,R,R,R							
17	2,2,0,42,3762	9530,-5810,-334,18796,-2006,1,8778,0,42,-3762,-9530,5810,424,-18796,1998,0,-8778,418,60,60,R,R,R,R							
18	2,2,0,42,3762	9530,-5810,-334,18794,-2006,1,8778,0,42,-3762,-9530,5810,424,-18794,1998,0,-8778,418,60,60,R,R,R,R							
19	2,2,0,42,3762	9530,-5810,-334,18792,-2006,1,8778,0,42,-3762,-9530,5810,424,-18792,1998,0,-8778,418,60,60,R,R,R,R							

Figura 2.7. Vista parcial de un archivo de trayectorias con extensión csv



## **3 El robot RH-2**

### **3.1 La serie de humanoides RH**

En el año 2002, el grupo de investigación Robotics Lab [20] de la Universidad Carlos III de Madrid emprende la creación de un robot humanoide, lanzando el proyecto RH-0 (acrónimo de Robot Humanoide-0). Entre los años 2002 y 2004 se trabajará en la construcción de este prototipo.

El robot humanoide **RH-0** (Figura 3.1, izquierda) contaba con 23 grados de libertad, cada uno con un motor encargado de su movimiento. Los 23 GDL estaban distribuidos del siguiente modo: 6 GDL en cada pierna (2 GDL en tobillo, 1 GDL en rodilla y 3 GDL en cadera), 4 GDL en cada brazo (2 GDL en hombro, 1 GDL en codo y 1 GDL en muñeca), 1 GDL en el tronco y 2 GDL en la cabeza [14].

Entre los años 2005 y 2007, el Robotics Lab trabaja en la segunda versión del robot humanoide, el **RH-1** (Figura 3.1, centro). Este tiene 1.45 m de altura y pesa 48.5 kg. Igual que su predecesor, tiene 21 grados de libertad.

Tanto el RH-0 como el RH-1 se desarrollaron tomando como modelo los prototipos más avanzados del momento (ASIMO de Honda y HRP-2P de Kawada) [14].

Aunque el **RH-1** mantiene el tamaño y grados de libertad del RH-0, introduce algunos cambios en el hardware. Su estructura se diseña en aluminio aeronáutico, que es un material fuerte y ligero. Tiene sensores de inercia, una cámara que incluye baterías y una conexión inalámbrica con la estación de trabajo. Tiene una autonomía de 30

minutos. La carcasa del robot se diseña principalmente para ocultar y proteger el hardware, y también para mejorar la estética del robot humanoide.

Puede caminar a una velocidad de 0.7 km/h. Puede responder a órdenes transmitidas por voz y gestos, gracias al micrófono que se conecta al robot por USB, y a la cámara que tiene colocada en su cabeza. A través de la información recogida por la cámara, puede calcular la distancia al usuario para reproducir el movimiento más adecuado para acercarse a él. También puede reconocer al usuario cruzando la imagen que recibe a través de la cámara con una base de datos de caras [10].

El principal problema del robot RH-1 [13] es que en cada paso debe soportar una fuerza de impacto de entre 700 y 800 Newton por segundo, lo que con el paso del tiempo deteriora de manera importante las articulaciones de las piernas. Por este motivo, y con el ánimo de buscar mejoras como la habilidad de subir y bajar escaleras, surge la idea de desarrollar la siguiente versión del humanoide, el **RH-2** (en la Figura 3.1, a la derecha se puede observar la parte inferior del humanoide). Este robot tendrá una altura de aproximadamente 1.65 m, altura que se asemeja más a la de un humano adulto.

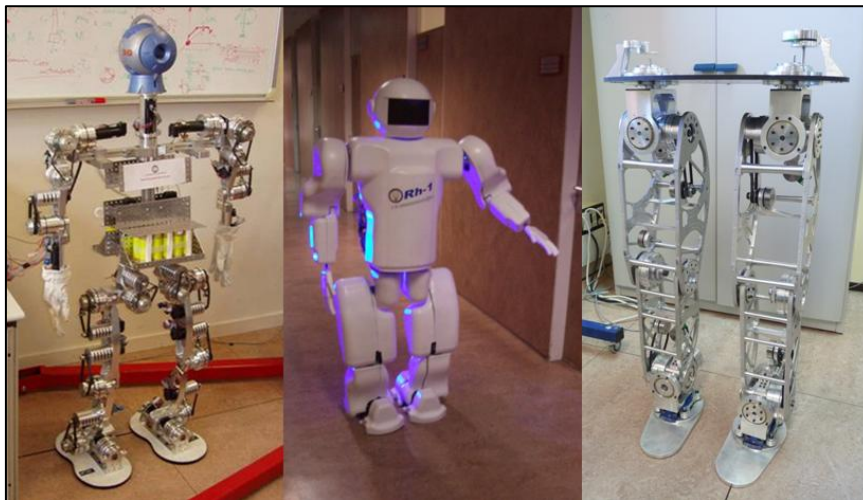


Figura 3.1. De izquierda a derecha, RH-0, RH-1 y RH-2

## 3.2 RH-2

### 3.2.1 Aspectos generales

En 2007 se comienza a trabajar en el diseño del RH-2, un robot del que se espera que pueda llegar a subir y bajar escalones, transportar objetos, etc. Se pretende además que camine de manera natural, pueda oír, imitar gestos y aprender habilidades.

El montaje del robot en su totalidad es una tarea compleja y es difícil que todo funcione perfectamente. Por eso se quiere realizar una primera fase del proyecto RH-2 para construir la parte inferior, las piernas, y construir un andador [14].

Actualmente (abril 2010), ya se ha construido y montado la parte inferior del robot. Una vez probado el andador y conseguida una caminata muy estable, se dará paso a la segunda fase. Se concretarán los detalles de diseño y se construirá la parte superior, haciendo hincapié en la relación del robot con el entorno. Para ello se definirán los sensores exoceptivos necesarios, cámaras, micrófonos, etc., y se estudiará la opción de incluir un microprocesador específico para el tratamiento de imagen y sonido [14].

### 3.2.2 Control del robot

El RH-2 tiene 28 grados de libertad. En la Tabla 3.1 se indican las articulaciones del robot y los movimientos asociados a las mismas en función de los ejes *Roll*, *Pitch* y *Yaw* referenciados en la Figura 2.3.

Articulación	Identificador	Movimiento
RLEG_HIP_R	1	Roll cadera
RLEG_HIP_P	2	Pitch cadera
RLEG_HIP_Y	3	Yaw cadera
RLEG_KNEE	4	Rodilla
RLEG_ANKLE_P	5	Pitch tobillo
RLEG_ANKLE_R	6	Roll tobillo
RARM_SHOULDER_P	7	Pitch brazo
RARM_SHOULDER_R	8	Roll brazo
RARM_SHOULDER_Y	9	Yaw brazo
RARM_ELBOW	10	Codo
RARM_WRIST_Y	11	Yaw muñeca
RARM_WRIST_R	12	Roll muñeca
LLEG_HIP_R	13	Roll cadera
LLEG_HIP_P	14	Pitch cadera
LLEG_HIP_Y	15	Yaw cadera
LLEG_KNEE	16	Rodilla
LLEG_ANKLE_P	17	Pitch tobillo
LLEG_ANKLE_R	18	Roll tobillo
LARM_SHOULDER_P	19	Pitch brazo
LARM_SHOULDER_R	20	Roll brazo
LARM_SHOULDER_Y	21	Yaw brazo
LARM_ELBOW	22	Codo
LARM_WRIST_Y	23	Yaw muñeca
LARM_WRIST_R	24	Roll muñeca
WAIST_P	25	Pitch cintura
WAIST_Y	26	Yaw cintura
CHEST_Y	27	Yaw cabeza
CHEST_P	28	Pitch cabeza

**Tabla 3.1. Movimiento asociado a cada articulación en el RH-2**

Como se desprende de la Tabla 3.1 la distribución de grados de libertad en el robot es la siguiente: 6 GDL en cada pierna, 6 GDL en cada brazo, 2 GDL en la cintura y 2 en la cabeza. En la Figura 3.2 se indica la localización de los grados de libertad y el criterio de signos positivo según Denavit-Hartenberg.



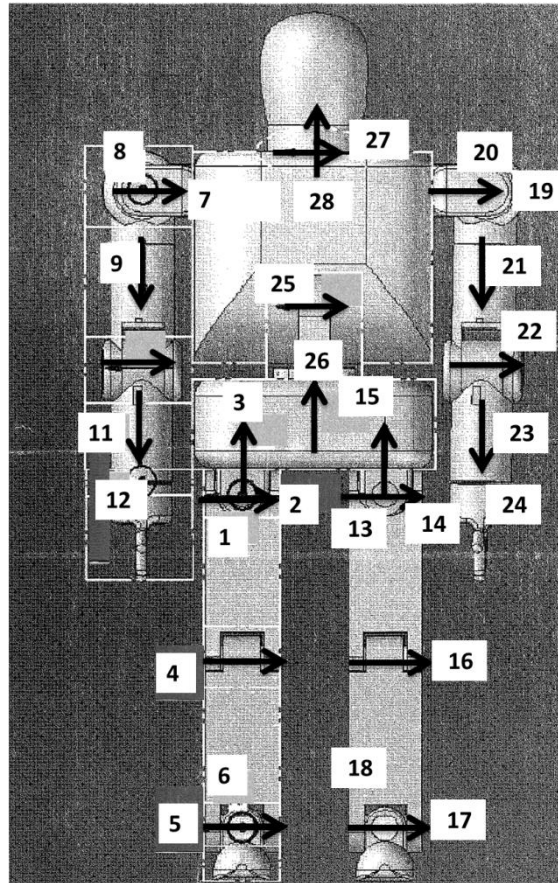


Figura 3.2. Distribución de GDL en el RH-2

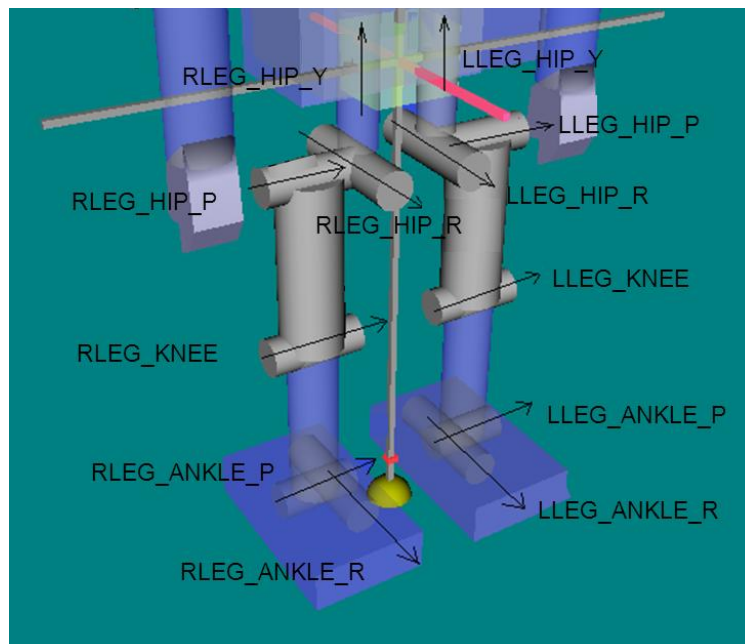


Figura 3.3. Articulaciones de las piernas del RH-2

En la Tabla 3.2 se indica el rango de movimientos permitido para cada articulación. Por el momento deben ser tomados como valores orientativos. A medida que se vaya desarrollando el humanoide, se irán concretando estos valores.

Articulación	Id	Movimiento	Ángulo mínimo	Ángulo máximo
RLEG_HIP_R	1	Roll cadera	-40	30
RLEG_HIP_P	2	Pitch cadera	-120	45
RLEG_HIP_Y	3	Yaw cadera	-30	24
RLEG_KNEE	4	Rodilla	0	150
RLEG_ANKLE_P	5	Pitch tobillo	-85	85
RLEG_ANKLE_R	6	Roll tobillo	-20	20
RARM_SHOULDER_P	7	Pitch brazo	-180	180
RARM_SHOULDER_R	8	Roll brazo	-60	60
RARM_SHOULDER_Y	9	Yaw brazo	-45	120
RARM_ELBOW	10	Codo	-100	100
RARM_WRIST_Y	11	Yaw muñeca	-105	105
RARM_WRIST_R	12	Roll muñeca	-40	55
LLEG_HIP_R	13	Roll cadera	-40	30
LLEG_HIP_P	14	Pitch cadera	-120	45
LLEG_HIP_Y	15	Yaw cadera	-30	24
LLEG_KNEE	16	Rodilla	0	150
LLEG_ANKLE_P	17	Pitch tobillo	-85	85
LLEG_ANKLE_R	18	Roll tobillo	-20	20
LARM_SHOULDER_P	19	Pitch brazo	-180	180
LARM_SHOULDER_R	20	Roll brazo	-60	60
LARM_SHOULDER_Y	21	Yaw brazo	-45	120
LARM_ELBOW	22	Codo	-100	100
LARM_WRIST_Y	23	Yaw muñeca	-105	105
LARM_WRIST_R	24	Roll muñeca	-40	55
WAIST_P	25	Pitch cintura	Ángulo por definir	Ángulo por definir
WAIST_Y	26	Yaw cintura	Ángulo por definir	Ángulo por definir
CHEST_Y	27	Yaw cabeza	Ángulo por definir	Ángulo por definir
CHEST_P	28	Pitch cabeza	Ángulo por definir	Ángulo por definir

**Tabla 3.2. Rango de movimiento permitido para cada articulación**

Tanto en el RH-0 como en el RH-1 se leía el encoder del motor sin tener datos de la posición absoluta en la que realmente se encontraba cada articulación. Es necesario un sistema para medir el error de posición entre la posición leída del motor y la posición real a la salida de las transmisiones. La implementación de los sensores de sincronismo para realizar el *homing* y buscar la posición de referencia de los motores no se llevó a cabo. Para ello será necesario introducir otro encoder en cada transmisión y diseñar el circuito de control y adquisición de datos. Otra de las medidas necesarias es la fuerza que ejercen los pies sobre el suelo. En los proyectos anteriores no se incluían estos sensores. Sin este parámetro y sin el error de posición de las articulaciones no se puede completar el ciclo de control de la caminata [14].

En la medida de lo posible, el ciclo de control de la caminata del robot debe cerrarse en el mínimo tiempo posible. Es imprescindible que se pueda realizar este lazo de control; el robot debe funcionar en un lazo cerrado para mantener siempre la estabilidad. Recurrir a sistemas lo más distribuidos posibles nos aseguran tiempos de cómputo más reducidos [14].

Los microprocesadores usados anteriormente se basaban en el protocolo ISA para la comunicación interna. Estos dispositivos fueron problemáticos y ahora se estudiarán distintos protocolos internos como PCI y se buscarán otras opciones de microprocesadores [14].

La información de los sensores que gestionan los microprocesadores debe ser actualizada en tiempo real. Las consultas entre los microprocesadores debe ser lo más rápida posible. El acceso desde un medio externo al sistema también debe ser eficaz para posibilitar un control del robot desde Internet u otro medio físico. Se han de estudiar las posibilidades de comunicación alámbrica e inalámbrica disponibles actualmente [14].

En definitiva se busca una solución para los problemas derivados de las anteriores versiones y, teniéndolos en cuenta, diseñar un nuevo sistema actualizado acorde a las nuevas tecnologías disponibles [14].

### 3.2.3 Modelo VRML del robot

El modelo VRML del robot RH-2 se ha diseñado tratando de reproducir con la máxima fidelidad el robot real. Como ya se ha comentado para el robot HOAP-3 en el capítulo anterior, cuanto más preciso es un modelo, más representativas de la realidad serán las simulaciones que se realicen con él. En este caso se han incluido todas las articulaciones del robot y se han respetado los parámetros originales, conservando las medidas de las piezas que lo forman, así como las masas de las articulaciones.

En la Figura 3.4 se puede observar este modelo. Además aparecen referenciadas cada una de las articulaciones con sus respectivos nombres.

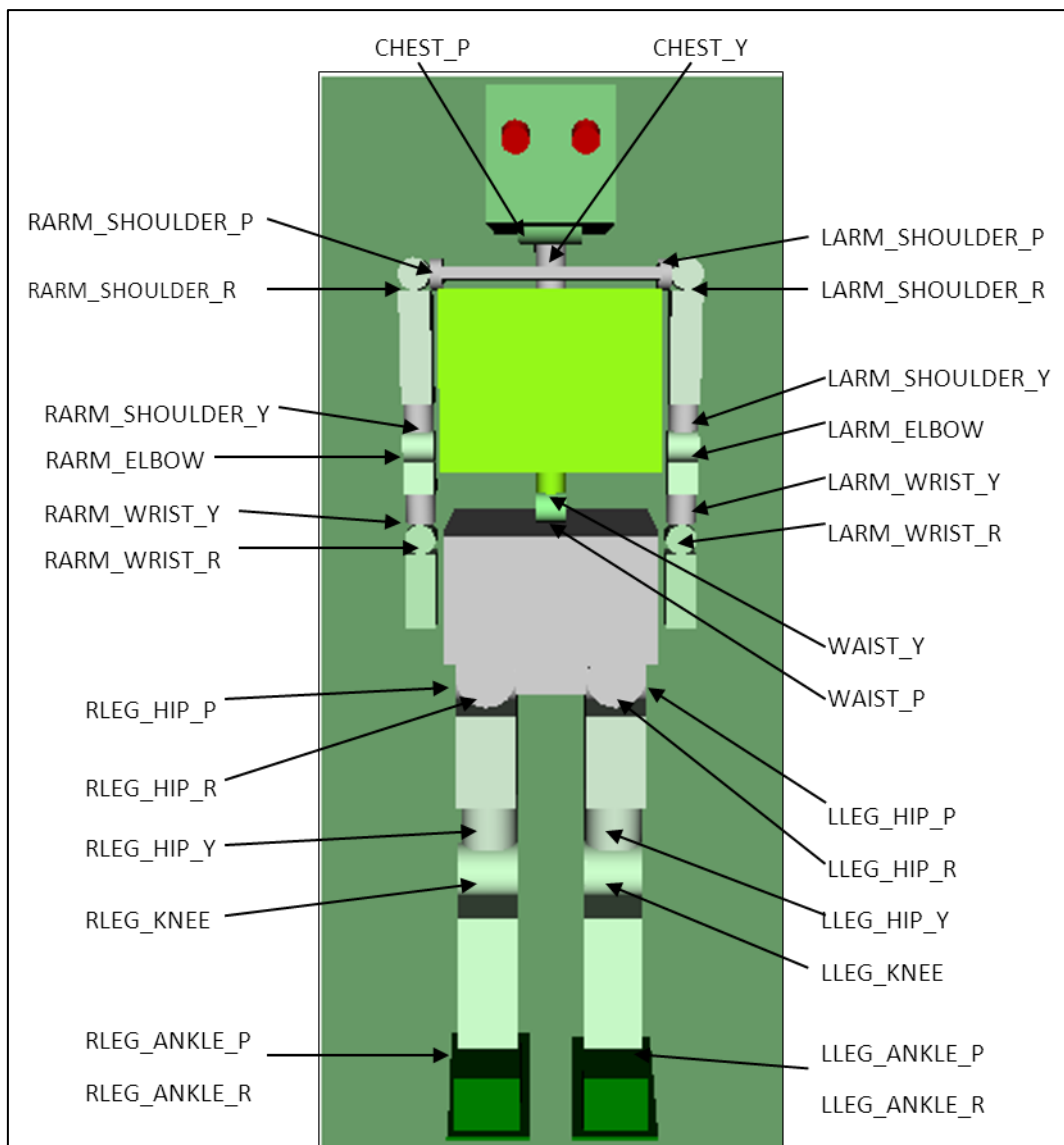


Figura 3.4. Modelo VRML del robot RH-2 [13]

Articulación RH-2	Identificador RH-2	Identificador modelo VRML
RLEG_HIP_R	1	0
RLEG_HIP_P	2	1
RLEG_HIP_Y	3	2
RLEG_KNEE	4	3
RLEG_ANKLE_P	5	4
RLEG_ANKLE_R	6	5
RARM_SHOULDER_P	7	6
RARM_SHOULDER_R	8	7
RARM_SHOULDER_Y	9	8
RARM_ELBOW	10	9
RARM_WRIST_Y	11	10
RARM_WRIST_R	12	11
LLEG_HIP_R	13	12
LLEG_HIP_P	14	13
LLEG_HIP_Y	15	14
LLEG_KNEE	16	15
LLEG_ANKLE_P	17	16
LLEG_ANKLE_R	18	17
LARM_SHOULDER_P	19	18
LARM_SHOULDER_R	20	19
LARM_SHOULDER_Y	21	20
LARM_ELBOW	22	21
LARM_WRIST_Y	23	22
LARM_WRIST_R	24	23
WAIST_P	25	24
WAIST_Y	26	25
CHEST_Y	27	26
CHEST_P	28	27

**Tabla 3.3. Correspondencia entre articulaciones del robot y el modelo**

En la Tabla 3.3 se muestra la relación entre los identificadores de las articulaciones del robot y del modelo en VRML.

### 3.2.4 Archivos de trayectorias

Para el simulador, la trayectoria del robot viene definida por tres archivos, que forman parte del controlador. Éstos indican los ángulos en radianes (*angle.dat*), velocidades en radianes/segundo (*vel.dat*) y aceleraciones en radianes/segundo<sup>2</sup> (*acc.dat*) que debe tomar cada una de las articulaciones a cada paso de simulación de la trayectoria completa [11].

El archivo *angle.dat* está compuesto por 29 columnas. La primera, es una secuencia de valores de tiempo que parte de cero y en cada fila se incrementa el valor del paso de la simulación en milisegundos. Esta será la base temporal de la simulación. Las otras 28 columnas corresponden a las 28 articulaciones del robot. En éstas, los valores de cada fila representan la posición que deberá marcar el encoder del motor correspondiente en el instante de tiempo que indica el valor de la primera columna para esa misma fila. El orden de las articulaciones es el indicado en la Tabla 3.4.

En el ejemplo de la Figura 3.5 los brazos están levantados 90°, quedando en posición perpendicular al tronco. Las articulaciones LARM\_SHOULDER\_P (columna 8) y RARM\_SHOULDER\_P (columna 18) toman el valor de -90° (-1.57 rad), signo negativo debido al criterio de signos que se define con los parámetros de Denavit- Hartenberg de la Figura 3.2.

G	H	I	J	K	L	M	N	O	P	Q	R	S	T
0	-1.57079633	0	0	0	0	0	0	0	0	0	0	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-7.13E-06	0	1.43E-05	-7.13E-06	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-2.84E-05	0	5.69E-05	-2.84E-05	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-6.39E-05	0	0.00012774	-6.39E-05	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00011332	0	0.00022664	-0.00011332	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00017671	0	0.00035342	-0.00017671	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00025395	0	0.0005079	-0.00025395	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00034495	0	0.00068991	-0.00034495	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00044964	0	0.00089928	-0.00044964	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00056793	0	0.00113585	-0.00056793	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00069972	0	0.00139945	-0.00069972	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00084495	0	0.0016889	-0.00084495	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00100352	0	0.00200704	-0.00100352	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00117535	0	0.0023507	-0.00117535	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00136035	0	0.0027207	-0.00136035	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00155845	0	0.00311689	-0.00155845	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00176955	0	0.0035391	-0.00176955	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00199357	0	0.00398715	-0.00199357	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00223044	0	0.00446088	-0.00223044	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00248006	0	0.00496013	-0.00248006	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00274236	0	0.00548472	-0.00274236	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00301724	0	0.00603449	-0.00301724	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00330464	0	0.00660928	-0.00330464	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00360446	0	0.00720891	-0.00360446	0	-1.57079633

Figura 3.5. Vista parcial de un archivo de trayectorias *angle.dat* para el RH-2 [12]

<b>Articulaciones RH-2</b>	<b>Identificador RH-2</b>	<b>Identificador modelo VRML</b>	<b>Columnas archivo .dat</b>
RLEG_HIP_R	1	0	2
RLEG_HIP_P	2	1	3
RLEG_HIP_Y	3	2	4
RLEG_KNEE	4	3	5
RLEG_ANKLE_P	5	4	6
RLEG_ANKLE_R	6	5	7
RARM_SHOULDER_P	7	6	8
RARM_SHOULDER_R	8	7	9
RARM_SHOULDER_Y	9	8	10
RARM_ELBOW	10	9	11
RARM_WRIST_Y	11	10	12
RARM_WRIST_R	12	11	13
LLEG_HIP_R	13	12	14
LLEG_HIP_P	14	13	15
LLEG_HIP_Y	15	14	16
LLEG_KNEE	16	15	17
LLEG_ANKLE_P	17	16	18
LLEG_ANKLE_R	18	17	19
LARM_SHOULDER_P	19	18	20
LARM_SHOULDER_R	20	19	21
LARM_SHOULDER_Y	21	20	22
LARM_ELBOW	22	21	23
LARM_WRIST_Y	23	22	24
LARM_WRIST_R	24	23	25
WAIST_P	25	24	26
WAIST_Y	26	25	27
CHEST_Y	27	26	28
CHEST_P	28	27	29

**Tabla 3.4. Articulaciones en archivos de trayectorias del RH-2**

Puesto que el robot está en fase de construcción, aún no se sabe exactamente el modo en que se le cargarán las trayectorias. En el caso del robot HOAP-3, se hace mediante archivos con extensión .csv, pero para el RH-2 no está definido aún. Por tanto, en esta interfaz nos limitaremos a generar archivos de trayectorias con



extensión *.dat* para usar en el simulador, quedando abierta la posibilidad de introducir los cambios necesarios en el programa para poder guardar las trayectorias en otro formato cuando se defina.



## **4 Funcionamiento de la interfaz**

### **4.1 Introducción**

La interfaz se ha diseñado con el objeto de cumplir las siguientes funcionalidades:

- Generación de nuevas trayectorias mediante la definición de valores angulares para las articulaciones del robot.
- Generación de nuevas trayectorias mediante la unión de otras existentes.
- Obtención de los correspondientes archivos de velocidad y aceleración a partir de los archivos de posición de una trayectoria.
- Estudio de trayectorias existentes.
- Conversión entre los formatos de archivo con que trabajan el robot real y el simulador OpenHRP3.

La herramienta pretende ser una plataforma de trabajo para los robots HOAP-3 y RH-2. Las características de dichos robots que es necesario conocer para diseñar trayectorias se detallan en los capítulos 2 y 3.

Esta interfaz (Figura 4.1) ha sido programada en MATLAB. En el capítulo 5, “Diseño de la interfaz” se explicará detalladamente y a nivel de programación el diseño de la misma.

La interfaz se estructura en pestañas que permiten presentar cada funcionalidad en una pantalla distinta. En cada una de estas pantallas se dispone de un panel que permite guardar las gráficas que se representen. Este proceso es análogo en todos los casos. Por ello se comentará en detalle una única vez y en una sección independiente.



Figura 4.1. Pantalla principal de la interfaz

## 4.2 Archivos necesarios

### 4.2.1 Equipos con el programa MATLAB instalado

En caso de tener el programa MATLAB instalado en el equipo de trabajo, será suficiente con hacer clic en el ejecutable “Generador\_trayectorias.exe” que se encuentra en la carpeta “Generador de trayectorias” del CD adjunto.

### 4.2.2 Equipos que no tienen instalado el programa MATLAB

Si el equipo no cuenta con el programa MATLAB, será necesario instalar las librerías del programa para poder trabajar con los ejecutables. Para ello, hay que hacer uso del MCR de MATLAB (*MATLAB Compiler Runtime*). En la carpeta “Generador de trayectorias”, hacer clic en el archivo “Generador\_trayectorias.pkg”. Esto pondrá en marcha el instalador del compilador, MCRInstaller. Éste programa instala el MCR si aún no está instalado.

Es importante que la versión del MCR que permitirá ejecutar el programa sea compatible con la versión del compilador de MATLAB donde se creó el ejecutable (MATLAB R2009a). Una vez instalado correctamente el compilador en tiempo real, para usar la interfaz solo habrá que hacer clic en el icono “Generador\_trayectorias.exe”.

### 4.2.3 Plataformas compatibles

La interfaz se ha programado en MATLAB versión R2009a, para sistemas operativos Windows de 32 bits. Por tanto, será compatible con otras versiones de Windows de 32 bits, pero no de 64 bits.

## 4.3 Generación de trayectorias

### 4.3.1 Aspectos generales

El objetivo es crear trayectorias a través de las cuales el robot describa los movimientos deseados por el usuario, para una vez creadas poder guardarlas en un formato que pueda ser utilizado por el simulador y en un formato que pueda ser utilizado por el robot.

Como se comentó en el capítulo anterior, el robot HOAP-3 trabaja con archivos en formato `.csv` (del inglés *'comma separated value'*, es decir, "archivo de valores separados por comas"). El simulador que emplearemos para realizar las simulaciones presentadas en este documento es OpenHRP3 y trabaja con archivos en formato `.dat`.

En el caso del robot RH-2, que está en fase de construcción, aún no se ha definido el formato de archivo que utilizará para ejecutar las trayectorias. Por el momento sólo se pueden realizar pruebas en plataformas de simulación, por lo tanto, en lo que respecta a este robot, nos limitaremos a trabajar con archivos en extensión `.dat` a nivel simulador OpenHRP3.

En la Figura 4.2 se ofrece la vista general de la pantalla de creación de nuevas trayectorias para el HOAP-3. La pantalla correspondiente al RH-2 es idéntica.

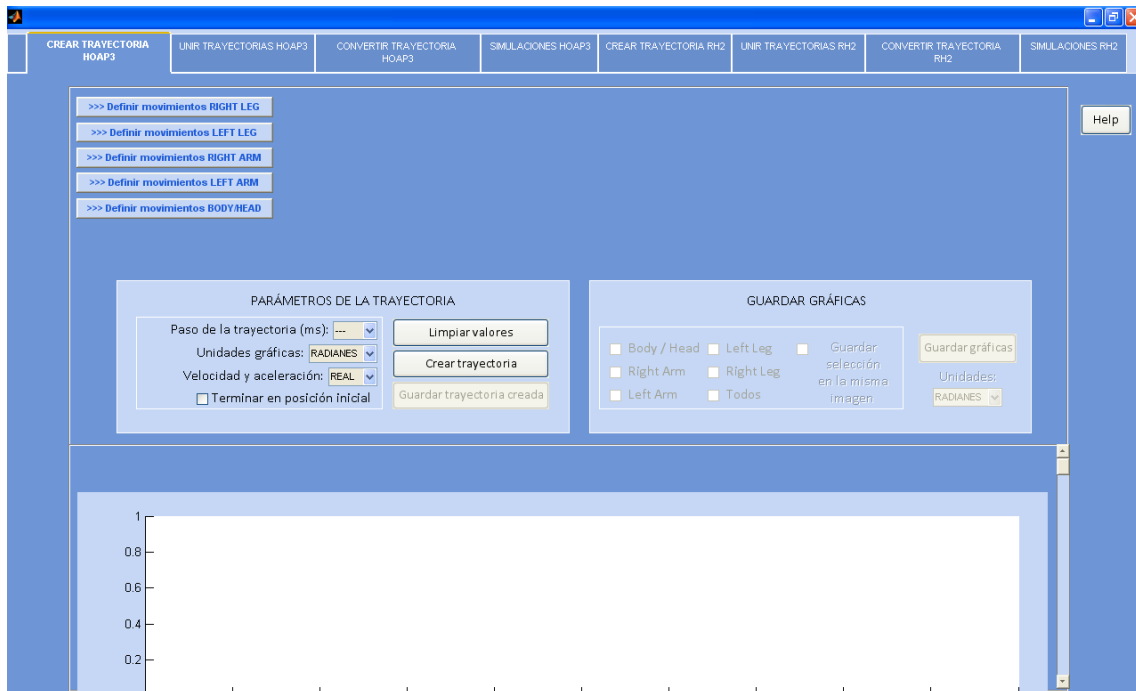


Figura 4.2. Pantalla de creación de nuevas trayectorias para el HOAP-3

### 4.3.2 Selección de parámetros de la trayectoria

Para crear la trayectoria es necesario indicar antes una serie de parámetros. En primer lugar, hay que indicar el paso de la trayectoria, que marcará la frecuencia con la que se envía una nueva instrucción al robot. Así mismo, este valor será el paso para la simulación en OpenHRP3. Puede tomar los siguientes valores: 1 ms, 2ms, 3ms, 4 ms y 5 ms.

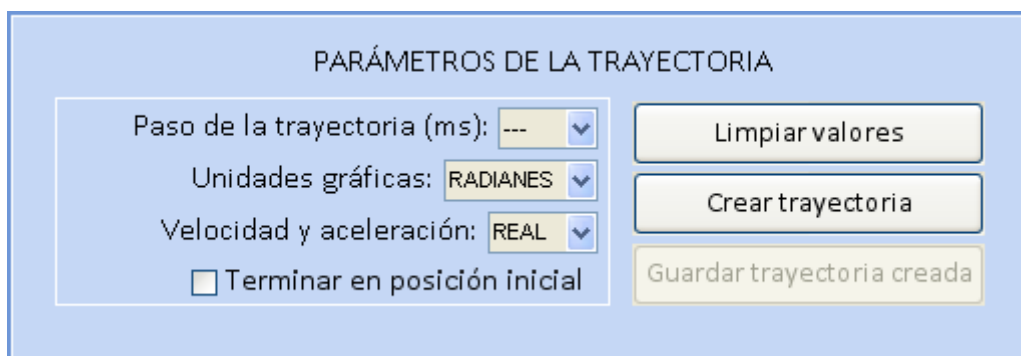
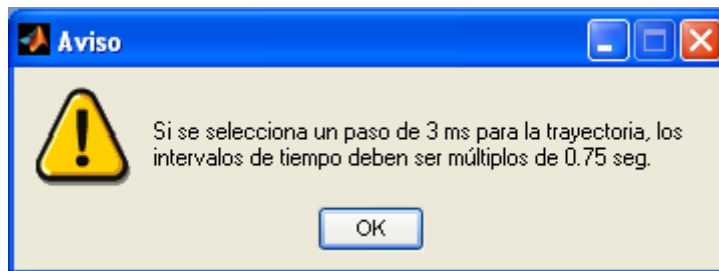


Figura 4.3. Parámetros de la trayectoria

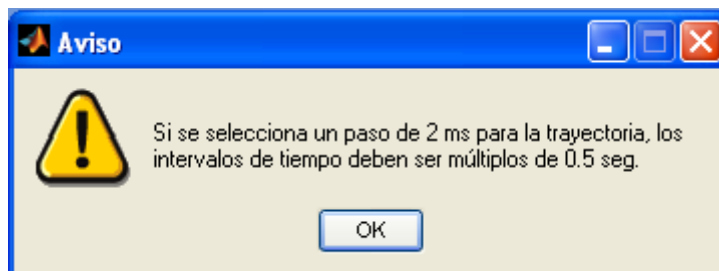
La selección del paso para la trayectoria lleva asociada ciertas limitaciones a la hora de elegir los intervalos de tiempo para los movimientos. Esto se debe a que hay que evitar intervalos de tiempo que no se puedan dividir en un número exacto de pasos. Al dejarse libertad total para definir por separado los movimientos de cada articulación, estas presentarán un número de movimientos que no tiene por qué coincidir entre sí.

Por cada uno de estos movimientos en que la división entre el intervalo de tiempo y el tiempo de muestreo no fuera exacta, se estarían perdiendo pasos para esa articulación. De este modo, la trayectoria tendría más pasos para unas articulaciones que para otras. Para evitar este tipo de situaciones, se fuerza que los intervalos de tiempo sean siempre múltiplos del tiempo de muestreo. Así, si se selecciona un paso de 3ms se obtendrá un mensaje de aviso como el de la Figura 4.4.



**Figura 4.4. Selección de paso 3 ms**

En cambio, si se selecciona un paso distinto de 3 ms, el mensaje será el de la Figura 4.5.



**Figura 4.5. Selección de paso distinta de 3 ms**

Hay que seleccionar las unidades en que se desea visualizar los valores que toman las articulaciones: radianes o grados. Es importante señalar que esto sólo afecta a la representación gráfica que se realiza a continuación para facilitar al usuario el diseño de la trayectoria, y en ningún caso afecta a las unidades en que se almacenan los datos en los archivos. En el caso del archivo *.dat* se almacenan en radianes porque es la unidad con la que trabaja el simulador, y en el caso del *.csv* se almacenan en pulsos, que es el modo en que el HOAP3 controla sus motores.

En algunos casos especiales, en que no se mueven las articulaciones de las piernas, pueden realizarse las simulaciones en OpenHRP3 con velocidad y aceleración nulas, dando sólo los valores de posición por medio del archivo *angle.dat* y cargando los archivos *vel.dat* y *acc.dat* con primera columna (paso de simulación) idéntica y el resto

de columnas a cero. Para ello, se ofrece la posibilidad al crear la trayectoria de generar estos archivos automáticamente. Como se puede ver en la Figura 4.3, se puede seleccionar “NULO” en el menú desplegable “Velocidad y aceleración”. Por defecto, este menú presenta la selección “REAL”, que permite obtener los valores de velocidad y aceleración reales para la trayectoria.

Es posible también imponer que la trayectoria termine con todas las articulaciones en la posición inicial. En caso de haber seleccionado un paso de 3 ms, el robot empleará 1.5 segundos en volver a la posición inicial. Si el paso es cualquier otro valor, será 1 segundo el tiempo para volver a la posición inicial. Esta limitación se debe a la necesidad de que todos los intervalos de tiempo se dividan en un número exacto de pasos.

Como podemos observar en la Figura 4.3, el botón “Guardar trayectoria creada” permanece deshabilitado hasta que se ha creado la trayectoria.

### 4.3.3 Definir los movimientos de las articulaciones

Para definir el movimiento que se desea que describa cada articulación, hay que hacer clic en los paneles desplegable de la esquina superior izquierda (Ver Figura 4.6).

Para ambos robots, las articulaciones están agrupadas en 5 partes del cuerpo del humanoide: *Right Leg*, *Left Leg*, *Right Arm*, *Left Arm* y *Body/Head*.



Figura 4.6. Definir movimientos

Según de qué robot se trate, los menús desplegables mostrarán la distribución de articulaciones propia de dicho robot. Por ejemplo, si se selecciona *Right Leg* del HOAP-3, se despliega un menú como el de la Figura 4.7.

>>> Definir movimientos RIGHT LEG																		
RLEG HIP Y	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG HIP R	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG HIP P	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	-18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG KNEE	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG ANKLE P	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	-28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG ANKLE R	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)

Figura 4.7. Menú *Right Leg*

Podemos observar que hay dos filas de cuadros de texto editables para cada articulación. La fila superior corresponde a valores de tiempo, y la inferior a valores de ángulos. Estos pares de valores tiempo-ángulo irán definiendo puntos de la trayectoria. A nivel interno, el programa hará una interpolación cúbica entre cada punto y el siguiente en la línea temporal. Al hacer esto para todos los puntos, se define una trayectoria completa.

El primer par de valores viene definido de antemano, e indica la posición inicial a la que por seguridad llevaremos al robot al comenzar cada trayectoria, evitando así comportamientos inesperados y problemas derivados de que no se encuentre en la posición de reposo al cargar la trayectoria. En el caso del RH-2, esa posición es 0° para todas las articulaciones.

Al introducirse un valor cualquiera de tiempo o de posición, se comprobará si se ha introducido previamente un valor para el paso. Si no se ha indicado el valor del paso, se mostrará el mensaje de error de la Figura 4.8.

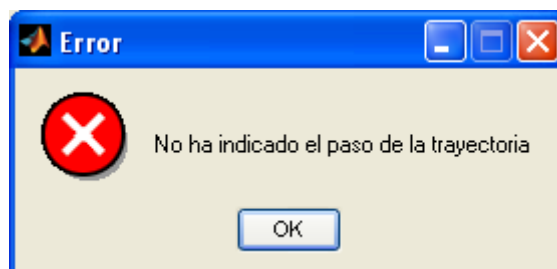


Figura 4.8. Mensaje falta el valor del paso

Los valores de posición que se pueden introducir deben pertenecer al rango descrito en los capítulos anteriores (Tabla 2.6 para el HOAP-3 y Tabla 3.2 para el RH-2). Hay que respetar el margen de  $1^\circ$  por encima del mínimo y  $1^\circ$  por debajo del máximo sobre la tabla del HOAP-3. El criterio a seguir para identificar los ángulos en el HOAP-3 es el de la Figura 2.4, pero considerando los brazos extendidos en paralelo al tronco como posición de reposo. Para el RH-2 el criterio es el de la Figura 3.2 y la Figura 3.3.

Al introducir un valor de posición se comprobará que éste está dentro del rango permitido. Si no lo está se mostrará un mensaje de error como el de la Figura 4.9, en el que se indica el rango permitido para la articulación en cuestión.

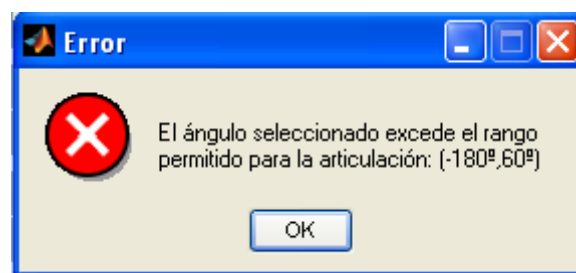


Figura 4.9. Mensaje para ángulo fuera de rango

Al introducir un **valor de tiempo**, si el valor del intervalo de tiempo que se está definiendo (la diferencia entre ese punto y el anterior) no es compatible con este valor, se mostrará un mensaje de error como el de la Figura 4.10.

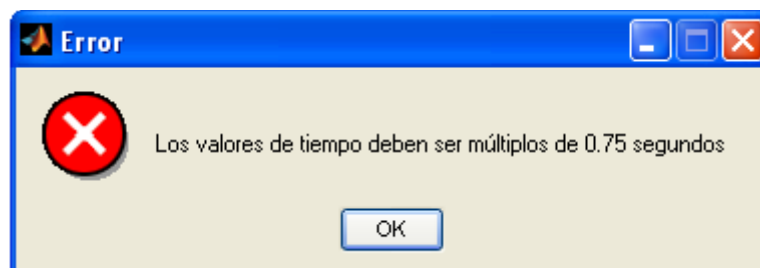


Figura 4.10. Mensaje valor de tiempo erróneo para paso 3 ms

Para cada articulación se puede definir el número de pares de puntos que se desee, no teniendo que ser igual para todas. El tiempo total de la trayectoria vendrá dado por la articulación que presente el par ángulo-tiempo con un valor de tiempo mayor. Para las articulaciones que tengan definidos ángulos hasta un tiempo menor que dicho tiempo total, el programa tomará el último par de puntos de la articulación y mantendrá el valor del ángulo constante hasta el tiempo total de la trayectoria.



Después, si se ha seleccionado “Terminar en posición inicial”, todas las articulaciones volverán a la posición inicial en un intervalo de 1.5 segundos si el paso seleccionado es 3 ms, o 1 segundo si el paso es distinto de 3 ms (ver punto 4.3.2).

Las articulaciones para las que no se indique ningún par tiempo-ángulo permanecerán toda la trayectoria en la posición inicial.

Si durante este proceso se desea eliminar todos los valores introducidos y comenzar de nuevo, es posible hacerlo haciendo clic en el botón “Limpiar valores” (ver Figura 4.3). Como confirmación se mostrará el mensaje de la Figura 4.11.

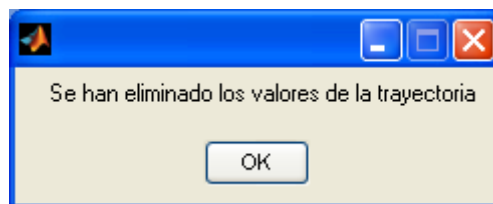


Figura 4.11. Mensaje valores eliminados

#### 4.3.4 Crear la trayectoria

Una vez definidos los valores tiempo-ángulo que se deseen y los parámetros de la trayectoria, se puede hacer clic en el botón “Crear trayectoria”.

En caso de no haber seleccionado un paso para la trayectoria, se mostrará el mensaje de error correspondiente (Figura 4.8).

Si todo es correcto, se mostrará un mensaje que nos indicará el tiempo final de la trayectoria (Figura 4.12). Ya se podrá observar la representación gráfica para cada articulación en el panel situado debajo de la zona de selección de parámetros.

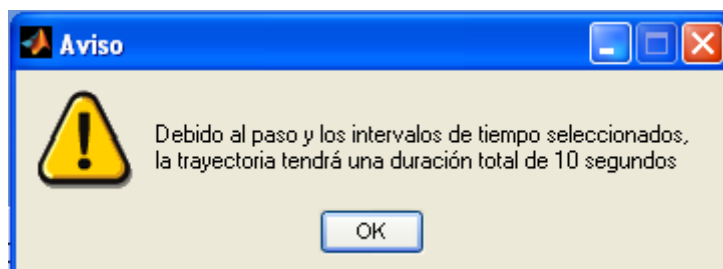


Figura 4.12. Mensaje con la duración de la trayectoria

### 4.3.5 Guardar la trayectoria

Como podemos observar en la Figura 4.3, el botón “Guardar trayectoria creada” permanece deshabilitado hasta que se ha creado la trayectoria.

Al hacer clic en “Guardar trayectoria creada” se abrirá un cuadro de diálogo como el de la Figura 4.13 para guardar el archivo en extensión *.dat*. Éste cuadro solicitará un nombre para dicho archivo, y sugerirá el último directorio al que ha accedido el programa.

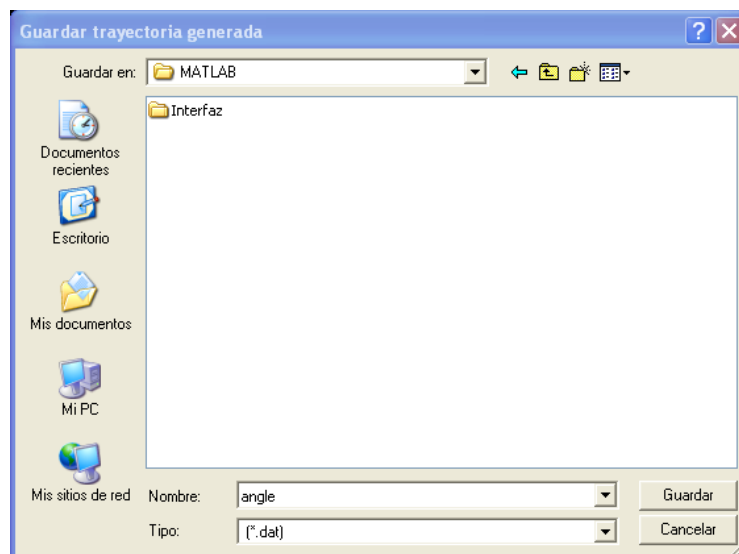


Figura 4.13. Cuadro de diálogo para guardar archivo con extensión *.dat*

Tras indicar el nombre y el lugar donde se desea guardar el archivo, si todo es correcto aparecerá un mensaje de confirmación como el de la Figura 4.14:

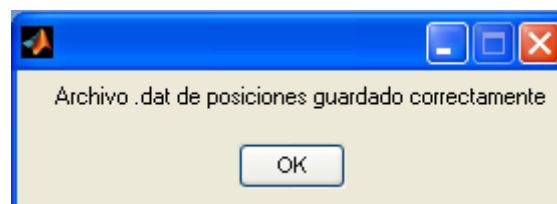
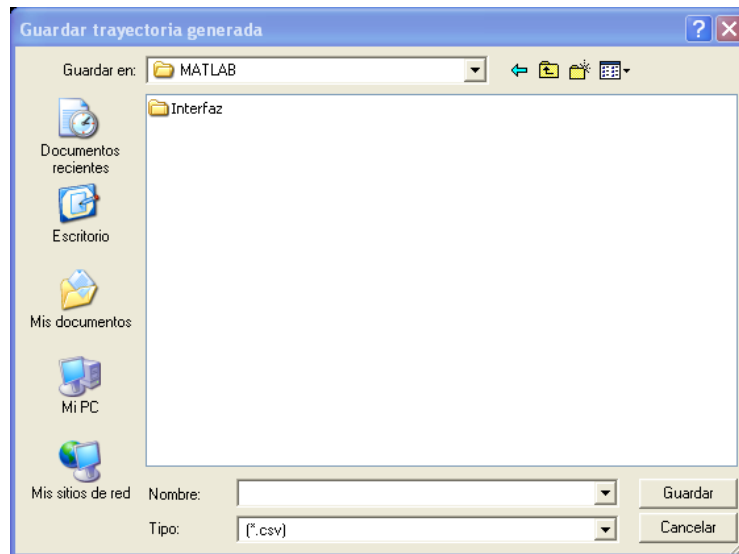


Figura 4.14. Mensaje archivo con extensión *.dat* guardado

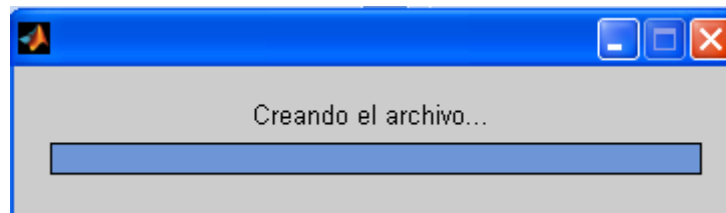
A continuación se guardan, siguiendo el mismo proceso, los archivos *vel.dat* y *acc.dat*, correspondientes a velocidad y aceleración, respectivamente. Las ventanas de diálogo mostrarán siempre la última ruta que se haya utilizado.

Si se trata de una trayectoria para el robot HOAP-3, a continuación se abre un cuadro de diálogo como el de la Figura 4.15 para guardar el archivo con extensión csv.



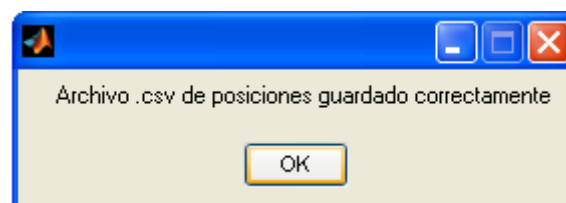
**Figura 4.15. Cuadro de diálogo para guardar archivo con extensión .csv**

Tras indicar el nombre y la ruta para el archivo, una barra indica el progreso de creación del archivo, como se puede ver en la Figura 4.16



**Figura 4.16. Barra de espera al guardar archivo con extensión .csv**

Si todo es correcto, se muestra el mensaje de la Figura 4.17.



**Figura 4.17. Mensaje archivo con extensión .csv guardado**

Tanto para el HOAP-3 como para el RH-2, al crear la trayectoria se genera un archivo con extensión .dat que contiene los puntos que se han indicado al definir la trayectoria.

Para cada articulación hay dos columnas, una para los valores de tiempo y otra para los valores de posición (Ver Figura 4.18).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	RLEG HIP Y	RLEG HIP Y	RLEG HIP R	RLEG HIP R	RLEG HIP P	RLEG HIP P	RLEG KNEE	RLEG KNEE	RLEG ANKLE P	RLEG ANKLE P	RLEG ANKLE R	RLEG ANKLE R	RARM SHOULDER P	RARM SHOULDER P
2	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición
3	0	0	0	0	0	-18	0	46	0	-28	0	2	0	0
4	-	-	7	0	7	-18	7	46	7	-28	7	2	2	2
5	-	-	9	2	9	-40	9	95	9	-54	9	2	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figura 4.18. Archivo de puntos de la trayectoria

Una vez guardados los archivos, se deshabilita el botón “Guardar trayectoria creada” hasta que se vuelva a hacer clic en “Crear trayectoria”.

## 4.4 Concatenación de trayectorias

### 4.4.1 Aspectos generales

Esta funcionalidad permite construir trayectorias a partir de otras existentes. Si tenemos diversas trayectorias preparadas, podemos hacer que el robot las ejecute una detrás de otra creando una única trayectoria a partir de ellas. En la Figura 4.19 se puede observar el aspecto general de la pantalla “Unir trayectorias HOAP-3”.

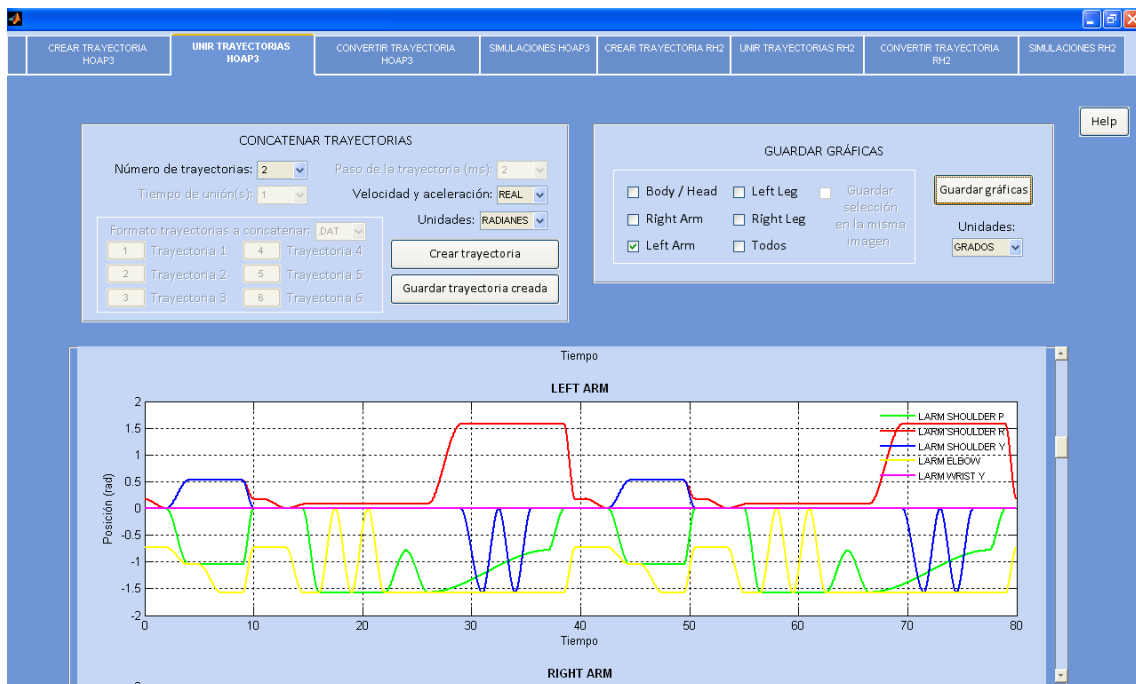


Figura 4.19. Pantalla de concatenación de trayectorias para HOAP-3

#### 4.4.2 Selección de parámetros

En primer lugar, como se observa en la Figura 4.20, será necesario seleccionar ciertos parámetros.

CONCATENAR TRAYECTORIAS

Número de trayectorias: 3 Paso de la trayectoria (ms): ----

Tiempo de unión(s): 1 Velocidad y aceleración: REAL

Formato trayectorias a concatenar: .DAT Unidades: RADIANES

1 Trayectoria 1 4 Trayectoria 4

2 Trayectoria 2 5 Trayectoria 5

3 Trayectoria 3 6 Trayectoria 6

Crear trayectoria

Guardar trayectoria creada

**Figura 4.20. Parámetros para la concatenación de trayectorias para el HOAP-3**

Hay que indicar el paso de la trayectoria que se va a crear, que debe ser el mismo que el de las trayectorias que se está uniendo. Si se trata de cargar las trayectorias antes de indicar el paso, un mensaje de error avisa al usuario de que no ha indicado este valor, impidiéndole continuar hasta haberlo hecho.

Es necesario seleccionar también el tiempo que se invertirá en la conexión de las dos trayectorias. Si el paso seleccionado es 3 ms, los tiempos de unión que se podrán seleccionar serán: 1.5 segundos y 3 segundos. Si el paso es otro cualquiera, se podrá escoger entre 1 segundo, 1.5 segundos, 2 segundos, 2.5 segundos y 3 segundos. En el tiempo de unión, el programa hará una interpolación cúbica entre el último punto de la primera trayectoria a concatenar y el primer punto de la siguiente.

Del mismo modo que en la generación de trayectorias, también hay que elegir en qué unidades se desea visualizar las gráficas (radianes por defecto) y si se desea obtener los archivos de velocidad y aceleración reales.

En el cuadro inferior podemos observar seis botones destinados a cargar las trayectorias que se desea concatenar. Inicialmente, estos seis botones permanecen deshabilitados hasta que se indica cuántas van a ser cargadas.

En el ejemplo de la Figura 4.20 se ha elegido concatenar tres trayectorias, y por ello aparecen tan sólo tres botones habilitados. Si en cualquier momento del proceso se cambiara el número de trayectorias, se habilitarían los botones correspondientes al nuevo número de trayectorias indicado, eliminándose cualquier trayectoria cargada previamente de la memoria del programa. Se puede observar que los botones para simular y guardar la trayectoria permanecen aún deshabilitados.

Una vez indicados estos parámetros, se puede proceder a cargar las trayectorias.

### 4.4.3 Carga de las trayectorias

En caso de estar creando una trayectoria para el robot HOAP-3, habrá que indicar el formato de las trayectorias que se van a introducir, bien *.dat* o bien *.csv* (por defecto *.dat*). En caso de tratarse del robot RH-2, no habría que indicar este parámetro ya que siempre se trataría de trayectorias en formato *.dat*.

Al cargar la trayectoria, se comprueba si el paso coincide con el de la trayectoria que se está creando. Si no coincide, se muestra el mensaje de error de la Figura 4.21 y no se carga la trayectoria.

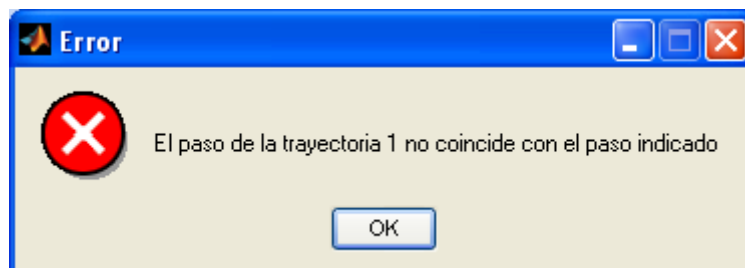


Figura 4.21. Mensaje de error al cargar la trayectoria

Si el archivo tiene el formato adecuado y el paso de la trayectoria corresponde con el indicado, se comprueba si todas las articulaciones se mueven dentro del rango de valores angulares permitido. Si ninguna articulación excede el rango de valores permitido, se muestra el mensaje de la Figura 4.22.

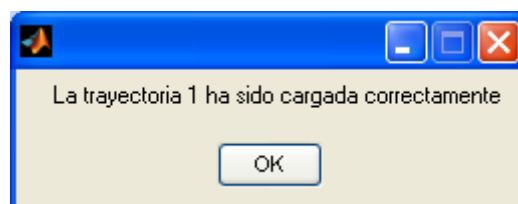
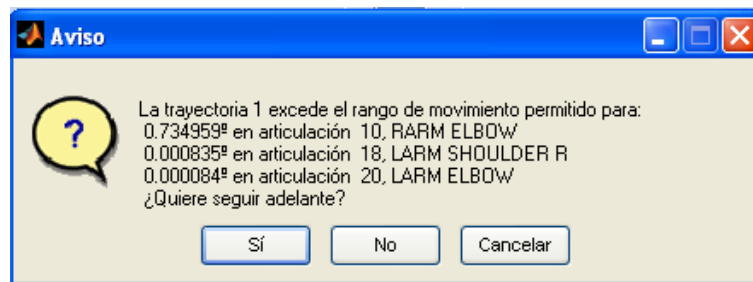


Figura 4.22. Mensaje de trayectoria cargada correctamente

Si alguna articulación excede este rango, se muestra un mensaje como el de la Figura 4.23 en el que se detalla qué articulaciones exceden dicho rango y en qué medida.



**Figura 4.23. Mensaje con las articulaciones fuera del rango permitido**

Este mensaje pide confirmación para seguir adelante cargando la trayectoria, o cancelar el proceso de carga. Si se acepta se cargará la trayectoria y se mostrará un mensaje como el de la Figura 4.22,

Una vez cargadas tantas trayectorias como se indicaron en el primer paso, (y como botones tenemos habilitados), el botón “Crear trayectoria” se habilitará.

#### **4.4.4 Creación de la trayectoria**

Si se hace clic en el botón, “Crear trayectoria”, se crea la trayectoria y se puede observar la representación gráfica en el panel inferior. Mediante la barra vertical que aparece a la derecha, se puede ir desplazando dicho panel para ver las gráficas correspondientes a las distintas articulaciones. Queda habilitado el botón “Guardar trayectoria creada”, situado a la derecha de la pantalla.

La selección del paso y los botones para cargar trayectorias se deshabilitarán para evitar que se modifique algún parámetro de la trayectoria creada antes de que sea guardada y evitar así posibles errores. Éstos permanecerán deshabilitados hasta que se vuelva a seleccionar un nuevo número de trayectorias, lo que provocaría la eliminación de las trayectorias cargadas obligando a comenzar de nuevo el proceso. Queda habilitada la selección de velocidad y aceleración real o nula, y la selección de las unidades. Es posible modificar estos parámetros y volver a hacer clic en “Crear trayectoria”.

Si se hace clic sobre el botón “Guardar trayectoria creada”, del mismo modo en que se ha descrito en el apartado anterior, se iniciará el proceso de guardar los archivos generados. Para el HOAP-3 se podrá guardar posición, velocidad y aceleración en *.dat*

y posición en .csv. Para el RH-2 se podrá guardar posición, velocidad y aceleración en .dat.

## 4.5 Conversión de trayectorias

### 4.5.1 Aspectos generales

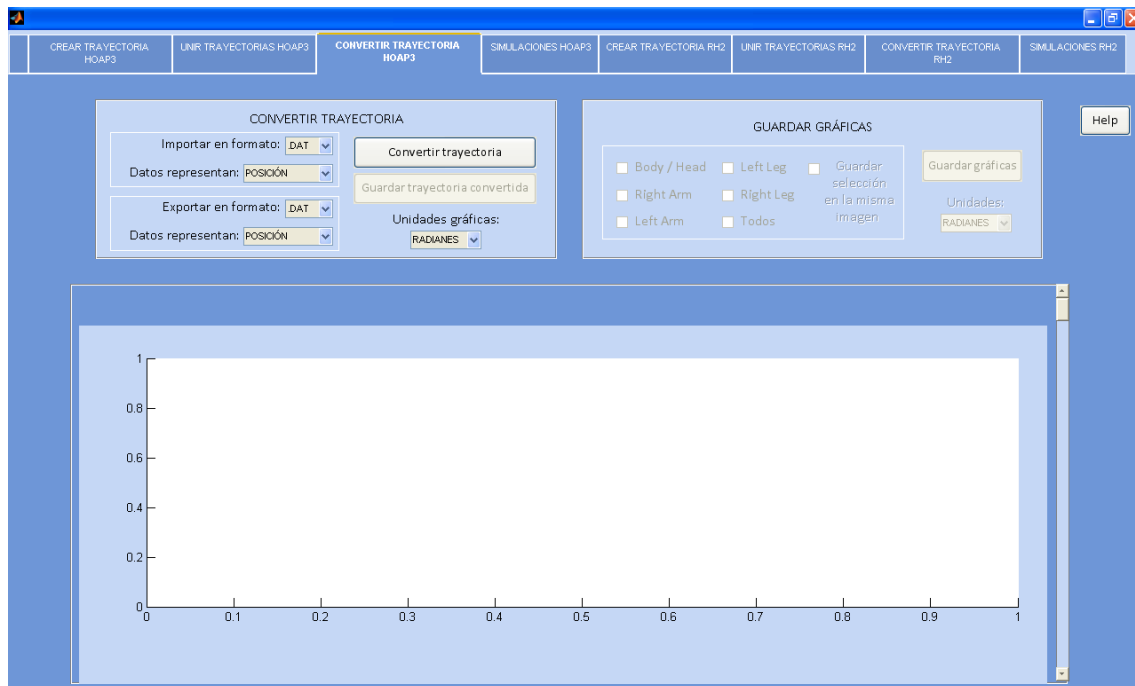


Figura 4.24. Vista general de la pantalla: "Convertir trayectorias HOAP-3"

En las pantallas "Convertir trayectorias RH2" y "Convertir trayectorias HOAP-3", es posible obtener a partir del archivo de posiciones los archivos de velocidad y aceleración que serán necesarios para realizar las simulaciones de las trayectorias. Para el HOAP-3 también es posible convertir los archivos entre los formatos *csv* y *dat*.

### 4.5.2 Selección de parámetros

En el caso del RH-2 bastará con indicar qué transformación se desea hacer, si se desea obtener velocidad o aceleración a partir de posición, o bien aceleración a partir de velocidad (Ver Figura 4.25).

En cambio, si se trata del HOAP-3, al ser posible también convertir los archivos de posiciones entre el formato *csv* (empleado por el robot) y el formato *dat* (empleado por el simulador), habrá que indicar la extensión de los archivos (Ver Figura 4.26).



CONVERTIR TRAYECTORIA

IMPORTAR ARCHIVO:  
Datos representan: POSICIÓN

EXPORTAR ARCHIVO:  
Datos representan: VELOCIDAD

Convertir trayectoria

Guardar trayectoria convertida

Unidades gráficas:  
RADIANES

Figura 4.25. Parámetros para convertir una trayectoria del RH-2

CONVERTIR TRAYECTORIA

Importar en formato: .DAT  
Datos representan: POSICIÓN

Exportar en formato: .DAT  
Datos representan: POSICIÓN

Convertir trayectoria

Guardar trayectoria convertida

Unidades gráficas:  
RADIANES

Figura 4.26. Parámetros para convertir una trayectoria del HOAP-3

Para obtener estos archivos, se hace la derivada de orden 1 o de orden 2 de los datos, según corresponda.

Habrá que seleccionar también, como se ha descrito en apartados anteriores, las unidades en que se desea visualizar las gráficas: grados o radianes.

Al cargar la trayectoria se comprueba que se cumple el rango de las articulaciones, del mismo modo al descrito con anterioridad al concatenar trayectorias. Una vez creada la trayectoria se habilitará el botón de “Guardar trayectoria” y los controles de selección de opciones para guardar las gráficas.

Una vez guardados los archivos, se deshabilita el botón “Guardar trayectoria convertida” para evitar posibles errores como guardar una trayectoria con unos parámetros almacenados en memoria tras haber modificado la selección. Si se desea volver a guardar el archivo, es necesario volver a hacer clic en el botón “Crear trayectoria”.

## 4.6 Simulación de trayectorias

### 4.6.1 Aspectos generales

Esta pantalla está destinada al estudio de las trayectorias. Permite observar una representación gráfica de los valores articulares de la trayectoria, y si así se desea, guardar los gráficos. En sentido estricto, no se trata de simulación de trayectorias, ya que estas se realizarán en OpenHRP-3.

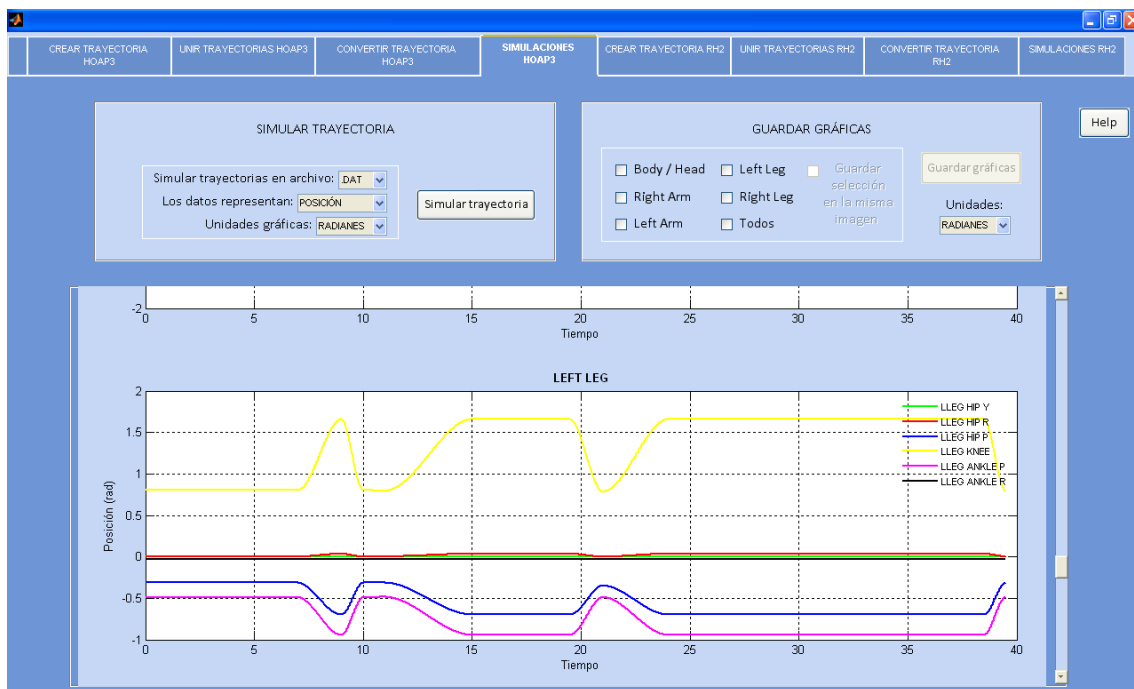


Figura 4.27. Vista general de la pantalla “Simular trayectorias HOAP-3”

### 4.6.2 Selección de parámetros

Para representar gráficamente los movimientos de las articulaciones, hay que indicar qué representan estos datos: posición, velocidad o aceleración. Hay que indicar también las unidades en que se desea ver las gráficas: grados o radianes (Figura 4.28).

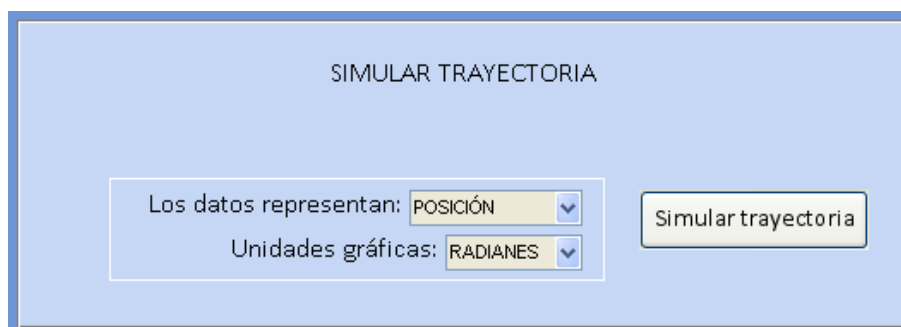


Figura 4.28. Parámetros de la simulación, para el RH-2

En el caso del HOAP-3 (Figura 4.29) hay que indicar también en qué formato se va a cargar la trayectoria: *dat* o *csv*.

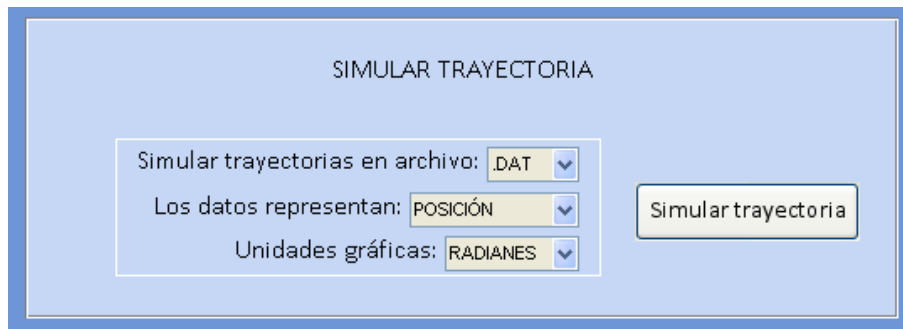


Figura 4.29. Parámetros de la simulación, para el HOAP-3

Una vez seleccionados los parámetros, se procede a cargar el archivo. En este proceso se toman todas las precauciones descritas anteriormente para otras funcionalidades de la interfaz, de este modo se comprueba que el formato del archivo es correcto y que las articulaciones se mueven dentro del rango permitido para cada una de ellas.

### 4.6.3 Representación gráfica

Del mismo modo que en las anteriores pantallas, en el panel inferior se puede observar la representación gráfica de los valores articulares durante la trayectoria. Es posible desplazar la barra vertical para ir mostrando las gráficas de los distintos grupos de articulaciones.

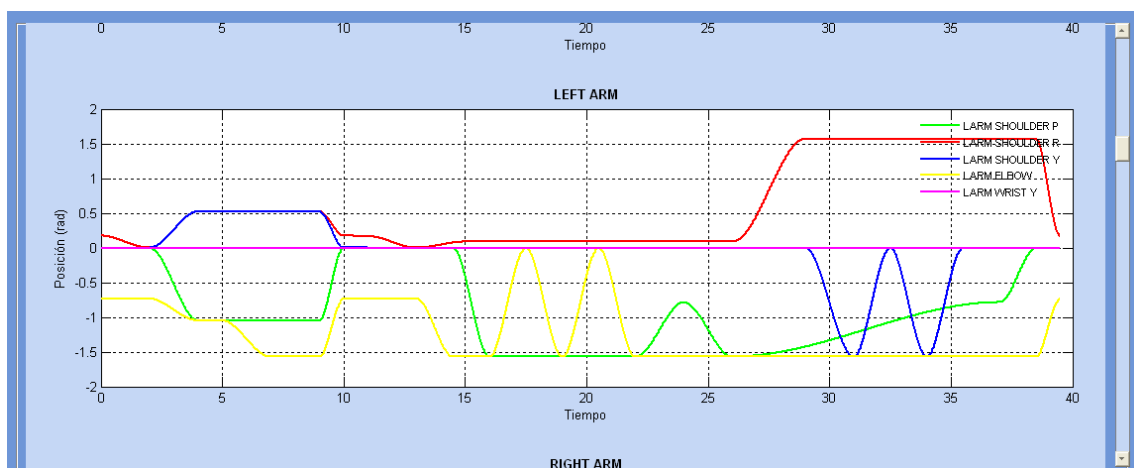


Figura 4.30. Detalle de la pantalla “Simular trayectoria HOAP-3”

## 4.7 Guardar gráficas



Figura 4.31. Panel "Guardar gráficas"

Para cada una de las funcionalidades de la interfaz se ofrece la posibilidad de guardar gráficas. Para ello, hay un panel como el de la Figura 4.31 en cada una de las pantallas. En el caso de la generación y concatenación de trayectorias, los controles de este panel permanecen deshabilitados hasta que se ha hecho clic en el botón "Crear trayectoria". En las pantallas "Convertir trayectoria" y "Simular trayectoria" no se habilitan hasta que se hace clic en el botón "Convertir trayectoria" y "Simular trayectoria", respectivamente.

Se puede seleccionar la sección del cuerpo de cuyas articulaciones se quiere guardar las gráficas. Si se selecciona más de una sección, se habilita la casilla "Guardar selección en la misma imagen".

En el ejemplo de la Figura 4.31, se han seleccionado todas estas secciones, y se ha indicado que se desean guardar todas las representaciones en una misma imagen. Se ha elegido ver los ángulos en radianes. Con estos parámetros se obtendría la imagen de la Figura 4.32. Queda habilitado un menú de herramientas de MATLAB que permite modificar las gráficas y guardarlas en distintos formatos.

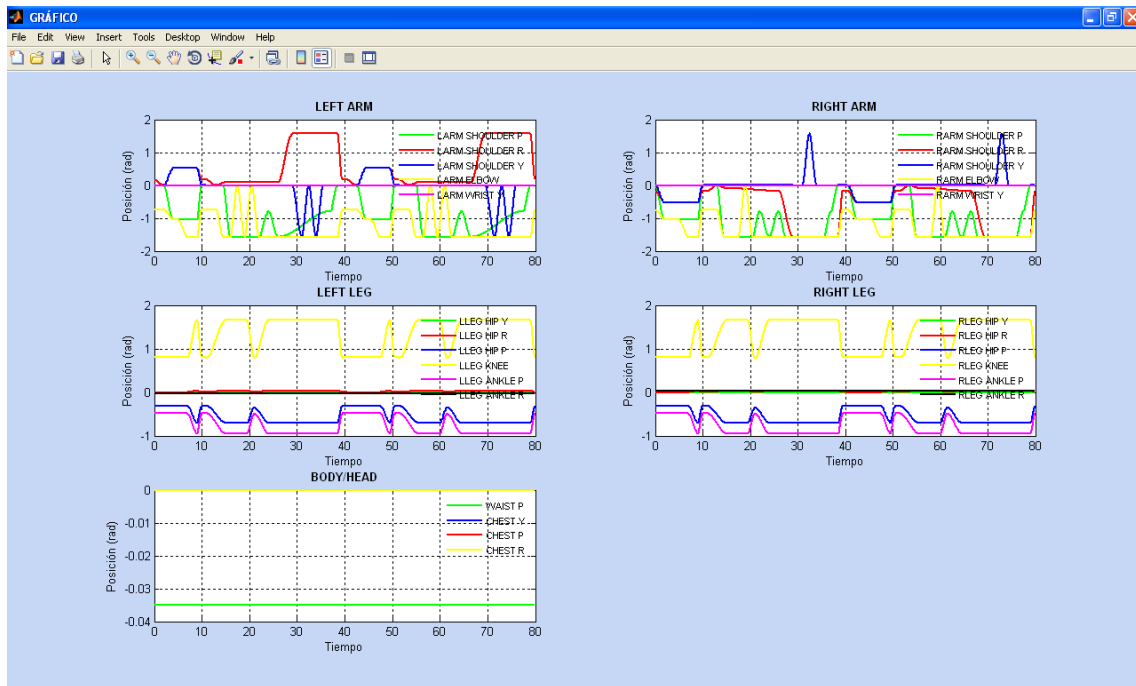


Figura 4.32: Ejemplo de gráficos tras crear la trayectoria

## 4.8 Documento de ayuda “Help”

Como ayuda para manejar la interfaz se incluye un documento en formato pdf. Éste es accesible haciendo clic en el botón “Help” disponible en la esquina superior derecha de todas las pantallas de la interfaz. En la Figura 4.33 se puede ver el botón correspondiente a la pantalla “Convertir trayectoria RH-2”.

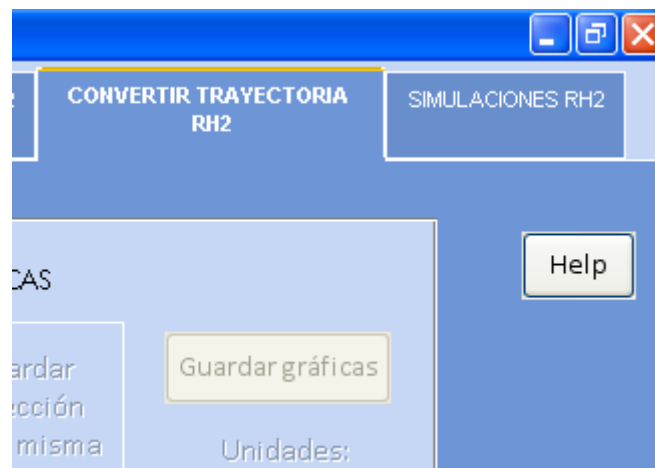


Figura 4.33. Botón Help en pantalla "Convertir trayectoria "RH-2".



## **5 Diseño de la interfaz**

### **5.1 MATLAB**

#### **5.1.1 Introducción**

**MATLAB**® es un lenguaje de programación interactivo propiedad de Mathworks™, que está siendo utilizado cada vez más en aquellas áreas científicas en las que es necesario llevar a cabo cálculos numéricos de gran envergadura [2]

Así, es usado en una amplia variedad de aplicaciones, incluyendo procesamiento de señal e imagen, comunicaciones, diseño de control, biología computacional, análisis financiero, estadística, álgebra lineal, etc. Existen diversas *toolboxes* específicamente diseñadas para las distintas áreas de aplicación, que contienen funciones especiales para resolver los problemas propios de estas [6].

El nombre MATLAB deriva del término inglés '*matrix laboratory*', ya que la principal característica de MATLAB es que permite manipular de manera sencilla y eficiente vectores y matrices [2].

Existen versiones de MATLAB compatibles con prácticamente todos los sistemas operativos, desde los utilizados en ordenadores personales, hasta los sistemas operativos UNIX empleados en estaciones de trabajo y grandes ordenadores. La última aplicación presentada, en 2010, es **MATLAB mobile**™, que permite utilizar MATLAB en los dispositivos móviles iPhone, iPad, y iPod touch.

Entre las funcionalidades de MATLAB, destaca el alto nivel de integración con aplicaciones externas y lenguajes, como C, C++, Fortran, Java™, COM, y Microsoft® Excel®. También hay que señalar que, al ser un lenguaje interactivo, para las aplicaciones que no implican tiempo real el usuario no necesita compilar el código como ocurre con la mayoría de los lenguajes de programación.

El sistema MATLAB está constituido por tres partes [6]:

- Herramientas del escritorio y Entorno de desarrollo: esta parte de MATLAB es el conjunto de herramientas que facilitan el trabajo con archivos y funciones. Muchas de estas herramientas son interfaces gráficas de usuario. Éstas incluyen: el escritorio MATLAB y la ventana de comandos, un editor y un depurador de código, así como exploradores para visualizar el contenido de ayuda, el entorno de trabajo y las carpetas.
- Librería de funciones matemáticas: esta librería es una amplia colección de algoritmos computacionales que van desde funciones elementales como suma, seno, coseno y aritmética compleja, hasta funciones más sofisticadas como la matriz inversa, cálculo de los autovalores de una matriz, funciones de Bessel o la transformada rápida de Fourier.
- El lenguaje: el lenguaje MATLAB es un lenguaje de alto nivel construido con matrices y vectores, con sentencias de control de flujo, funciones, estructuras de datos, y funcionalidades de entrada y salida y programación orientada a objetos.

A continuación se comentan algunos de los elementos y comandos del lenguaje MATLAB que han tenido más relevancia para el diseño de la interfaz, con el objeto de facilitar la comprensión del proceso de programación de la misma, el cual será analizado en detalle en la sección 5.2, “Diseño de la Interfaz”.

### 5.1.2 Variables en MATLAB

MATLAB permite asignar nombres variables a valores numéricos. En general, se puede dar un nombre variable a cualquier expresión mediante una sentencia de la forma:

```
>> variable = expresion
```



En “expresión” pueden aparecer valores numéricos, otras variables, matrices, operadores y funciones.

### 5.1.2.1 Tipos de variables

En MATLAB hay tres tipos de variables: locales, globales y persistentes [4].

- Variables locales: cada función tiene sus propias variables. Éstas están separadas de las de otras funciones (excepto en el caso de las funciones anidadas, que se estudiará más adelante), y de las del espacio de trabajo. Las variables definidas en una función no permanecen en memoria entre una llamada a la función y la siguiente, a no ser que sean declaradas *globales* o *persistentes*.
- Variables globales: si varias funciones declaran la misma variable como *global*, todas ellas comparten una copia de dicha variable. Cualquier asignación que se haga a dicha variable en una función estará disponible para las demás funciones que la hayan declarado como *global*. Aunque no es obligatorio hacerlo, suelen escribirse en letras mayúsculas para mayor claridad. En los manuales de MATLAB se recomienda utilizarlas lo menos posible, ya que su uso entraña cierto riesgo. Errores derivados de la declaración de una variable global en una función bajo un nombre que ya ha sido utilizado para declarar una variable global distinta en otra función pueden ser muy difíciles de detectar.
- Variables persistentes: sólo pueden ser declaradas en funciones, y sólo tendrán acceso a ellas las funciones donde han sido declaradas. Cuando termina la ejecución de una función, MATLAB no borra estas variables de la memoria como haría con las variables locales, así su valor persiste en las sucesivas llamadas a la función.

### 5.1.2.2 Alcance de una variable

MATLAB almacena las variables en un espacio de memoria llamado *workspace* o espacio de trabajo. Las variables creadas de manera interactiva y las creadas al ejecutar scripts se almacenan en el *espacio base de trabajo*. Pero las funciones no usan este *espacio base de trabajo*, porque cada función tiene su propio espacio de trabajo. A continuación se muestran algunas de las opciones para ampliar el alcance de una variable:

- Usar variables globales
- Pasar la variable a la función que necesita usarla, como un argumento extra en la llamada a la función. Esto se puede hacer creando una estructura o una matriz de celdas que contenga todas las variables que se quieren pasar a la función, para mayor comodidad.
- Uso de funciones anidadas: el alcance de una variable local se extiende a todas las funciones que estén anidadas dentro de ella.

Para eliminar los valores de variables almacenadas en el espacio de trabajo, se emplea el comando `clear` seguido del nombre de la variable. Para eliminar los valores de todas las variables, se emplea el comando `clear all` [2].

### 5.1.2.3 Nombrando las variables

Los nombres de las variables deben empezar con una letra seguida de una serie de caracteres alfanuméricos. Es importante señalar que MATLAB es *'case sensitive'*, es decir, distingue entre mayúsculas y minúsculas. Los nombres de las variables pueden tener toda la longitud que se quiera, pero MATLAB sólo usará los primeros N caracteres e ignorará el resto. N es el número devuelto al ejecutar la función `namelengthmax`, tecleando dicho nombre en la ventana de comandos.

Por ello es muy importante asegurarse de que el nombre de cada variable es único en sus primeros N caracteres para que MATLAB las pueda distinguir, el resto de caracteres pueden servir para facilitar al usuario la identificación de la variable [4].

### 5.1.3 Tipos de datos en MATLAB

MATLAB trabaja esencialmente con un único objeto: **matrices numéricas** rectangulares formadas por números reales o complejos. De este modo, un escalar en MATLAB es una matriz de dimensiones 1x1, y un vector es una matriz de una única fila o columna [2]. El siguiente ejemplo muestra la declaración de una matriz:

```
>> Matriz = [1 2 3; 4 5 6]
```

```
Matriz =
```

```
1 2 3  
4 5 6
```

También existe la posibilidad de crear **variables de texto**. Para ello se introduce el texto entre comillas al asignar el valor a la variable, del siguiente modo.

```
>> Variable_de_texto = 'Texto'
```

```
Variable_de_texto =  
Texto
```

Este tipo de variables son en realidad un vector fila, quedando almacenado cada carácter de texto en una componente del vector. Las variables de texto se pueden convertir en numéricas mediante la sentencia *str2num* y las numéricas en variables de texto mediante la sentencia *num2str*.

Las conocidas como **matrices de celdas (cell array)** son matrices multidimensionales cuyos elementos son matrices numéricas o cadenas de caracteres de dimensiones cualesquiera. Para declarar una matriz de celdas, se encierran entre llaves todos los elementos de la misma, separando los elementos de una misma fila con espacios o comas, usando el carácter “punto y coma” o *return* para indicar el final de una fila. En el siguiente ejemplo se muestra la declaración de una matriz de celdas:

```
>> matriz_celdas = {[1 2; 3 4],1,'texto';[1 2],[ 'texto1';'texto2'],[1;2]}
```

```
matriz_celdas =  
 [2x2 double] [ 1] 'texto'  
 [1x2 double] [2x6 char] [2x1 double]
```

Mediante el comando *num2cell*, las matrices numéricas se pueden convertir en matrices de celdas.

Una **matriz de tipo estructura** es un conjunto heterogéneo de datos distribuidos en distintos grupos llamados campos. Para declarar una matriz de tipo estructura se van definiendo campos y asignándoles valores. Estos valores pueden ser matrices celda, o bien celdas con un único escalar o valor numérico. También es posible hacer que el contenido de uno de los campos sea una matriz de tipo estructura. El siguiente ejemplo muestra el proceso de declaración de una estructura:

```
>> estructura = struct ('campo1','hola','campo2',123)
```

```
estructura =  
    campo1: 'hola'  
    campo2: 123
```

Para referirnos al contenido de un campo de la estructura, se emplea el nombre de la estructura seguido de un punto y el nombre del campo:

```
>> primer_campo = estructura.campo1  
  
primer_campo =  
hola
```

Del mismo modo se puede crear un nuevo campo, o modificar el valor de uno existente:

```
>> estructura.campo3 = 'nuevo texto'  
  
estructura =  
    campo1: 'hola'  
    campo2: 123  
    campo3: 'nuevo texto'
```

Una matriz de tipo estructura se puede convertir en una matriz de celdas por medio de la sentencia *struct2cell*, y una matriz de celdas se puede convertir en una estructura por la sentencia *cell2struct* [2].

Dentro de las **variables de tipo numérico**, existen distintos tipos:

- *Double*. Coma flotante, precisión doble.
- *Single*. Coma flotante, precisión simple.
- *Uint8*. Números naturales de 8 bits. Valores entre 0 y 255.
- *Uint16*. Números naturales de 16 bits. Valores entre 0 y 65535.
- *Uint32*. Números naturales de 32 bits. Valores entre 0 y 4294967295.
- *Uint64*. Números naturales de 64 bits. Valores entre 0 y 18446744073709551615.
- *Int8*. Números enteros de 8 bits. Valores entre -128 y 127.

- *Int16*. Números enteros de 16 bits. Valores entre -32768 y 32767.
- *Int32*. Números enteros de 32 bits. Valores entre -2147483648 y 2147483647.
- *Int64*. Números enteros de 64 bits. Valores entre 0 y 18446744073709551615.

Por defecto, MATLAB utiliza la doble precisión para almacenar variables numéricas.

### 5.1.4 Programación con MATLAB

La programación con MATLAB es muy similar a la realizada con otros lenguajes de programación. Podemos encontrar las sentencias habituales para el control condicional (sentencias *if/ else/ elseif*, y sentencias *switch/ case/ otherwise*) y el control por ciclos (*for, while, continue* y *break*).

Para expresar condiciones se emplean los siguientes operadores relacionales: igual “==”, menor o igual “<=”, mayor o igual “>=”, menor “<”, mayor “>”, distinto “~=”, y los siguientes operadores lógicos: AND “&”, NOT “~” y OR “|”

#### 5.1.4.1 Tipos de archivos

Aunque MATLAB puede usarse de un modo interactivo, su utilización más habitual consiste en la ejecución de secuencias de sentencias que deberán almacenarse en archivos. En este sentido trabaja de manera similar a cualquier lenguaje de programación [2].

Estos archivos se conocen como *M-File* y tienen extensión “.m”. Son compilados de manera automática durante su primera ejecución. Hay dos tipos de ficheros *M-File*: los *ficheros script* y las *funciones*.

Los **ficheros script** consisten en una serie de sentencias y comandos de MATLAB. Cuando se ejecuta un fichero script, todas las variables utilizadas en el mismo quedan almacenadas en memoria durante la sesión actual de trabajo.

Las **funciones** deben siempre comenzar con la siguiente sintaxis:

```
function [argumentos de salida] = nombre_funcion (argumentos de entrada)
```

Otro requisito que debe cumplir la función, es que el nombre del archivo debe ser igual que el nombre de la función en la declaración de la primera línea. En el ejemplo de arriba, el archivo se llamaría “nombre\_funcion.m”.

La diferencia principal entre una función y un script es que las variables que se definen y manipulan dentro de la función sólo permanecen en memoria mientras dura la ejecución de la misma; por tanto, las únicas variables que permanecen en memoria en la sesión de trabajo son los argumentos de entrada y salida.

Para interrumpir la ejecución de un script o una función, se emplea la sentencia *return*.

Los archivos de tipo función pueden contener funciones adicionales con nombres diferentes, que se irán escribiendo unas a continuación de otras. La primera función es la *función principal* y las demás son **subfunciones**. Cada subfunción debe comenzar con su propia sentencia de declaración de función.

#### 5.1.4.2 Sintaxis de las funciones

En la declaración de una función, si no hay argumentos de salida se pueden omitir los corchetes y el signo “=”. Si sólo hay un argumento de entrada no es necesario corchetes, y si no hay argumentos de entrada no es necesario poner el paréntesis.

Si el número de argumentos de la función es variable, se pueden emplear las variables *varargin* y *varargout*.

```
function varargout = nombre_funcion (varargin)
```

La variable *varargin* es una matriz de celdas que contiene todos los argumentos opcionales de la función. Recoge todos los argumentos que se introducen al producirse la llamada a la función. La variable *varargout* es una matriz de celdas que recoge todos los argumentos de salida a los que se haya asignado valores dentro de la función.

Siguiendo con el ejemplo anterior, podríamos extraer las variables que nos interesan de la matriz *varargin* del siguiente modo.

```
[variable_1, variable_2] = varargin {[2,5]};
```

Y podemos asignar valores a las variables de la matriz *varargout* que consideremos.

```
varargout {1} = resultado;
```

## 5.1.5 Creación de interfaces gráficas con MATLAB

### 5.1.5.1 Qué es una GUI

Las aplicaciones gráficas de MATLAB permiten la representación gráfica de vectores y matrices, pudiéndose comentar e imprimir estas gráficas. Incluye funciones de alto nivel para la visualización de datos en dos o tres dimensiones, procesamiento de imagen, animación, etc. También incluye funciones de bajo nivel que permiten al usuario personalizar completamente la apariencia de los gráficos, e incluso construir interfaces gráficas de usuario para las aplicaciones de MATLAB [6].

Una interfaz gráfica de usuario, del inglés *GUI* (*'Graphical user interface'*) es una plataforma gráfica que contiene controles, llamados componentes, que permiten al usuario llevar a cabo tareas interactivas. Estos controles llevan asociados unos *callbacks*, que son funciones que se ejecutan al llevar a cabo el usuario una determinada acción, por ejemplo hacer clic en un botón. Al utilizar una GUI, el usuario no necesita escribir comandos para realizar tareas, ni necesita conocer los detalles de cómo se realizan las mismas.

### 5.1.5.2 Cómo construir una GUI

Hay dos maneras de construir una *GUI* [5]:

- Usando la herramienta interactiva **GUIDE**, abreviatura del término inglés *'GUI development environment'*.
- **Mediante programación**. Creando archivos de código que generan *GUIs*, como funciones o scripts.

Al comenzar el proceso de creación de una GUI mediante la herramienta GUIDE, aparece una figura que podemos ir poblando de controles y objetos por medio de un editor gráfico. GUIDE crea un archivo de código asociado a esta figura, que contiene los *callbacks* de la GUI y sus componentes. GUIDE guarda ambos, la figura (como un archivo *.fig*), y el código (como un archivo *.m*). Al abrir uno de ellos se abre el otro para arrancar la GUI.

Si se crea la GUI mediante programación, lo que se hace es crear un archivo de código en el que se definen todos los elementos y comportamientos de la GUI. Al ejecutar el archivo, se crea una figura en la que aparecen los elementos programados.

Normalmente, la figura no se guarda entre sesiones porque el programa crea una nueva cada vez que se arranca el programa.

### 5.1.5.3 GUIDE vs Programación

Según qué método se emplee, los archivos de código creados tienen un aspecto muy distinto. Los archivos de las *GUIs* creadas programando son generalmente más largos, ya que definen de manera explícita todas las propiedades de la figura y sus controles, así como los *callbacks*. En cambio, GUIDE guarda la mayoría de definiciones y propiedades en la propia figura. En el archivo de código sólo guarda los *callbacks* y otras funciones que inicializan la *GUI* cuando se abre.

La elección de un método u otro estará basada en el tipo de aplicación que se desea crear, y la experiencia del desarrollador. Por ejemplo, para crear *GUIs* sencillas con pocos elementos, es recomendable hacerlo mediante programación, ya que se puede definir completamente cada componente con una simple llamada a una función. Si la *GUI* es de una complejidad moderada, ya empieza a ser más recomendable usar la herramienta GUIDE, porque simplificará notablemente el proceso. En cambio, para *GUIs* complejas con un gran número de componentes o que requieran interacción con otras *GUIs*, se recomienda hacerlo programando. De este modo el desarrollador tiene el control total sobre el diseño de la interfaz

### 5.1.5.4 Componentes de una GUI

En una primera aproximación, se puede decir que una *GUI* está basada en objetos gráficos. El objeto principal es una figura. De hecho, se puede decir que en MATLAB, una *GUI* es en sí una figura. Las figuras pueden contener los siguientes objetos gráficos [2]:

- Objetos axes: definen una región dentro de la figura, en la cual se representarán imágenes, dibujos en dos o tres dimensiones, se escribirán textos, etc.
- Objetos uicontrol: abreviatura del término inglés 'user interface control'. Son cajas de texto y botones que permiten ejecutar una determinada acción, programada por el usuario, cuando se activan con el ratón.



- Objetos *uimenu*: abreviatura del término inglés ‘*user interface menu*’. Son los menús creados por el usuario que se añaden al menú ya existente en la parte superior de la figura.
- Objetos *uicontextmenu*: abreviatura del término inglés ‘*user interface context menu*’. Son menús que se activan con el botón derecho del ratón y que pueden asociarse a cualquiera de los objetos anteriores.

Al crear estos objetos, MATLAB les asigna un número como identificador (*handle*).

#### 5.1.5.5 Propiedades de los objetos gráficos

Los objetos gráficos tienen una serie de propiedades que determinan su aspecto en la pantalla. Algunas de las más importantes son su posición en la jerarquía de objetos gráficos (*parent*, *children*), su visibilidad (*on*, *off*), el color, la posición, etc. Si no se indican otros valores de manera explícita, estas propiedades tomarán los valores que el objeto gráfico tenga definidos por defecto.

A continuación se indican algunas funciones necesarias para interactuar con las propiedades de los objetos gráficos [2]:

- *get (h)*: donde “*h*” es el *handle* del objeto gráfico, permite obtener un listado de todas las propiedades del objeto gráfico.
- *set (h)*: proporciona un listado por pantalla de todas las propiedades del objeto con los posibles valores que puede tomar cada propiedad.
- *get (h,'propiedad')*: permite conocer el valor de la propiedad “*propiedad*” del objeto “*h*”.
- *set (h,'propiedad')*: permite conocer los valores que puede tomar la propiedad “*propiedad*” del objeto “*h*”.
- *set (h,'propiedad', nuevovalor)*: permite asignar el valor “*nuevovalor*”, a la propiedad “*propiedad*” del objeto “*h*”.

### 5.1.5.6 Uicontrols:

Como se comentó antes, los objetos *uicontrol* son cajas de texto y botones que permiten ejecutar acciones programadas por el usuario, cuando se activan con el ratón. Distinguimos los siguientes tipos de *uicontrol* [2]:

- Push button: al hacer clic en ellos se ejecuta una determinada acción.
- Toggle button
- Radio button
- Checkbox
- Edit text: cajas de texto editables.
- Static text: cajas de texto no editables.
- Slider: permite ejecutar acciones dependientes de un parámetro numérico, cuyo valor varía al trasladar el botón a lo largo de una barra rectangular por medio del ratón.
- Listbox
- Popup menú
- Frame: marco cuya función es agrupar grupos de botones.

Los *uicontrols* de tipo *Toggle button*, *Radio Button* y *Checkbox* permiten realizar una acción al hacerse clic en ellos. Esta acción se mantiene hasta que se vuelve a hacer clic de nuevo en el mismo. Suelen emplearse para elegir entre opciones.

Los de tipo *Listbox* y *Popup Menu*, muestran una lista de alternativas que harán que se ejecute una determinada acción programada al seleccionar una de ellas con el ratón.

Se puede crear *uicontrols* por la función “*uicontrol*”, cuya sintaxis es la siguiente:

```
h1 = uicontrol (identificadorpadre,'propiedad1','valor1','propiedad2','valor2',...)
```

donde *identificadorpadre* es el identificador de la figura a la que se añade el *uicontrol*.

Las propiedades más relevantes que presentan los *uicontrols* son las siguientes:

- BackgroundColor: es el color del objeto

- Callback: especifica la acción a realizar por MATLAB al actuar sobre el botón con el ratón.
- Enable: permite desactivar el *uicontrol* de modo que no sea posible actuar sobre él con el ratón. Toma valores *on* y *off*.
- Position: determina posición y tamaño del *uicontrol* dentro de la figura. Se define mediante un vector de cuatro elementos: [x y anchura altura]. La propiedad “units” establece las unidades en que se mide la posición: *inches*, *centimeters*, *normalized*, *points*, *pixels* y *characters*.
- String: es el texto que aparece en el *uicontrol*. El tamaño, tipo, color y grosor de la fuente, el ángulo de la letra, pueden modificarse mediante las propiedades *FontSize*, *FontName*, *ForegroundColor*, *Fontweight* y *FontAngle* respectivamente.
- Style: es el tipo de *uicontrol*: *pushbutton*, *togglebutton*, *radio-button*, *checkbox*, *edit*, *text*, *slider*, *frame*, *listbox* y *popupmenu*.
- Value: en los *uicontrols* de tipo *Checkbox*, *Radio Button* y *Toogle Button* toma el valor “1” al estar seleccionado, y el valor “0” en caso contrario.
- TooltipString: permite introducir un comentario que se mostrará en un recuadro amarillo al acercarnos con el ratón al *uicontrol*.

## 5.2 Diseño de la interfaz

### 5.2.1 Estructura del programa

En primer lugar, hay que señalar que para desarrollar esta interfaz se ha optado por la opción programática. La complejidad de la *GUI* que se quiere diseñar, y la necesidad de emplear elementos no disponibles en la herramienta *GUIDE*, como es el caso de las pestañas (*tabs*), hacen que sea la mejor opción.

Para definir las variables del programa se emplean esencialmente dos estructuras. Una, llamada “H”, contiene las variables relativas al robot HOAP-3, y la otra, llamada “R”, contiene las variables relativas al robot RH-2.

De modo análogo, para definir las funciones y los *callbacks* se emplean las letras H y R seguidas de un guión bajo antes del nombre de la función. De este modo, tendremos por ejemplo las funciones “H\_representar” y “R\_representar”, para el HOAP-3 y el RH-2, respectivamente.

En la Figura 5.1 y la Figura 5.2 se muestra la organización del programa en las distintas funciones.



Figura 5.1. Estructura del programa (I)



Figura 5.2. Estructura del programa (II)

## 5.2.2 Funciones empleadas

### 5.2.2.1 Función *main*

La función *main* comprende el inicio del programa y la creación de la pantalla principal. Tiene anidada la función *CreateTab1* (ver punto 5.2.2.4), en la que se hacen la mayor parte de definiciones de la interfaz. A continuación se muestran las primeras líneas de código ya que a través de ellas se puede señalar alguno de los aspectos más importantes de la interfaz.

```
function main

    global ULTIMO_PATH
    global FACTOR_RADIANES
    global ANCHO
    global ALTO

    % Añadir las carpetas internas a la ruta de búsqueda
    ruta_actual = fileparts(mfilename('fullpath'));
    addpath(genpath(ruta_actual));

    FACTOR_RADIANES = pi/180;
    ULTIMO_PATH=ruta_actual;
```

La variable *ULTIMO\_PATH* se emplea para almacenar la última ruta a la que accede el programa. Esto facilita el proceso de guardar archivos, ya que normalmente interesa guardar todos los archivos de una trayectoria en una misma carpeta. Así, una vez elegida una ruta para guardar un archivo (mediante un cuadro de diálogo), al guardar los siguientes archivos se mostrará la misma ruta por defecto. Del mismo modo, cuando se estén cargando archivos de una carpeta, una vez cargado el primero se mostrará la ruta elegida en las sucesivas cargas. Para el primer intercambio de archivos, como aún no se ha indicado ninguna ruta, se mostrará por defecto la ruta donde está guardado el programa de la interfaz. Para ello se inicializa la variable al valor *ruta actual*. Esta variable se declara como global para posibilitar el acceso a ella de todos los programas que necesitan realizar intercambio de archivos con el usuario.

También se declara como global la variable *FACTOR\_RADIANES*, que equivale al factor de conversión para pasar de grados a radianes, que será utilizado en múltiples ocasiones por los diversos programas de la interfaz.

Mediante las siguientes sentencias se captura el tamaño de la pantalla en la que se está ejecutando el programa. Las variables *ALTO* y *ANCHO* han sido declaradas globales porque serán empleadas más adelante en diversos procesos para permitir

que el tamaño de los elementos de la interfaz se ajuste al de la pantalla correctamente.

```
screensize = get(0, 'ScreenSize');  
ANCHO = screensize (3);  
ALTO = screensize (4);
```

Dada la cantidad de objetos gráficos que tiene que cargar MATLAB, el programa necesita cierto tiempo para cargar la interfaz por completo. Mientras se cargan todos los objetos gráficos se muestra una pantalla azul con el mensaje: “Cargando interfaz”. Esta pantalla es la figura *espera* que se define con el tamaño de la pantalla completa mediante las siguientes sentencias:

```
espera = figure(...  
    'Name', '', ...  
    'NumberTitle', 'off', ...  
    'Menubar', 'none', ...  
    'Units', 'pixels', ...  
    'Position', [0,0,ANCHO,ALTO], ...  
    'Color', [0.4314,0.5882,0.8431]);  
  
uicontrol(...  
    'Parent', espera, ...  
    'Units', 'normalized', 'Position', [0.06,0.45,0.8,0.2], ...  
    'Style', 'Text', 'BackgroundColor', [0.4314,0.5882,0.8431], ...  
    'Fontname', 'Courier New', 'FontSize', 32, 'Fontweight', 'Bold', ...  
    'ForegroundColor', [1 1 1], 'FontAngle', 'normal', ...  
    'String', 'Cargando interfaz...');  
  
maximize (espera)
```

La figura se maximiza mediante la función **maximize** [18] (Ver punto 5.2.2.2).

La base de toda la interfaz es la figura *interfaz*, que se define a continuación. Cuando esta figura y todos sus controles están cargados por completo, la figura *espera* desaparece.

```
interfaz = figure(...  
    'Visible', 'off', ...  
    'Name', '', ...  
    'NumberTitle', 'off', ...  
    'Menubar', 'none', ...  
    'Units', 'normalized', ...  
    'Color', [0.7725,0.8431,0.9608], ...  
    'Position', [0,0,1,1]);
```

### 5.2.2.2 Función *maximize*

Esta función está disponible en la sección de intercambio de archivos de la página web MATLAB Central [18].

Haciendo una llamada a esta función se muestra a pantalla completa una figura, que se debe pasar como parámetro. La sintaxis es la siguiente:

```
maximize (figure)
```

En este caso, se realizará una llamada a esta función dentro del *main* (ver punto 5.2.2.1), para maximizar la pantalla temporal de espera. Y una vez cargados todos los controles de la interfaz, ésta también se maximizará.

### 5.2.2.3 Función *uitabpanel*

Para la implementación de las pestañas, al no estar disponibles en MATLAB en forma de *uicontrols*, se ha recurrido a la función *uitabpanel* [19] (ver punto 5.2.2.3). Esta función permite crear paneles con diversas formas a partir de unos parámetros definidos por el usuario.

Dentro del programa *main* (ver punto 5.2.2.1), una vez creada la figura *interfaz*, se realiza una llamada a esta función para crear el panel de la pantalla principal. Este panel será una figura con otras figuras dentro, que serán accesibles haciendo clic en las pestañas definidas en el extremo superior de esta. Así se tiene una pantalla para cada funcionalidad de la interfaz, con su nombre en la correspondiente pestaña.

```
uitabpanel(...  
    'Parent',interfaz,...  
    'TabPosition','lefttop',...  
    'Units','Pixels',...  
    'Position',[0,0,1.1*ANCHO,0.93*ALTO],...  
    'PanelBorderStyle','line',...  
    'Title',{'','CREAR TRAYECTORIA HOAP3',...  
    'UNIR TRAYECTORIAS HOAP3','CONVERTIR TRAYECTORIA HOAP3',...  
    'SIMULACIONES HOAP3','CREAR TRAYECTORIA RH2',...  
    'UNIR TRAYECTORIAS RH2','CONVERTIR TRAYECTORIA RH2',...  
    'SIMULACIONES RH2'},...  
    'CreateFcn',@CreateTab1);
```

Dentro de los paneles correspondientes a la pestaña “Crear trayectoria HOAP3” y “Crear trayectoria RH2” se crean una serie de paneles desplegables que servirán para introducir los puntos de la trayectoria que se quiere crear. A continuación se muestra el código para el caso del HOAP-3:



```
htab(1) = uitabpanel(...
    'Parent',hpanel(2),...
    'Style','popup',...
    'Units','normalized',...
    'Position',[0.05,0,0.8,0.98],...
    'FrameBackgroundColor',[0.4314,0.5882,0.8431],...
    'FrameBorderStyle','etchedin',...
    'Title',{'>>> Definir movimientos RIGHT LEG'},...
    '>>> Definir movimientos LEFT LEG',...
    '>>> Definir movimientos RIGHT ARM',...
    '>>> Definir movimientos LEFT ARM',...
    '>>> Definir movimientos BODY/HEAD'},...
    'PanelHeights',[0.0313*ALTO,0.0313*ALTO,...
    0.0187*ALTO,0.0187*ALTO,0.0187*ALTO],...
    'HorizontalAlignment','left',...
    'FontWeight','bold',...
    'TitleBackgroundColor',get(hstatus,'BackgroundColor'),...
    'TitleForegroundColor',[0.1294,0.3647,0.8510],...
    'PanelBackgroundColor',[0.7725,0.8431,0.9608],...
    'PanelBorderStyle','line',...
    'CreateFcn',@H_popup); % Crear tab con popup panels
```

Así se definen cuatro paneles, uno para cada parte del cuerpo del robot, distinguiendo brazos, piernas y tronco/cabeza. Así se crean las funciones *H\_popup* / *R\_popup* (ver punto 5.2.2.36).

En todas las pantallas de la interfaz se crea un panel que sirve para mostrar las gráficas. Éste se sitúa en la parte inferior, y tiene una barra de desplazamiento vertical para poder ver las distintas gráficas. Las siguientes sentencias corresponden a la creación de uno de estos paneles, el de la pantalla “Crear trayectoria HOAP3”.

```
htab(1) = uitabpanel(...
    'Parent',hpanel(2),...
    'Style','popup',...
    'Units','normalized',...
    'Position',[0.05,0,0.8,0.4],...
    'FrameBackgroundColor',[0.4314,0.5882,0.8431],...
    'FrameBorderStyle','etchedin',...
    'Title',{' '},...
    'PanelHeights',0.125*ALTO,...
    'HorizontalAlignment','left',...
    'FontWeight','bold',...
    'TitleBackgroundColor',get(hstatus,'BackgroundColor'),...
    'TitleForegroundColor',[0.1294,0.3647,0.8510],...
    'PanelBackgroundColor',[0.7725,0.8431,0.9608],...
    'PanelBorderStyle','line',...
    'CreateFcn',@H_panel_nueva); % Crear tab para las gráficas
```

Así se crean las funciones *H\_panel\_nueva* / *R\_panel\_nueva*, *H\_panel\_simular* / *R\_panel\_simular*, *H\_panel\_concatenar* / *R\_panel\_concatenar*, *H\_panel\_convertir* / *R\_panel\_convertir* (ver punto 5.2.2.14)

#### 5.2.2.4 Función *CreateTab1*

Esta función contiene la mayoría de las definiciones de los objetos gráficos de la interfaz. Como puede verse en la Figura 5.2, también contiene todos los *callbacks* de los objetos definidos en ella.

#### 5.2.2.5 Función *averiguar\_derivada*

Esta función se emplea cuando se quiere convertir una trayectoria. El objetivo es averiguar cuál es la transformación que quiere hacer el usuario.

Los parámetros de entrada de la función son las variables *datos\_entrada* y *datos\_salida*, que indican qué tipo de datos se están cargando y qué tipo de datos se quieren obtener tras la transformación. Antes de la llamada a la función hay que averiguar estos parámetros, leyendo la selección que ha hecho el usuario en las listas desplegables de la interfaz:

```
aux2 = get(H.convertir_datos_in, 'Value');  
string_list2 = get(H.convertir_datos_in, 'String');  
datos_entrada = string_list2{aux2};  
  
aux3 = get(H.convertir_datos_out, 'Value');  
string_list3 = get(H.convertir_datos_out, 'String');  
datos_salida = string_list3{aux3};
```

Después se puede hacer la llamada a la función del siguiente modo:

```
derivada = averiguar_derivada (datos_entrada,datos_salida);
```

La función compara el valor de *datos\_entrada* y *datos\_salida* con las cadenas de caracteres “POSICIÓN”, “VELOCIDAD” y “ACELERACIÓN”, y así se decide si es necesario derivar una, dos, o ninguna vez. La función devuelve el valor *derivada* que indica el número de veces que hay que derivar.

#### 5.2.2.6 Función *derivar*

Esta función se emplea al convertir una trayectoria. Al llamar a la función hay que pasarle dos parámetros, el primero de ellos una matriz que contiene la trayectoria que se quiere transformar, y el segundo el orden de la derivada que se requiere aplicar.

La función devuelve como parámetro de salida una matriz que contiene la derivada de los datos introducidos. El significado físico de esta derivada es obtener la velocidad como derivada de la posición, y la aceleración como derivada de la velocidad.

```
matriz_derivada = derivar (matriz_sin_derivar,derivada)
```

### 5.2.2.7 Función *interpolar*

Al ejecutarse la función **calcular\_trayectoria** (ver punto 5.2.2.11), se hace una llamada a **interpolar** por cada dos valores de posición de cada articulación. La sintaxis de la llamada es la siguiente:

```
[q,d_q,dd_q] = interpolar (paso, T, q0, q1);
```

donde *paso* es el paso de la trayectoria, *T* es el tiempo del intervalo en que se hace la interpolación, y *q0* y *q1* son la posición inicial y final del intervalo, respectivamente. En *q* se obtiene la posición interpolada, en *d\_q* la velocidad interpolada y en *dd\_q* la aceleración interpolada.

```
function [trayectoria, d_trayectoria, dd_trayectoria] = interpolar  
(Ts, T, p0, p1)  
  
x = 0 : Ts : T;  
auxiliar = size(x,2);  
a = -2*(p1-p0)/T^3;  
b = 3*(p1-p0)/T^2;  
trayectoria = a * x.^3 + b * x.^2 + p0*ones(1,auxiliar);  
d_trayectoria = 3 * a * x.^2 + 2 * b * x;  
dd_trayectoria = 6 * a * x + 2 * b;
```

end

La interpolación que se realiza es una interpolación cúbica que garantiza la existencia de la segunda derivada matemática de los datos. De este modo, la posición es una curva suave y la velocidad y la aceleración son continuas. Al crear trayectorias por este método, nos aseguramos de que no habrá picos de aceleración que puedan dañar los motores del robot.

La ecuación empleada es en realidad una particularización de la siguiente ecuación:

$$x = x_0 + v_0 \times t + c \times t^2 + d \times t^3$$

$$c = \frac{3 \times x_1 - x_0 - T \times 2 \times v_0 + v_1}{T^2}; \quad d = \frac{2 \times x_0 - x_1 + T \times (v_0 + v_1)}{T^3}$$

tomando la posición y la velocidad inicial el valor cero:

$$x_0 = 0; \quad v_0 = 0$$

### 5.2.2.8 Función *H\_comprobar\_rango* / *R\_comprobar\_rango*

Esta función sirve para comprobar si para una determinada trayectoria, las articulaciones del robot se mueven dentro del rango permitido. La sintaxis de una llamada a la función es la siguiente:

```
[excede, numero_joints, grados, articulacion]=  
R_comprobar_rango(matriz_entrada)
```

La única variable de entrada es una matriz cuyas columnas representan las articulaciones del robot y cuyas filas las posiciones que éstas toman. Si se detecta que alguna articulación sobrepasa el límite permitido, la variable *excede* toma el valor "1". La variable *numero\_joints* indica el número de articulaciones que sobrepasan su rango permitido. El vector *articulación* tiene tantos elementos como articulaciones sobrepasan el rango. Cada elemento indica el número de una de estas articulaciones. En el vector *grados*, cada elemento toma el valor en grados en que la articulación indicada en el mismo elemento del vector *articulación* sobrepasa el límite.

### 5.2.2.9 Función *identificar\_joints*

Cuando gracias a la función *H\_comprobar\_rango* / *R\_comprobar\_rango* (ver punto 5.2.2.8) se ha detectado que alguna articulación excede el rango de movimiento permitido, es necesario saber cuál es nombre de dicha articulación para poder mostrar un mensaje por pantalla. Para ello se realiza una llamada a esta función, indicando el número de la articulación que excede el rango permitido, y el robot del que se trata. La función devuelve la variable *nombre*, a la que asigna su valor a partir de una lista ordenada de nombres de articulaciones de cada robot de que dispone.

La sintaxis para llamar a la función es la siguiente:

```
nombre = identificar_joints (numero,robot)
```

### 5.2.2.10 Función *comprobar\_paso*

Cuando se está creando una trayectoria a partir de la unión de otras, hay que comprobar que el valor del paso es el mismo para todas, y el mismo que se ha seleccionado en el panel. Para ello, se llama a esta función, del siguiente modo:

```
indice =comprobar_paso (matriz,paso,numero)
```

La variable *numero* indica qué trayectoria es dentro del grupo de trayectorias que se está cargando. Así se puede mostrar un mensaje al usuario indicándole que el paso de una determinada trayectoria no coincide con el esperado.

### 5.2.2.11 Función *calcular\_trayectoria*

La sintaxis es la siguiente:

```
[posicion, velocidad, aceleracion, numero_pasos]=calcular_trayectoria(...  
    paso, tiempo_maximo, numero_valores, ...  
    entrada_radianes, entrada_tiempo, vuelta)
```

Cuando se está creando una trayectoria, se almacenan todos los valores de posición indicados por el usuario en una matriz, y todos los valores de tiempo en otra. Las columnas de estas matrices son las articulaciones del robot y las filas contienen las posiciones. Después se realiza una llamada a la función ***calcular\_trayectoria*** para cada articulación del robot, pasando como parámetro la columna de la matriz que le corresponde a dicha articulación.

La variable *paso* es el paso de la trayectoria y *tiempo\_máximo* es el valor de tiempo del último par posición-tiempo que se ha definido para la trayectoria. El vector *entrada\_radianes* comprende todos los valores de posición de la articulación definidos en radianes. Del mismo modo, *entrada\_tiempo* contiene todos los valores de tiempo. La variable *vuelta* toma el valor “1” si se ha seleccionado la opción de regresar a la posición inicial, y “0” si no se ha seleccionado.

En esta función se comprueba cuantos pares posición-tiempo han sido introducidos para la articulación que corresponde en la llamada a la función. Si el último valor de tiempo definido es menor que el *tiempo\_maximo* se añade un par de valores, cuyo tiempo es igual a *tiempo\_maximo* y posición es igual a la del último punto definido. Es decir, se mantiene constante la posición para la articulación hasta el último instante que ha sido definido para alguna articulación. Si *vuelta* vale “1”, se añade un punto de tiempo 1’5 segundos, si el paso de la trayectoria es 3 ms, o 1 segundo si el paso es distinto de 3 ms.

La función devuelve la variable *posición*, que es un vector que contiene las posiciones de la articulación durante la trayectoria, la variable *velocidad*, que contiene las velocidades y la variable *aceleración* que contiene las aceleraciones.

### 5.2.2.12 Función *guardar\_dat*

Para guardar los archivos de trayectorias en extensión *.dat* se emplea esta función. Hay que pasar cuatro parámetros a la función: la matriz que se quiere guardar, el nombre con el que se quiere guardar, el mensaje que se quiere mostrar en el cuadro de diálogo para guardarla y el mensaje de confirmación al haberla guardado. Los mensajes que se mostrarán al guardar un archivo serán distintos dependiendo de si corresponden a posición, velocidad o aceleración. Un ejemplo de llamada a la función para guardar un archivo de posiciones:

```
guardar_dat (matriz_posiciones_simulador, '\angle', ...  
            'Guardar trayectoria generada en .dat', ...  
            'Archivo .dat de posiciones guardado correctamente');
```

En caso de haber seleccionado en la lista desplegable de la interfaz que se quieren obtener los archivos de velocidad y aceleración nulas para el simulador, en lugar de guardar las matrices creadas por la función **calcular trayectoria** (ver punto 5.2.2.11), se crea una matriz de ceros del mismo tamaño, conservando para la primera columna los valores originales (paso de la simulación).

Dentro de la función **guardar\_dat**, para guardar los valores se emplea la función **dlmwrite** con una precisión de 15 valores en coma flotante. Esto es necesario para no perder la suavidad de las curvas obtenidas en la interpolación al recortar decimales.

### 5.2.2.13 Función *guardar\_valores*

Al crear una trayectoria, se guardan en un archivo con extensión *.dat* los pares tiempo-posición definidos por el usuario. De este modo se puede saber después cuales son los puntos a partir de los que se han realizado las interpolaciones. Esto puede ser útil cuando se quieren crear trayectorias similares a otras creadas anteriormente. Para cada articulación se reservan dos columnas, la primera para los valores de tiempo y la segunda para los valores de posición. En la primera fila de cada columna se indica a qué articulación corresponden los valores. La sintaxis de la llamada a la función es la siguiente:

```
guardar_valores(matriz_tiempos, matriz_posiciones, mensaje1, mensaje2,  
               nombre_archivo, robot)
```

Hay que pasar a la función la matriz con las posiciones y la matriz con los tiempos, el mensaje que se quiere mostrar en el cuadro de diálogo mediante el que se guardará el archivo, el mensaje de confirmación al guardar, y el nombre de archivo sugerido.

También hay que indicar de qué robot se trata para que el programa pueda escribir los nombres de las articulaciones en el archivo.

#### 5.2.2.14 Funciones para los paneles

Las siguientes funciones sirven para crear los ejes de coordenadas en los que se representarán las gráficas:

- H\_panel\_simular / R\_panel\_simular,
- H\_panel\_nueva / R\_panel\_nueva,
- H\_panel\_concatenar / R\_panel\_concatenar,
- H\_panel\_convertir / R\_panel\_convertir

Dentro de la función se declara la variable global *EJES*, una estructura que contendrá los identificadores de todos los ejes de la interfaz. Los campos de la estructura serán: *H\_simular*, *H\_nueva*, *H\_concatenar*, *H\_convertir*, *R\_simular*, *R\_nueva*, *R\_concatenar*, *R\_convertir*.

Así, por ejemplo la variable *EJES.H\_simular* es un vector cuya definición tiene lugar dentro de la función ***H\_panel\_simular*** (ver punto 5.2.2.14), y contiene los identificadores de los ejes del panel situado en la pantalla “Simular trayectoria HOAP-3”. Cada elemento del vector contiene el identificador de un eje de coordenadas. A continuación se muestra la definición de los dos primeros elementos de este vector.

```
EJES.H_simular(1)=axes(...  
    'Parent',hpanel(1),...  
    'Position',[0.07,0.03,0.9,0.15]);  
  
EJES.H_simular(2)=axes(...  
    'Parent',hpanel(1),...  
    'Position',[0.07,0.23,0.9,0.15]);
```

#### 5.2.2.15 Función *H\_guardar\_grafico* / *R\_guardar\_grafico*

Esta función crea una nueva figura en una ventana independiente en la que representa las gráficas que se hayan seleccionado. La variable *matriz* contiene los datos que se quiere representar, la variable *unidades*, puede ser radianes o grados y la variable *tipo\_datos* puede ser “POSICIÓN”, “VELOCIDAD” o “ACELERACIÓN”. Las variables *body\_head*, *left\_arm*, *right\_arm*, *left\_leg* y *right\_leg* toman el valor “1”, si se ha seleccionado que se quieren las gráficas de ese grupo de articulaciones y “0” si no se

ha seleccionado. Del mismo modo, si se ha indicado que todas las seleccionadas se guarden juntas, la variable *todos\_juntos* tomará el valor “1”.

```
H_guardar_grafico (matriz, unidades, tipo_datos, body_head, left_arm, ...  
                 right_arm, left_leg, right_leg, todos_juntos)
```

La nueva ventana es una figura de MATLAB, por lo que se dispone de una barra de herramientas que permite modificar las gráficas como se desee, además de guardarla como imagen en diversos formatos.

#### 5.2.2.16 Función *H\_representar* / *R\_representar*

Al crear, concatenar, simular y convertir trayectorias se representa en los paneles inferiores de la pantalla las gráficas resultantes. Para llevar a cabo estas representaciones se realizan llamadas a esta función. La sintaxis es la siguiente:

```
H_representar (matriz_representar, unidades, tipo_datos, panel)
```

El parámetro de entrada *matriz\_representar* es una matriz con la trayectoria que se quiere representar. La variable *unidades* indica si se ha seleccionado radianes o grados para visualizar las gráficas. La variable *tipo\_datos* toma los valores “ACELERACIÓN”, “VELOCIDAD” o “POSICIÓN”, y la variable *panel* es una variable que se emplea para indicarle a la función donde se quieren las gráficas: en el panel de la pantalla de creación de trayectorias, en el de la pantalla de simulación, etc. Por ejemplo, para que las gráficas aparecieran en el panel de la pantalla “Concatenar trayectorias HOAP-3”, *panel* debería tomar el valor: “EJES.H\_concatenar”.

#### 5.2.2.17 Función *H\_convertir\_dat*

Esta función se emplea cuando hay que convertir una trayectoria del formato con el que trabaja el robot al formato con el que trabaja el simulador. Para ello se siguen las tablas y parámetros de conversión indicados en el punto 2.2.2. de este documento. Es importante señalar que en esta función se suman 90° al *pitch* del brazo izquierdo y se restan 90° al *pitch* del brazo derecho, para compensar las diferencias entre el modelo del simulador y el robot real.

#### 5.2.2.18 Función *H\_convertir\_csv*

Esta función se emplea cuando hay que convertir una trayectoria del formato con el que trabaja el simulador al formato con el que trabaja el robot. También se emplea después de crear una nueva trayectoria, ya que éste proceso se hace en radianes, que son las unidades con las que guardamos las trayectorias en los archivos *.dat*.





al leerlos con esta función tienen el formato de las matrices de celdas. Por ello hay que convertirlos al formato *double* mediante el comando **cell2mat**.

#### 5.2.2.20 Función **H\_guardar\_csv**

Esta función se emplea cuando se quiere guardar las trayectorias en el formato que emplea el HOAP-3. Como parámetros de entrada hay que pasar la variable *matriz\_posiciones\_csv* que contiene la trayectoria en valores decimales, el nombre del archivo y los mensajes que se desea mostrar en el cuadro de diálogo y como confirmación al guardar el archivo.

```
H_guardar_csv (matriz_posiciones_csv, '\posicion', ...  
             'Guardar trayectoria generada en csv', ...  
             'Archivo .csv de posiciones guardado correctamente');
```

Para crear el archivo se emplea la función **fopen** y para escribir los datos la función **fprintf**. Después de escribir los valores para los motores se añaden cuatro letras “R” para los sensores.

#### 5.2.2.21 Función **H\_cambiar\_signo**

Como se comenta en el Capítulo 2.2, hay articulaciones cuyos motores están colocados en sentido contrario a los parámetros de Denavit\_Hartenberg desarrollados para el robot. El programa realiza los cambios necesarios para que el usuario no tenga que tener estas diferencias en cuenta. Estos cambios se realizan en esta función, donde se multiplica por el valor “-1” las articulaciones que lo necesitan (ver Tabla 2.5).

#### 5.2.2.22 Función **H\_concatenar\_paso\_Callback / R\_concatenar\_paso\_Callback**

Esta función se ejecuta al cambiar la selección del valor del paso cuando se está concatenando trayectorias. Sirve para actualizar la lista desplegable de valores para el tiempo de unión de las trayectorias. Si se ha seleccionado un valor de 3 ms para el paso, la lista desplegable de valores para el tiempo de unión será: “1.5” y “3”. Si se ha seleccionado un paso distinto de 3 ms, la lista de valores será “1”, “1,5”, “2”, “2.5” y “3”.

#### 5.2.2.23 Función **H\_concatenar\_numero\_Callback / R\_concatenar\_numero\_Callback**

Esta función se ejecuta cuando se modifica la selección del número de trayectorias. Se emplea para averiguar el número de trayectorias que se quieren concatenar, y así habilitar ese mismo número de botones para cargar las trayectorias. Si en algún

momento se cambia la selección, se deshabilitan todos los botones y se vuelven a habilitar sólo los que corresponden.

#### **5.2.2.24 Función *H\_concatenar\_cargar\_Callback* / *R\_concatenar\_cargar\_Callback***

Esta función se ejecuta cuando se hace clic en uno de los botones para cargar trayectoria de la pantalla “Concatenar trayectoria”. Antes de almacenar ningún valor, se comprueba que la trayectoria cumple todos los requisitos necesarios. Se comprueba que el paso es el que se ha indicado antes, y se comprueba que tiene una columna para cada articulación del robot. Si no cumple los requisitos, se muestra un mensaje. Si la trayectoria es correcta, se carga y se muestra un mensaje de confirmación.

En esta función se declara la variable global *REGISTRO\_HOAP* / *REGISTRO\_RH2* para llevar la cuenta de las trayectorias que ya se han cargado. También se declaran las variables globales *TH1*, *TH2*, *TH3*, *TH4*, *TH5* y *TH6* en el caso del HOAP-3, y *TR1*, *TR2*, *TR3*, *TR4*, *TR5* y *TR6* en el caso del RH-2. Estas variables sirven para almacenar cada una de las seis trayectorias que es posible cargar.

Cada vez que se carga una trayectoria, el correspondiente elemento del vector *REGISTRO\_HOAP* / *REGISTRO\_RH2* toma el valor “1” y la variable correspondiente (*TH1* / *TR1*, *TH2* / *TR2*, etc.) almacena la trayectoria que se ha cargado.

Después se recurre a la variable *REGISTRO\_HOAP* / *REGISTRO\_RH2* para habilitar o no el botón de creación de la trayectoria: si el valor de esta variable es igual al número de trayectorias que se quería cargar, quiere decir que ha terminado el proceso y se habilita el botón “Crear trayectoria”.

#### **5.2.2.25 Función *H\_concatenar\_crear\_Callback* / *R\_concatenar\_crear\_Callback***

Esta función se ejecuta al hacer clic en el botón “Crear trayectoria” de la pantalla “Unir trayectorias HOAP-3” / “Unir trayectorias RH-2”.

En primer lugar, averigua los parámetros indicados por el usuario para la unión de trayectorias: el paso, el número de trayectorias y el tiempo de unión. Después, para cada articulación del robot hace una interpolación entre el último punto de una trayectoria y el primero de la siguiente. Se hace una llamada a la función *interpolar* (ver punto 5.2.2.7) para cada unión entre dos trayectorias. Así, si se trata de unir tres

trayectorias, se harán dos interpolaciones, la primera entre la trayectoria uno y la trayectoria dos, y la segunda entre la trayectoria dos y la trayectoria tres.

Después se llama a la función **H\_representar / R\_representar** (ver punto 5.2.2.16) para ver las gráficas de la trayectoria creada, y se habilita el botón “Guardar trayectoria”.

#### **5.2.2.26 Función *H\_concatenar\_guardar\_Callback / R\_concatenar\_guardar\_Callback***

Esta función se ejecuta al hacer clic en el botón “Guardar trayectoria” de la pantalla “Unir trayectorias HOAP-3” / “Unir trayectorias RH-2”.

En primer lugar, se guarda el archivo de posiciones haciendo una llamada a la función **guardar\_dat** (ver punto 5.2.2.12). Después se comprueba si se ha seleccionado guardar la velocidad y aceleración reales, o archivos con valores nulos para hacer las simulaciones. Si se han elegido los archivos de valores nulos, se crea una matriz de ceros que conserve la primera columna original (paso de la simulación) y se guardan mediante **guardar\_dat**. Si se ha seleccionado que se quiere los archivos de velocidad y aceleración reales, se llama a la función **derivar** (ver punto 5.2.2.6) una vez, para el archivo de velocidades y dos veces para el archivo de aceleraciones.

En el caso del HOAP-3, después se llama a la función **H\_convertir\_csv** (ver punto 5.2.2.18) y **H\_guardar\_csv** (ver punto 5.2.2.20) para tener el archivo de posiciones de la trayectoria en el formato con el que trabaja el robot.

#### **5.2.2.27 Función *H\_convertir\_simular\_Callback / R\_convertir\_simular\_Callback***

Esta función se ejecuta al hacer clic en el botón “Simular trayectoria” de la pantalla “Convertir trayectoria”. En primer lugar se averigua la selección de los parámetros para la conversión que ha hecho el usuario: el formato de entrada para los datos, el tipo de datos de entrada, el tipo de datos de salida, y las unidades en que se quiere ver las gráficas. Con estos datos se llama a la función **averiguar\_derivada** (ver punto 5.2.2.5), para saber qué tipo de transformación hay que hacer. El orden de la derivada que hay que hacer se almacena en la variable *derivada* para su uso posterior.

Si el formato de datos de entrada es .csv, se llama a la función **H\_leer\_csv** (ver punto 5.2.2.19) y después a la función **H\_convertir\_dat** (ver punto 5.2.2.17). Una vez hecho esto se llama a la función **H\_comprobar\_rango** (ver punto 5.2.2.8) para ver si la

trayectoria cumple con las restricciones de movimiento de las articulaciones. Si la trayectoria no está dentro del rango permitido se llama a la función **identificar\_joints** (ver punto 5.2.2.9) para poder mostrar un mensaje indicando para qué articulaciones no se cumple con las restricciones y en cuantos grados se exceden. Si la trayectoria sí que cumple con las restricciones, la variable local *matriz\_leida* toma el valor “1”.

Si el formato de entrada de los datos fuera *.dat*, sólo habría que llamar a la función **H\_comprobar\_rango**, y del mismo modo que antes a la función **identificar\_joints** si fuese necesario. Igual que antes, si la trayectoria cumple con las restricciones, la variable local *matriz\_leida* toma el valor “1”.

Después de estas operaciones, si el valor de la variable *matriz\_leida* tiene el valor “1”, y el valor de la variable *derivada* es distinto de “0”, se llama a la función **derivar** (ver punto 5.2.2.6). En el caso del HOAP-3, para representar los datos con el mismo criterio de signos que se indicó en el Capítulo 2.2, se llama a la función **H\_cambiar\_signo** (ver punto 5.2.2.21) y después a la función **H\_representar** (ver punto 5.2.2.16).

#### 5.2.2.28 Función **H\_convertir\_guardar\_Callback**

Esta función se ejecuta cuando se hace clic en el botón “Guardar trayectoria” de la pantalla “Convertir trayectoria HOAP-3”. En primer lugar se averigua qué representan los datos obtenidos en la conversión y en qué formato se quiere guardar el archivo.

Si el formato elegido es *.csv*, se llama a la función **H\_convertir\_csv** (ver punto 5.2.2.18) y después a **H\_guardar\_csv** (ver punto 5.2.2.20) para guardar el archivo que se ha convertido. Si el formato es *.dat*, se llama a la función **guardar\_dat** (ver punto 5.2.2.12).

#### 5.2.2.29 Función **R\_convertir\_guardar\_Callback**

Esta función se ejecuta cuando se hace clic en el botón “Guardar trayectoria” de la pantalla “Convertir trayectoria RH-2” En primer lugar se averigua qué representan los datos obtenidos en la transformación, para poder asignar el nombre correspondiente al archivo al guardarlo. Después se llama a la función **guardar\_dat** (ver punto 5.2.2.12).

### 5.2.2.30 Función *H\_convertir\_seleccion\_in\_Callback* /

#### *R\_convertir\_seleccion\_in\_Callback*

Esta función se ejecuta cada vez que se modifica la selección de las listas desplegadas de los parámetros de la conversión de trayectorias. Estos son el formato de los datos de entrada en la conversión de trayectorias, el formato de salida (.dat o .csv) y el tipo de datos que se quiere convertir.

Esta función se encarga de actualizar las listas desplegadas en función de la selección de formato o del tipo de datos que se ha hecho, de manera que nunca se puedan seleccionar combinaciones erróneas. Por ejemplo, si el formato de entrada es .csv, en la lista desplegable del tipo de datos de entrada solo aparecerá "POSICIÓN".

### 5.2.2.31 Funciones para seleccionar gráficos

Las siguientes funciones son muy similares entre sí, solo difieren en las variables que emplean:

- H\_nueva\_seleccion\_grafico\_Callback /  
R\_nueva\_seleccion\_grafico\_Callback,
- H\_convertir\_seleccion\_grafico\_Callback /  
R\_convertir\_seleccion\_grafico\_Callback,
- H\_simular\_seleccion\_grafico\_Callback /  
R\_simular\_seleccion\_grafico\_Callback,
- H\_concatenar\_seleccion\_grafico\_Callback /  
R\_concatenar\_seleccion\_grafico\_Callback,

Cada función se ejecuta al hacer clic en alguna de las casillas seleccionables del cuadro "Guardar gráficas" de la pantalla correspondiente. Si alguna de ellas está seleccionada, se habilita el botón guardar gráficos y la lista desplegable para seleccionar las unidades. Si no hay ninguna seleccionada, estos controles permanecen deshabilitados.

En caso de haber más de una casilla seleccionada, se habilita "Guardar selección en la misma imagen", para agrupar las gráficas seleccionadas en la misma figura. Si en cualquier momento se selecciona la casilla "Todos", aparecen seleccionadas automáticamente todas las casillas de selección de gráficas.

### 5.2.2.32 Funciones para guardar gráficos

Estas funciones son las acciones asociadas a los botones correspondientes de “Guardar gráficas” que hay en cada pantalla de la interfaz.

- H\_nueva\_guardar\_grafico\_Callback /  
R\_nueva\_guardar\_grafico\_Callback,
- H\_convertir\_guardar\_grafico\_Callback /  
R\_convertir\_guardar\_grafico\_Callback,
- H\_simular\_guardar\_grafico\_Callback /  
R\_simular\_guardar\_grafico\_Callback,
- H\_concatenar\_guardar\_grafico\_Callback /  
R\_concatenar\_guardar\_grafico\_Callback

En primer lugar, se averigua qué grupos de articulaciones se han seleccionado para guardar las gráficas, y las unidades en las que se quiere que estén las gráficas. Después se realiza una llamada a la función **H\_guardar\_grafico / R\_guardar\_grafico** (ver punto 5.2.2.15) pasándole estos parámetros.

### 5.2.2.33 Función **H\_simular\_formato\_Callback**

Esta función se ejecuta cada vez que se modifica la selección de formato de archivo en la pantalla “Simular trayectoria HOAP-3”. Si se selecciona “CSV”, la lista desplegable para seleccionar el tipo de datos queda reducida a un único valor: “POSICIÓN”. Si se selecciona “DAT”, esta lista se amplía a los valores: “POSICIÓN”, “VELOCIDAD” y “ACELERACIÓN”.

### 5.2.2.34 Función **H\_simular\_Callback**

Esta función se ejecuta al hacer clic en el botón “Simular trayectoria”, de la pantalla “Simular trayectoria HOAP-3”. En primer lugar, se averigua en qué formato se quiere cargar la trayectoria. Si se trata de formato .csv, se llama a la función **H\_leer\_csv** (ver punto 5.2.2.19). Después se llama a la función **H\_comprobar\_rango** (ver punto 5.2.2.8). Si la trayectoria excede el rango de movimiento permitido para alguna articulación, se llama a **identificar\_joints** (ver punto 5.2.2.9) para poder mostrar un mensaje indicando qué articulaciones superan el rango permitido y en cuantos grados

lo hacen. Después se llama a ***H\_cambiar\_signo*** (ver punto 5.2.2.21) y a ***H\_representar*** (ver punto 5.2.2.16) para ver las gráficas.

Si se trata de formato .dat, se llama a la función ***H\_comprobar\_rango*** y del mismo modo que antes se llama a ***identificar\_joints*** si es necesario. Después se llama a ***H\_cambiar\_signo*** y a ***H\_representar***.

#### **5.2.2.35 Función *R\_simular\_Callback***

Esta función se ejecuta al hacer clic en el botón “Simular trayectoria”, de la pantalla “Simular trayectoria RH-2”. Después se llama a la función ***R\_comprobar\_rango*** (ver punto 5.2.2.8). Si la trayectoria excede el rango de movimiento permitido para alguna articulación, se llama a ***identificar\_joints*** (ver punto 5.2.2.9) para poder mostrar un mensaje indicando qué articulaciones superan el rango permitido y en cuantos grados lo hacen. Después se llama a ***R\_representar*** (ver punto 5.2.2.16) para ver las gráficas.

#### **5.2.2.36 Función *H\_popup / R\_popup***

En esta función se definen todos los controles que ocupan el panel de la pantalla de creación de nuevas trayectorias. En la Figura 5.2 puede verse como también contiene todos los *callbacks* asociados a estos controles. Este panel contiene una serie de paneles que se despliegan haciendo clic en ellos, donde están las cajas de texto editables para introducir los puntos que definen las trayectorias.

#### **5.2.2.37 Función *H\_nueva\_paso\_Callback / R\_nueva\_paso\_Callback***

Esta función se ejecuta cuando se modifica la selección de la lista desplegable de valores para el paso de la trayectoria. En ella se averigua el valor del paso, y en función de este valor se muestra un aviso por pantalla que indica la duración que deben tener los intervalos de tiempo de la trayectoria. Si el paso es 3 ms, los intervalos deben ser múltiplos de 0.75 segundos, y si el paso es distinto de 3 ms, los intervalos deben ser múltiplos de 0.5 segundos.

#### **5.2.2.38 Función *H\_nueva\_valor\_Callback / R\_nueva\_valor\_Callback***

Esta función se ejecuta cada vez que se introduce algún valor de tiempo o posición en las cajas de texto editables. Se realizan una serie de comprobaciones de los datos. Si se trata de un valor de tiempo, se comprueba que sea mayor que el del par de puntos anterior, que la diferencia entre este y el anterior sea menor que veinte segundos, etc. Si se trata de un valor de posición, se comprueba que esté definido el par de valores anterior, que éste esté dentro del rango de movimiento permitido, etc.



### 5.2.2.39 Función *H\_nueva\_crear\_Callback / R\_nueva\_crear\_Callback*

Una vez introducidos los parámetros necesarios para crear una trayectoria, al hacer clic en el botón “Guardar trayectoria” se ejecuta la función **H\_nueva\_crear\_Callback / R\_nueva\_crear\_Callback**. Esta función lee los valores de los parámetros seleccionados por el usuario. También recoge todos los valores de posición y tiempo introducidos. Si algún valor de tiempo no tiene el correspondiente valor de posición definido o viceversa, ese par de valores es eliminado y en el propio recuadro de la interfaz aparecerá el signo “-” que indica que no hay valor definido.

Después se realizan las llamadas a la función **calcular\_trayectoria** (ver punto 5.2.2.11) y a la función **H\_representar / R\_representar** (ver punto 5.2.2.16).

### 5.2.2.40 Función *H\_nueva\_limpiar\_Callback / R\_nueva\_limpiar\_Callback*

Esta función ejecuta la acción asociada al botón “Limpiar valores”. Si se hace clic en él, se eliminan todos los tiempos o posiciones que se hayan introducido hasta el momento y se muestra un mensaje de confirmación.

### 5.2.2.41 Función *H\_nueva\_guardar\_Callback / R\_nueva\_guardar\_Callback*

Esta función ejecuta la acción asociada a los botones “Guardar trayectoria”.

Si se trata del RH-2, en la función **R\_nueva\_guardar\_Callback** se efectúan llamadas a la función **guardar\_dat** (ver punto 5.2.2.12) para guardar los archivos.

Si se trata del HOAP-3, en primer lugar se llama a la función **H\_cambiar\_signo** (ver punto 5.2.2.21) para adecuar los datos al convenio de signos del robot y el simulador. Después se llama a las funciones **guardar\_dat**, **H\_convertir\_csv** (ver punto 5.2.2.18) y **H\_guardar\_csv** (ver punto 5.2.2.20).

### 5.2.2.42 Función *Help\_Callback*

Esta función se ejecuta al hacer clic en el botón “Help” que se pueden encontrar en cada una de las pantallas de la interfaz.

```
function Help_Callback (varargin)
    open('Help.pdf');
end
```



Esta función abre el archivo "Help.pdf". Este documento se puede encontrar en el anexo de esta memoria.

## **6 Creación de una trayectoria**

### **6.1 Introducción**

Para comprobar que la interfaz funciona correctamente, se han creado cinco trayectorias para el robot HOAP-3. Estas cinco trayectorias se han unido por medio de la interfaz para formar una trayectoria de mayor duración, y comprobar así que la unión de trayectorias funciona correctamente. Las trayectorias comprenden una secuencia de movimientos sencillos que involucran a la mayoría de las articulaciones del robot.

En el CD adjunto a esta memoria se pueden encontrar los archivos de las trayectorias, así como los vídeos de la simulación y la prueba en el robot. También se pueden encontrar los archivos con puntos que dan lugar a cada trayectoria. Para realizar las simulaciones se ha eliminado la velocidad y la aceleración de todas las articulaciones excepto las de las piernas. Por tanto, se ha dado valor "0" a todas las posiciones de las columnas que corresponden a articulaciones que no pertenecen a las piernas. Sin embargo, los archivos que se aportan en el CD conservan las velocidades y aceleraciones originales de todas las articulaciones.

A continuación se comenta brevemente en qué consiste cada trayectoria. Se aportan algunas imágenes de la simulación en OpenHRP3, así como gráficas del comportamiento de las articulaciones. No se aportan las gráficas de las articulaciones de la cabeza y el tronco, ya que éstas no se mueven durante estas trayectorias, permaneciendo todo el tiempo en la posición inicial.

## 6.2 Trayectoria 1: “Sube y baja”

En esta trayectoria, tras flexionar los brazos repetidas veces, el robot termina con los brazos extendidos hacia arriba. Tras ello, flexiona las rodillas varias veces subiendo y bajando el tronco, a la vez que estira y recoge de manera alternada un brazo y el contrario. Después de esto, vuelve a subir el tronco y mueve los brazos imitando dar palmadas. En la Figura 6.1 se muestra el comienzo del archivo con los puntos definidos para crear esta trayectoria.

	RLEG HIP Y	RLEG HIP X	RLEG HIP R	RLEG HIP P	RLEG KNEE	RLEG ANKLE P	RLEG ANKLE R	RARM SHOULDER P	RARM SHOULDER R
1	0	0	0	0	0	0	0	0	0
2	-	-	7	0	7	-18	7	46	7
3	-	-	9	2	9	-40	9	95	9
4	-	-	11	0	11	-18	11	46	11
5	-	-	13	2	13	-40	13	95	13
6	-	-	15	0	15	-18	15	46	15
7	-	-	17	2	17	-40	17	95	17
8	-	-	19	0	19	-18	19	46	19
9	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-

Figura 6.1. Puntos definidos para la trayectoria “Sube y baja”

En la Figura 6.2 se puede observar la posición inicial del robot en la trayectoria “sube y baja”, que es también la posición de reposo del robot. En la Figura 6.3 se captura un instante de la segunda parte de la trayectoria, en la que está flexionando las rodillas y recogiendo el brazo izquierdo.

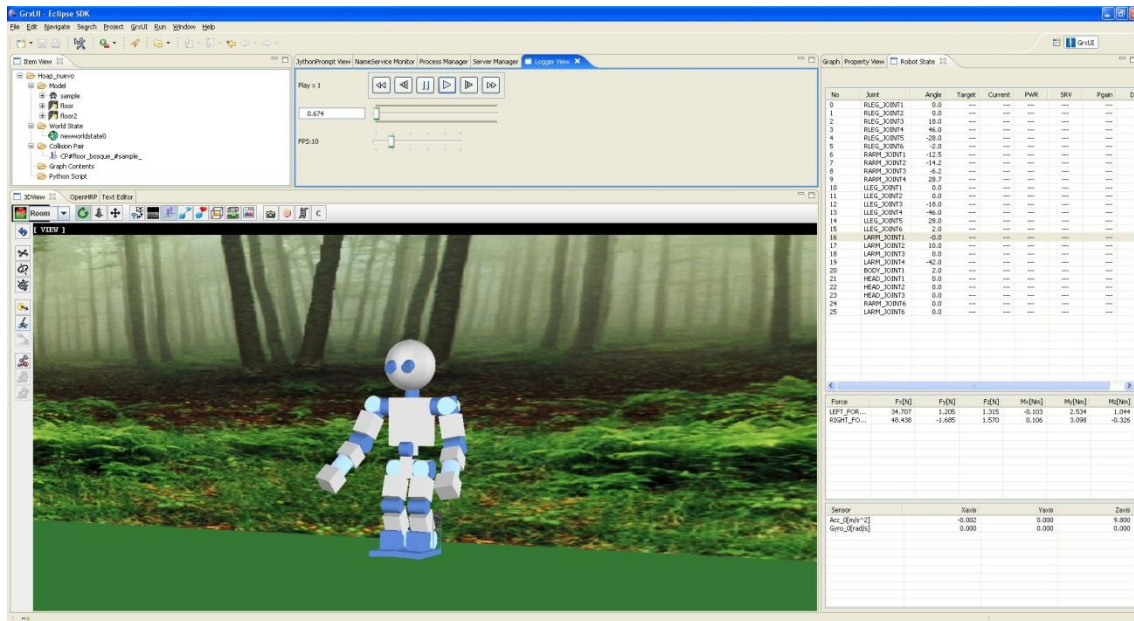


Figura 6.2. Posición inicial en la trayectoria "Sube y baja"

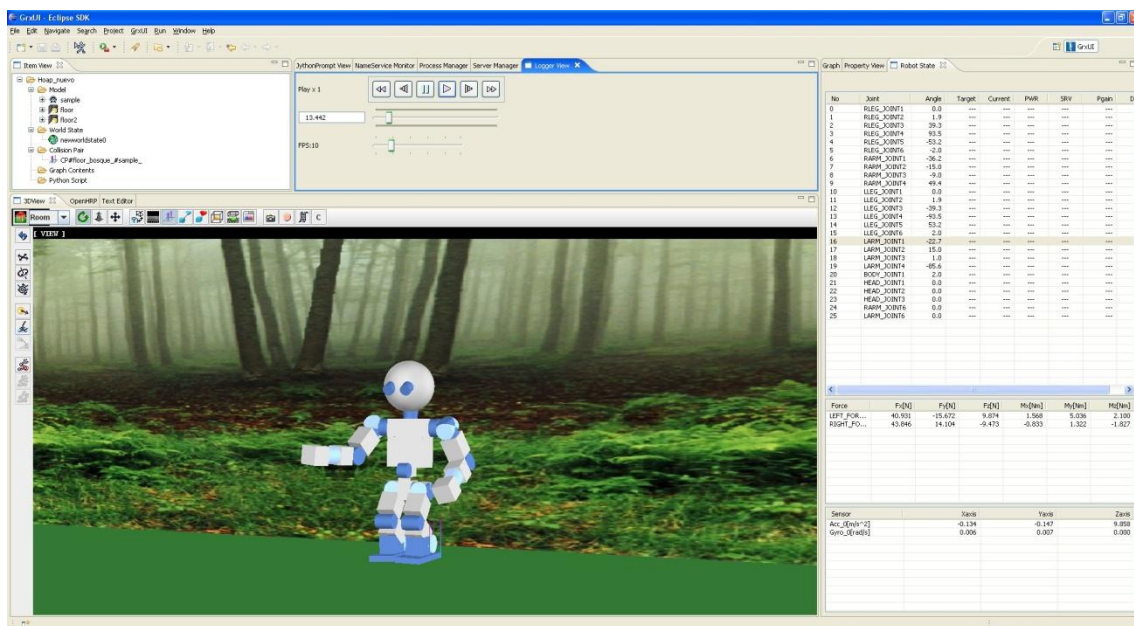


Figura 6.3. Trayectoria "Sube y baja": rodillas flexionadas

En la Figura 6.4 se puede observar las gráficas que representan las posiciones que toman las articulaciones de los brazos y piernas del robot en esta trayectoria.

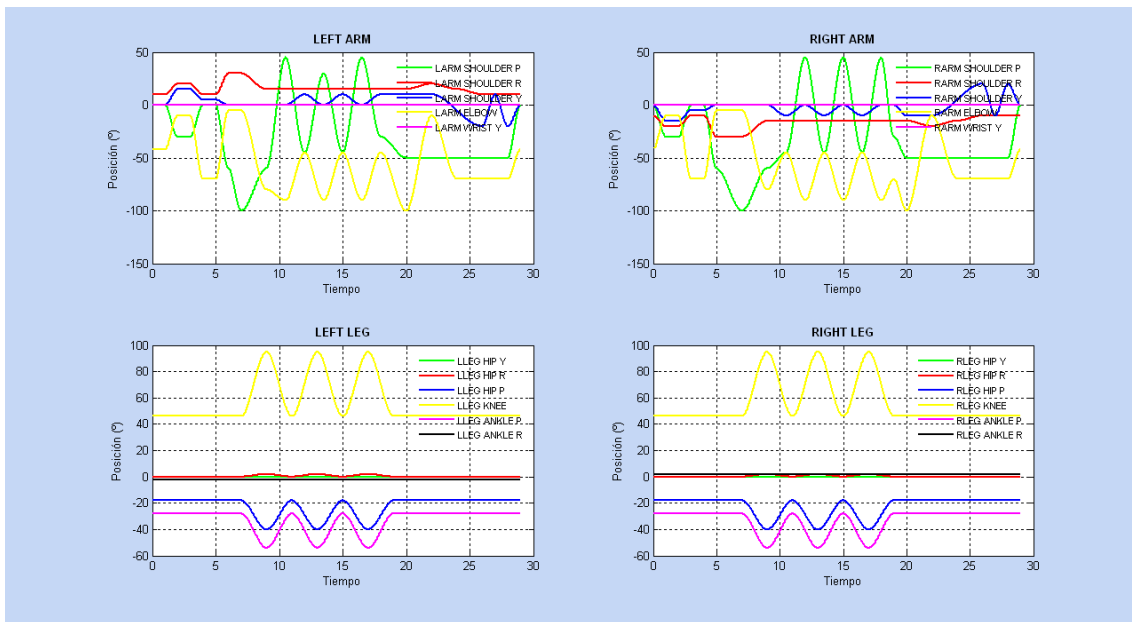


Figura 6.4. Gráficas de posición en la trayectoria "Sube y baja"

En la gráfica de la Figura 6.4 se puede comprobar que las dos piernas realizan los mismos movimientos. Se aprecia el movimiento de flexión de las rodillas (color amarillo) que tiene lugar entre los segundos 7 y 17, oscilando entre una flexión de  $46^{\circ}$  y  $95^{\circ}$ . También se puede observar el movimiento de las articulaciones del tobillo (colores azul y morado), que acompañan la flexión de las piernas del mismo modo que las rodillas.

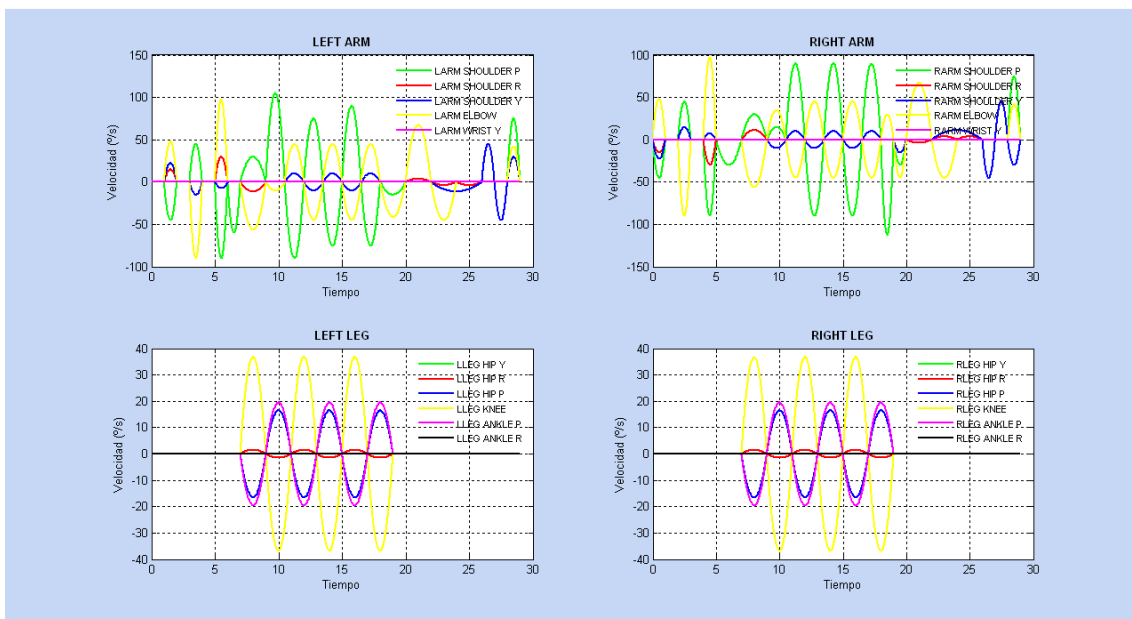


Figura 6.5. Gráficas de velocidad en la trayectoria "Sube y baja"

En la Figura 6.5 se puede observar la velocidad de las articulaciones durante la trayectoria, y en la Figura 6.6 la aceleración de las mismas. Como es de suponer, en este caso también la velocidad y la aceleración serán idénticas para ambas piernas.

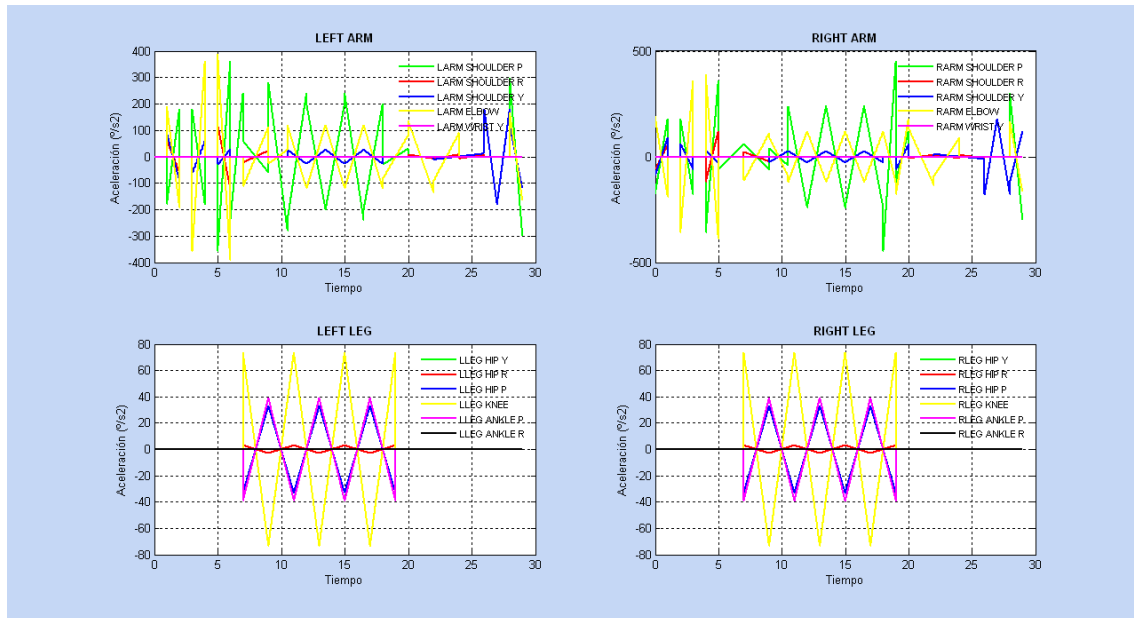


Figura 6.6. Gráficas de aceleración en la trayectoria "Sube y baja"

Se puede observar la diferencia de velocidad entre articulaciones como el codo (color amarillo) y el *pitch* del brazo (color verde) que se mueven más rápido y articulaciones como el *yaw* del brazo (color azul) o el *roll* de la pierna (color rojo). Éstas últimas varían mucho menos su posición que las primeras, y por tanto se ven sometidas a velocidades y aceleraciones muy inferiores a las primeras.

## 6.3 Trayectoria 2: "Saludo derecha"

En esta trayectoria el robot coloca el brazo izquierdo flexionado frente al tronco, y dispone al brazo izquierdo levantado hacia el lado derecho, en paralelo al tronco. Con el brazo izquierdo fijo en su posición, sube y baja el brazo derecho, a modo de saludo.

En la Figura 6.7 se puede ver una imagen de la simulación de esta trayectoria, correspondiente al instante en que ha colocado los brazos en las posiciones mencionadas y está comenzando a levantar el brazo derecho.

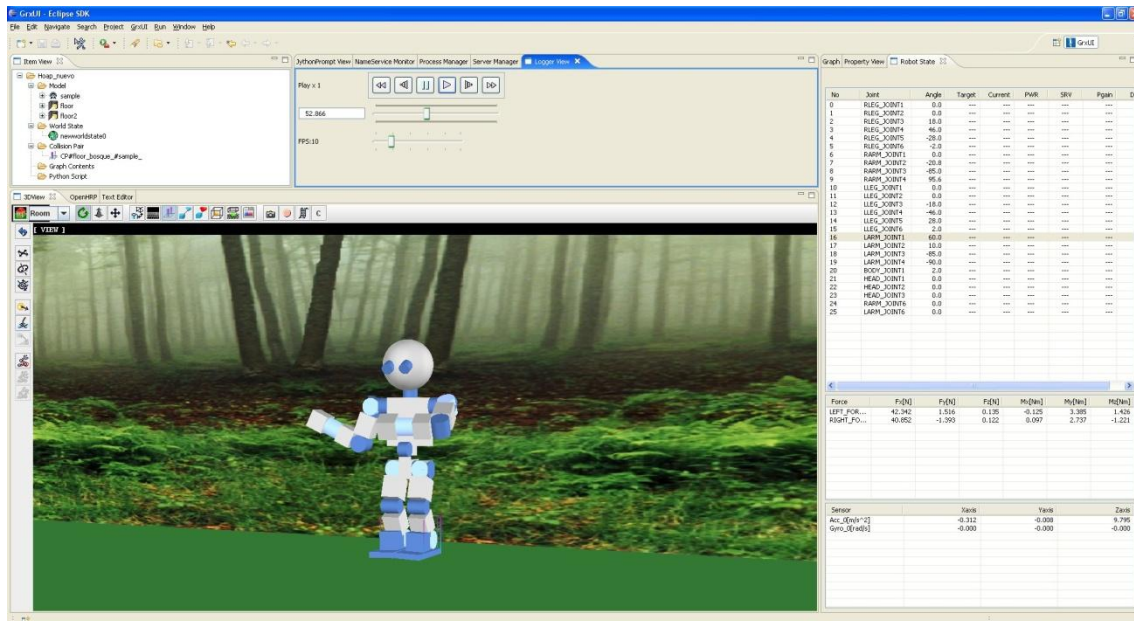


Figura 6.7. Trayectoria "Saludo derecha"

En esta trayectoria las piernas no se mueven. El robot permanece en la posición inicial, en la cual tiene las rodillas ligeramente flexionadas, y se limita a mover los brazos en esta posición. Por ello, no se aportan las gráficas de las articulaciones de las piernas.

En la gráfica de la Figura 6.8 se puede ver cómo el brazo izquierdo permanece inmóvil a partir de los 2 segundos de comenzar la trayectoria. Se puede ver la flexión del codo derecho (color amarillo) entre los segundos 5 y 9.

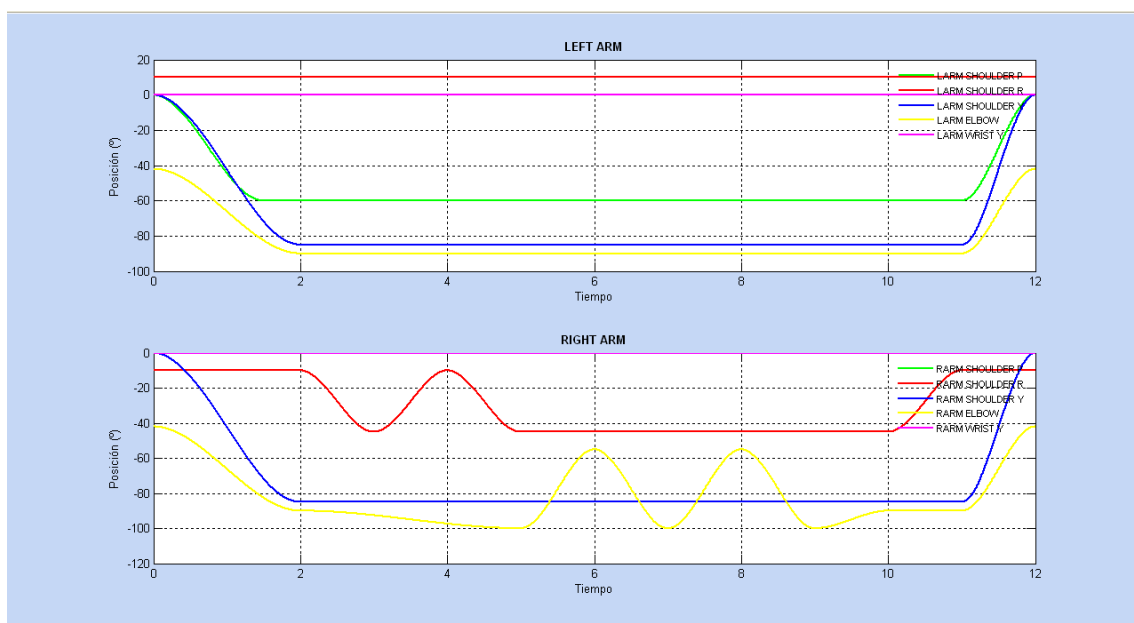


Figura 6.8. Gráficas de posición de los brazos en la trayectoria "Saludo derecha"



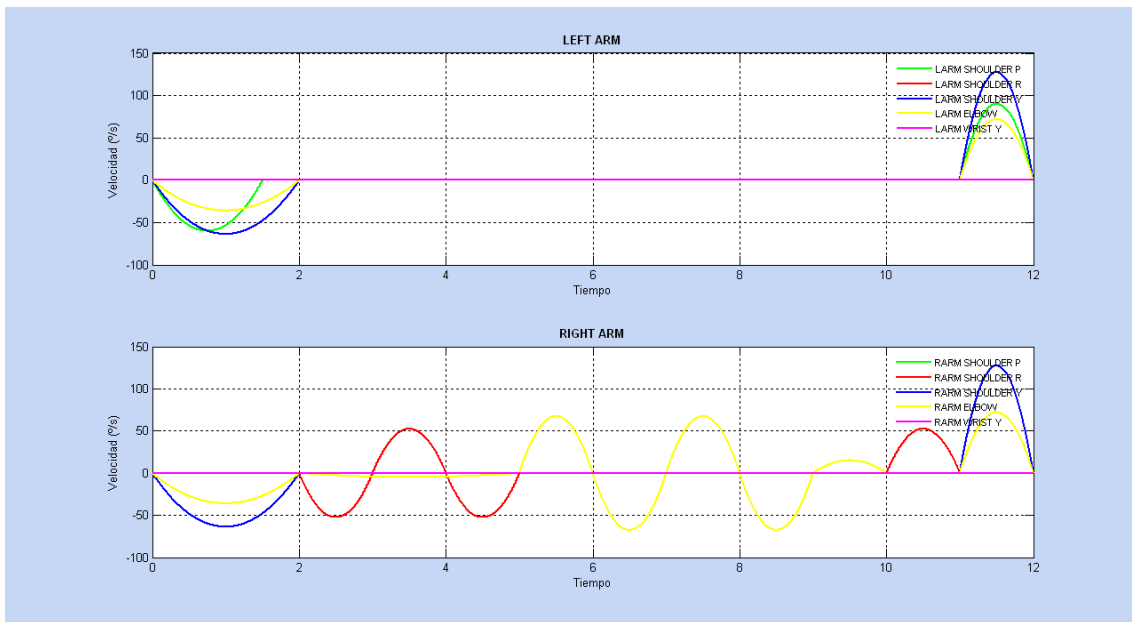


Figura 6.9. Gráficas de velocidad de los brazos en la trayectoria "Saludo derecha"

En la gráfica de la Figura 6.9 se ve claramente como las articulaciones del brazo izquierdo se mueven durante los primeros segundos, y al final para recobrar la posición de reposo del robot. El resto del tiempo, entre los segundos 2 y 11, éstas permanecen inmóviles.

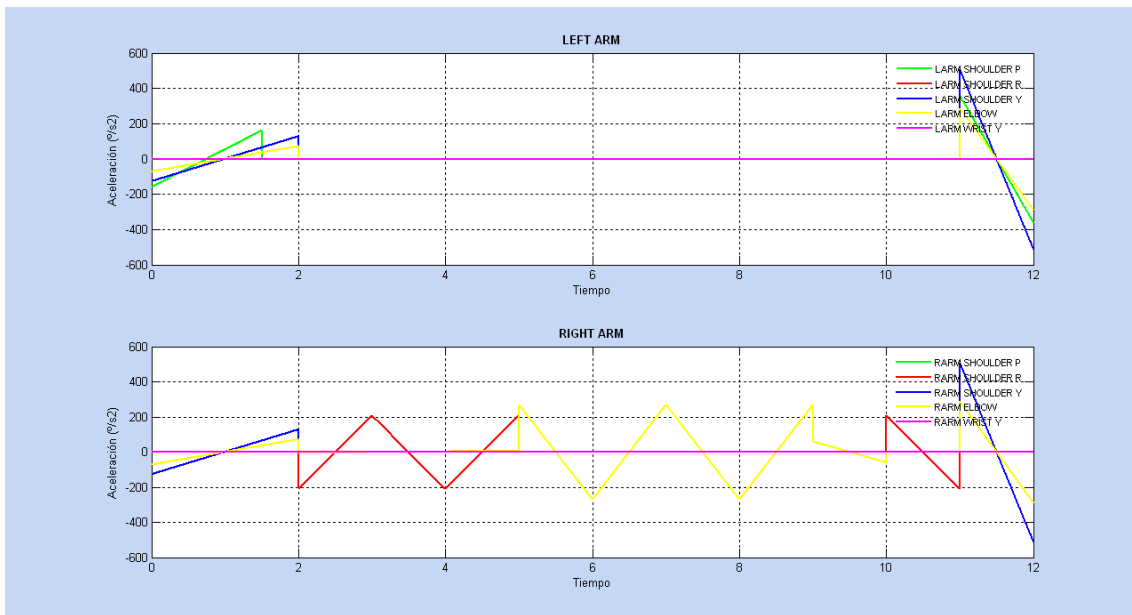


Figura 6.10. Gráficas de aceleración de los brazos en la trayectoria "Saludo derecha"

## 6.4 Trayectoria 3: "Saludo izquierda"

En esta trayectoria, el robot realiza movimientos equivalentes a los de la descrita en el punto anterior, "Saludo derecha", pero intercambiando los movimientos de un brazo por los del otro. Si en la anterior trayectoria dejaba inmóvil el brazo izquierdo para mover el derecho, ahora es el izquierdo el que queda inmóvil para mover el derecho.

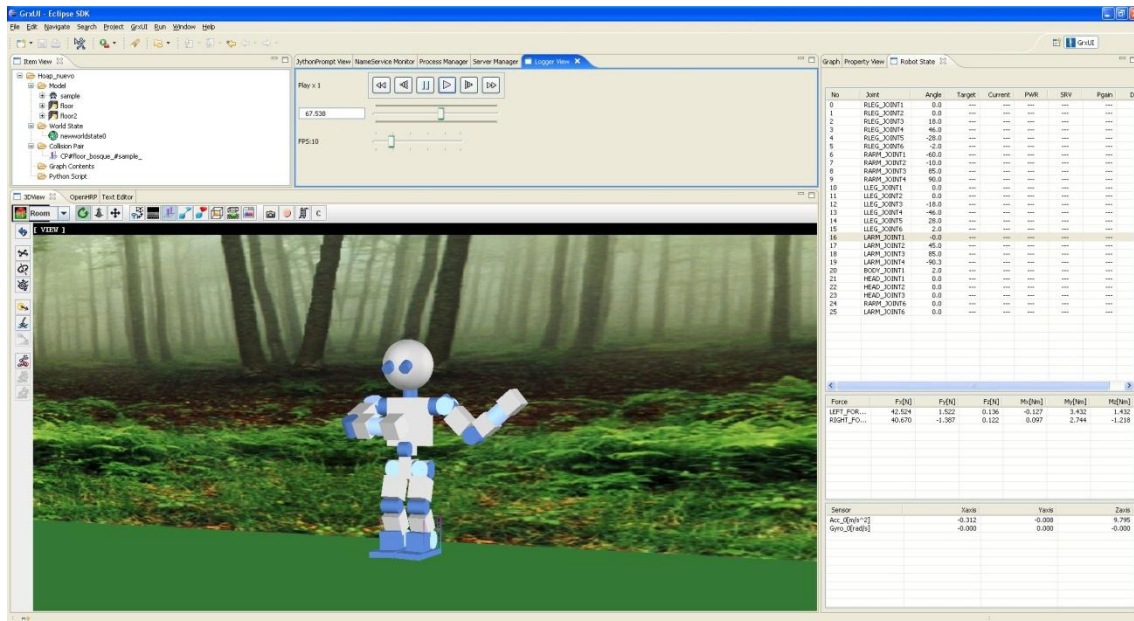


Figura 6.11. Trayectoria "Saludo izquierda"

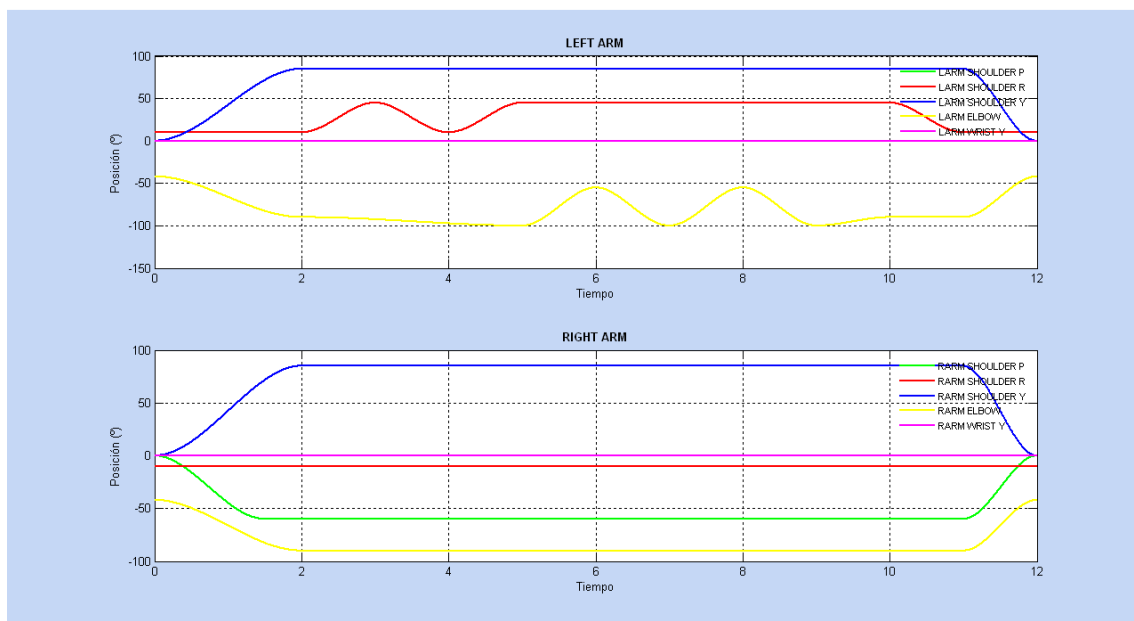


Figura 6.12. Gráficas de posición en la trayectoria "Saludo izquierda"

Al igual que en la trayectoria "Saludo derecha", el robot mantiene las piernas flexionadas en la posición inicial, moviendo sólo los brazos. Por este motivo no se muestran las gráficas de las articulaciones de las piernas.

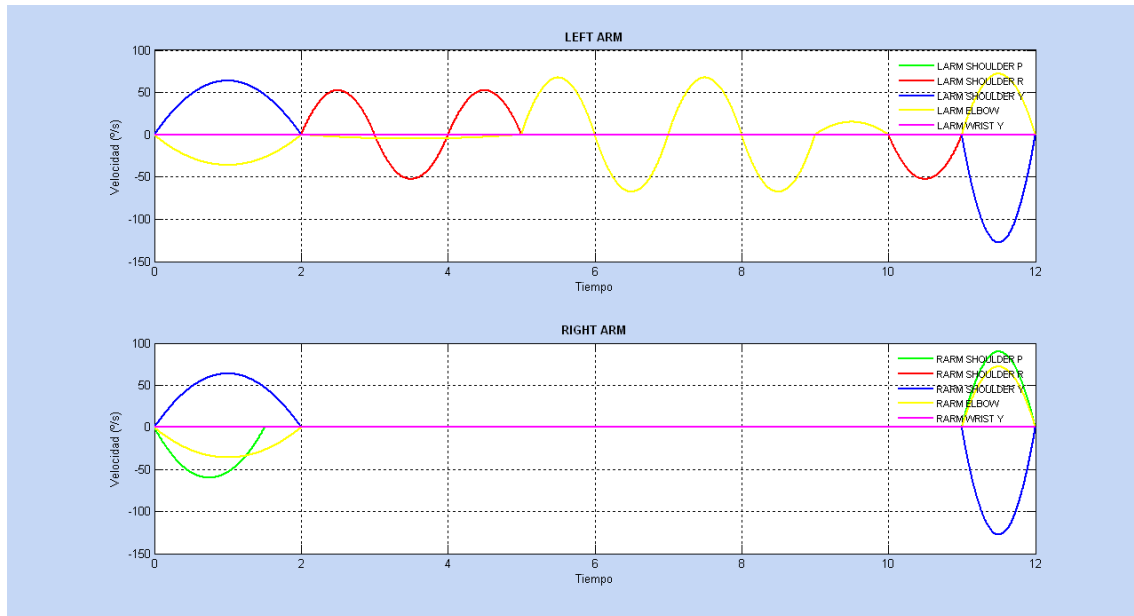


Figura 6.13. Gráficas de velocidad en la trayectoria "Saludo izquierda"

En la gráfica de la Figura 6.8 se puede ver como el brazo derecho permanece inmóvil a partir de los 2 segundos de comenzar la trayectoria. Se puede ver la flexión del codo izquierdo (color amarillo) entre los segundos 5 y 9.

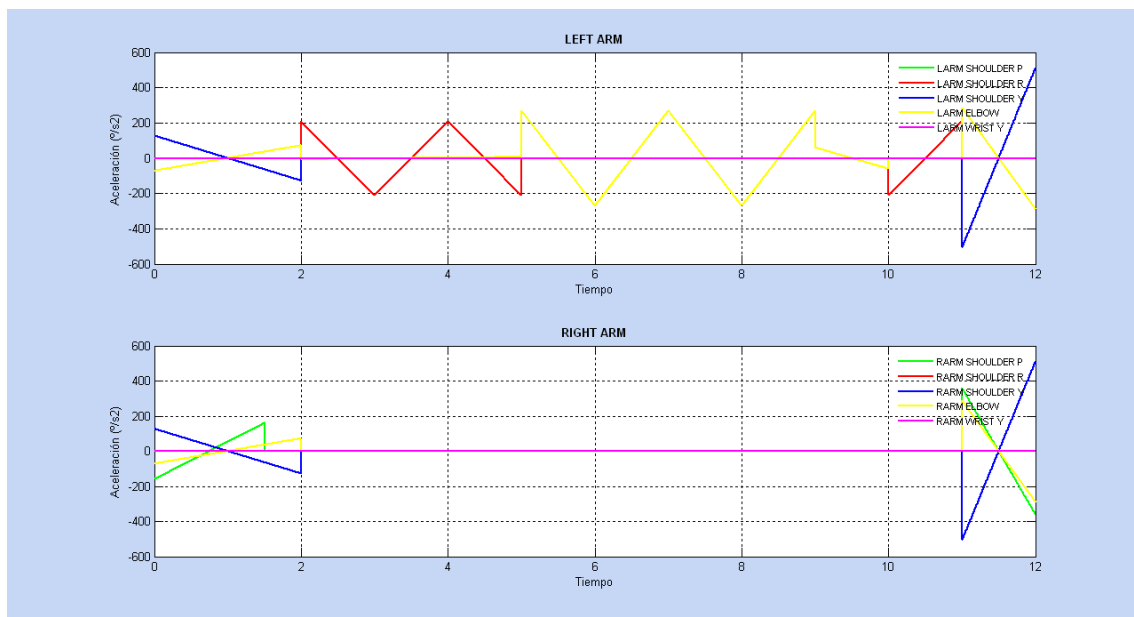


Figura 6.14. Gráficas de aceleración en la trayectoria "Saludo izquierda"

## 6.5 Trayectoria 4: "Gimnasia"

En esta trayectoria, el robot parte de la posición de reposo en que tiene las piernas ligeramente flexionadas. A continuación flexiona aún más las piernas, a la vez que eleva los dos brazos hasta la altura de los hombros, con los codos flexionados.

En la Figura 6.15 y la Figura 6.16 se muestran dos instantes de la trayectoria.

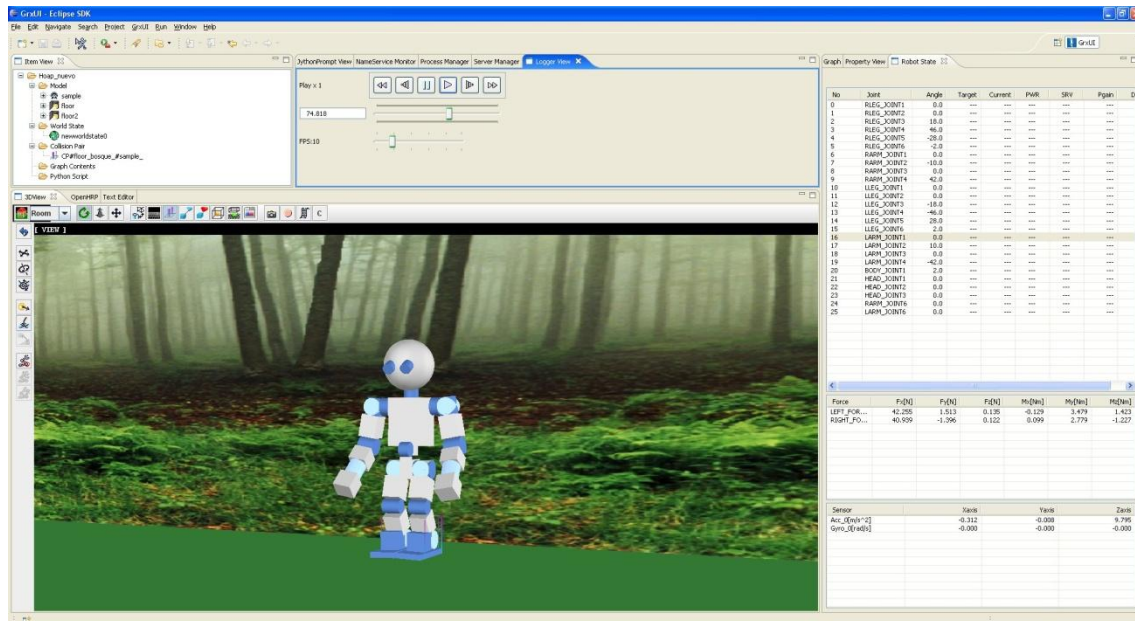


Figura 6.15. Comienzo de la trayectoria "Gimnasia"

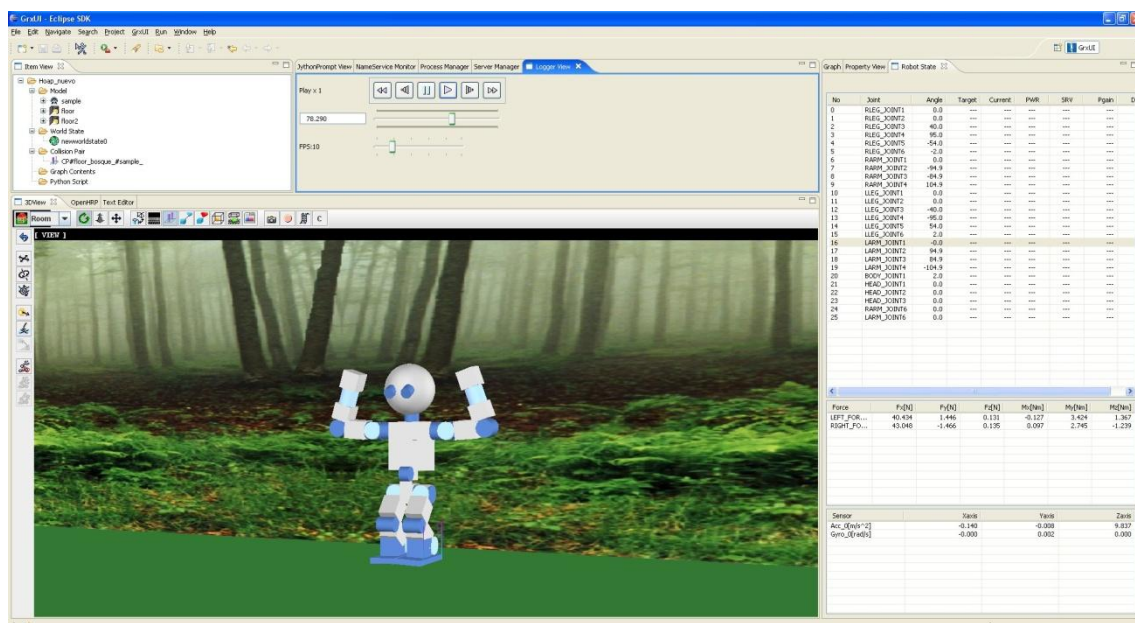


Figura 6.16. Brazos arriba en la trayectoria "Gimnasia"

En la gráfica de la Figura 6.17 se puede comprobar como las piernas se flexionan tres veces, si se observa el movimiento de las rodillas (en color amarillo), cuya posición varía entre  $46^\circ$  y  $95^\circ$ . Como en las trayectorias anteriores, el movimiento en ambas piernas es idéntico. Los movimientos de los dos brazos también son simétricos. Si bien se puede observar que las gráficas tienen sentido contrario, es debido a los criterios de signos definidos en la Tabla 2.4.

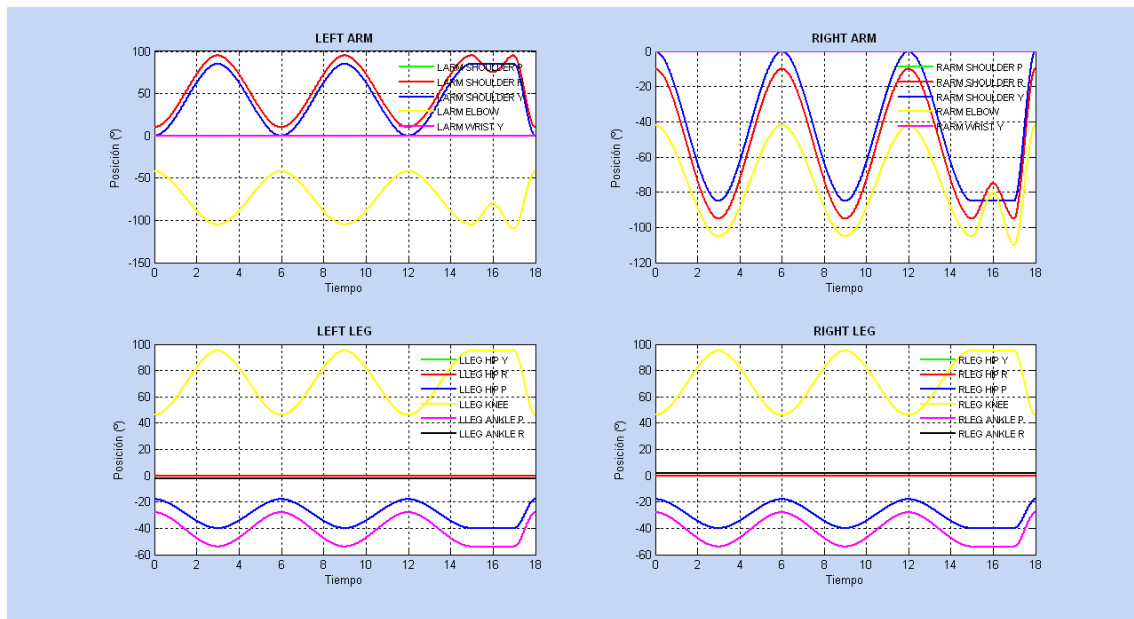


Figura 6.17. Gráficas de posición en la trayectoria "Gimnasia"

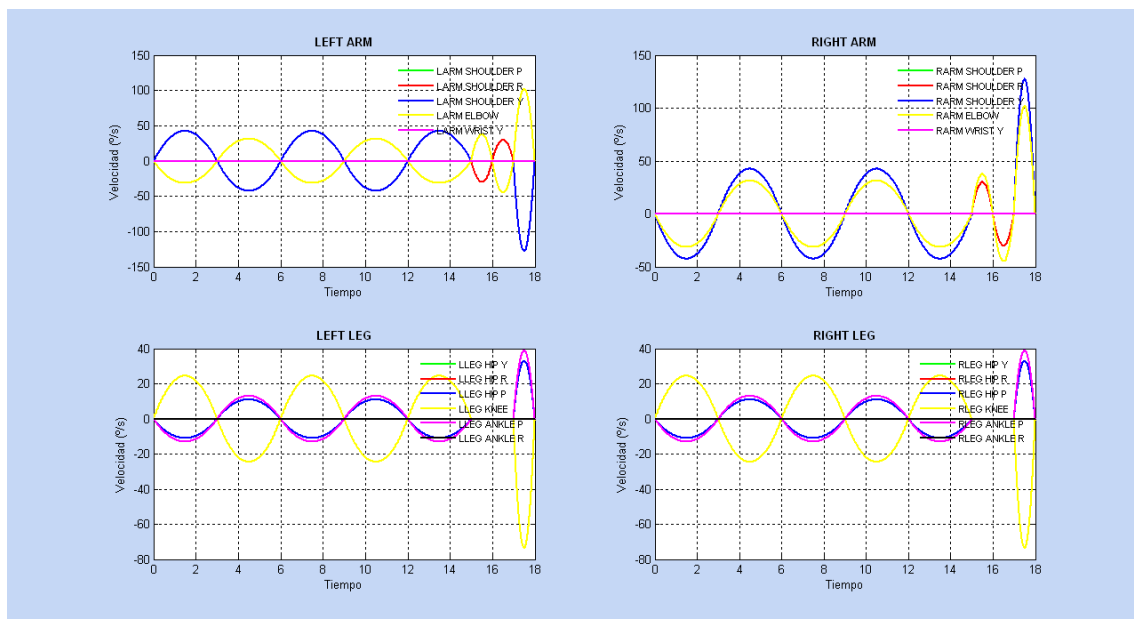


Figura 6.18. Gráficas de velocidad en la trayectoria "Gimnasia"

En el último segundo, se aprecian velocidades y aceleraciones mayores que en el resto de la trayectoria. Esto es debido a que el robot vuelve a la posición inicial en un tiempo de un segundo, y por ello algunas articulaciones tienen que hacer un movimiento igual que el que han realizado en instantes anteriores, pero en un tiempo menor.

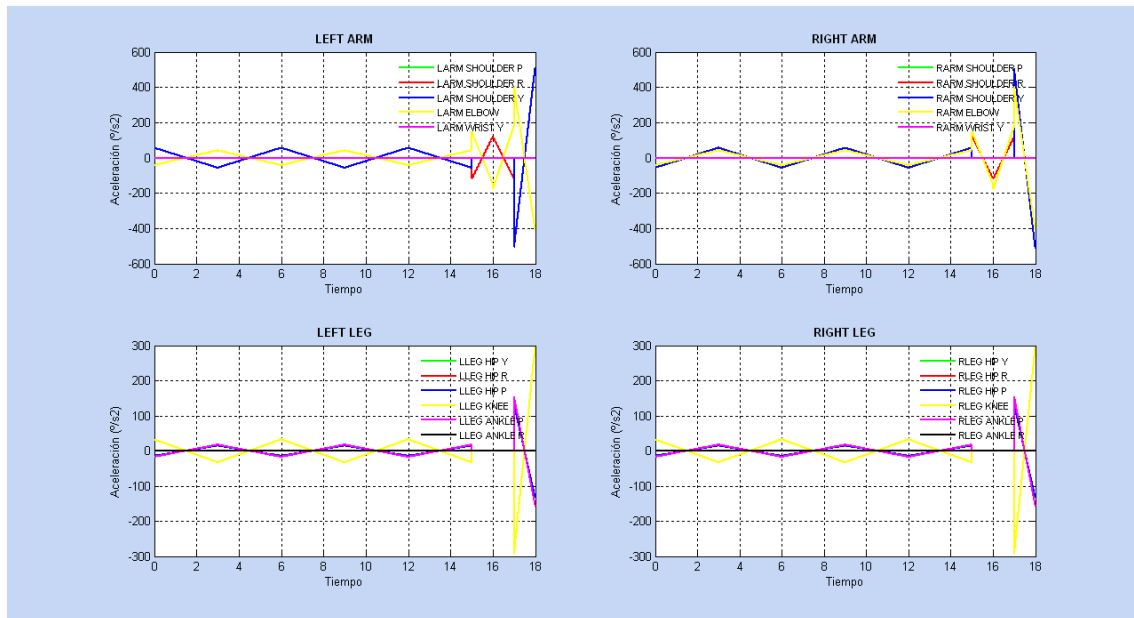


Figura 6.19. Gráficas de aceleración en la trayectoria “Gimnasia”

## 6.6 Trayectoria 5: “Flexión brazos”

En esta trayectoria, el robot permanece ligeramente agachado todo el tiempo, sin mover las piernas. Al comenzar la trayectoria levanta ambos brazos por encima de los hombros, y comienza una secuencia de movimientos alternando la flexión del codo de un brazo y el contrario. Después eleva los dos brazos y los flexiona dos veces a la vez que los dirige hacia delante. Por último, con los brazos colocados hacia delante los vuelve a flexionar y a estirar.

En la Figura 6.20 y la Figura 6.21 se muestra la primera parte de la trayectoria en que se estiran y flexionan los brazos de manera alternada.

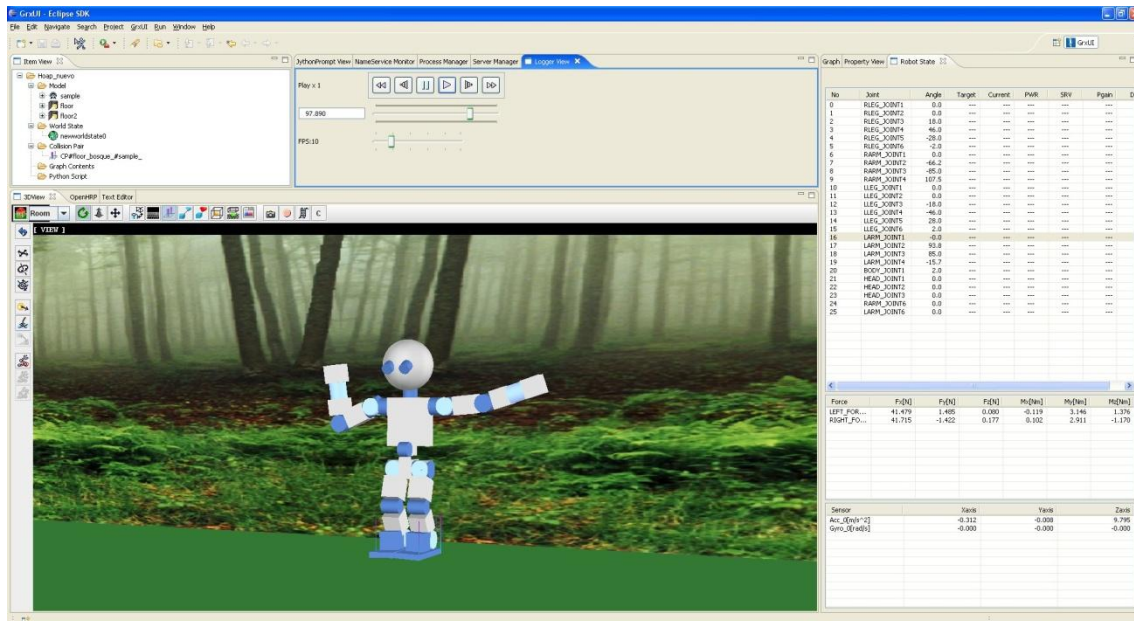


Figura 6.20. Brazo izquierdo extendido en la trayectoria "Flexión brazos"

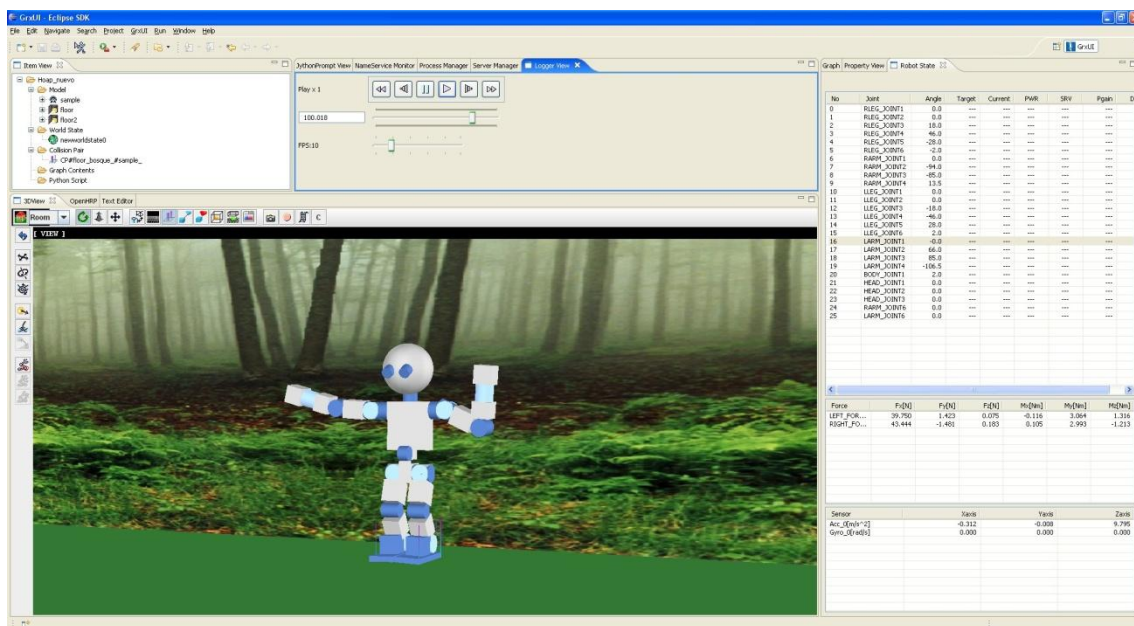


Figura 6.21. Brazo derecho extendido en la trayectoria "Flexión brazos"

En la Figura 6.22 se muestra un instante de la parte de la trayectoria en que se flexionan y estiran ambos brazos a la vez.

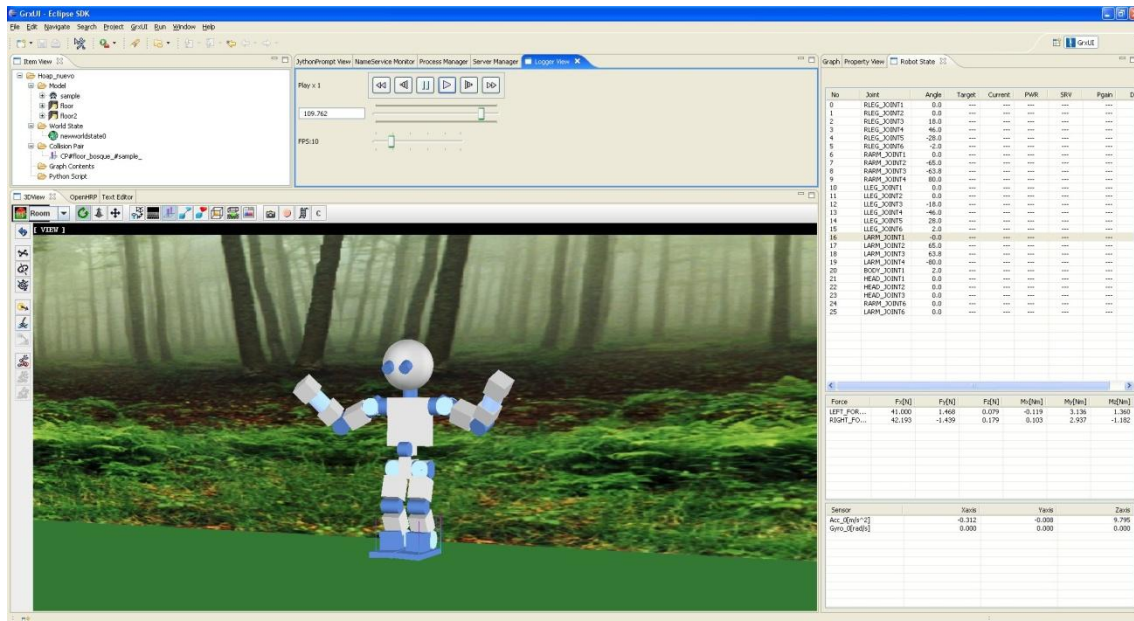


Figura 6.22. Brazos extendidos en la trayectoria "Flexión brazos"

A continuación se muestran las gráficas de posición, velocidad y aceleración de los brazos. No se aportan gráficas de las piernas ni de la cabeza porque sus articulaciones no se mueven, como se comentó antes.

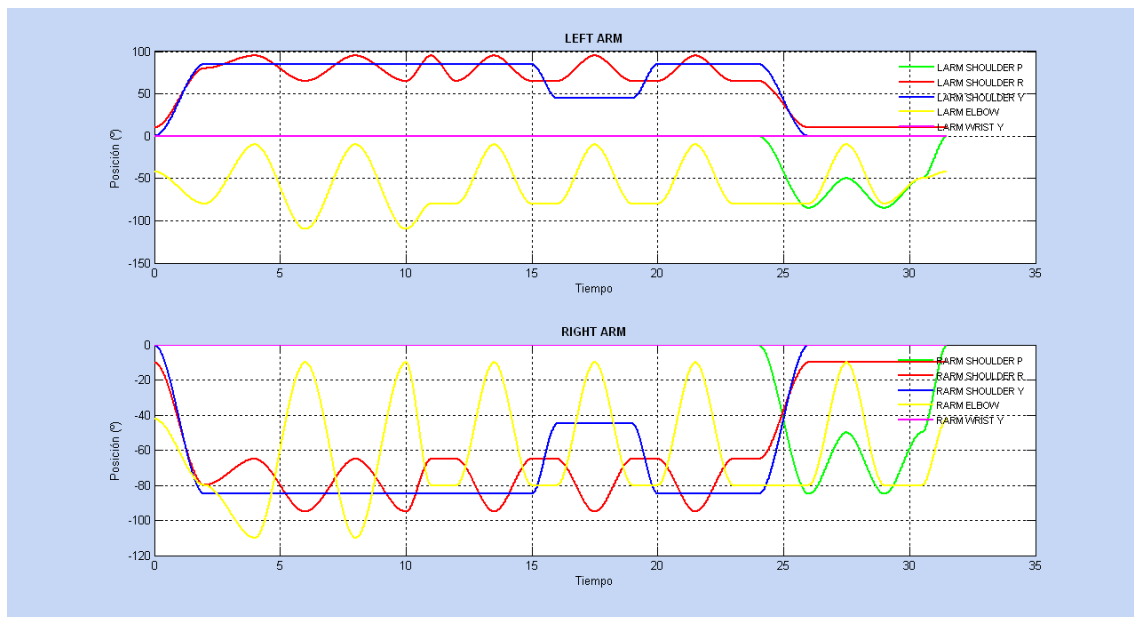


Figura 6.23. Gráficas de posición para los brazos en la trayectoria "Flexión brazos"



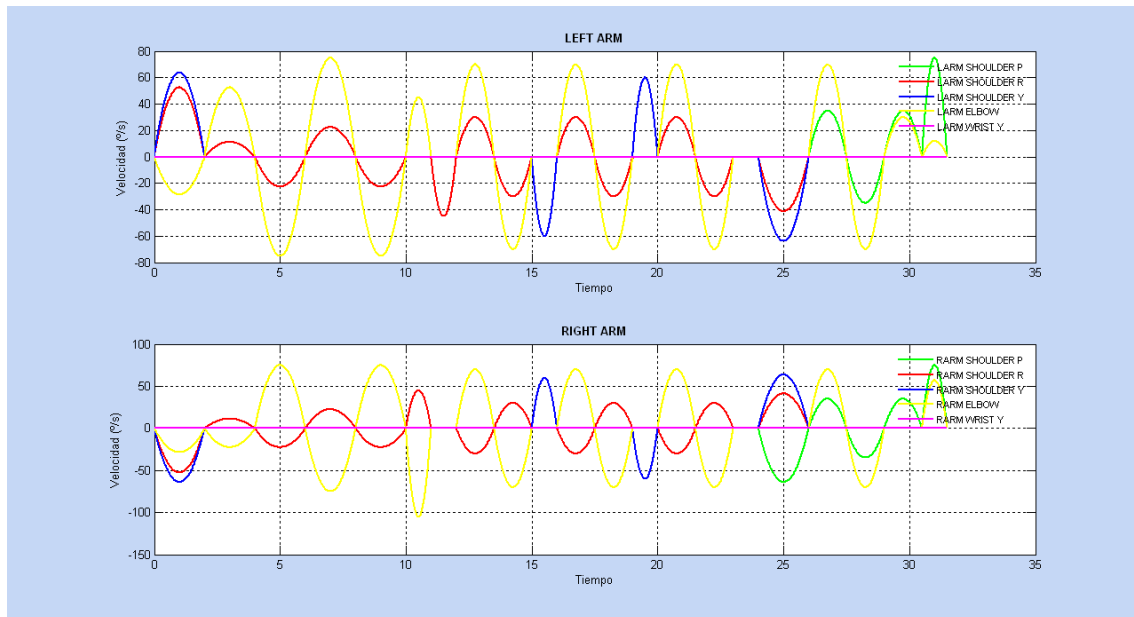


Figura 6.24. Gráficas de velocidad para los brazos en la trayectoria “Flexión brazos”

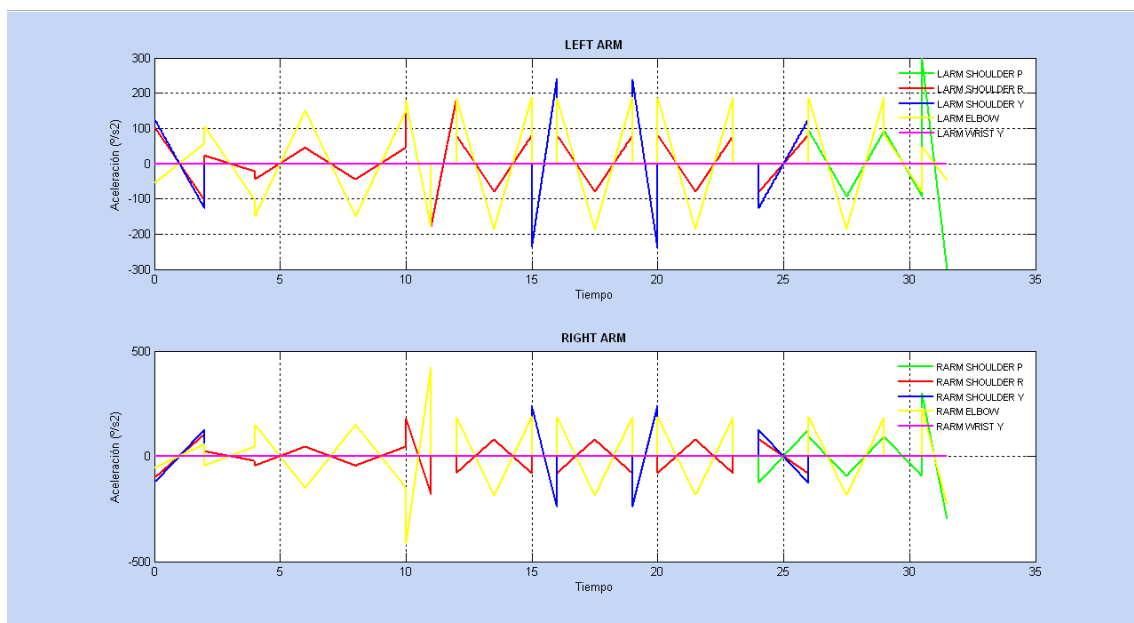


Figura 6.25. Gráficas de aceleración para los brazos en la trayectoria “Flexión brazos”

## 6.7 Trayectoria 6: “Baile”

A partir de las cinco trayectorias descritas anteriormente, se crea la trayectoria “Baile”. Para ello se une la siguiente secuencia de trayectorias: trayectoria 1, trayectoria 4, trayectoria 2, trayectoria 3, trayectoria 4 y trayectoria 5. A continuación se muestran las posiciones más representativas de la trayectoria (Figura 6.26 y Figura 6.27).



Figura 6.26. Secuencia de movimientos en la trayectoria "Baile" (I)



**Figura 6.27. Secuencia de movimientos en la trayectoria "Baile" (II)**

En el CD adjunto, se aporta el vídeo de la simulación de esta trayectoria en OpenHRP3 y el video de la prueba en el HOAP-3. También se aporta un montaje en el que se pueden ver simultáneamente la prueba en el robot y la simulación en OpenHRP3.



## **7 Conclusiones**

Trabajando con la interfaz, se puede comprobar cómo la creación de trayectorias básicas por el método propuesto es relativamente sencilla e intuitiva, si bien un conocimiento más exhaustivo de la cinemática del robot permitiría elaborar trayectorias más complejas.

Al diseñar trayectorias por este procedimiento, hay que observar ciertas precauciones. Es importante evitar colisiones entre las articulaciones del robot, para ello hay que tener en cuenta el tamaño de las extremidades. Así mismo, en el caso del HOAP-3, cuyo modelo VRML no tiene en cuenta las dimensiones de la carcasa protectora, para evitar que los brazos choquen con dicha carcasa es importante dejar un poco de margen al hacer determinados movimientos.

También hay que procurar no definir movimientos excesivamente bruscos que podrían dañar los motores. Por ello, una vez creada la trayectoria, es conveniente observar las gráficas de la aceleración que soportan las articulaciones para cerciorarse de que los movimientos exigidos no comprometen las posibilidades del robot.

Si bien, diseñar trayectorias por este método es relativamente sencillo, hay que señalar que es muy difícil conseguir que el robot camine. Al perder el doble apoyo de las piernas, su estabilidad se ve muy comprometida y los movimientos a realizar para mantenerse estable no se desprenden del simple examen del robot. Para trayectorias más complejas que impliquen caminata del robot, o simplemente desplazamiento lateral de las caderas, habría que aplicar cinemática inversa y otros procedimientos de control más complejos, como control por ZMP.



Por tanto, las trayectorias creadas mediante esta interfaz se centrarán principalmente en las articulaciones superiores, empleando sólo las de las piernas para realizar flexiones y desplazamientos en el plano vertical que no comprometan tanto la estabilidad del robot.

## **8 Trabajos futuros**

Sobre la herramienta presentada cabría realizar ciertas mejoras que se enuncian a continuación:

- Modificación de la posición de partida:

En las trayectorias que estamos generando, el robot parte siempre de la posición de reposo definida. Esto es una medida de seguridad que tiene mucho sentido al crear trayectorias independientes. Pero cuando el objetivo final es crear una trayectoria mediante la unión de varias trayectorias independientes, sería interesante poder cargar una posición inicial deseada, y así poder hacer coincidir directamente la posición final de una trayectoria con la posición inicial de la siguiente. Esto facilitaría la unión de trayectorias sin tener que hacer pasar al robot por la posición de reposo en cada transición.

- Aumentar el número de movimientos por trayectoria:

Actualmente, el número de movimientos que se pueden definir para cada articulación en una trayectoria se limita a veinte. Podría ser útil aumentar este número de movimientos. Para ello, habría que plantearse otro método para recoger los valores introducidos por el usuario. Por el método actual, cuadros de texto editables para introducir los puntos, nos enfrentamos a un problema muy habitual en el diseño de *GUIs*: el espacio a la hora de distribuir controles por la pantalla.

- Integración de la cinemática inversa:

Como se ha comentado en las conclusiones de este trabajo, no es viable la creación por este método de trayectorias que exijan al robot caminar. Ésta es la gran limitación que tiene la herramienta. Por diversos motivos, siendo éste el de mayor peso, sería



muy interesante integrar el método de la cinemática inversa en esta interfaz. Se supliría así dicha limitación, pudiéndose emplear la cinemática inversa para estos casos. En cualquier caso, integrar la funcionalidad de la cinemática inversa significaría crear una herramienta más potente y versátil.





## **9 Anexo - Documento de ayuda**



## DOCUMENTO DE AYUDA

# Interfaz para la generación de trayectorias para las plataformas robóticas HOAP-3 y RH-2





# 1 Funcionamiento de la interfaz

## 1.1 Generación de trayectorias

PARÁMETROS DE LA TRAYECTORIA

Paso de la trayectoria (ms): ---

Unidades gráficas: RADIANES

Velocidad y aceleración: REAL

Terminar en posición inicial

Limpiar valores

Crear trayectoria

Guardar trayectoria creada

Figura 9.1. Parámetros de la trayectoria

- **Paso de la trayectoria**

El paso de la trayectoria marcará la frecuencia con que se envía una nueva instrucción al robot. Así mismo, este valor será el paso para la simulación en OpenHRP3. Puede tomar los siguientes valores: 1 ms, 2ms, 3ms, 4 ms y 5 ms.

La selección del paso lleva asociada ciertas limitaciones a la hora de elegir los intervalos de tiempo para los movimientos. Si se selecciona un paso de 3 ms, los intervalos de tiempo que se introduzcan deberán ser múltiplos de 0.75 segundos. Si el paso es distinto de 3ms, deberán ser múltiplos de 0.5 segundos. Al cambiar la selección del paso se mostrará un mensaje como el de la Figura 9.2.

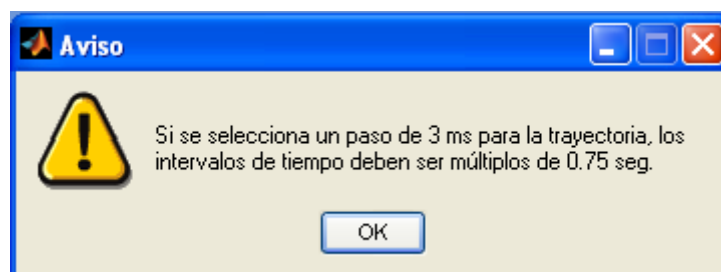


Figura 9.2. Selección de paso 3 ms

- **Unidades de las gráficas:**

Hay que seleccionar las unidades en las que se desea visualizar los valores que toman las articulaciones: radianes o grados. Es importante señalar que esto sólo afecta a la representación gráfica que se realiza a continuación para facilitar al usuario el diseño de la trayectoria, y en ningún caso afecta a las unidades en que se almacenan los datos en los archivos.

- **Velocidad y aceleración:**

En algunos casos especiales, en que no se mueven las articulaciones de las piernas, pueden realizarse las simulaciones en OpenHRP3 con velocidad y aceleración nulas, dando sólo los valores de posición por medio del archivo *angle.dat* y cargando los archivos *vel.dat* y *acc.dat* con primera columna (paso de simulación) idéntica y el resto de columnas a cero. Para ello, se ofrece la posibilidad al crear la trayectoria de generar estos archivos automáticamente. Como se puede ver en la Figura 9.1, se puede seleccionar “NULO” en el menú desplegable “Velocidad y aceleración”. Por defecto, este menú presenta la selección “REAL”, que permite obtener los valores de velocidad y aceleración reales para la trayectoria.

- **Terminar en posición inicial:**

Es posible imponer que la trayectoria termine con todas las articulaciones en la posición inicial. En caso de haber seleccionado un paso de 3 ms, el robot empleará 1.5 segundos en volver a la posición inicial. Si el paso es cualquier otro valor, será 1 segundo el tiempo para volver a la posición inicial.

- **Definir movimientos:**

Para definir el movimiento que se desea que describa cada articulación, hay que hacer clic en los paneles desplegables de la esquina superior izquierda. (Ver Figura 9.3).

Según de qué robot se trate, los menús desplegables mostrarán la distribución de articulaciones propia de dicho robot. Si seleccionamos una parte del cuerpo del HOAP-3, por ejemplo *Right Leg*, se despliega un menú como el de la Figura 9.4.



Figura 9.3. Definir movimientos

>>> Definir movimientos RIGHT LEG																		
RLEG HIP Y	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG HIP R	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG HIP P	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	-18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG KNEE	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG ANKLE P	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	-20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)
RLEG ANKLE R	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Tiempo (s)
	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Ángulo(º)

Figura 9.4. Menú *Right Leg* del HOAP-3

Para cada articulación hay dos filas de cuadros de texto editables. La fila superior corresponde a valores de tiempo, y la inferior a valores de ángulos. Estos pares de valores tiempo-ángulo irán definiendo puntos de la trayectoria. A nivel interno, el programa hará una interpolación cúbica entre cada uno de los puntos indicados y el siguiente en la línea temporal. Al hacer esto para todos los puntos, se define una trayectoria completa.

El primer par de valores viene definido de antemano, e indica la posición inicial a la que llevamos al robot al comenzar cada trayectoria. En el caso del RH-2, esa posición es 0º para todas las articulaciones.

Para identificar los ángulos, se considera como posición de reposo del RH-2 la posición de la. Figura 9.46. Se debe seguir el criterio de signos de la misma figura. Para el HOAP-3, se considera posición de reposo la de la Figura 9.42, pero

considerando los brazos extendidos en paralelo al tronco. El criterio de signos para el HOAP-3 será el de la misma figura.

Al introducirse un valor cualquiera de tiempo o de posición, se comprobará si se ha introducido previamente un valor para el paso. Si no se ha indicado el valor del paso, se mostrará el mensaje de error de la Figura 9.5.

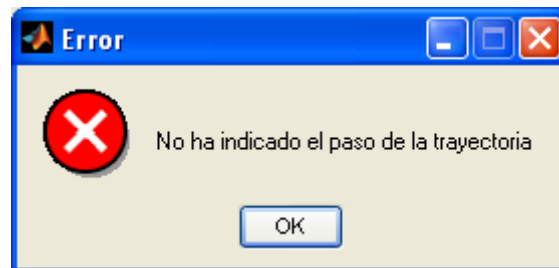


Figura 9.5. Mensaje de paso no indicado

Los **valores de posición** que se pueden introducir deben pertenecer al rango descrito en la Tabla 9.2 (para el HOAP-3) y en la Tabla 9.5 (para el RH-2). Hay que respetar también un margen de  $1^\circ$  por encima del mínimo y  $1^\circ$  por debajo del máximo.

Al introducir un valor de posición se comprobará que éste está dentro del rango permitido. Si no lo está se mostrará un mensaje de error como el de la Figura 9.6, en el que se indica el rango permitido para la articulación.

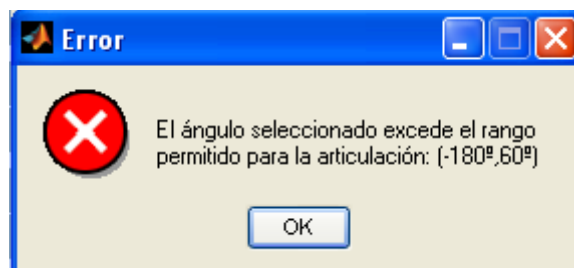


Figura 9.6. Mensaje de ángulo fuera de rango

Al introducir un **valor de tiempo**, si el valor del intervalo de tiempo que se está definiendo no es compatible con éste valor, se mostrará un mensaje de error similar al de la Figura 9.7.



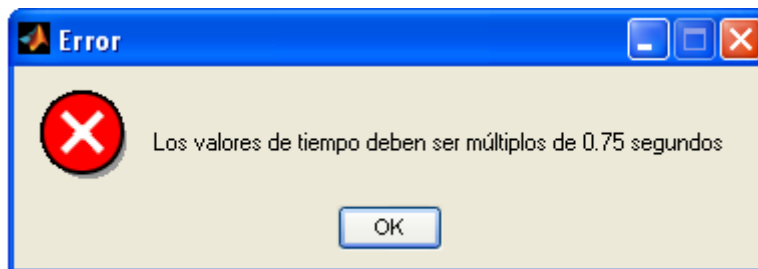


Figura 9.7. Mensaje de valor de tiempo erróneo para paso 3 ms

Para cada articulación se puede definir el número de pares de puntos que se desee, no teniendo que ser igual para todas. El tiempo total de la trayectoria vendrá dado por la articulación que presente el par ángulo-tiempo con un valor de tiempo mayor. Para las articulaciones que tengan definidos ángulos hasta un tiempo menor, el programa tomará el último par de puntos de la articulación y mantendrá el valor del ángulo constante hasta dicho tiempo total de la trayectoria.

Después, si se ha seleccionado “Terminar en posición inicial”, todas las articulaciones volverán a la posición inicial en un intervalo de 1.5 segundos si el paso seleccionado es 3 ms, o 1 segundo si el paso es distinto de 3 ms.

Las articulaciones para las que no se indique ningún par tiempo-ángulo, permanecerán toda la trayectoria en la posición inicial.

- **Limpiar valores:**

Si durante este proceso se desea eliminar todos los valores introducidos y comenzar de nuevo, es posible hacerlo haciendo clic en el botón “Limpiar valores” (ver Figura 4.3). Como confirmación se mostrará el mensaje de la Figura 9.8

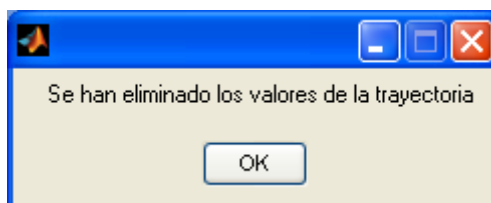


Figura 9.8. Mensaje de valores eliminados

- **Crear trayectoria:**

Una vez definidos los valores tiempo-ángulo que se deseen y los parámetros de la trayectoria, se puede hacer clic en el botón “Crear trayectoria”. En caso de no haber introducido ningún valor de tiempo y posición, se mostrará el mensaje de la Figura 9.9.

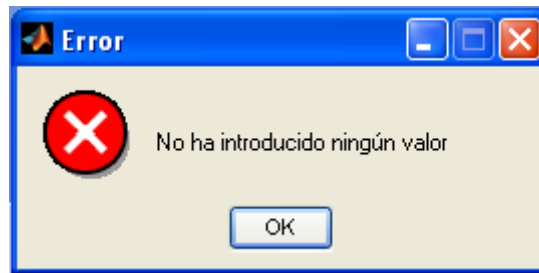


Figura 9.9. Mensaje de falta de valores

Si no se ha seleccionado un paso para la trayectoria, se mostrará el mensaje de error correspondiente (Figura 9.5). Si todo es correcto, se mostrará un mensaje que nos indicará el tiempo final de la trayectoria (Figura 9.10). Ya se podrá observar la representación gráfica para cada articulación en el panel situado debajo de la zona de selección de parámetros.

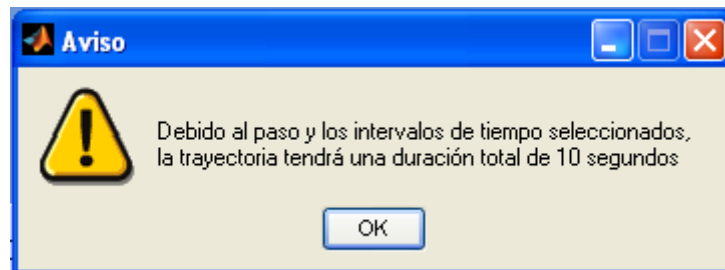
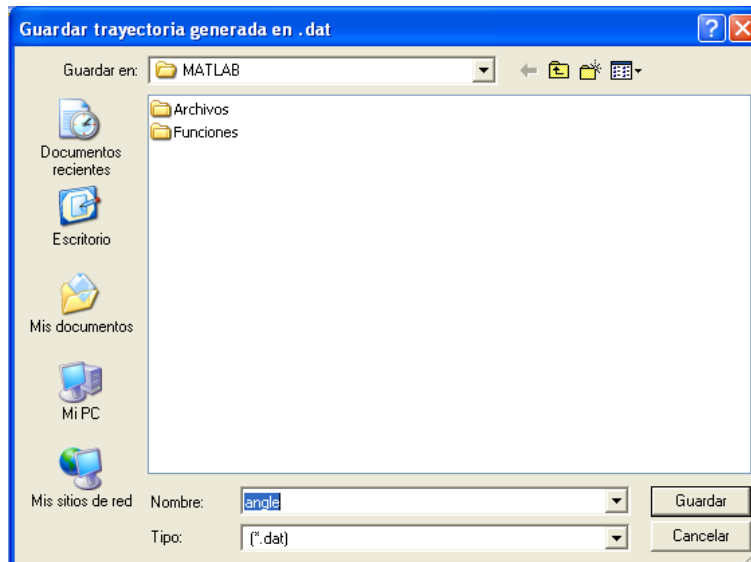


Figura 9.10. Mensaje con duración de la trayectoria

- **Guardar trayectoria creada:**

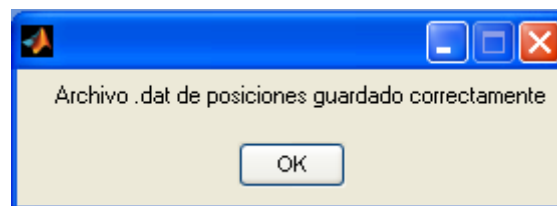
Como podemos observar en la Figura 9.1, el botón “Guardar trayectoria creada” permanece deshabilitado hasta que se ha creado la trayectoria.

Al hacer clic en “Guardar trayectoria creada”, se abrirá un cuadro de diálogo como el de la Figura 9.11 para guardar el archivo en extensión *.dat*. Éste cuadro solicitará un nombre para dicho archivo, y sugerirá el último directorio al que ha accedido el programa.



**Figura 9.11. Cuadro de diálogo para guardar archivo con extensión .dat**

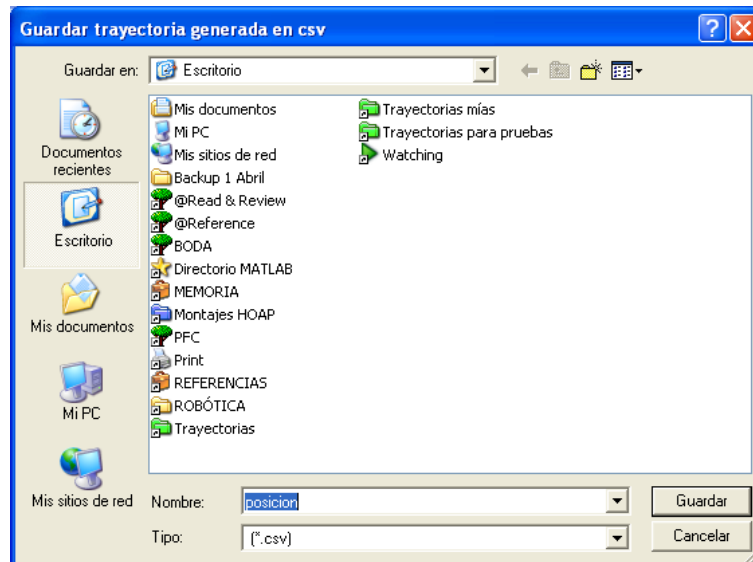
Tras indicar el nombre y el lugar donde se desea guardar el archivo, si todo es correcto aparecerá un mensaje de confirmación como el siguiente:



**Figura 9.12. Mensaje de archivo con extensión .dat guardado**

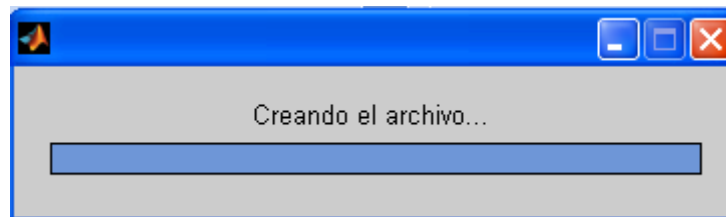
A continuación se guardan, siguiendo el mismo proceso, los archivos *vel.dat* y *acc.dat*, correspondientes a velocidad y aceleración respectivamente. Las ventanas de diálogo mostrarán siempre la última ruta que se haya utilizado.

Si se trata de una trayectoria para el robot HOAP-3, a continuación se abre un cuadro de diálogo como el de la Figura 9.13 para guardar el archivo con extensión *csv*.



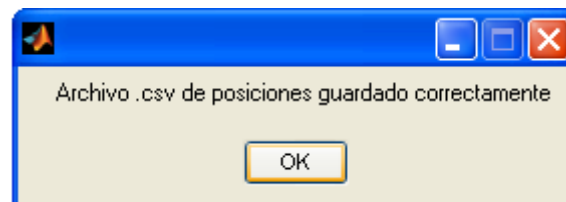
**Figura 9.13. Cuadro de diálogo para guardar archivo con extensión .csv**

Tras indicar el nombre y la ruta para el archivo, una barra indica el progreso de creación del archivo, como se puede ver en la Figura 9.14.



**Figura 9.14. Barra de espera guardar archivo con extensión .csv**

Si todo es correcto, se muestra el mensaje de la Figura 9.15.



**Figura 9.15. Mensaje de archivo con extensión .csv guardado**

Tanto para el HOAP-3 como para el RH-2, al crear la trayectoria se genera un archivo con extensión .dat que contiene los puntos que se han indicado al definir la trayectoria. Para cada articulación hay dos columnas, una para los valores de tiempo y otra para los valores de posición. (Ver Figura 9.16).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	RLEG HIP Y	RLEG HIP Y	RLEG HIP R	RLEG HIP R	RLEG HIP P	RLEG HIP P	RLEG KNEE	RLEG KNEE	RLEG ANKLE P	RLEG ANKLE P	RLEG ANKLE R	RLEG ANKLE R	RARM SHOULDER P	RARM S
2	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición
3	0	0	0	0	0	-18	0	46	0	-28	0	2	0	
4	-	-	7	0	7	-18	7	46	7	-28	7	2	2	
5	-	-	9	2	9	-40	9	95	9	-54	9	2	-	
6	-	-	-	-	-	-	-	-	-	-	-	-	-	
7	-	-	-	-	-	-	-	-	-	-	-	-	-	
8	-	-	-	-	-	-	-	-	-	-	-	-	-	
9	-	-	-	-	-	-	-	-	-	-	-	-	-	
10	-	-	-	-	-	-	-	-	-	-	-	-	-	
11	-	-	-	-	-	-	-	-	-	-	-	-	-	
12	-	-	-	-	-	-	-	-	-	-	-	-	-	
13	-	-	-	-	-	-	-	-	-	-	-	-	-	

Figura 9.16. Archivo de puntos de la trayectoria

Una vez guardados los archivos, se deshabilita el botón “Guardar trayectoria creada” hasta que se vuelva a hacer clic en “Crear trayectoria”.

## 1.2 Concatenación de trayectorias

CONCATENAR TRAYECTORIAS

Número de trayectorias:  Paso de la trayectoria (ms):

Tiempo de unión(s):  Velocidad y aceleración:

Formato trayectorias a concatenar:  Unidades:

Trayectoria 1     Trayectoria 4

Trayectoria 2     Trayectoria 5

Trayectoria 3     Trayectoria 6

Figura 9.17. Parámetros para la concatenación de trayectorias para el HOAP-3

- **Paso de la trayectoria:**

El paso debe ser el mismo que el de las trayectorias que se están uniendo. Si se trata de cargar las trayectorias antes de indicar el paso, un mensaje de error avisa al usuario de que no ha indicado este valor, impidiéndole continuar hasta haberlo hecho.

- **Tiempo de unión:**

Hay que seleccionar también el tiempo que se invertirá en la conexión de las dos trayectorias. Si el paso seleccionado es 3 ms, los tiempos de unión que se podrán seleccionar serán: 1.5 segundos y 3 segundos. Si el paso es otro cualquiera, se podrá escoger entre 1 segundo, 1.5 segundos, 2 segundos, 2.5 segundos y 3 segundos. En el tiempo de unión, el programa hará una interpolación cúbica entre el último punto de la primera trayectoria a concatenar y el primer punto de la siguiente.

- **Unidades de las gráficas:**

Hay que seleccionar las unidades en las que se desea visualizar los valores que toman las articulaciones: radianes o grados. Es importante señalar que esto sólo afecta a la representación gráfica que se realiza a continuación para facilitar al usuario el diseño de la trayectoria, y en ningún caso afecta a las unidades en que se almacenan los datos en los archivos.

- **Velocidad y aceleración:**

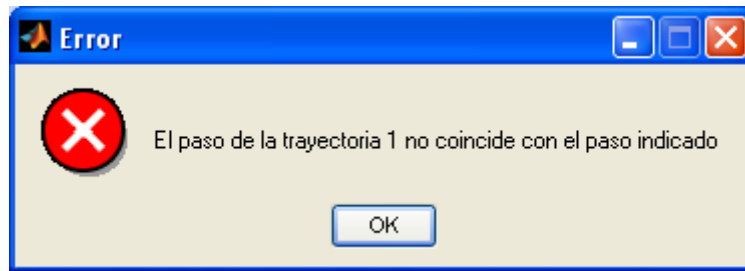
En algunos casos especiales, en que no se mueven las articulaciones de las piernas, pueden realizarse las simulaciones en OpenHRP3 con velocidad y aceleración nulas, dando sólo los valores de posición por medio del archivo *angle.dat* y cargando los archivos *vel.dat* y *acc.dat* con primera columna (paso de simulación) idéntica y el resto de columnas a cero. Para ello, se ofrece la posibilidad al crear la trayectoria de generar estos archivos automáticamente. Como se puede ver en la Figura 9.17, se puede seleccionar “NULO” en el menú desplegable “Velocidad y aceleración”. Por defecto, este menú presenta la selección “REAL”, que permite obtener los valores de velocidad y aceleración reales para la trayectoria.

- **Trayectorias a concatenar:**

En el cuadro inferior, podemos observar seis botones destinados a cargar las trayectorias que se desea concatenar. Inicialmente, estos seis botones permanecen deshabilitados hasta que se indica cuantas van a ser cargadas. En el caso del HOAP-3, hay que indicar qué formato tienen las trayectorias (*.dat* o *.csv*). En el caso del RH-2, siempre será en *.dat* y por tanto no es necesario indicarlo.

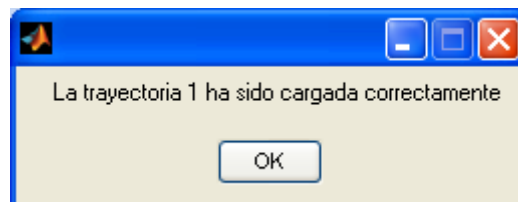
En el ejemplo de la Figura 9.17 se ha elegido concatenar tres trayectorias, y por ello aparecen tan sólo tres botones habilitados. Si en cualquier momento del proceso se cambiara el número de trayectorias, se habilitarían los botones correspondientes al nuevo número de trayectorias indicado, eliminándose cualquier trayectoria cargada previamente de la memoria del programa

Al cargar la trayectoria, se comprueba si el paso coincide con el de la trayectoria que se está creando. Si no coincide, se muestra el mensaje de error de la Figura 9.18 y no se carga la trayectoria.



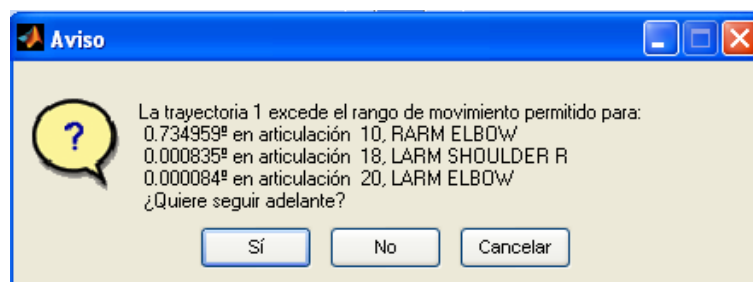
**Figura 9.18. Mensaje de error al cargar trayectoria**

Si el archivo tiene el formato adecuado y el paso de la trayectoria corresponde con el indicado, se comprueba si todas las articulaciones se mueven dentro del rango de valores angulares permitido. Si ninguna articulación excede el rango de valores permitido, se muestra el mensaje de la Figura 9.19.



**Figura 9.19. Mensaje de trayectoria cargada correctamente**

Si alguna articulación excede este rango, se muestra un mensaje como el de la Figura 9.20 en el que se detalla qué articulaciones exceden dicho rango y en qué medida.



**Figura 9.20. Mensaje que muestra articulaciones fuera de rango permitido**

Este mensaje pide confirmación para seguir adelante cargando la trayectoria, o cancelar el proceso de carga. Si se acepta se cargará la trayectoria y se mostrará un mensaje como el de la Figura 9.19.

Una vez cargadas tantas trayectorias como se indicara en el primer paso, (y como botones tenemos habilitados), el botón “Crear trayectoria” se habilitará.

- **Crear trayectoria:**

Si se hace clic en este botón, se crea la trayectoria y se puede observar la representación gráfica en el panel inferior. Mediante la barra vertical que aparece a la derecha, se puede ir desplazando dicho panel para ver las gráficas correspondientes a las distintas articulaciones. Queda habilitado el botón “Guardar trayectoria creada”, situado a la derecha de la pantalla.

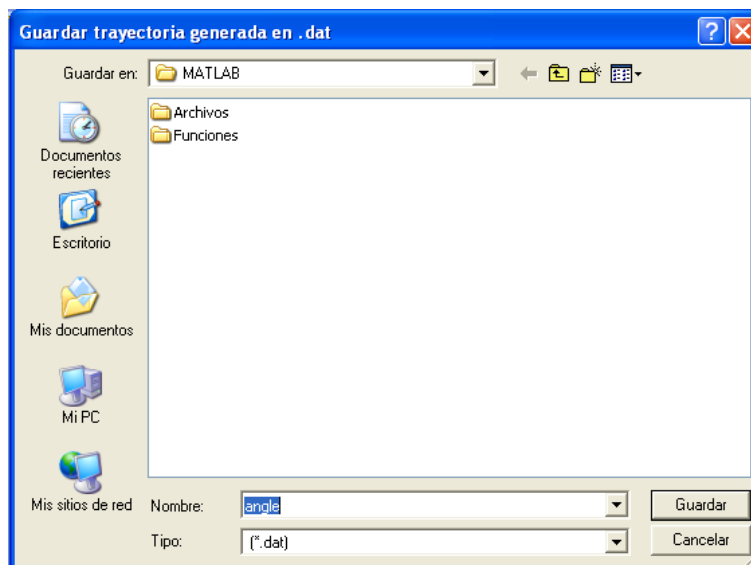
La selección del paso y los botones para cargar trayectorias se deshabilitarán para evitar que se modifique algún parámetro de la trayectoria creada antes de que sea guardada y evitar así posibles errores. Éstos permanecerán deshabilitados hasta que se vuelva a seleccionar un nuevo número de trayectorias, lo que provocaría la eliminación de las trayectorias cargadas obligando a comenzar de nuevo el proceso.

Queda habilitada la selección de velocidad y aceleración real o nula, y la selección de las unidades. Es posible modificar estos parámetros y volver a hacer clic en “Crear trayectoria”.

- **Guardar trayectoria creada:**

Como podemos observar en la Figura 9.17, el botón “Guardar trayectoria creada” permanece deshabilitado hasta que se ha creado la trayectoria.

Al hacer clic en “Guardar trayectoria creada”, se abrirá un cuadro de diálogo como el de la Figura 9.21 para guardar el archivo en extensión *.dat*. Éste cuadro solicitará un nombre para dicho archivo, y sugerirá el último directorio al que ha accedido el programa.



**Figura 9.21. Cuadro de diálogo para guardar archivo con extensión *.dat***



Tras indicar el nombre y el lugar donde se desea guardar el archivo, si todo es correcto aparecerá un mensaje de confirmación como el siguiente:

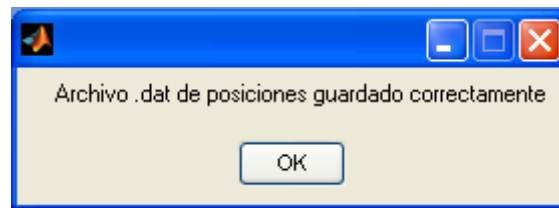


Figura 9.22. Mensaje de archivo con extensión .dat guardado

A continuación se guardan, siguiendo el mismo proceso, los archivos *vel.dat* y *acc.dat*, correspondientes a velocidad y aceleración respectivamente. Las ventanas de diálogo mostrarán siempre la última ruta que se haya utilizado.

Si se trata de una trayectoria para el robot HOAP-3, a continuación se abre un cuadro de diálogo como el de la Figura 9.23.

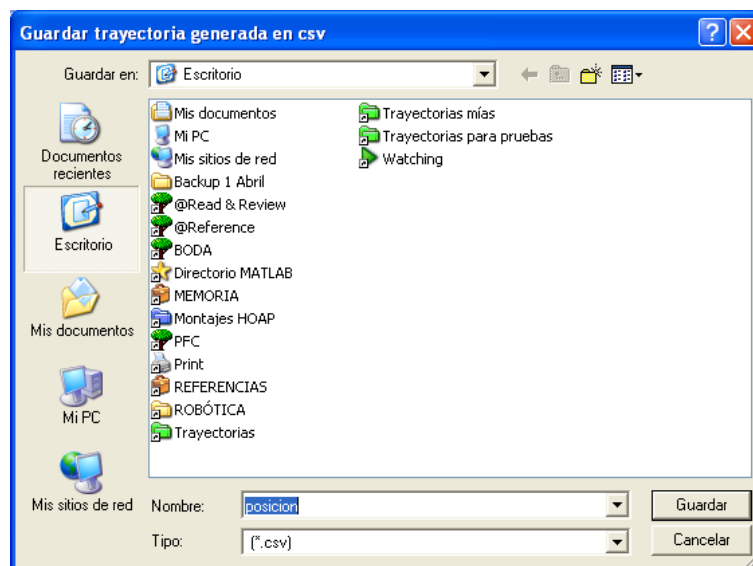


Figura 9.23. Cuadro de diálogo para guardar archivo con extensión .csv

Una barra indica el progreso de creación del archivo (Ver Figura 9.24).

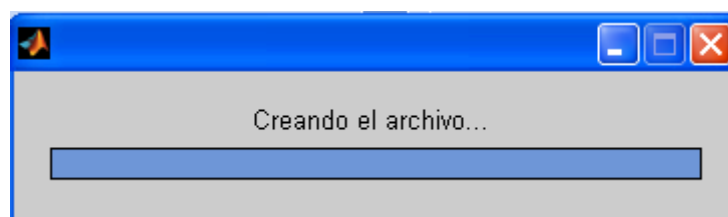


Figura 9.24. Barra de espera para guardar archivo con extensión .csv

Si todo es correcto, se muestra el mensaje de la Figura 9.25.

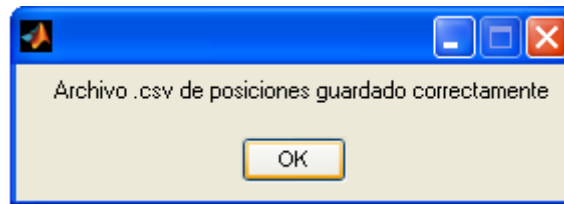


Figura 9.25. Mensaje de archivo con extensión .csv guardado

Una vez guardados los archivos, se vuelve a deshabilitar el botón “Guardar trayectoria creada”, para evitar posibles errores como guardar una trayectoria con unos parámetros almacenados en memoria tras haber modificado los parámetros en el panel de selección. Permanecerá deshabilitado hasta que se vuelva a hacer clic en “Crear trayectoria”.

### 1.3 Conversión de trayectorias

- **Parámetros de la conversión:**

En las pantallas “Convertir trayectorias RH2” y “Convertir trayectorias HOAP-3”, es posible obtener a partir del archivo de posiciones los archivos de velocidad y aceleración que serán necesarios para realizar las simulaciones de las trayectorias.

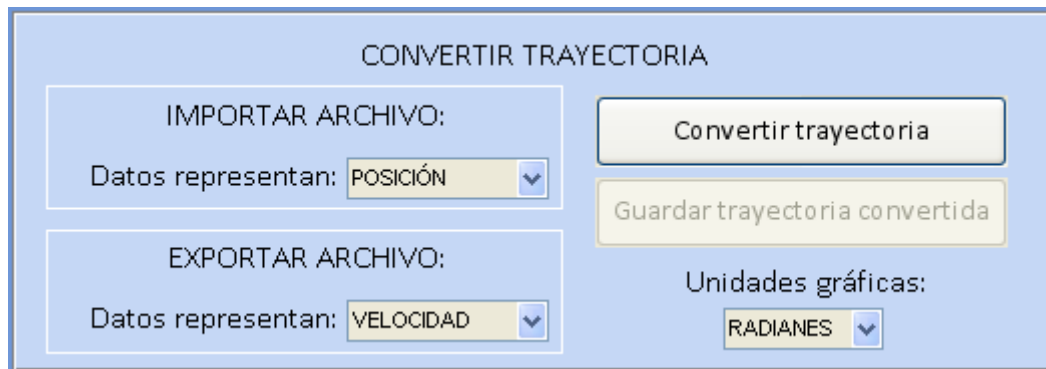
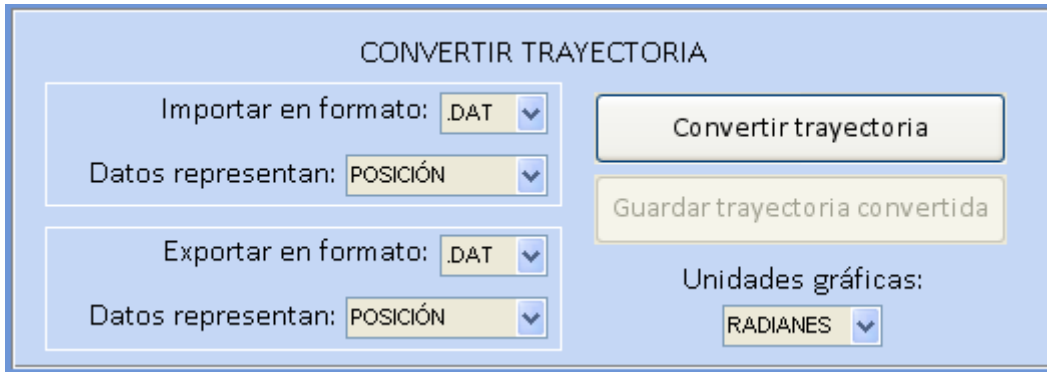


Figura 9.26. Parámetros para convertir trayectoria del RH-2

En el caso del HOAP-3, también es posible convertir los archivos de posiciones entre el formato *csv* empleado por el robot y el formato *dat* empleado por el simulador. (Ver Figura 9.27).



CONVERTIR TRAYECTORIA

Importar en formato: .DAT

Datos representan: POSICIÓN

Exportar en formato: .DAT

Datos representan: POSICIÓN

Unidades gráficas: RADIANES

Convertir trayectoria

Guardar trayectoria convertida

Figura 9.27. Parámetros para convertir trayectoria del HOAP-3

Para obtener estos archivos, se hace la derivada de orden 1 o de orden 2 de los datos, según corresponda.

- **Unidades de las gráficas:**

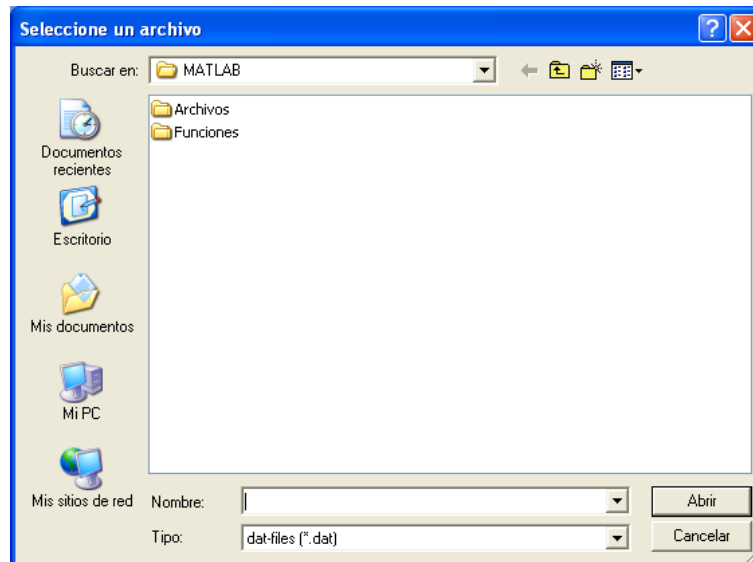
Hay que seleccionar las unidades en las que se desea visualizar los valores que toman las articulaciones: radianes o grados. Es importante señalar que esto sólo afecta a la representación gráfica que se realiza a continuación para facilitar al usuario el diseño de la trayectoria, y en ningún caso afecta a las unidades en que se almacenan los datos en los archivos.

- **Importar en formato / Exportar en formato:**

En el caso del HOAP-3 se incluyen estos controles que permiten realizar la transformación entre el formato de archivo usado por el robot y el del simulador. Puede convertirse de *dat* a *csv* y de *csv* a *dat*.

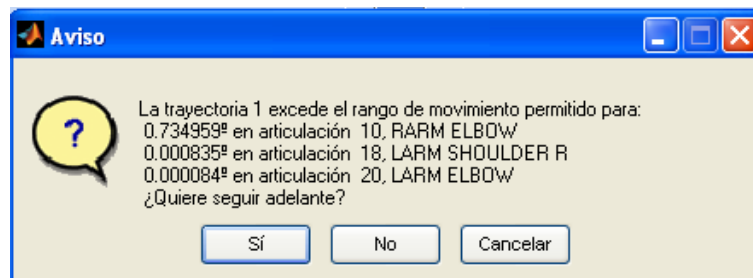
- **Convertir trayectoria:**

Al hacer clic en el botón “Convertir trayectoria”, se abre un cuadro de diálogo como el de la Figura 9.28 que nos permite seleccionar el archivo de trayectorias que queremos convertir.



**Figura 9.28. Cuadro de diálogo para seleccionar archivo**

Al cargar la trayectoria se comprueba si cumple el rango de movimiento permitido para las articulaciones. Si alguna articulación excede este rango, se muestra un mensaje como el de la Figura 9.29 en el que se detalla qué articulaciones exceden dicho rango y en qué medida. Este mensaje pide confirmación para seguir adelante cargando la trayectoria, o cancelar el proceso.

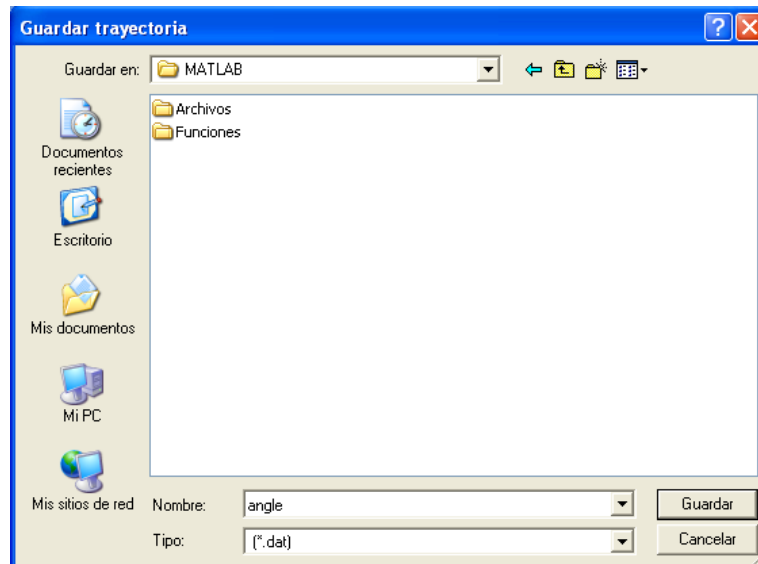


**Figura 9.29. Mensaje que muestra articulaciones fuera de rango permitido**

Una vez cargada la trayectoria, aparece en el panel inferior la representación gráfica de la conversión realizada y se habilita el botón “Guardar trayectoria convertida”.

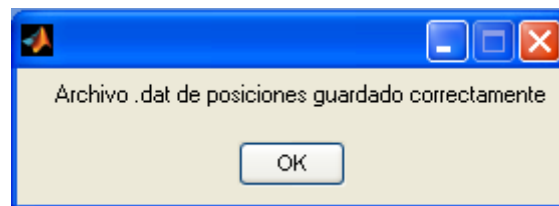
- **Guardar trayectoria convertida:**

Al hacer clic en el botón “Guardar trayectoria convertida”, se abre un cuadro de diálogo como el de la Figura 9.30 para guardar el archivo en extensión *dat*. Éste muestra la última ruta a la que accedió el programa y solicita un nombre para el archivo.



**Figura 9.30. Cuadro de diálogo para guardar archivo .dat**

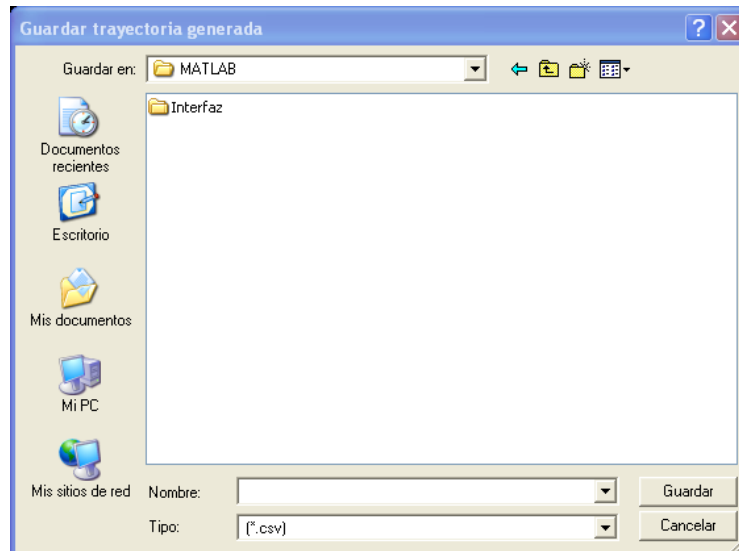
Tras indicar el nombre y el lugar donde se desea guardar el archivo, si todo es correcto aparece un mensaje de confirmación como el siguiente:



**Figura 9.31. Mensaje de archivo con extensión .dat guardado**

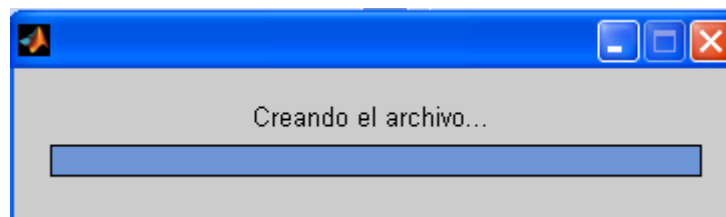
A continuación, se guardan los archivos *vel.dat* y *acc.dat*, correspondientes a velocidad y aceleración respectivamente. La ventana de diálogo muestra la ruta en que se acaba de guardar el archivo *angle.dat*. De manera análoga, se muestran mensajes de confirmación al guardar los archivos.

Si se trata del robot HOAP-3, a continuación se abre un cuadro de diálogo como el de la Figura 9.32, para guardar el archivo con extensión *csv*.

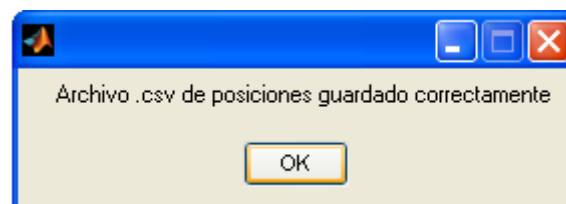


**Figura 9.32. Cuadro de diálogo para guardar archivo con extensión .csv**

Tras indicar nombre y ruta para el archivo, una barra indica el progreso de creación del mismo (Figura 9.33). Al terminar, se muestra el mensaje de la Figura 9.34.



**Figura 9.33. Barra de espera para guardar archivo con extensión .csv**



**Figura 9.34. Mensaje de archivo con extensión .csv guardado**

Una vez guardados los archivos, se deshabilita el botón “Guardar trayectoria convertida” para evitar posibles errores como guardar una trayectoria con unos parámetros almacenados en memoria tras haber modificado la selección. Si se desea volver a guardar el archivo, es necesario volver a hacer clic en el botón “Crear trayectoria”.

## 1.4 Simulación de trayectorias

### 1.4.1 Selección de parámetros:

Para representar gráficamente los movimientos de las articulaciones, hay que indicar qué representan estos datos: posición, velocidad o aceleración. Hay que indicar también las unidades en que se desea ver las gráficas: grados o radianes.

SIMULAR TRAYECTORIA

Los datos representan: POSICIÓN

Unidades gráficas: RADIANES

Simular trayectoria

Figura 9.35. Parámetros para simulación del RH-2

En el caso del HOAP-3 (Ver Figura 9.36) hay que indicar también en qué formato se va a cargar la trayectoria: *dat* o *csv*.

SIMULAR TRAYECTORIA

Simular trayectorias en archivo: .DAT

Los datos representan: POSICIÓN

Unidades gráficas: RADIANES

Simular trayectoria

Figura 9.36. Parámetros para simulación del HOAP-3

### 1.4.2 Simular trayectoria:

Una vez seleccionados los parámetros, se procede a cargar el archivo. Al cargar la trayectoria se comprueba si cumple el rango de movimiento permitido para las articulaciones. Si alguna articulación excede este rango, se muestra un mensaje como el de la Figura 9.37 en el que se detalla qué articulaciones exceden dicho rango y en qué medida. Este mensaje pide confirmación para seguir adelante cargando la trayectoria, o cancelar el proceso.

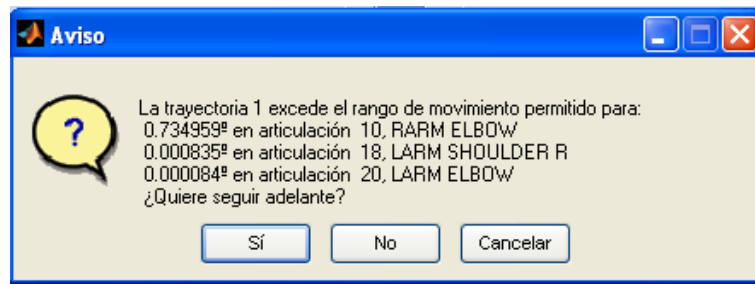


Figura 9.37. Mensaje que muestra articulaciones fuera de rango permitido

### 1.4.3 Representación gráfica:

En el panel inferior se puede observar la representación gráfica de los valores articulares durante la trayectoria. Es posible desplazar la barra vertical para ir mostrando las gráficas de los distintos grupos de articulaciones.

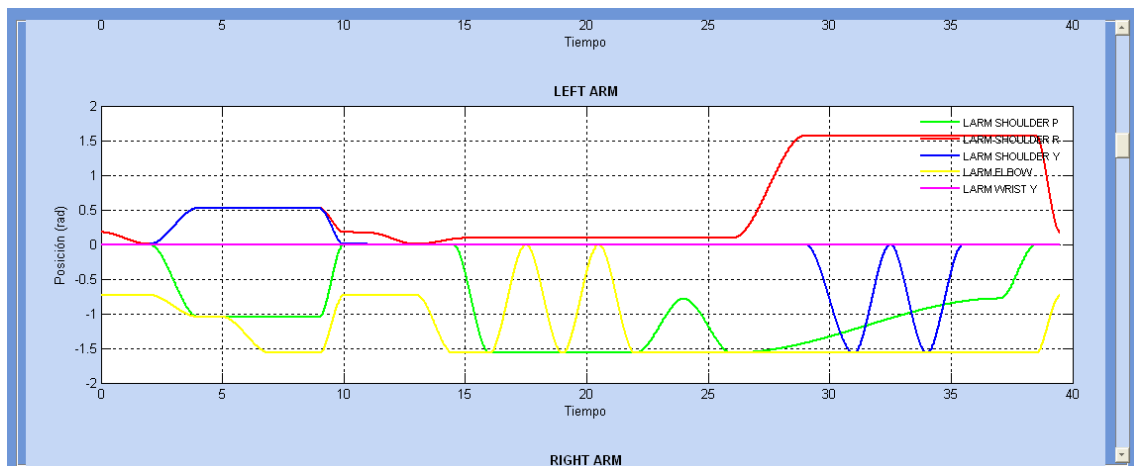


Figura 9.38. Detalle del panel de simulaciones

## 1.5 Guardar gráficas:

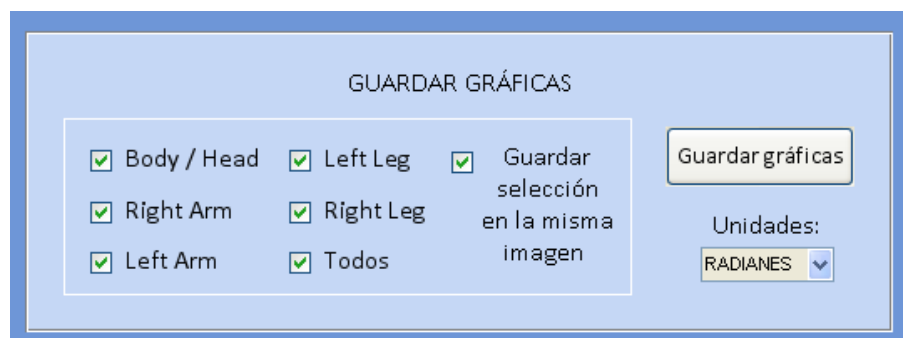


Figura 9.39. Panel "Guardar gráficas"



Para cada una de las funcionalidades de la interfaz, se ofrece la posibilidad de guardar gráficas. Para ello, hay un panel como el de la Figura 9.39, en cada una de las pestañas. En el caso de la generación y concatenación de trayectorias, los controles de este panel permanecen deshabilitados hasta que se ha hecho clic en el botón “Crear trayectoria”. En la pantalla “Convertir trayectoria”, y “Simular trayectoria”, no se habilitan hasta que se hace clic en el botón “Convertir trayectoria” y “Simular trayectoria” respectivamente.

Se puede seleccionar la sección del cuerpo de cuyas articulaciones se quiere guardar las gráficas. Si se selecciona más de una sección, se habilita la casilla “Guardar selección en la misma imagen”.

En el ejemplo de la Figura 9.39, se han seleccionado todas estas secciones, y se ha indicado que se desea guardar todas las representaciones en una misma imagen. Se ha elegido ver los ángulos en radianes.

Con estos parámetros se obtendría la imagen de la Figura 9.40. Queda habilitado un menú de herramientas de MATLAB que permite modificar las gráficas y guardarlas en distintos formatos.

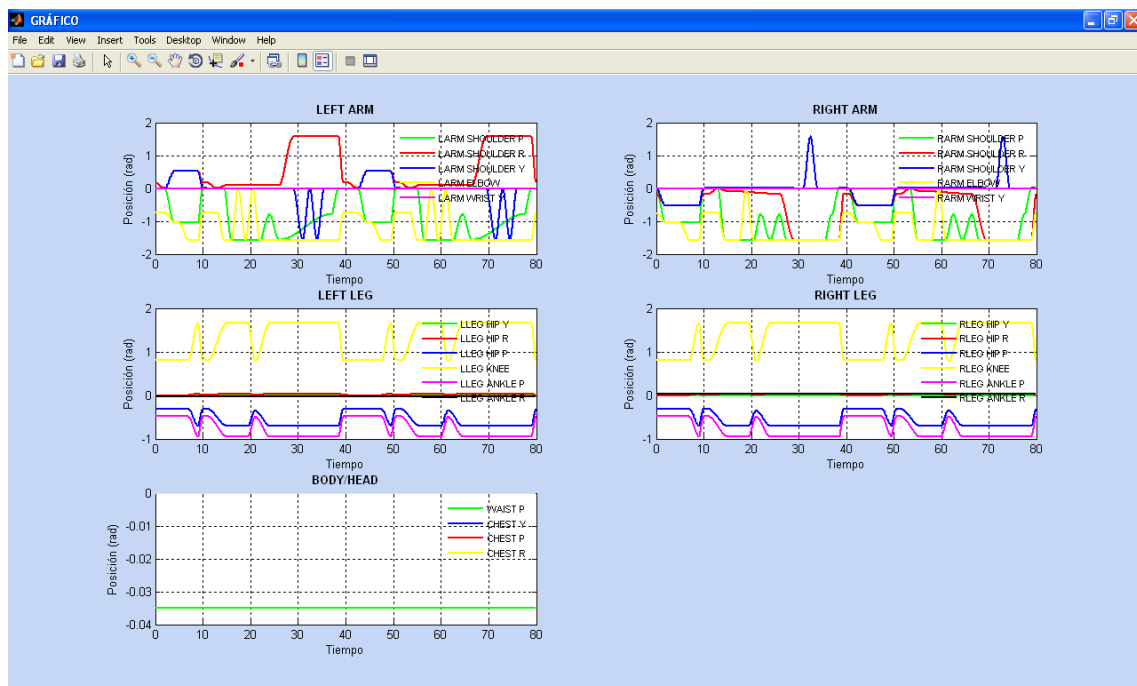


Figura 9.40. Ejemplo de gráficos tras crear la trayectoria

## 2 El robot HOAP-3

### 2.1 Estructura del robot

El HOAP-3 cuenta con 28 grados de libertad dispuestos según el diagrama de la Figura 9.41. En la Tabla 9.1 se refieren los movimientos asociados a cada una de las articulaciones en función de los ejes *Yaw*, *Pitch* y *Roll*.



Figura 9.41. Grados de libertad del HOAP-3

Articulación	Identificador	Movimiento
RLEG_JOINT1	1	Right hip joint torsion
RLEG_JOINT2	2	Right hip joint roll
RLEG_JOINT3	3	Right hip joint pitch
RLEG_JOINT4	4	Right knee
RLEG_JOINT5	5	Right ankle pitch
RLEG_JOINT6	6	Right ankle roll
RARM_JOINT1	7	Right shoulder pitch
RARM_JOINT2	8	Right shoulder roll
RARM_JOINT3	9	Right shoulder torsion
RARM_JOINT4	10	Right elbow
LLEG_JOINT1	11	Left hip joint torsion
LLEG_JOINT2	12	Left hip joint roll
LLEG_JOINT3	13	Left hip joint pitch
LLEG_JOINT4	14	Left knee
LLEG_JOINT5	15	Left ankle pitch
LLEG_JOINT6	16	Left ankle roll
LARM_JOINT1	17	Left shoulder pitch
LARM_JOINT2	18	Left shoulder roll
LARM_JOINT3	19	Left shoulder torsion
LARM_JOINT4	20	Left elbow
BODY_JOINT1	21	Waist pitch
HEAD_JOINT1	22	Head torsion
HEAD_JOINT2	22	Head pitch
HEAD_JOINT3	22	Head roll
LARM_JOINT5	23	Left fingers open/close
RARM_JOINT5	23	Right fingers open/close
LARM_JOINT6	23	Left hand torsion
RARM_JOINT6	23	Right hand torsion

**Tabla 9.1. Movimiento asociado a cada articulación en el HOAP-3**

En el modelo no se han incluido los grados de libertad correspondientes a las manos. Éstos están asociados a las articulaciones LARM\_JOINT5 y RARM\_JOINT5, que sirven para abrir y cerrar los dedos permitiendo así agarrar objetos.

Por otro lado, hay que señalar que no es posible asignar valores a las articulaciones CHEST\_Y, CHEST\_P, CHEST\_R, RARM\_WRIST\_Y y LARM\_WRIST\_Y. Estas articulaciones permanecerán en reposo (posición 0°) durante toda la trayectoria.

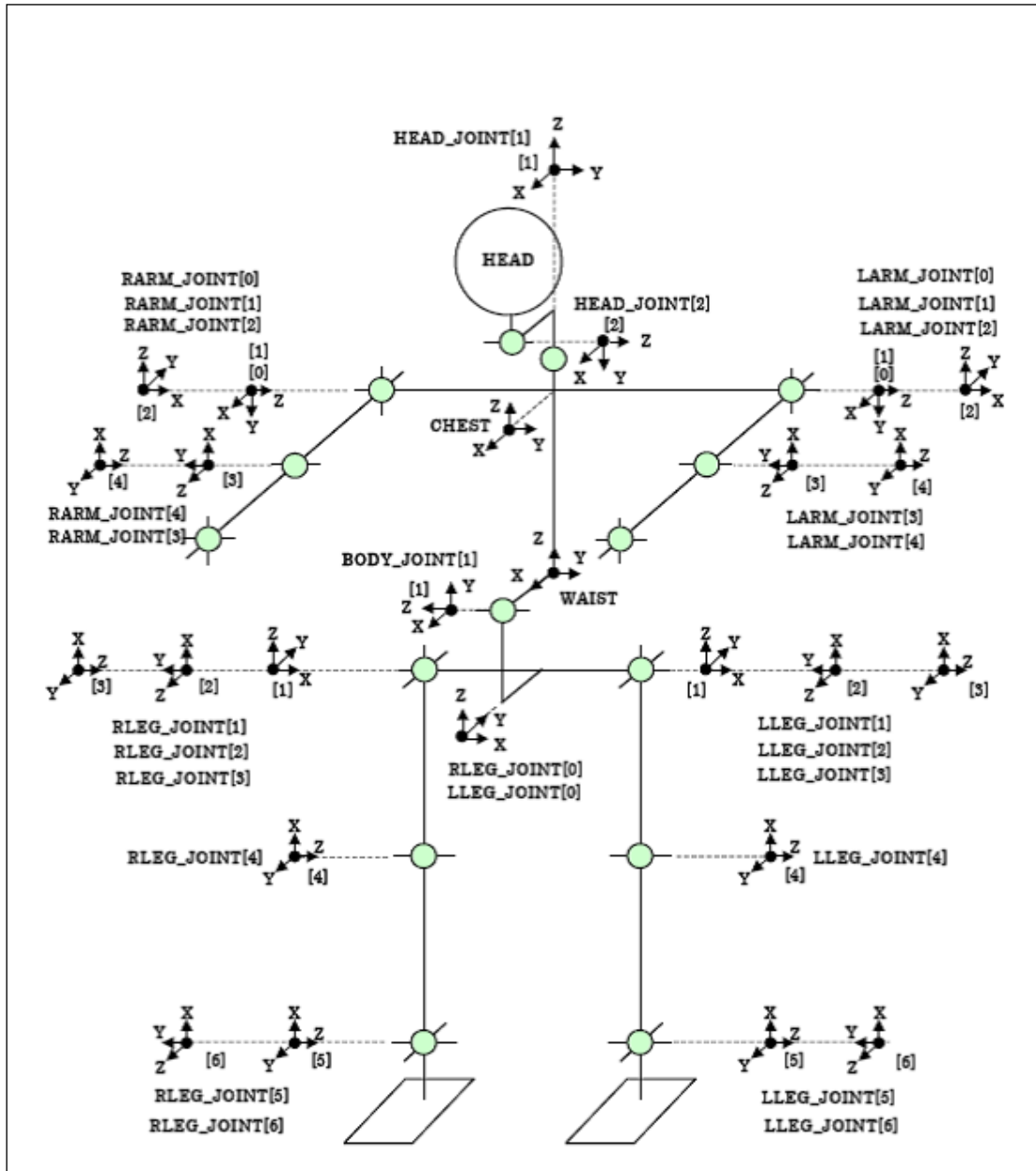
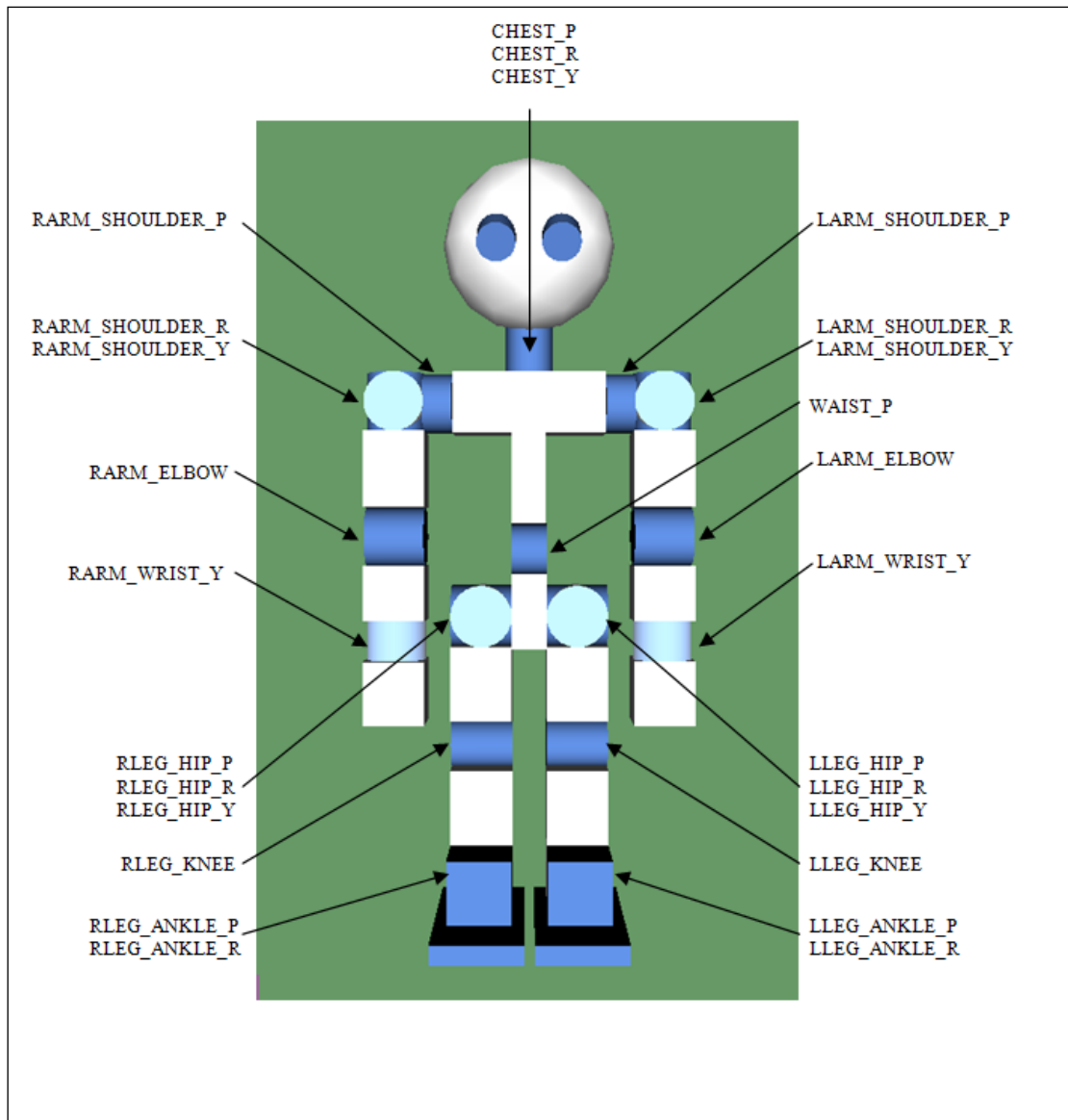


Figura 9.42. Definición de parámetros de DH en el HOAP-3

En la Figura 9.42 se indica el criterio de signos según los parámetros de Denavit-Hartenberg. Algunos motores están dispuestos en sentido contrario a este criterio, pero este hecho ya ha sido tenido en cuenta al diseñar la interfaz. Por tanto, para crear trayectorias por medio de esta aplicación hay que ceñirse a los criterios de Denavit-Hartenberg en todos los casos.

Mientras que la posición de reposo para el HOAP-3 (es decir, donde todos los encoder de los motores marcan 0°) corresponde a la posición en que el robot tiene los brazos extendidos en perpendicular al tronco, en el modelo VRML se ha definido la posición de reposo cuando el robot tiene los brazos extendidos en paralelo al tronco. Estas diferencias pueden apreciarse examinando la Figura 9.42 y la Figura 9.43. Para elaborar trayectorias con esta interfaz se debe considerar como posición inicial la del modelo VRML, reflejada en la Figura 9.43.



**Figura 9.43. Articulaciones en el modelo VRML del HOAP-3**

En la Tabla 9.2 se puede consultar el rango de movimiento permitido para las articulaciones del robot. Es importante señalar que en la práctica, por seguridad, deber reducirse este rango de movimientos en 1° por encima del mínimo y 1° por debajo del

máximo. Siguiendo esta directriz, y a modo de ejemplo, el rango real de movimiento para la articulación RLEG\_JOINT1 quedará reducido a  $(-90^{\circ}, 30^{\circ})$ .

Articulación	Id	Movimiento asociado	Valor mínimo	Valor máximo
RLEG_JOINT1	1	Right hip joint torsion	-91	31
RLEG_JOINT2	2	Right hip joint roll	-31	21
RLEG_JOINT3	3	Right hip joint pitch	-82	71
RLEG_JOINT4	4	Right knee	-1	130
RLEG_JOINT5	5	Right ankle pitch	-61	61
RLEG_JOINT6	6	Right ankle roll	-25	25
RARM_JOINT1	7	Right shoulder pitch	-181	61
RARM_JOINT2	8	Right shoulder roll	-96	1
RARM_JOINT3	9	Right shoulder torsion	-91	91
RARM_JOINT4	10	Right elbow	-115	1
LLEG_JOINT1	11	Left hip joint torsion	-31	91
LLEG_JOINT2	12	Left hip joint roll	-21	31
LLEG_JOINT3	13	Left hip joint pitch	-82	71
LLEG_JOINT4	14	Left knee	-1	130
LLEG_JOINT5	15	Left ankle pitch	-61	61
LLEG_JOINT6	16	Left ankle roll	-25	25
LARM_JOINT1	17	Left shoulder pitch	-181	61
LARM_JOINT2	18	Left shoulder roll	-1	96
LARM_JOINT3	19	Left shoulder torsion	-91	91
LARM_JOINT4	20	Left elbow	-115	1
BODY_JOINT1	21	Waist pitch	-90	1

**Tabla 9.2. Rango de movimiento de las articulaciones del HOAP-3 para el usuario**

## 2.2 Archivos de trayectorias

### 2.2.1 Archivos .dat:

Para el simulador, la trayectoria del robot viene definida por tres archivos que forman parte del controlador. Éstos indican los ángulos en radianes (*angle.dat*), velocidades en radianes/segundo (*vel.dat*) y aceleraciones en radianes/segundo<sup>2</sup> (*acc.dat*) que debe tomar cada una de las articulaciones a cada paso de simulación de la trayectoria completa.

El archivo *angle.dat* está compuesto por 27 columnas. La primera, es una secuencia de valores de tiempo que parte de cero y en cada fila se ve aumentada en el valor del paso de la simulación en milisegundos. Ésta será la base temporal de la simulación. Las otras 26 columnas corresponden a las 26 articulaciones del robot. En éstas, los valores de cada fila representan la posición que deberá marcar el encoder del motor correspondiente en el instante de tiempo que indica el valor de la primera columna para esa misma fila. El orden de las articulaciones es el indicado en la Tabla 9.3.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	0	0.003340343	-0.308564191	0	0.796421296	-0.484934305	0.027891865	0	-0.167017153	8,35E+00	-0.734958983	0	0.003340343	-0.311486991
2	0.002	0.003262688	-0.308690561	-7,80E+00	0.796583023	-0.484986705	0.02789176	0	-0.167017153	8,35E+00	-0.734958983	0	0.003340343	-0.311486991
3	0.004	0.003189373	-0.308816904	-0.000155512	0.796745215	-0.485039505	0.027887312	0	-0.167017153	8,35E+00	-0.734958983	0	0.003344693	-0.311486991
4	0.006	0.003124228	-0.308943462	-0.000232656	0.796908375	-0.485092964	0.027874692	0	-0.167017153	8,35E+00	-0.734958983	0	0.003357223	-0.311486991
5	0.008	0.003066428	-0.309070183	-0.000309393	0.797072394	-0.485147015	0.027865476	0	-0.167017153	8,35E+00	-0.734958983	0	0.003377105	-0.311486711
6	0.01	0.003015468	-0.309197035	-0.000385725	0.797237202	-0.485201635	0.02782792	0	-0.167017153	8,35E+00	-0.734958983	0	0.003403833	-0.311486505
7	0.012	0.002971036	-0.309323997	-0.000461654	0.797402753	-0.485256791	0.027794587	0	-0.167017153	8,35E+00	-0.734958983	0	0.003437092	-0.311486248
8	0.014	0.002932936	-0.309451054	-0.000537183	0.797569017	-0.485312465	0.027754924	0	-0.167017153	8,35E+00	-0.734958983	0	0.003476685	-0.311485938
9	0.016	0.002901041	-0.309578197	-0.000612312	0.797735973	-0.485368658	0.027709059	0	-0.167017153	8,35E+00	-0.734958983	0	0.003522482	-0.311485574
10	0.018	0.002875265	-0.309705415	-0.000687044	0.797903602	-0.485425348	0.02765708	0	-0.167017153	8,35E+00	-0.734958983	0	0.003574396	-0.311485157
11	0.02	0.002855548	-0.309832714	-0.00076138	0.798071891	-0.485482531	0.027599046	0	-0.167017153	8,35E+00	-0.734958983	0	0.003632366	-0.311484684
12	0.022	0.002841842	-0.309960076	-0.000835321	0.798240826	-0.485540202	0.027535007	0	-0.167017153	8,35E+00	-0.734958983	0	0.003696342	-0.311484155
13	0.024	0.002834113	-0.310087495	-0.000908865	0.798410395	-0.485598352	0.027464998	0	-0.167017153	8,35E+00	-0.734958983	0	0.003766288	-0.311483567
14	0.026	0.002832328	-0.31021498	-0.000982024	0.798580587	-0.485656975	0.027389053	0	-0.167017153	8,35E+00	-0.734958983	0	0.00384217	-0.311482908
15	0.028	0.002836459	-0.310342512	-0.001054785	0.798751388	-0.485716064	0.0273072	0	-0.167017153	8,35E+00	-0.734958983	0	0.003923958	-0.311482188
16	0.03	0.00284648	-0.310470094	-0.001127164	0.798922788	-0.485775614	0.027219466	0	-0.167017153	8,35E+00	-0.734958983	0	0.004011625	-0.3114814
17	0.032	0.002862367	-0.310597716	-0.001199151	0.799094772	-0.485835616	0.027125878	0	-0.167017153	8,35E+00	-0.734958983	0	0.004105143	-0.31148054
18	0.034	0.0028884094	-0.310725375	-0.00127075	0.799267329	-0.485896065	0.027026461	0	-0.167017153	8,35E+00	-0.734958983	0	0.004204488	-0.311479605
19	0.036	0.002911638	-0.310853065	-0.001341964	0.799440446	-0.485956952	0.02692124	0	-0.167017153	8,35E+00	-0.734958983	0	0.004309633	-0.311478591
20	0.038	0.002944975	-0.31098078	-0.001412794	0.799614108	-0.48601827	0.026810239	0	-0.167017153	8,35E+00	-0.734958983	0	0.004420553	-0.311477495
21	0.04	0.002984081	-0.311108515	-0.001483241	0.799788301	-0.486080013	0.026693483	0	-0.167017153	8,35E+00	-0.734958983	0	0.004537222	-0.311476305
22	0.042	0.003028934	-0.311236262	-0.001553305	0.799963012	-0.486142171	0.026570996	0	-0.167017153	8,35E+00	-0.734958983	0	0.004659617	-0.311475035

Figura 9.44. Vista parcial de un archivo de trayectorias *angle.dat*

De manera análoga, el archivo *vel.dat* y *acc.dat* mostrarán los valores de velocidad y aceleración respectivamente, que deben tomar las articulaciones a cada valor de tiempo de la primera columna.

<b>Articulación HOAP-3</b>	<b>Articulación modelo VRML</b>	<b>Columnas archivo .dat</b>	<b>Columnas archivo .csv</b>
RLEG_JOINT1	RLEG_HIP_Y	2	3
RLEG_JOINT2	RLEG_HIP_R	3	4
RLEG_JOINT3	RLEG_HIP_P	4	5
RLEG_JOINT4	RLEG_KNEE	5	6
RLEG_JOINT5	RLEG_ANKLE_P	6	7
RLEG_JOINT6	RLEG_ANKLE_R	7	8
RARM_JOINT1	RARM_SHOULDER_P	8	9
RARM_JOINT2	RARM_SHOULDER_R	9	10
RARM_JOINT3	RARM_SHOULDER_Y	10	11
RARM_JOINT4	RARM_ELBOW	11	12
LLEG_JOINT1	LLEG_HIP_Y	12	13
LLEG_JOINT2	LLEG_HIP_R	13	14
LLEG_JOINT3	LLEG_HIP_P	14	15
LLEG_JOINT4	LLEG_KNEE	15	16
LLEG_JOINT5	LLEG_ANKLE_P	16	17
LLEG_JOINT6	LLEG_ANKLE_R	17	18
LARM_JOINT1	LARM_SHOULDER_P	18	19
LARM_JOINT2	LARM_SHOULDER_R	19	20
LARM_JOINT3	LARM_SHOULDER_Y	20	21
LARM_JOINT4	LARM_ELBOW	21	22
BODY_JOINT1	WAIST_P	22	23
HEAD_JOINT1	CHEST_Y	23	24
HEAD_JOINT2	CHEST_P	24	24
HEAD_JOINT3	CHEST_R	25	24
RARM_JOINT6	RARM_WRIST_Y	26	25
LARM_JOINT6	LARM_WRIST_Y	27	25

**Tabla 9.3. Articulaciones en archivos de trayectorias para el HOAP-3**



### 2.2.2 Archivos .csv:

El robot puede funcionar en dos modos: online y offline. En el modo offline le cargaremos una secuencia de comandos que el robot irá ejecutando paso a paso. Estos archivos deben estar en formato *csv*, acrónimo del inglés ‘*comma separated value*’, o archivo de valores separados por comas. Cada fila del archivo corresponde a un comando que se le manda al robot, especificando el valor que debe tomar cada articulación. Las unidades son pulsos decimales, siendo el factor de conversión:

$$1^\circ = 209 \text{ pulsos.}$$

La frecuencia con la que el robot ejecuta cada comando es el paso de la simulación.

Cada comando está compuesto por 29 valores separados por comas. La estructura es la siguiente:

Modo de funcionamiento, paso de la simulación, posición motor 1, posición motor 2, posición motor 3, y así hasta indicar la posición de los 23 motores.

Los valores para los primeros 21 motores corresponden a las primeras 21 articulaciones (posiciones 3 a 23 del archivo). Según el valor asignado al motor 22 (posición 24 del archivo) se estará moviendo una de las tres articulaciones de la cabeza, y según el valor asignado al motor 23 (posición 25 del archivo) se estará moviendo una de las dos muñecas.

Como se ha comentado, no podemos definir posiciones para algunas de las articulaciones mediante esta interfaz, pero en el archivo de trayectorias hay que asignar valores a todas. Por tanto, al crear trayectorias por medio de la interfaz, el programa asignará automáticamente el valor de  $0^\circ$  a las articulaciones que no puede controlar el usuario, permaneciendo constante este valor durante toda la trayectoria. Para los motores 22 y 23, la conversión es distinta a la indicada antes. En este caso, la posición  $0^\circ$  equivale a 60 pulsos. Por ello las posiciones 24 y 25 del archivo tendrán siempre un valor de 60.

Como estamos generando trayectorias para su uso en el modo offline, el valor del modo de funcionamiento debe ser “2” para todos los comandos de la trayectoria.

El valor del paso de la simulación que se puede elegir para crear trayectorias con esta interfaz puede variar entre 1 ms y 5 ms, pero debe ser igual para todos los comandos de la trayectoria.

Después de los 25 valores descritos más arriba, hay cuatro valores que corresponden a los sensores. En el modo de funcionamiento offline, tomarán el valor R.

En la Figura 9.45 se puede observar la estructura que se ha indicado. En este ejemplo, el paso de la simulación elegido para la trayectoria es “2”. A continuación se reflejan los valores de la primera línea.

2, 2, 0, 42, 3762, 9530, -5810, -334, 18810, -2006, 1, 8778, 0, 42, -3762, -9530, 5810, 424, -18810, 1998, 0, -8778, 418, 60, 60, R, R, R, R

Estos valores constituyen el primer comando que recibirá el robot. Éste se emplea para llevarlo a la posición inicial, y es igual para cualquier trayectoria generada para el HOAP-3 a través de esta interfaz.

	A	B	C	D	E	F	G	H	I
1	2,2,0,42,3762	9530,-5810,-334,18810,-2006,1,8778,0,42,-3762,-9530,5810,424,-18810,1998,0,-8778,418,60,60,R,R,R,R							
2	2,2,0,42,3762,9530,-5810,-334,18810,-2006,1,8778,0,42,-3762,-9530,5810,424,-18810,1998,0,-8778,418,60,60,R,R,R,R								
3	2,2,0,42,3762,9530,-5810,-334,18810,-2006,1,8778,0,42,-3762,-9530,5810,424,-18810,1998,0,-8778,418,60,60,R,R,R,R								
4	2,2,0,42,3762,9530,-5810,-334,18809,-2006,1,8778,0,42,-3762,-9530,5810,424,-18809,1998,0,-8778,418,60,60,R,R,R,R								
5	2,2,0,42,3762,9530,-5810,-334,18809,-2006,1,8778,0,42,-3762,-9530,5810,424,-18809,1998,0,-8778,418,60,60,R,R,R,R								
6	2,2,0,42,3762,9530,-5810,-334,18809,-2006,1,8778,0,42,-3762,-9530,5810,424,-18809,1998,0,-8778,418,60,60,R,R,R,R								
7	2,2,0,42,3762,9530,-5810,-334,18808,-2006,1,8778,0,42,-3762,-9530,5810,424,-18808,1998,0,-8778,418,60,60,R,R,R,R								
8	2,2,0,42,3762,9530,-5810,-334,18807,-2006,1,8778,0,42,-3762,-9530,5810,424,-18807,1998,0,-8778,418,60,60,R,R,R,R								
9	2,2,0,42,3762,9530,-5810,-334,18806,-2006,1,8778,0,42,-3762,-9530,5810,424,-18806,1998,0,-8778,418,60,60,R,R,R,R								
10	2,2,0,42,3762,9530,-5810,-334,18805,-2006,1,8778,0,42,-3762,-9530,5810,424,-18805,1998,0,-8778,418,60,60,R,R,R,R								
11	2,2,0,42,3762,9530,-5810,-334,18804,-2006,1,8778,0,42,-3762,-9530,5810,424,-18804,1998,0,-8778,418,60,60,R,R,R,R								
12	2,2,0,42,3762,9530,-5810,-334,18803,-2006,1,8778,0,42,-3762,-9530,5810,424,-18803,1998,0,-8778,418,60,60,R,R,R,R								
13	2,2,0,42,3762,9530,-5810,-334,18802,-2006,1,8778,0,42,-3762,-9530,5810,424,-18802,1998,0,-8778,418,60,60,R,R,R,R								
14	2,2,0,42,3762,9530,-5810,-334,18801,-2006,1,8778,0,42,-3762,-9530,5810,424,-18801,1998,0,-8778,418,60,60,R,R,R,R								
15	2,2,0,42,3762,9530,-5810,-334,18799,-2006,1,8778,0,42,-3762,-9530,5810,424,-18799,1998,0,-8778,418,60,60,R,R,R,R								
16	2,2,0,42,3762,9530,-5810,-334,18797,-2006,1,8778,0,42,-3762,-9530,5810,424,-18797,1998,0,-8778,418,60,60,R,R,R,R								
17	2,2,0,42,3762,9530,-5810,-334,18796,-2006,1,8778,0,42,-3762,-9530,5810,424,-18796,1998,0,-8778,418,60,60,R,R,R,R								
18	2,2,0,42,3762,9530,-5810,-334,18794,-2006,1,8778,0,42,-3762,-9530,5810,424,-18794,1998,0,-8778,418,60,60,R,R,R,R								
19	2,2,0,42,3762,9530,-5810,-334,18792,-2006,1,8778,0,42,-3762,-9530,5810,424,-18792,1998,0,-8778,418,60,60,R,R,R,R								

Figura 9.45. Vista parcial de un archivo de trayectorias con extensión csv

## 3 El robot RH-2

### 3.1 Estructura del robot

El RH-2 tiene 28 grados de libertad. En la Figura 9.46 se muestra una representación del robot indicando la localización de los grados de libertad y el criterio positivo de signos según Denavit-Hartenberg.

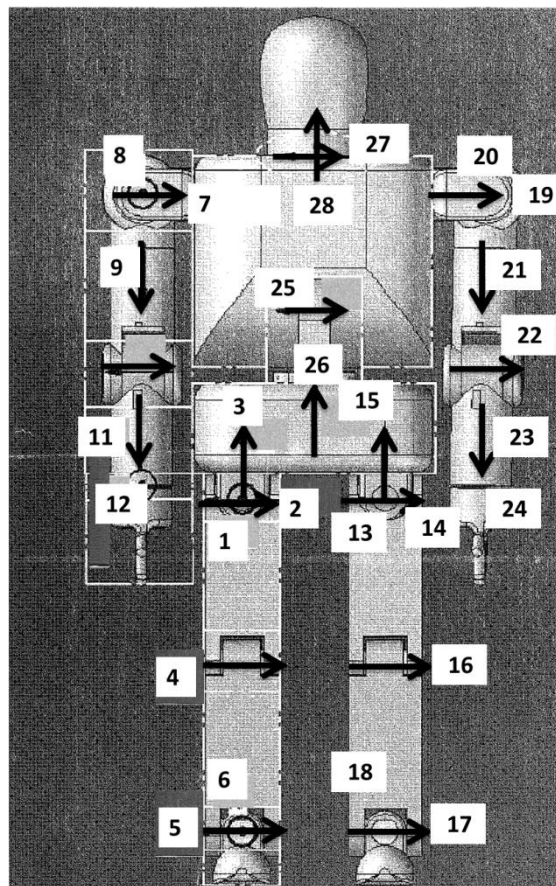


Figura 9.46. Distribución de GDL en el RH-2

Articulación	Identificador	Movimiento
RLEG_HIP_R	1	Roll cadera
RLEG_HIP_P	2	Pitch cadera
RLEG_HIP_Y	3	Yaw cadera
RLEG_KNEE	4	Rodilla
RLEG_ANKLE_P	5	Pitch tobillo
RLEG_ANKLE_R	6	Roll tobillo
RARM_SHOULDER_P	7	Pitch brazo
RARM_SHOULDER_R	8	Roll brazo
RARM_SHOULDER_Y	9	Yaw brazo
RARM_ELBOW	10	Codo
RARM_WRIST_Y	11	Yaw muñeca
RARM_WRIST_R	12	Roll muñeca
LLEG_HIP_R	13	Roll cadera
LLEG_HIP_P	14	Pitch cadera
LLEG_HIP_Y	15	Yaw cadera
LLEG_KNEE	16	Rodilla
LLEG_ANKLE_P	17	Pitch tobillo
LLEG_ANKLE_R	18	Roll tobillo
LARM_SHOULDER_P	19	Pitch brazo
LARM_SHOULDER_R	20	Roll brazo
LARM_SHOULDER_Y	21	Yaw brazo
LARM_ELBOW	22	Codo
LARM_WRIST_Y	23	Yaw muñeca
LARM_WRIST_R	24	Roll muñeca
WAIST_P	25	Pitch cintura
WAIST_Y	26	Yaw cintura
CHEST_Y	27	Yaw cabeza
CHEST_P	28	Pitch cabeza

**Tabla 9.4. Movimiento asociado a cada articulación en el RH-2**

En la Tabla 9.5 se indica el rango de movimientos permitido para cada articulación. Éstos deben ser tomados como valores orientativos, ya que están basados en simulaciones por ordenador. A medida que se vaya avanzando en el desarrollo del humanoide, se irá concretando estos valores.

Articulación	Id	Movimiento	Ángulo mínimo	Ángulo máximo
RLEG_HIP_R	1	Roll cadera	-40	30
RLEG_HIP_P	2	Pitch cadera	-120	45
RLEG_HIP_Y	3	Yaw cadera	-30	24
RLEG_KNEE	4	Rodilla	0	150
RLEG_ANKLE_P	5	Pitch tobillo	-85	85
RLEG_ANKLE_R	6	Roll tobillo	-20	20
RARM_SHOULDER_P	7	Pitch brazo	-180	180
RARM_SHOULDER_R	8	Roll brazo	-60	60
RARM_SHOULDER_Y	9	Yaw brazo	-45	120
RARM_ELBOW	10	Codo	-100	100
RARM_WRIST_Y	11	Yaw muñeca	-105	105
RARM_WRIST_R	12	Roll muñeca	-40	55
LLEG_HIP_R	13	Roll cadera	-40	30
LLEG_HIP_P	14	Pitch cadera	-120	45
LLEG_HIP_Y	15	Yaw cadera	-30	24
LLEG_KNEE	16	Rodilla	0	150
LLEG_ANKLE_P	17	Pitch tobillo	-85	85
LLEG_ANKLE_R	18	Roll tobillo	-20	20
LARM_SHOULDER_P	19	Pitch brazo	-180	180
LARM_SHOULDER_R	20	Roll brazo	-60	60
LARM_SHOULDER_Y	21	Yaw brazo	-45	120
LARM_ELBOW	22	Codo	-100	100
LARM_WRIST_Y	23	Yaw muñeca	-105	105
LARM_WRIST_R	24	Roll muñeca	-40	55
WAIST_P	25	Pitch cintura	Ángulo por definir	Ángulo por definir
WAIST_Y	26	Yaw cintura	Ángulo por definir	Ángulo por definir
CHEST_Y	27	Yaw cabeza	Ángulo por definir	Ángulo por definir
CHEST_P	28	Pitch cabeza	Ángulo por definir	Ángulo por definir

**Tabla 9.5. Rango de movimiento permitido para cada articulación del RH-2**

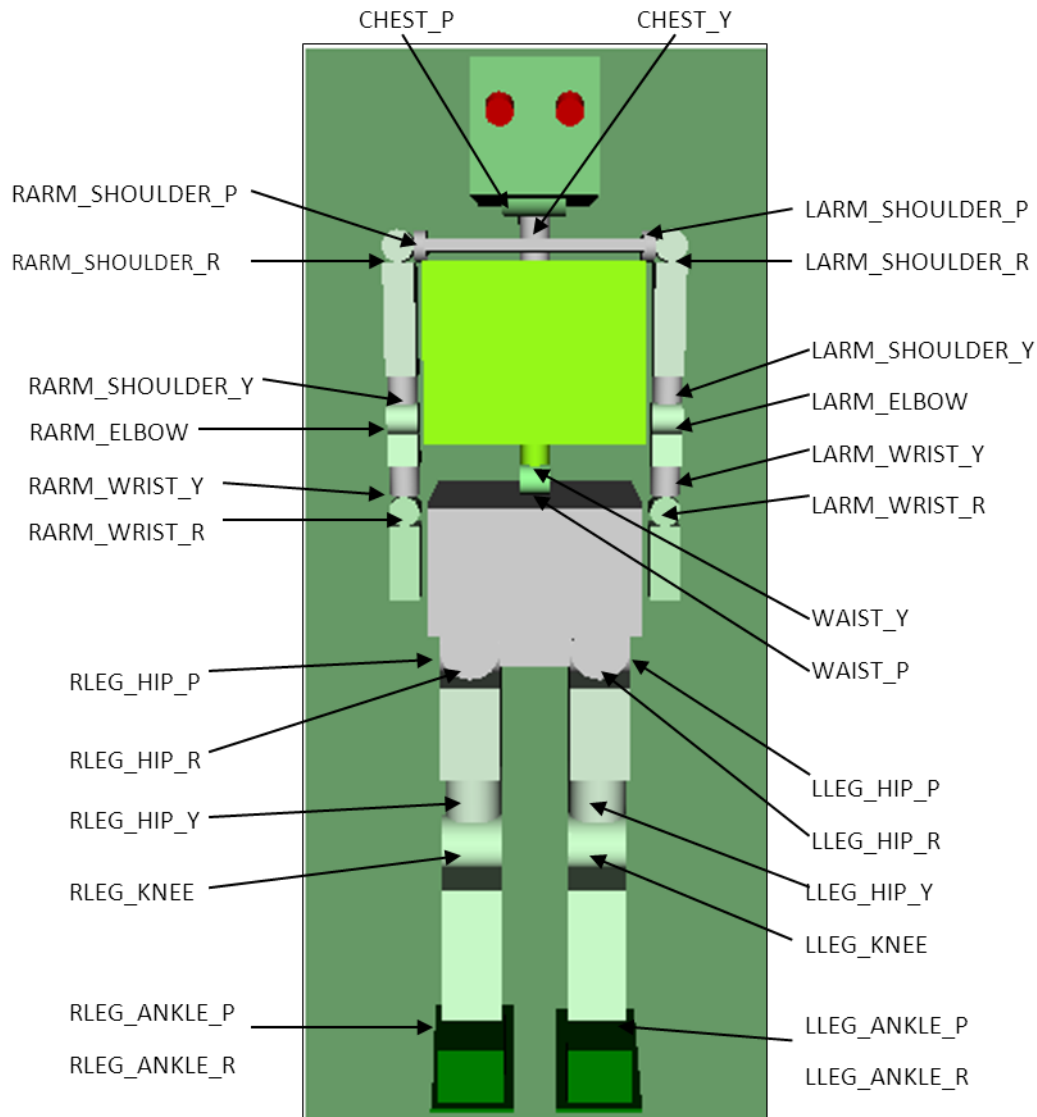


Figura 9.47. Modelo VRML del robot RH-2

## 3.2 Archivos de trayectorias

Para el simulador, la trayectoria del robot viene definida por tres archivos, que forman parte del controlador. Éstos indican los ángulos en radianes (*angle.dat*), velocidades en radianes/segundo (*vel.dat*) y aceleraciones en radianes/segundo<sup>2</sup> (*acc.dat*) que debe tomar cada una de las articulaciones a cada paso de simulación de la trayectoria.

El archivo *angle.dat* está compuesto por 29 columnas. La primera, es una secuencia de valores de tiempo que parten de cero y en cada fila se ven aumentados en el valor del paso de la simulación en milisegundos. Ésta será la base temporal de la simulación. Las otras 28 columnas corresponden a las 28 articulaciones del robot. En éstas, los valores de cada fila representan la posición que deberá marcar el encoder

del motor correspondiente en el instante de tiempo que indica el valor de la primera columna para esa misma fila. El orden de las articulaciones es el indicado en la Tabla 9.6.

En el ejemplo de la Figura 9.48, los brazos están levantados  $90^\circ$ , quedando en posición perpendicular al tronco. Las articulaciones LARM\_SHOULDER\_P (columna 8) y RARM\_SHOULDER\_P (columna 18), toman el valor de  $-90^\circ$  ( $-1.57$  rad), signo negativo debido al criterio de signos que se define con los parámetros de Denavit-Hartenberg de la Figura 9.46.

G	H	I	J	K	L	M	N	O	P	Q	R	S	T
0	-1.57079633	0	0	0	0	0	0	0	0	0	0	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-7.13E-06	0	1.43E-05	-7.13E-06	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-2.84E-05	0	5.69E-05	-2.84E-05	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-6.39E-05	0	0.00012774	-6.39E-05	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00011332	0	0.00022664	-0.00011332	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00017671	0	0.00035342	-0.00017671	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00025395	0	0.0005079	-0.00025395	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00034495	0	0.00068991	-0.00034495	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00044964	0	0.00089928	-0.00044964	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00056793	0	0.00113585	-0.00056793	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00069972	0	0.00139945	-0.00069972	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00084495	0	0.0016899	-0.00084495	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00100352	0	0.00200704	-0.00100352	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00117535	0	0.0023507	-0.00117535	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00136035	0	0.0027207	-0.00136035	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00155845	0	0.00311689	-0.00155845	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00176955	0	0.0035391	-0.00176955	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00199357	0	0.00398715	-0.00199357	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00223044	0	0.00446088	-0.00223044	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00248006	0	0.00496013	-0.00248006	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00274236	0	0.00548472	-0.00274236	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00301724	0	0.00603449	-0.00301724	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00330464	0	0.00660928	-0.00330464	0	-1.57079633
0	-1.57079633	0	0	0	0	0	0	-0.00360446	0	0.00720891	-0.00360446	0	-1.57079633

Figura 9.48. Archivo de trayectorias angle.dat para el RH-2

<b>Articulaciones RH-2</b>	<b>Identificador RH-2</b>	<b>Identificador modelo VRML</b>	<b>Columnas archivo .dat</b>
RLEG_HIP_R	1	0	2
RLEG_HIP_P	2	1	3
RLEG_HIP_Y	3	2	4
RLEG_KNEE	4	3	5
RLEG_ANKLE_P	5	4	6
RLEG_ANKLE_R	6	5	7
RARM_SHOULDER_P	7	6	8
RARM_SHOULDER_R	8	7	9
RARM_SHOULDER_Y	9	8	10
RARM_ELBOW	10	9	11
RARM_WRIST_Y	11	10	12
RARM_WRIST_R	12	11	13
LLEG_HIP_R	13	12	14
LLEG_HIP_P	14	13	15
LLEG_HIP_Y	15	14	16
LLEG_KNEE	16	15	17
LLEG_ANKLE_P	17	16	18
LLEG_ANKLE_R	18	17	19
LARM_SHOULDER_P	19	18	20
LARM_SHOULDER_R	20	19	21
LARM_SHOULDER_Y	21	20	22
LARM_ELBOW	22	21	23
LARM_WRIST_Y	23	22	24
LARM_WRIST_R	24	23	25
WAIST_P	25	24	26
WAIST_Y	26	25	27
CHEST_Y	27	26	28
CHEST_P	28	27	29

**Tabla 9.6. Articulaciones en archivos de trayectorias del RH-2**



# **10 Referencias**

## **MANUALES**

- [1] "HOAP-3 Instruction Manual". FUJITSU Automation Ltd.
- [2] Carrera Amuriza, A.R., Martínez Nebreda, M. (2004). "Introducción a MATLAB y a la creación de interfaces gráficas." Servicio Editorial de la Universidad del País Vasco.
- [3] Barrientos, A., Peñín, L.F., Balaguer, C., y Aracil, R. (2007). "Fundamentos de robótica" (2a ed.). Mc Graw-Hill.
- [4] 'MATLAB 7 Programming Fundamentals.' Mathworks. Septiembre de 2010.
- [5] 'MATLAB 7 Creating Graphical User Interfaces.' Mathworks. Septiembre de 2010.
- [6] 'MATLAB 7 Getting Started Guide.' Mathworks. Septiembre de 2010.
- [7] 'MATLAB 7 Graphics.' Mathworks. Septiembre de 2010.
- [8] 'MATLAB 7 Data Import and Export.' Mathworks. Septiembre de 2010.
- [9] 'MATLAB 7 Programming tips.' Mathworks. Septiembre de 2010.

## **TESIS DE DOCTORADO**

- [10] Arbulú Saavedra, M.R. "Stable locomotion of humanoid robots based on mass concentrated model". Universidad Carlos III de Madrid. 2008

## **PROYECTOS FIN DE CARRERA**

- [11] Ramos Cambero, T. "Simulación de la plataforma robótica HOAP-3 en el simulador OPENHRP3". Universidad Carlos III de Madrid. 2009
- [12] De Torre Doblás, C. "Caminata del robot humanoide RH-2 en la plataforma de simulación OpenHRP3". Universidad Carlos III de Madrid. 2010

[13] Sierra Molina, R.M. "Simulación de la plataforma robótica RH-2 en el simulador OpenHRP3". Universidad Carlos III de Madrid. 2010

[14] Navarro Criado, A. "Análisis de los accionadores del robot humanoide RH-2". Universidad Carlos III de Madrid. 2009

## **DIRECCIONES DE INTERNET**

[15] Documentación de MATLAB (28 de Marzo de 2010)

<http://www.mathworks.com/help/techdoc/index.html>

[16] Documentación de MATLAB en formato pdf (28 de Marzo de 2010)

[http://www.mathworks.com/help/techdoc/matlab\\_product\\_page2.html](http://www.mathworks.com/help/techdoc/matlab_product_page2.html)

[17] MATLAB Central (28 de Marzo de 2010)

<http://www.mathworks.com/matlabcentral/>

[18] Archivo maximize.m (MATLAB Central) (28 de Marzo de 2010)

<http://www.mathworks.se/matlabcentral/fileexchange/25471-maximize>

[19] Archivo uitabpanel.m (MATLAB Central) (28 de Marzo de 2010)

<http://www.mathworks.com/matlabcentral/fileexchange/11546>

[20] Web Robotics Lab (28 de Marzo de 2010)

<http://roboticslab.uc3m.es/roboticslab/>

## **OTROS**

[21] "HOAP-3 Press release". FUJITSU Automation Ltd.

[22] "HOAP-3 Specification Sheet". FUJITSU Automation Ltd.