

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA
MÁSTER EN INFORMÁTICA INDUSTRIAL

DESARROLLO DE UN SIMULADOR DE
LA CINEMÁTICA DE LA PIERNA DE UN
ROBOT HUMANOIDE

Curso 2012/2013

Alumno/a:

Adolfo Biosca Molina

Director/es:

Dr. Francisco Rodríguez Díaz
Dr. Ramón González Sánchez



TRABAJO FIN DE MÁSTER
MÁSTER EN INFORMÁTICA INDUSTRIAL
POSGRADO EN INFORMÁTICA



DESARROLLO DE UN SIMULADOR DE LA CINEMÁTICA DE
LA PIERNA DE UN ROBOT HUMANOIDE

por

Adolfo Biosca Molina

Para la obtención del

Título del Máster en Informática Industrial
Posgrado en Informática

Directores

Autor

Dr. Francisco Rodríguez Díaz
Dr. Ramón González Sánchez.

Adolfo Biosca Molina

Índice

Nomenclaturas	1
1. Introducción	2
2. Fundamentos matemáticos del modelado de robots humanóides	3
2.1 Grupos y álgebras de Lie para robots	4
2.2 Grupo de Lie especial euclídeo SE(3) y álgebra se(3)	4
2.3 Movimientos screw y twist	5
2.4 Problema PADEN-KAHAN-UNO, rotación alrededor de un eje	5
2.5 Fórmula del producto de exponenciales POE	5
3. Implementación de un simulador para el robot Robonova-I	6
3.1 Adaptación de los conceptos matemáticos al robot humanoide Robonova-I	7
4. Resultados	7
4.1 Movimiento de la pierna del robot a través de un eje	8
4.1.1 Movimiento de la pierna del robot a través de un eje usando exclusivamente Paden-Kahan Uno	8
4.1.2 Movimiento de la pierna del robot a través de un eje usando Paden-Kahan Uno especificando una posición no alcanzable	9
4.1.3 Movimiento de la pierna del robot a través de un eje usando Paden-Kahan Uno en combinación con POE	10
4.1.4 Movimiento de la pierna del robot a través de un eje usando Paden-Kahan Uno en combinación con POE especificando una posición no alcanzable	10
4.2 Movimiento de la pierna del robot a través de dos ejes	11
4.2.1 Movimiento de la pierna del robot con dos motores a través del plano X-Z	11
4.2.2 Movimiento de la pierna del robot con dos motores a través del plano X-Z especificando una posición no alcanzable.	13
4.2.3 Movimiento hacia delante de la pierna del robot con dos motores a través del plano Y-Z	14
4.2.4 Movimiento hacia atrás de la pierna del robot con dos motores a través del plano Y-Z	16
4.2.5 Movimiento hacia delante de la pierna del robot con dos motores a través del plano Y-Z usando un algoritmo recursivo	18
4.2.6 Movimiento hacia atrás de la pierna del robot con dos motores a través del plano Y-Z usando un algoritmo recursivo	19
4.3 Movimiento de la pierna del robot a través de tres ejes	20
5. Simulación de un paso	23
6. Conclusiones y Trabajos Futuros	27
7. Referencias	28
Apéndice A: Descripción del robot Robonova-I	29
Datos del robot	29
Sensores	29
Actuadores	29

Hardware:	29
Datos técnicos de la placa de control.....	30
Software:	30
<i>Apéndice B: Funciones implementadas en Matlab</i>	30

Desarrollo de un simulador de la cinemática de la pierna de un robot humanoide

Adolfo Biosca Molina
 Máster en Informática Industrial
 Postgrado en Informática
 Escuela Superior de Ingeniería
 Departamento de Informática
 Universidad de Almería

Abstract: The study of kinematic modelling of humanoid robots constitutes a challenge, mainly due to the number of degrees of freedom of these systems. The complexity of the problem is proportional to the number of joints and the number of engines. This paper shows a simulator related to the motion of one leg of one humanoid robot. For that purpose a kinematic model has been studied and implemented. The simulator is based on the mathematical tools: Paden-Kahan One and Product of Exponentials. In particular, the robot Robonova-I has been used for modelling purposes. This robot has five degrees of freedom in each leg. With the aim of showing the operation of the simulator proposed, the cases of leg movements in one, two and three axes, as well as the simulation of a full step are tried.

Resumen: El estudio de la cinemática de un robot humanoide supone un reto, debido a la cantidad de grados de libertad del sistema. La complejidad del problema es proporcional a la cantidad de articulaciones, aumentando exponencialmente conforme aumenta el número de motores. En este trabajo, se presenta un simulador de la cinemática de un robot que ayudará a implementar los modos de caminar del mismo. Se basa en las herramientas matemáticas de Paden-Kahan Uno y el Producto de Exponenciales para resolver el problema de la cinemática de una pierna. Como caso práctico, se ha utilizado el robot Robonova-I, que dispone de cinco grados de libertad en cada pierna. Para mostrar el buen funcionamiento de esta herramienta, se analizarán los casos de los movimientos de la pierna del robots en uno, dos y tres ejes, así como la simulación de un paso completo.

Keywords: Kinematic Model, Paden Kahan, POE

Palabras Clave: Modelo Cinemático, Algoritmo, Paden Kahan, POE

Nomenclaturas

Los acrónimos utilizados por orden de aparición son:

Acrónimo	Descripción
GDL	Grado de libertad
DH	Formulación de parámetros de Denavit-Hartenberg
SE	Grupo Especial Euclídeo
SO	Grupo Especial Ortogonal
PK1	Problema de Paden-Kahan Uno
POE	Fórmula del producto de exponenciales

De igual forma se presenta la simbología matemática usada.

Símbolo	Descripción
R	Conjunto de los números reales
g	Matriz de transformación
T	Vector de traslación
D	Matriz de rotación
se	Álgebra de Lie Asociada a un SE
ξ^{\wedge}	<i>Twist</i>
ω	Vector de dirección del <i>twist</i>
r	Punto del eje del <i>twist</i>
so	Álgebra de Lie Asociada a SO
θ	Ángulo del giro
$e^{\omega^{\wedge}\theta}$	Matriz de Transformación del <i>twist</i>
p	Punto inicial del método de Paden-Kahan Uno
k	Punto final del método de Paden-Kahan Uno
u	Vector que une r y p
v	Vector que une r y k.
g(0)	Matriz de transformación inicial
g(θ)	Matriz de transformación final

1. Introducción

Un robot humanoíde es aquel cuya forma y movimientos se asemejan a los del ser humano. Estos robots están especialmente orientados a los casos donde la similitud con el ser humano sea un tema prioritario. De esta forma hay robots trabajando en el sector educativo [1], servicios (por ejemplo, en lugar de recepcionistas humanos [2,3,4] o vigilancia [5]) y tareas de búsqueda y rescate [6,7]. La tendencia es que en el futuro puedan utilizarse para el cuidado de personas dependientes [8]. En la figura 1 pueden observarse los robots Actroid [9](a), desarrollado por la Universidad de Osaka y la empresa Kokoro, y el robot HRP-4C [10,11](b), creado por el *National Institute of Advanced Industrial Science and Technology* de Japón, donde la prioridad es lograr el mayor parecido posible en su aspecto con el ser humano. Actroid se está empleando como recepcionista desde que se usara por primera vez en la Exposición Mundial de Aichi de 2005 [12], mientras que HRP-4C tiene como objetivo ser modelo de ropa [13]

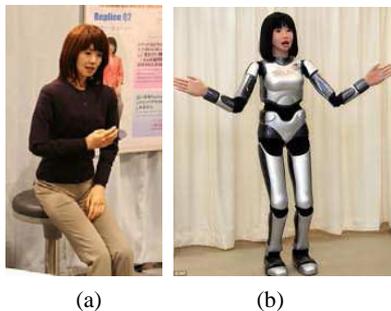


Figura 1: Robots Actroid y HRP-4C

Algunas peculiaridades de estos robots bípedos son: La superficie de apoyo con el suelo es muy reducida, sobre todo en los casos en los que uno de los pies está en el aire, dejando la estabilidad de todo el sistema sólo sobre el otro pie. Esto se relaciona con los conceptos de estabilidad estática y dinámica. La primera establece las condiciones en las que el robot mantendría el equilibrio cuando está parado, mientras que la estabilidad dinámica indica la capacidad para desplazarse sin caerse [14]. Para que el robot mantenga el equilibrio debe ser estable estática y/o dinámicamente [15]. A esto hay que añadir el gran número de grados de libertad que debe tener cada articulación, lo cual conlleva el problema de la alimentación de los motores. Conforme aumenta la cantidad de energía necesaria para alimentar dichos motores, también lo hace el tamaño de la batería, y en estos tipos de robots el espacio disponible es muy limitado.

Actualmente se desarrollan de forma paralela numerosos proyectos de investigación centrados en la locomoción de robots humanoídes. De esta forma, algunas de las empresas más relevantes en el campo de la ingeniería, como Honda o Toyota llevan años investigando en este campo. En la figura 2-a se puede ver el robot ASIMO [16], desarrollado por la empresa Honda, que consta de 57 grados de libertad y es capaz de correr a 9 km/h y caminar sobre superficies irregulares. En la figura 2-b se muestra el Partner Robot [17] de Toyota, capaz de correr a 7 km/h y de tocar la trompeta o el violín, en la figura 2-c se observa el robot desarrollado por Fujitsu, denominado HOAP [18], un robot de código abierto con un sistema operativo Linux, capaz de caminar sobre un terreno plano, traer objetos pequeños y seguir una

pelota. En la figura 2-d, el robot de la imagen es NAO [19], un robot con 25 GDL desarrollado por Aldebaran robotics, empleado actualmente con fines educativos.

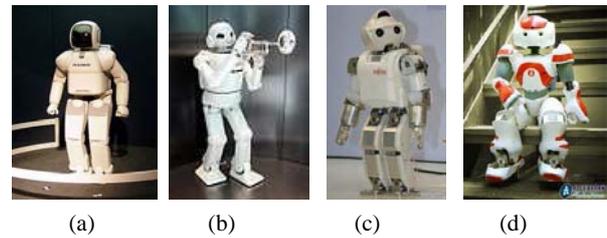


Figura 2: Robot ASIMO, Partner Robot, HOAP y NAO.

Otro ámbito donde se ha desarrollado esta área de la robótica ha sido el universitario, destacando las universidades de Japón. Así, la Universidad de Waseda fue la pionera sacando su Wabot-1 [20] en 1973 (Figura 3-a). Este prototipo incorporaba un control de las extremidades, un sistema de visión y un sistema de conversación. El WABOT-1 era capaz de comunicarse con una persona en japonés y de calcular las distancias a los objetos usando receptores externos. Además poseía ojos, orejas y boca artificiales. Los desarrolladores estimaron que WABOT-1 tenía la facultad mental de un niño de un año y medio. Otras universidades japonesas también desarrollan proyectos en locomoción de robótica humanoíde, como las de Tokio y Tohoku. Así, en la figura 3-b se muestra el robot H7 [21] de la Universidad de Tokio, con 35 grados de libertad, incluyendo 6 en cada pierna, mientras que en la figura 3-c se ve el SAIKA-3 [22], de la Universidad de Tohoku, formado por 30 GDL.

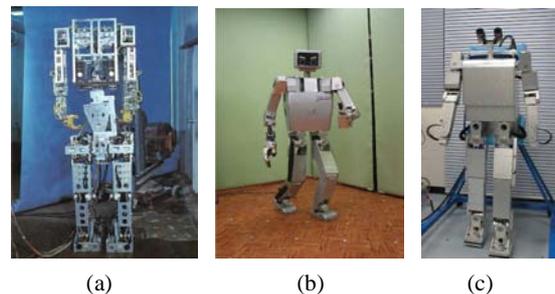


Figura 3: Robots WABOT-1, H y SAIKA-3

En España existen algunos proyectos en marcha. Cabe destacar el proyecto de la Universidad Carlos III de Madrid, el robot RH1 [23] (Figura 4-a), formado por 21 grados de libertad, incluyendo 6 en cada pierna. La empresa PAL Robotics, con sede en Barcelona, también desarrolla la serie REEM [24]. Actualmente, el REEM-B (figura 4-b) es capaz de reconocer rostros, comunicarse con las personas, caminar e incluso subir escaleras o sentarse.

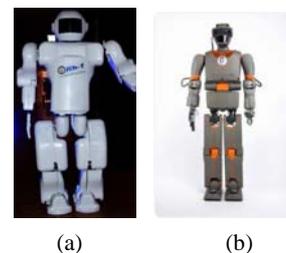


Figura 4: Robot RH1 y REEM-B

En el caso del robot humanoide Robonova-I, fabricado por Hitec, que se utilizará como modelo para el simulador, dispone de 16 grados de libertad, 3 en cada brazo y 5 en cada pierna.



Figura 5: Robot Robonova-I

En lo referente a simuladores de robots humanoides, hay que destacar OpenHRP3. Es un simulador y una biblioteca de control de movimiento para los robots humanoides desarrollados en el Instituto Nacional de Ciencia Industrial Avanzada y Tecnología, Kanehiro (2004) [25, 26]. Esta es una herramienta gratuita que puede simular la dinámica directa de las cadenas abiertas y cerradas de sistemas robotizados. Incluye librerías de posición, relación fuerza / par, visión artificial y sensores de inclinación. Estos son esenciales para el desarrollo del control de sistemas robóticos. El software también incluye controladores para la locomoción, equilibrio, un generador de patrones de marcha, así como sistemas de detección de colisiones evitación de las mismas. OpenHRP tiene una interfaz de visualización 3D para mostrar la simulación. El software desarrollado en OpenHRP puede ser utilizado directamente en los robots humanoides HRP, pero no en otros robots humanoides. La razón principal es que varias características de OpenHRP no pueden ser modificadas por el usuario. Por ejemplo, los modelos de actuadores son fijos, y la posición de las articulaciones del robot no puede modificarse. Además, existe muy poca documentación disponible para OpenHRP y desde que esta herramienta es gratuita, no hay apoyo por parte de los desarrolladores.

Otros simuladores han sido desarrollados por Honda, Sony o Fujitsu para sus robots humanoides (ASIMO, QRIO y HOAP-1). También existen paquetes software comerciales como Webots[27], Adams [28] y RoboWorks [29], útiles para el modelado de sistemas multicuerpo, pero carece de modelos para actuadores, sensores e impactos. Ninguno de estos paquetes proporcionan control total sobre todas las características de interés [30].

Existen numerosas investigaciones en el ámbito universitario en las que se desarrollan simuladores para robots humanoides con diferentes métodos. En la Universidad Carlos III de Madrid se desarrolló el simulador RobManSim [31] (Figura 6) utilizando Algebra de Lie para resolver la cinemática del robot RH0. La ventaja que tiene este robot es que los ejes de giro de los motores de la cintura y el tobillo coinciden, por lo que se puede utilizar Paden-Kahan Dos para resolver el problema, algo que no es posible para el Robonova-I. Dicho simulador ha sido desarrollado con el software Matlab-Simulink, incluyendo el entorno 3D. Hay que destacar que además del modelo cinemático, este simulador implementa la dinámica del robot.

Otros proyectos utilizan otros métodos para resolver el problema cinemático de un robot humanoide. Así en [32] los autores

utilizan DH para resolver el problema para el robot KHR-4, mientras que en [30] se aplican las fórmulas de Newton-Euler. En ambos casos se obtienen unas fórmulas para el movimiento mucho más complejas que usando Algebras de Lie y el Producto de Exponenciales, lo que deriva en un incremento en el tiempo de cómputo.

La principal contribución de este trabajo se enmarca en el desarrollo de un simulador para las extremidades de robots humanoides. Para ello se ha desarrollado un toolbox para Matlab. Estas herramientas se ejecutan de forma eficiente.



Figura 6: Simulador RobManSim para RH0

Este trabajo está organizado de la siguiente forma. En la sección 2 se presentan los fundamentos matemáticos en los que se basa el simulador, concretamente el Algebra de Lie y el Producto de Exponenciales. A continuación se aplicará al caso concreto del robot Robonova-I, para posteriormente mostrar algunos resultados del movimiento de la pierna del robot en uno, dos y tres ejes, así como la simulación de un paso. Finalmente, se indican las conclusiones y futuros trabajos a realizar.

2. Fundamentos matemáticos del modelado de robots humanoides.

En esta sección se presentan los fundamentos matemáticos utilizados para modelar el comportamiento cinemático de las piernas de un robot humanoide. Hay que indicar que la descripción de estas herramientas está extraída parcialmente de [31]. El tratamiento del movimiento utilizado en este trabajo se basa en la teoría de Grupos de Lie y de *screws* presentadas en el punto 2.1. Se desvía del estándar utilizado en la mayoría de trabajos de robótica, que siguen prefiriendo la formulación de parámetros de Denavit-Hartenberg (DH)[32]

Las ventajas de la teoría mecánica de *screws* y la matemática de Grupos de Lie asociada son las siguientes:

- Permiten una descripción realmente geométrica del movimiento que facilita el análisis mecánico.
- Permiten una descripción sin singularidades debidas al uso de coordenadas locales (como sucede por el contrario con las representaciones de ángulos de Euler y las de DH), ya que es posible el uso de sólo dos sistemas coordenados de referencia, el de la base y el de la herramienta.
- Permite usar la misma representación matemática para diferentes tipos de movimientos, esto es, tanto para traslaciones como para rotaciones.

Para representar el movimiento de un sólido rígido se usará la transformación exponencial de un *twist*, que proporciona el movimiento relativo del mismo.

La interpretación de esta transformación no es la relación entre los puntos de un sistema de coordenadas a otro, sino la relación entre las coordenadas iniciales de los puntos con sus coordenadas finales, tras aplicar un movimiento al sólido rígido.

Puesto que las cadenas abiertas de sólidos rígidos (por ejemplo robots manipuladores) están constituidas mediante la conexión de diferentes articulaciones (típicamente de revolución o prismáticas), usando eslabones rígidos, el movimiento queda restringido a un subgrupo del Grupo Especial Euclídeo SE(3), haciendo de la exponencial de un *twist* la representación natural para el análisis mecánico de estos sistemas.

En este proyecto se van a tener las siguientes asunciones:

- Se asume que el robot humanoíde se mueve en una superficie plana (sin desnivel) y libre de obstáculos.
- No se consideran las restricciones derivadas de la composición de movimientos: En este proyecto solo se consideran las restricciones individuales de cada uno de los motores individualmente.

2.1 Grupos y álgebras de Lie para robots

Un álgebra de Lie es una estructura algebraica que se usa para estudiar objetos geométricos tales como grupos de Lie y variedades diferenciables. Existen tres conceptos importantes que se deben conocer:

- Una variedad (Manifold) es un espacio matemático abstracto, en el que todos los puntos tienen una vecindad que se asemeja al espacio Euclídeo. Cuando se habla de los "manifold", es importante tratar el término de la dimensión. Por ejemplo, las líneas son unidimensionales, y los planos son de dos dimensiones. Además, otro ejemplo de una variedad es la superficie de una esfera, que no es un plano, pero pequeñas porciones de la superficie son topológicamente equivalentes a porciones de un plano Euclídeo. Dicha variedad es diferenciable, informalmente hablando, si cada uno de sus puntos tiene espacio tangente, es decir, no tiene "picos" ni "filos". Las Variedades diferenciales se usan en matemáticas para describir objetos geométricos y son la herramienta más general y natural para el estudio de la diferenciabilidad.
- Un grupo de Lie es una variedad analítica real o compleja que es también un grupo, tal que tiene un operador de grupo que cumple las propiedades: asociativa, identidad, inversa y grupo cerrado. Además, ambas estructuras son infinitamente diferenciables. Algunos ejemplos de grupos de Lie con sus respectivas operaciones son los siguientes: el espacio Euclídeo R^n con la suma ordinaria de vectores, los números reales no nulos R^x con la multiplicación, los números reales positivos $R>0$ con la multiplicación y también grupos de matrices invertibles con la operación de producto de matrices, como el grupo especial ortogonal SO(3) de todas las rotaciones en el espacio tridimensional. En el contexto de este proyecto, se utilizará el grupo SE(3), cuyos elementos son matrices de transformación homogéneas 4x4 donde se definen la posición y orientación en el espacio de cada elemento.

➤ Un álgebra de Lie: A es un espacio vectorial sobre un cierto cuerpo F junto con una operación binaria $[\cdot, \cdot] : A \times A \rightarrow A$, llamado el corchete de Lie (Lie bracket), que satisface las propiedades siguientes:

- es bilineal, es decir, $[a x + b y, z] = a [x, z] + b [y, z]$ y $[z, a x + b y] = a [z, x] + b [z, y]$ para todo a, b en F y todo x, y, z en A.
- satisface la identidad de Jacobi, es decir, $[[x, y], z] + [[z, x], y] + [[y, z], x] = 0$ para todo x, y, z en A.
- $[x, x] = 0$ para todo x en A.

Se puede observar que la primera y tercera propiedad juntas implican $[x, y] = - [y, x]$ para todo x, y en A ("anti-simetría"). Además, la multiplicación representada por el corchete de Lie no es, en general, asociativa, es decir, $[[x, y], z]$ no es necesariamente igual a $[x, [y, z]]$. A cada grupo de Lie, podemos asociar un álgebra de Lie que captura totalmente la estructura local del grupo, por lo menos si el grupo de Lie es conexo. Por ejemplo, el espacio euclídeo R^3 se convierte en un álgebra de Lie con el corchete de Lie dado por el producto vectorial. En este trabajo se usará el grupo se(3), que captura la estructura de SE(3). En éste grupo están incluidos los *screws* y los *trists* que serán presentados posteriormente.

2.2 Grupo de Lie especial euclídeo SE(3) y álgebra se(3)

El Grupo Especial Euclídeo SE(3), es el Grupo de Lie de dimensión seis, correspondiente a las transformaciones en R^3 . La representación del movimiento de un sólido rígido en el espacio Euclídeo tridimensional es el espacio de matrices 4x4 de transformación homogénea g, pertenece al Grupo de Lie SE(3). Entonces, la posición y orientación de un sólido rígido queda descrita por un elemento del SE(3), consistente en una matriz de la forma g (1), donde T representa el vector de translación y D la matriz de rotación espacial. Cabe destacar que T es un vector de dimensión 3, mientras que D es una matriz 3x3, 0 representa un vector de dimensión 3 compuesto por 0 y 1 es la escala.

$$g = \begin{bmatrix} D & T \\ 0 & 1 \end{bmatrix} \in SE(3) / SE(3) = \{(D, T) : T \in R^3, D \in SO(3)\} \in R^{3 \times 3} \quad (1)$$

El grupo SE(3) tiene asociada el álgebra de Lie se(3), que puede ser identificada con la matriz ξ^\wedge llamada *twists*, dadas por (2,3), junto con la operación *Lie Bracket* dada por (4).

$$\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2)$$

$$\xi^\wedge = \begin{bmatrix} \omega^\wedge & v \\ 0 & 1 \end{bmatrix} \in se(3) / se(3) = \{(v, \omega^\wedge) : v \in R^3, \omega^\wedge \in so(3)\} \in R^{3 \times 3} \quad (3)$$

$$[\xi_1^\wedge, \xi_2^\wedge] = \xi_1^\wedge \xi_2^\wedge - \xi_2^\wedge \xi_1^\wedge \quad (4)$$

Donde ω es el vector de dirección del *twist* y v el producto de $-\omega$ por un punto r perteneciente al eje del *twist*. Las matrices antisimétricas de la forma ω^\wedge (5), son el álgebra de Lie so(3), correspondiente al Grupo Ortogonal Especial de Lie SO(3), que representa todas las rotaciones en el espacio tridimensional. Se observa que las matrices de so(3), no son sino una transformación que convierte el producto vectorial en R^3 en un producto matricial (5), por lo que dado un vector ω es trivial la operación para

convertirlo en matriz ω^\wedge . La operación inversa consiste en dada una matriz ω^\wedge , obtener el vector asociado ω .

$$\omega^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} / \forall \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \wedge v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \Rightarrow \omega \times v = \omega^\wedge v \quad (5)$$

Se puede extraer del álgebra de Lie (del *twist* ξ^\wedge que es una matriz 4x4), el vector de seis dimensiones que parametriza el *twist*, este es ξ . Se hará con el sencillo operador (6 (a)). La operación inversa se ejecuta con el operador (6(b)).

$$(\xi^\wedge)^\vee = \begin{bmatrix} \omega^\wedge & v \\ 0 & 0 \end{bmatrix}^\vee = \begin{bmatrix} v \\ \omega \end{bmatrix} = \xi \quad (6(a))$$

$$(\xi)^\wedge = \begin{bmatrix} v \\ \omega \end{bmatrix}^\wedge = \begin{bmatrix} \omega^\wedge & v \\ 0 & 0 \end{bmatrix} = \xi^\wedge \quad (6(b))$$

La transformación exponencial, como ya se ha comentado anteriormente, es la principal conexión entre el grupo de Lie SE(3) y su correspondiente álgebra se(3) (por cada g perteneciente a SE(3), existe un ξ^\wedge perteneciente a se(3), tal que $\exp(\xi^\wedge) = g$). Dado un determinado *twist* ($\xi = (v, \omega)$), y una determinada magnitud θ del movimiento, la fórmula exponencial vendrá dada por (7). De esta forma, la transformación exponencial para un *twist*, equivale a una transformación homogénea y representa el movimiento de un sólido rígido.

$$e^{\xi^\wedge \theta} = \begin{bmatrix} e^{\omega^\wedge \theta} & (I - e^{\omega^\wedge \theta})(\omega \times v) + \omega \omega^\wedge v \theta \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} D & T \\ 0 & 1 \end{bmatrix} \in SE(3); \omega \neq 0 \quad (7)$$

$$e^{\xi^\wedge \theta} = \begin{bmatrix} I & v \theta \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} D & T \\ 0 & 1 \end{bmatrix} \in SE(3); \omega = 0$$

Se puede invertir la operación exponencial para una matriz de transformación homogénea, aplicando la operación logaritmo de matrices (8), para obtener el *twist* ξ^\wedge , y la magnitud θ del movimiento correspondiente.

$$\ln(e^{\omega^\wedge \theta}) = \ln \begin{bmatrix} e^{\omega^\wedge \theta} & (I - e^{\omega^\wedge \theta})(\omega \times v) + \omega \omega^\wedge v \theta \\ 0 & 1 \end{bmatrix} = \ln \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = \xi^\wedge \theta \quad (8)$$

En las fórmulas anteriores, la matriz de rotación D se calcula como la exponencial de una matriz ω^\wedge de so(3), que no es sino la representación de la rotación de un cuerpo en el espacio R^3 alrededor de un eje ω una cierta magnitud (ángulo) θ , que se puede calcular de una forma eficiente con la conocida fórmula de Rodrigues (9).

$$e^{\omega^\wedge \theta} = I + \omega \omega^\wedge \text{sen} \theta + \omega^\wedge^2 (1 - \text{cos} \theta) \quad (9)$$

2.3 Movimientos screw y twist

El movimiento de rotación sobre un eje con una traslación a lo largo de ese mismo eje se llama *screw*, por el parecido que tiene dicho movimiento con el realizado por un tornillo. Se comprende, que un *twist* está relacionado con esta geometría, de manera que un *twist* es la versión infinitesimal de un *screw*. Es importante destacar la siguiente propiedad de los *screws*:

- No afectación del giro sobre su propio eje: Si el movimiento *screw* es solo rotativo, entonces la aplicación del mismo a un punto r situado en el eje del *twist*, no produce ningún efecto sobre ese punto, lo que matemáticamente implica que el

producto de la exponencial de un *twist* de rotación sobre un punto de su propio eje es igual al propio punto. Esta propiedad es de utilidad para simplificar el problema cuando se emplee POE.

2.4 Problema PADEN-KAHAN-UNO, rotación alrededor de un eje

Este problema geométrico canónico resuelve de forma directa y cerrada el problema cinemático inverso que se plantea cuando se tiene un movimiento de giro alrededor de un eje y se quiere conocer cuál debe ser el ángulo necesario para obtener dicho movimiento. Este método solo contempla el problema de la traslación de un punto desde una posición a otra a través de una rotación alrededor de un eje. Para relacionar dicho problema con la orientación de la herramienta debe utilizarse POE, como se verá más adelante.

El problema consiste en encontrar el valor del ángulo θ a girar alrededor de un eje ω para mover un punto p hasta llevarlo a una posición dada por el punto k (Figura 7).

El problema se expresa en forma de Lie según la ecuación (10), que es la exponencial de un *twist* correspondiente a un giro de valor θ aplicado al punto p es igual al punto k . La solución viene dada por la ecuación que se muestra en (11).

$$e^{\xi^\wedge \theta} \cdot p = k \wedge \xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times r \\ \omega \end{bmatrix} \quad (10)$$

$$\theta = a \tan \left[\omega^T (u \times v'), u^T \cdot v' \right] \wedge \begin{cases} u' = u - \omega \omega^T u \\ v' = v - \omega \omega^T v \end{cases} \quad (11)$$

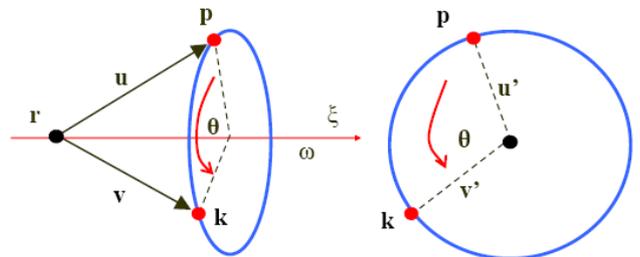


Figura 7: Problema de Paden-Kahan Uno – Rotación alrededor de un eje

2.5 Fórmula del producto de exponenciales POE

Los métodos de Paden-Kahan resuelven el problema de la traslación de un punto, pero para relacionar de forma conjunta la orientación y la traslación es necesario combinarlos con Fórmula del producto de exponenciales (POE). Esta fórmula tiene el objetivo de describir de una forma geométrica, la cinemática de un sólido rígido.

La fórmula del POE, cuya ecuación es (12), es una descripción para la cinemática directa de un sólido rígido que sufre una concatenación de n movimientos *screws* (caracterizados por sus

correspondientes *twists* ξ y magnitudes θ), una vez que se elige una posición de referencia $g(0)$ para un sistema coordinado. En el caso de la pierna del robot, la matriz $g(\theta)$ es la posición y orientación del pie tras aplicar n rotaciones a la matriz $g(0)$, siendo esta la matriz de transformación inicial de dicho pie.

$$g(\theta) = \prod_{i=1}^n e^{\xi_i \wedge \theta_i} \cdot g(0) \quad (12)$$

La transformación exponencial puede ser interpretada como un operador que relaciona puntos de un sólido rígido, desde sus coordenadas iniciales a sus nuevas coordenadas en el sistema de referencia, después de que el sólido haya sufrido un movimiento de tipo *screw*. Como una generalización de la transformación exponencial, el POE representa el sistema de referencia tras una concatenación de *screws* que han afectado al movimiento del sólido rígido. El POE se utiliza para la formulación de los problemas cinemáticos tanto directos como inversos.

3. Implementación de un simulador para el robot Robonova-I

En este apartado se discute el simulador para el robot Robonova-I implementado, utilizando las herramientas matemáticas introducidas previamente. El entorno elegido para este fin ha sido Matlab [33], por estar especialmente orientado a cálculos matriciales y por disponer del toolbox ROBOMAN [31], destinada a robots humanoides usando Algebra de Lie. Debe tenerse en cuenta que los motores tienen el siguiente rango de actuación:

$$\begin{aligned} \theta_1 &\in [-90^\circ, 20^\circ] & (18) \\ \theta_2 &\in [-90^\circ, 90^\circ] \\ \theta_3 &\in [-100^\circ, 30^\circ] \\ \theta_4 &\in [-60^\circ, 110^\circ] \\ \theta_5 &\in [-15^\circ, 45^\circ] \end{aligned}$$

Para la visualización de los resultados se ha implementado las funciones siguientes:

- **DibujarInicial:** Esta función dibuja el sistema inicial en 3 dimensiones y las proyecciones sobre los ejes X-Z e Y-Z. Hace uso de las funciones Representar, DibujarEje, Dibujar3d y DibujarEje3d. Utiliza como parámetros la secuencia de motores inicial del robot a representar (Inicial), la matriz de transformación inicial del pie (MTI), la dirección del eje de giro (DE) y un punto de éste (PE) y el rango de representación de las gráficas.

```
function DibujarInicial(Inicial, (19)
MTI, DE, PE, Rango)
```

- **DibujarMov:** Mediante esta función se representa el movimiento a través de un eje tanto en los planos X-Y-Z como en sus proyecciones sobre X-Y e Y-Z. Hace uso de las funciones Representar y Dibujar3d. A los parámetros de la función anterior se le incluyen la secuencia de motores después del giro (Final), la matriz de transformación final del pie (MTF), la posición de los motores sobre los que se van a efectuar giros antes y después del giro representado

(MotoresInicial y MotoresFinal) y el mensaje a representar, en caso que fuese necesario.

```
function DibujarMov(Inicial, Final, (20)
MTI, MTF, DE, PE, MotoresInicial,
MotoresFinal, Rango, Mensaje)
```

- **Dibujar3d:** Con esta función se dibujan las representaciones del sistema sobre los planos X-Y-Z. Usa la función DibujarEje3d. Se añade el parámetro Color, una variable booleana para indicar el conjunto de colores en los que se va a representar, siendo 0 el conjunto azul-amarillo, usado para los estados iniciales del sistema, y 1 el conjunto rojo-verde, empleado para los estados finales del movimiento.

```
function Dibujar3d(Ejes, Titulo, (21)
Matriz, Direccion, Punto, Motores,
Rango, Color)
```

- **DibujarEje3d:** dibuja un conjunto de ejes adicionales a la representación sobre los planos X-Y-Z. El parámetro Ejes es una matriz de direcciones con los vectores de dirección de todos los nuevos ejes a representar, mientras que Puntos es una matriz de puntos pertenecientes a los ejes.

```
function DibujarEje3d(Eje, Punto, (22)
Rango)
```

- **Representar:** Dibuja la proyección del sistema sobre un plano. Utiliza la función DibujarEje. Utiliza como parámetros dos matrices con las coordenadas de los puntos que se van a representar (X e Y), el título de la gráfica (Titulo), las etiquetas de los ejes (EtiquetaX y EtiquetaY), la matriz de transformación del pie (Matriz), la dirección del eje de giro (Direccion), un punto del eje (Punto), Los ejes sobre que definen el plano (EjeX,y EjeY), expresados como un número entero (1-eje X, 2-eje Y y 3-eje Z), la posición de los motores (Motores), el rango de representación de la gráfica (Rango), el conjunto de colores (Color) expresados de igual que en Dibujar3d, el mensaje a mostrar (Mensaje) y una variable booleana, Leyenda, que indica si se quiere mostrar la leyenda.

```
function Representar(X, Y, Titulo, (23)
EtiquetaX, EtiquetaY, Matriz,
Direccion, Punto, EjeX, EjeY,
Motores, Rango, Color, Mensaje,
Leyenda)
```

- **DibujarEje:** Dibuja un eje en una representación sobre un plano. Utiliza como parámetros el vector de dirección y un punto del eje (Eje, Punto), los ejes de los planos (EjeX,y EjeY) expresados como un número entero (1-eje

X, 2-eje Y y 3-eje Z) y el rango de representación de las gráficas (Rango).

```
function DibujarEje(Eje, Punto, EjeX, (24)
EjeY, Rango)
```

3.1 Adaptación de los conceptos matemáticos al robot humanoide Robonova-I

En este apartado se van a especificar las herramientas matemáticas planteadas en el apartado anterior para el robot humanoide Robonova-I. Se está trabajando en SE(3), donde todos los elementos tienen la forma (1). Esto implica que todas las posiciones deben estar definidas de esta forma. Así, la posición inicial del pie tendría la siguiente forma, expresando las unidades en centímetros.

$$g = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -19 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (13)$$

En la figura 8 se representan dichas medidas en el robot.



Figura 8: Posición inicial de la herramienta

Este robot tiene 5 GDL en cada pierna, que tienen asociados sus correspondientes *twists*. Los valores del vector de dirección del giro ω y del punto r perteneciente al eje son los siguientes:

$$r_1 = \begin{bmatrix} 2.5 \\ 0 \\ 0 \end{bmatrix} \wedge \omega_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \wedge r_2 = \begin{bmatrix} 2.5 \\ 0 \\ -4 \end{bmatrix} \wedge \omega_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \wedge r_3 = \begin{bmatrix} 2.5 \\ 2.9 \\ -10 \end{bmatrix} \wedge \omega_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (14)$$

$$r_4 = \begin{bmatrix} 2.5 \\ 0 \\ -15 \end{bmatrix} \wedge \omega_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \wedge r_5 = \begin{bmatrix} 2.5 \\ 0 \\ -19 \end{bmatrix} \wedge \omega_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

En la siguiente figura se representan los ejes de giro en dirección X (a) e Y (b).

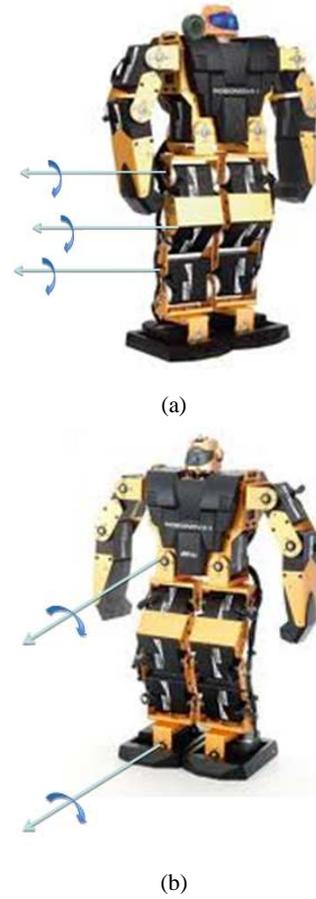


Figura 9: Ejes de los wists del robot Robonova1

Cada *twist* está parametrizado por sus valores ω y v . El valor de ω es conocido, y para calcular v se realiza en producto escalar entre $-\omega$ y r .

$$v_n = -\omega_n \times r_n \quad (15)$$

Para obtener el valor del *twist* ξ^\wedge se utiliza la operación (6(b)).

$$\xi^\wedge = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (16)$$

$$\xi^\wedge_n = \begin{bmatrix} \omega^\wedge_n & v_n \\ 0 & 0 \end{bmatrix} \quad (17)$$

Este *twist* es un álgebra de Lie de $se(3)$. Para obtener matriz de la forma (1) resultante de realizar un giro de magnitud θ a través del *twist*, se hace uso de la función exponencial, mediante la fórmula (7). Esta nueva matriz $e^{\xi^\wedge \theta}$ conseguida es un elemento de SE(3).

4. Resultados

En este apartado se mostrarán algunos resultados del movimiento de la pierna del robot en uno, dos y tres ejes describiendo las funciones implementadas en Matlab y la formulación matemática asociada.

4.1 Movimiento de la pierna del robot a través de un eje

En esta sección se va resolver un sistema formado por una sola articulación usando solamente Paden-Kahan Uno (PK1) y la combinación de PK1 y POE para comparar los resultados obtenidos. El problema planteado consiste en una rotación a través del *twist* 1, cuyo eje de giro está representado en la figura 10.



Figura 10: Representación del eje de giro del *twist* 1

El valor del punto del eje r y la dirección de giro ω es el siguiente, expresadas las unidades en centímetros:

$$r = \begin{pmatrix} 2.5 \\ 0 \\ 0 \end{pmatrix} \wedge \omega = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (25)$$

Se puede obtener el *twist* ξ que describe el movimiento a través de un giro sobre este eje. Para ello hay que obtener los valores de ω y v que parametrizan el *twist*. El valor de ω es conocido, mientras que el de v se calcula aplicando la fórmula (15). Este *twist* es un Algebra de Lie de $se(3)$. Para su implementación en Matlab se dispone de la función `revolutetwist` de ROBOMAN que, a partir de r y ω , devuelve una matriz de la forma (2). La definición de dicha función es la siguiente, donde q y w son r y ω respectivamente.

$$\text{function } x = \text{revolutetwist}(q, w) \quad (26)$$

4.1.1 Movimiento de la pierna del robot a través de un eje usando exclusivamente Paden-Kahan Uno

En este apartado se ha implementado el método de Paden-Kahan Uno (PK1) para el robot Robonova-I. Este método resuelve el problema de obtener el ángulo necesario para la traslación de un punto de una posición p a otra k a través un eje. Para ello se ha desarrollado la siguiente función :

$$\text{function } PK1(PI, PF, eje, Restricciones) \quad (27)$$

Los parámetros que utiliza son el punto inicial y final (PI y PF), el eje sobre el que se producirá el giro, expresado por un número entero entre 1 y 5, ordenados de forma descendente desde la cadera y la variable booleana *Restricciones*, que indica si

desea que se usen las restricciones de los motores. En la figura 11 se representa la posición inicial de la pierna y el eje sobre el que se realizará el giro. El problema planteado consiste en trasladar un punto p situado en la posición inicial del pie hasta el punto k siguiente:

$$p = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix} \wedge k = \begin{pmatrix} 21.5 \\ 0 \\ 0 \end{pmatrix} \quad (28)$$

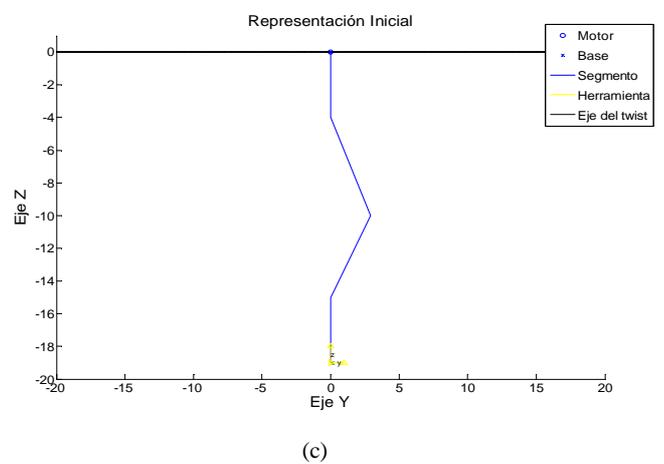
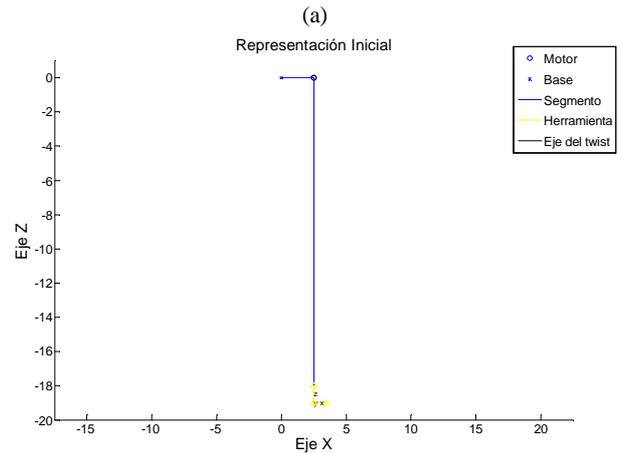
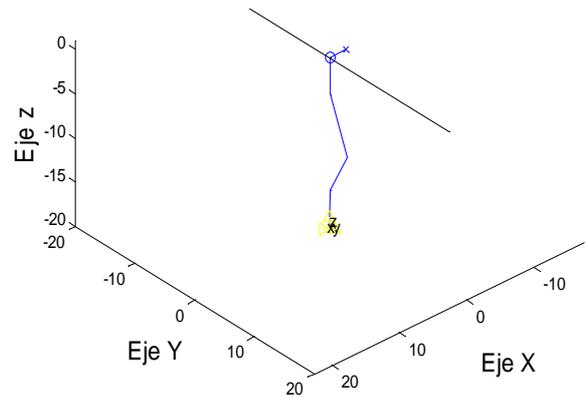


Figura 11: Proyección sobre lo planos X-Z e Y-Z inicial

Con estos datos se puede aplicar Paden-Kahan Uno (11) y obtener el ángulo deseado para trasladar la herramienta hasta la posición deseada. Esto se hace con la función `padenkahanone` que usa como parámetros el vector expresado como un vector 6x1 obtenido en (26), el punto p (p) y el punto k (q) y devuelve el ángulo en formato [-180°, 180°].

$$\text{function } t = \text{padenkahanone}(x, p, q) \quad (29)$$

Tras aplicar esta función, se obtiene el siguiente valor para el ángulo de giro necesario para provocar el movimiento deseado.

$$u = \begin{pmatrix} 0 \\ 0 \\ -19 \end{pmatrix} \wedge v = \begin{pmatrix} 19 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \theta = -90^\circ \quad (30)$$

Se obtiene un giro de -90° , es decir, 90° en sentido contrario a las agujas del reloj. Este giro está dentro del rango de actuación del motor. En la figura 12 se puede observar la representación del giro sobre el plano X-Z.

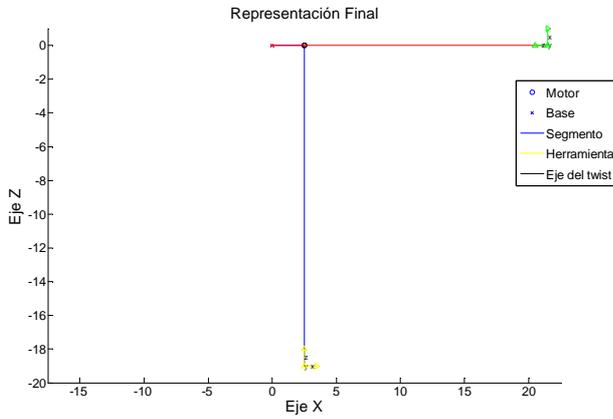


Figura 12: Representación del giro sobre Y-Z

Para comprobar los resultados se aplica POE (12) para la cinemática directa, siendo $g(\theta)$ la matriz de transformación tras aplicar el giro y $g(0)$ la matriz de transformación inicial del pie.

$$g(\theta) = e^{\xi^{\wedge} \theta} \cdot g(0) \quad (31)$$

Para implementar esto, primero hay que calcular la exponencial de ξ^{\wedge} , $e^{\xi^{\wedge} \theta}$, aplicando la fórmula (7). Para ello, se dispone de la función de ROBOMAN `twistexp`, a la que se le pasan como parámetros el vector (x) obtenido en (26) y el ángulo (t) obtenido en (30) expresado en radianes.

$$\text{function } g = \text{twistexp}(x, t) \quad (32)$$

Tras aplicar esta función se obtiene la siguiente matriz de transformación para el giro de un ángulo θ a través de un *twist* ξ^{\wedge} . Esta matriz es un elemento de SE(3).

$$e^{\xi^{\wedge} \theta} = \begin{pmatrix} 0 & 0 & -1 & 2.5 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -2.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (33)$$

Con estos datos se puede aplicar POE (31) para obtener la posición y orientación del pie tras aplicar el giro.

$$g(\theta) = e^{\xi^{\wedge} \theta} \cdot g(0) = \begin{pmatrix} 0 & 0 & -1 & 21.5 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (34)$$

Se verifica que se alcanza la posición deseada. El tiempo empleado para los cálculos ha sido de 0.0156 segundos, un resultado muy eficiente para sistemas en tiempo real. En la figura 13 se comparan los resultados obtenidos con el sistema real.

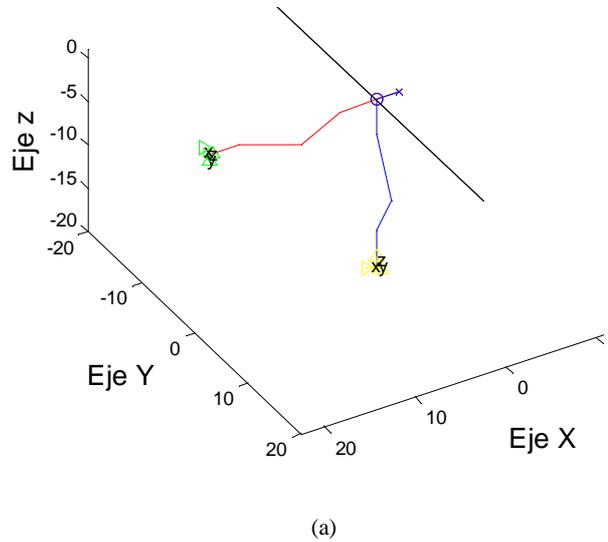


Figura 13: Representación del giro

4.1.2 Movimiento de la pierna del robot a través de un eje usando Paden-Kahan Uno especificando una posición no alcanzable

El objetivo de este ejemplo es comprobar como reacciona el método cuando existe un error en la especificación de la traslación del pie. Se ha establecido una posición no alcanzable, siendo ésta la propuesta en el apartado anterior pero incrementando 1 cm. en dirección del eje X la posición final, la cual no se puede conseguir con los *twists* que se disponen. El punto inicial p es el del apartado 4.1.1. (28), mientras que el valor de k es el siguiente:

$$k = \begin{pmatrix} 22.5 \\ 0 \\ 0 \end{pmatrix} \quad (35)$$

Tras aplicar PK1 (11) mediante la función (29) con éste nuevo valor de k se obtiene el ángulo para el *twist*.

$$u = \begin{pmatrix} 0 \\ 0 \\ -19 \end{pmatrix} \wedge v = \begin{pmatrix} 19 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \theta = -90^\circ \quad (36)$$

Se observa que se obtiene el mismo ángulo que en el apartado 4.1.1. (30). PK1 aporta el ángulo que se debe girar un *twist* para poner un punto p en la dirección de otro punto k . Si las posiciones están bien definidas, este giro provocará que coincidan dichos puntos pero, si se establece una posición errónea, PK1 traslada el punto p hasta la posición más próxima posible al punto k especificado, que es el punto alcanzable más próximo en la dirección de k . Una vez obtenido el ángulo, al aplicar POE para el problema cinemático directo (31), obteniéndose la posición y orientación real del pie tras sufrir el giro.

4.1.3 Movimiento de la pierna del robot a través de un eje usando Paden-Kahan Uno en combinación con POE

En este apartado se resolverá el problema de la traslación del pie a través de un giro sobre el *twist* 1 aplicando PK1 en combinación con POE para comparar los resultados con los del apartado 4.1.1. Con este fin se ha desarrollado la función `PK1conPOE`, que utiliza como parámetros los vectores de traslación inicial y final (VTI y VTF), el eje sobre el que se realizará el giro (eje), expresado como un número que indica la posición del motor en orden descendente desde la cintura, las matrices de rotación inicial y final (MTI y MTF) y la variable booleana *Restricciones* que indica si se desean usar las restricciones de los motores.

```
function PK1conPOE(VTI, VTF, eje, MTI, (37)
MTF, Restricciones)
```

Aplicando POE (13), el problema inicial es el siguiente:

$$g(\theta) = e^{\xi \wedge \theta} \cdot g(0) \quad (38)$$

$g(0)$ es un elemento de SE(3), por lo que debe tener la forma (1). El vector de traslación T se corresponde con el punto k deseado, mientras que el valor de D es la matriz de rotación inicial.

$$g(0) = \begin{bmatrix} D & T \\ 0 & 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -19 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (39)$$

Para implementar esta conversión de una matriz 3x3 y un punto a una matriz 4x4 se dispone de la función `rptohomogeneous` de ROBOMAN, donde r se corresponde con D y p con T .

```
function r = rptohomogeneous(r,p) (40)
```

La matriz de transformación final, $g(\theta)$, al igual que $g(0)$, es un elemento de SE(3) y debe tener la forma (1). Se conoce la posición T a la que se desea trasladar el pie, el punto k , pero al no tener ningún GDL situado en dicho pie, no se puede corregir la orientación tras sufrir el giro. Esto implica que debe conocerse la orientación antes de calcular el ángulo que se va a girar, lo cual no es posible. En este caso se va a establecer una orientación D igual a la inicial, sabiendo que ésta no es alcanzable. La matriz $g(\theta)$ obtenida mediante (40) es la siguiente:

$$g(\theta) = \begin{bmatrix} D & T \\ 0 & 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 21.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (41)$$

Para obtener el valor de k , a partir de la fórmula del POE se pasa $g(0)$ al lado izquierdo y se aplica a ambos lados el punto p .

$$g(\theta) = e^{\xi \wedge \theta} \cdot g(0) \quad (42)$$

$$g(\theta) \cdot g(0)^I = e^{\xi \wedge \theta}$$

$$g(\theta) \cdot g(0)^I \cdot p = e^{\xi \wedge \theta} \cdot p$$

El lado izquierdo de la ecuación es un punto k calculable, por lo que se tiene una ecuación de la forma requerida para poder aplicar Paden-Kahan Uno (11).

$$k = e^{\xi \wedge \theta} \cdot p \rightarrow PK1 \quad (43)$$

$$k = g(\theta) \cdot g(0)^I \cdot p = \begin{pmatrix} 21.5 \\ 0 \\ 0 \end{pmatrix}$$

El valor de k es el mismo que utilizando exclusivamente Paden-Kahan Uno (29), lo que implica que se obtendrán los mismos resultados. Se puede observar que, aunque se ha especificado un valor para D imposible, cuando POE está formado por un solo giro, un error en la especificación de la orientación de la herramienta no deriva en un error en la traslación de la misma. El tiempo de cómputo ha sido de 0.0313 segundos. Éste tiempo de cómputo dobla el obtenido usando solamente PK1, debido a las operaciones necesarias para obtener el punto k .

4.1.4 Movimiento de la pierna del robot a través de un eje usando Paden-Kahan Uno en combinación con POE especificando una posición no alcanzable

En este apartado se estudia cómo reacciona PK1 en combinación con POE cuando se establece una posición no alcanzable. El ejemplo es similar al del apartado 4.1.2, pero incluyendo POE. El

punto de destino del apartado anterior se incrementa en 1 centímetro en dirección del eje X, haciendo el movimiento imposible. El valor de $g(0)$ es similar al caso anterior, mientras que $g(\theta)$ es:

$$g(\theta) = \begin{pmatrix} 1 & 0 & 0 & 22.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (44)$$

Al igual que el apartado previo, al no conocer la orientación del pie, D, se utiliza la orientación inicial. El nuevo valor de k, obtenido de igual forma que en dicho apartado (42, 43), es el siguiente:

$$k = g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 21.5 \\ 0 \\ 0 \end{pmatrix} \quad (45)$$

Nuevamente se vuelven a obtener los mismos resultados. Al igual que usando exclusivamente PK1, en caso de establecer una posición imposible, PK1 obtiene el ángulo necesario para acercar el punto final lo máximo posible al especificado, obteniendo el valor de k alcanzable en la dirección deseada al aplicar POE para el problema cinemático directo.

4.2 Movimiento de la pierna del robot a través de dos ejes

En este apartado se va a estudiar el caso del movimiento de la pierna del robot a través de dos giros realizados consecutivamente sobre dos ejes paralelos. Se pueden diferenciar el caso en el que uno de los motores está dedicado a la orientación de la herramienta y el que sendos *twists* están dedicados a la traslación del pie.. El primero de los casos resuelve el movimiento de la pierna del robot para el movimiento lateral, es decir, en el plano X-Z, mientras que el caso de dos motores destinados a la traslación es un paso previo necesario para implementar el movimiento sobre tres ejes, que modela el movimiento frontal, en el plano Y-Z. Para ambos supuestos, se ha implementado la función PK1POE2, que tiene un parámetro adicional a (37), la variable *Rekursivo*, que indica si se desea utilizar el algoritmo recursivo que se describirá en el apartado 4.2.3.

```
function PK1POE2(VTI, VTF, (46)
ejes, MTI, MTF, Restricciones,
Rekursivo)
```

4.2.1 Movimiento de la pierna del robot con dos motores a través del plano X-Z

En este apartado se va resolver el problema del movimiento lateral de la pierna derecha del robot humanoide. Al problema planteado en el apartado 4.1 se le añade un segundo giro provocado por el *twist* 5. Al estar esta última articulación situada en el pie y tener dirección de giro paralela al *twist* 1, se dedicará a la orientación del pie, buscando siempre que mantenga la orientación respecto al suelo. Éste último GDL, al no afectar a la traslación se puede considerar un GDL Virtual y separarlo del

problema de la traslación, ya que se sabe que siempre debe corregir el giro realizado por el *twist* 1, por lo que tendrán el mismo valor pero con signo contrario. En la figura 14 puede observarse el sistema inicial.

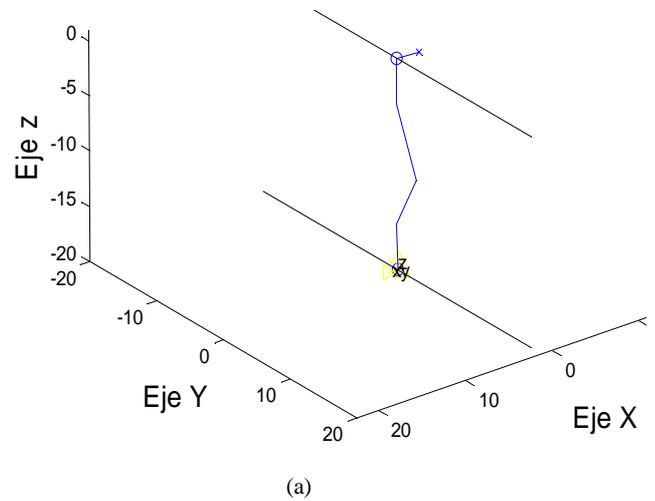
Este sistema está formado por los *twists* 1 y 5, definidos por los siguientes parámetros:

$$r_1 = \begin{pmatrix} 2.5 \\ 0 \\ 0 \end{pmatrix} \wedge \omega_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \wedge r_2 = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix} \wedge \omega_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (47)$$

En la figura 15 se representan las proyecciones sobre los planos X-Z e Y-Z en el estado inicial. Aplicando POE (12) se tienen las siguientes ecuaciones iniciales:

$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) \quad (48)$$

$$g(\theta) \cdot g(0)^T = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2}$$



(b)

Figura 14: Representación X-Y-Z Inicial

Aplicando POE (12) se tienen las siguientes ecuaciones iniciales:

$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) \quad (48)$$

$$g(\theta) \cdot g(0)^T = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2}$$

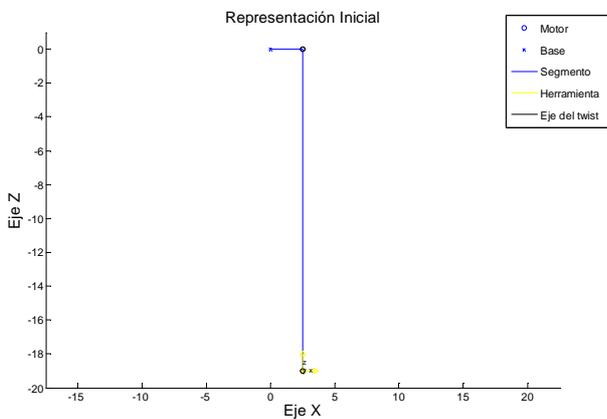
La matriz de transformación inicial es la misma que en el apartado 4.1.3. (41). Se busca que el pie mantenga en todo momento su orientación respecto al suelo, lo que implica que D no debe variar, mientras que T es la posición a la que se desea trasladar el pie.

$$g(\theta) = \begin{bmatrix} D & T \\ 0 & 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 21.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (49)$$

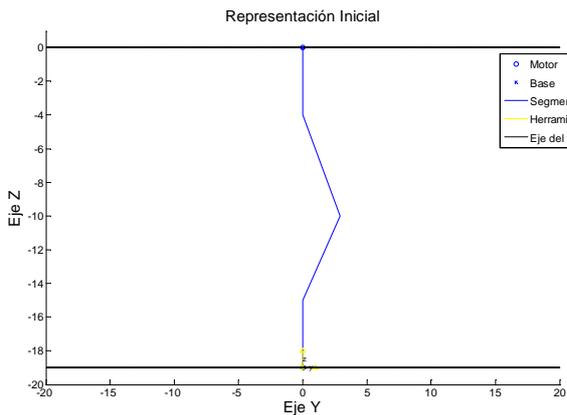
Según la propiedad 1 de los screw “No afección del giro sobre su propio eje”, si se aplica un punto p contenido en el eje 2, dicho punto no se verá afectado por el giro del eje, por lo que se puede eliminar de la ecuación del POE. De esta forma se tiene una ecuación similar a la del apartado 4.1.3. (42).

$$p = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix} \Rightarrow \quad (50)$$

$$g(\theta) \cdot g(0)^T \cdot p = e^{\xi_1 \wedge \theta_1} \cdot p$$



(a)



(b)

Figura 15: Proyección sobre Y-Z Inicial

La ecuación resultante tiene un lado izquierdo k conocido, lo que deja la ecuación en la forma (10) para poder aplicar PK1.

(51)

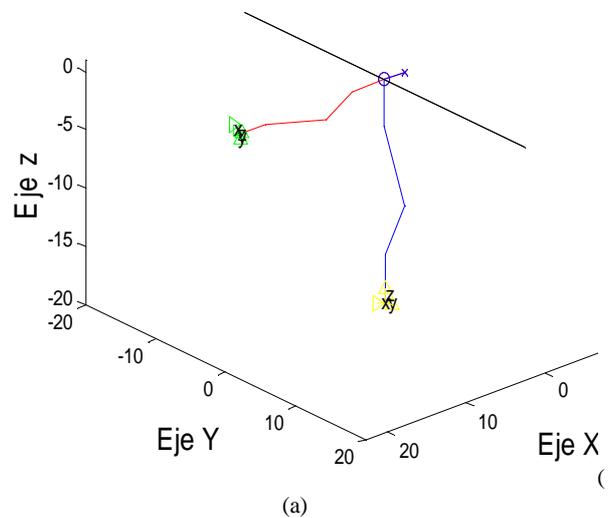
$$k = e^{\xi_1 \wedge \theta_1} \cdot p \rightarrow PK1 \Rightarrow$$

$$k = g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 21.5 \\ 0 \\ 0 \end{pmatrix}$$

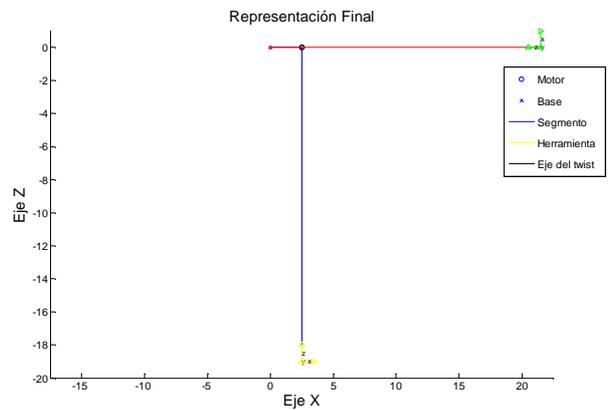
Con estos datos se aplica PK1 (11) mediante la función (29) para obtener el ángulo para el twist 1. El resultado ha sido un valor de $\theta_1 = -90^\circ$.

$$u = \begin{pmatrix} 0 \\ 0 \\ -19 \end{pmatrix} \wedge v = \begin{pmatrix} 19 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \theta_1 = -90^\circ \quad (52)$$

En la siguiente figura se puede ver el giro realizado, tanto en los planos X-Y-Z (a) como en su proyección sobre el plano X-Z (b).



(a)



(b)

Figura 16: Representación del giro 1

Para obtener el valor de θ_2 hay que aplicar a la ecuación un punto p que no esté contenido en el twist 2, ya que el valor de $e^{\xi_1 \wedge \theta_1}$ se calcula fácilmente usando (7,32).

$$p = \begin{pmatrix} 2.5 \\ 0 \\ -18 \end{pmatrix} \wedge e^{\omega_1 \theta_1} = \begin{pmatrix} 0 & 0 & -1 & 2.5 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -2.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (53)$$

Se pasa $g(0)$ y $e^{\xi_1 \wedge \theta_1}$ al lado izquierdo de la ecuación y se aplica a ambos lados el punto p .

$$e^{\xi_1 \wedge \theta_1} \cdot g(\theta) \cdot g(0)^T \cdot p = e^{\xi_2 \wedge \theta_2} \cdot p \quad (54)$$

El lado izquierdo de la ecuación es un valor k calculable, lo que deja la fórmula de la forma (10).

$$k = e^{\xi_1 \wedge \theta_1} \cdot p \rightarrow PK1 \Rightarrow \quad (55)$$

$$k = e^{\xi_1 \wedge \theta_1} \cdot g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 1.5 \\ 0 \\ -19 \end{pmatrix}$$

Con estos datos se vuelve a aplicar PK1 (11,29), obteniendo un valor de $\theta_2=90^\circ$.

$$u = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \wedge v = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \theta_2 = 90^\circ \quad (56)$$

Aplicando POE (12) se verifica que el resultado obtenido es el deseado:

$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) = \begin{pmatrix} 1 & 0 & 0 & 21.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (57)$$

En la figura 17 se representa el giro sobre los planos X-Y-Z.

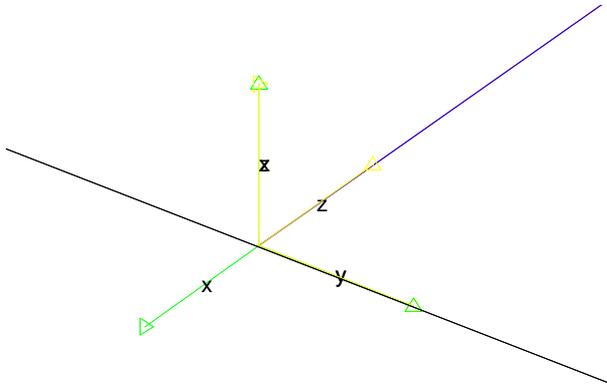


Figura 17: Representación del giro 2

El motor 5 no puede realizar el giro que se le pide, por lo que se aplica el giro máximo, $\theta_2=45^\circ$. Se vuelve a calcular $e^{\xi_2 \wedge \theta_2}$ con el nuevo ángulo y se aplica POE, alcanzando la siguiente posición:

$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) = \begin{pmatrix} 0.7071 & 0 & -0.7071 & 21.5 \\ 0 & 1 & 0 & 0 \\ 0.7071 & 0 & 0.7071 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (58)$$

En la figura 16 se observa el movimiento de la herramienta

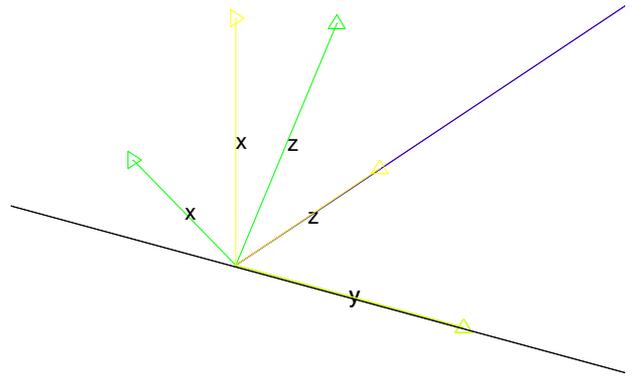


Figura 18: Representación del giro 2 sobre Y-Z



Figura 19: Posición alcanzada por el sistema real

El tiempo empleado para los cálculos ha sido 0.1406 segundos, lo cual resulta muy eficiente para un sistema en tiempo real. En la figura 19 se muestra la posición final del sistema real.

4.2.2 Movimiento de la pierna del robot con dos motores a través del plano X-Z especificando una posición no alcanzable.

En este apartado se va a comprobar el funcionamiento del método cuando se establece una posición objetivo que no puede alcanzarse con los GDL disponibles. Para ello, se extiende el problema 4.1.2. añadiéndole el *twist* 5 para la orientación del pie. El valor de $g(\theta)$ deseado es el siguiente:

$$g(\theta) = \begin{pmatrix} 1 & 0 & 0 & 22.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (59)$$

El punto p coincide con la posición inicial del pie. Tras operar de igual forma que en el apartado 4.2.1. (51), se obtienen los siguientes resultados:

$$p = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix} \Rightarrow \quad (60)$$

$$k = g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 21.5 \\ 0 \\ 0 \end{pmatrix}$$

Con estos datos se puede aplicar PK1 (11) obteniendo el siguiente ángulo:

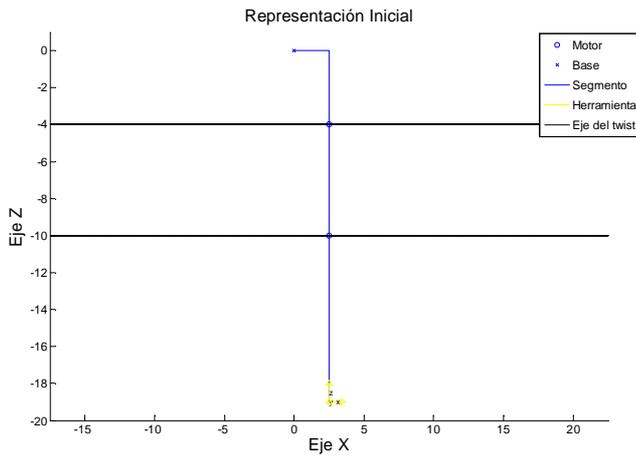
$$u = \begin{pmatrix} 0 \\ 0 \\ -19 \end{pmatrix} \wedge v = \begin{pmatrix} 19 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \theta_1 = -90^\circ \quad (61)$$

Se obtiene el mismo valor para el primer giro que en el caso anterior (52). Cómo este motor es el único dedicado a la traslación, el resto del problema será igual que en el apartado 4.2.1, comprobando que en este caso también se corrige el error en la especificación de la posición final.

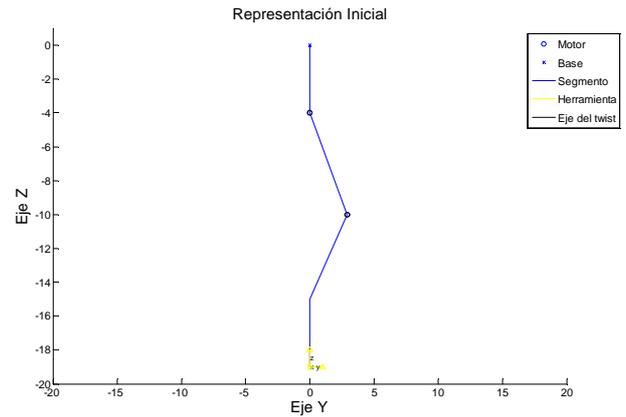
4.2.3 Movimiento hacia delante de la pierna del robot con dos motores a través del plano Y-Z

En este apartado se va a estudiar el sistema formado por dos motores usados para la traslación del pie en el plano Y-Z, es decir, los dos motores con dirección de giro paralela al eje X más cercanos a la cintura del robot. En la Figura 20 se representan las proyecciones de los ejes de giro. Este sistema está formado por los *twists* 2 y 3, definidos por los siguientes parámetros:

$$r_1 = \begin{pmatrix} 2.5 \\ 0 \\ -4 \end{pmatrix} \wedge \omega_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \wedge r_2 = \begin{pmatrix} 2.5 \\ 2.9 \\ -10 \end{pmatrix} \wedge \omega_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (62)$$



(a)



(b)

Figura 20: Proyecciones sobre Y-Z Inicial

La matriz de Transformación Inicial $g(0)$ es la misma que en los ejemplos anteriores (49), mientras que la matriz de rotación final utilizada será la matriz de rotación inicial, si bien el sistema no dispone de un GDL en la herramienta para orientarla. En la figura 21 se representa el sistema inicial en contraposición al sistema real. El problema planteado consiste en encontrar los ángulos que deben girar los *twists* para alcanzar $g(\theta)$.

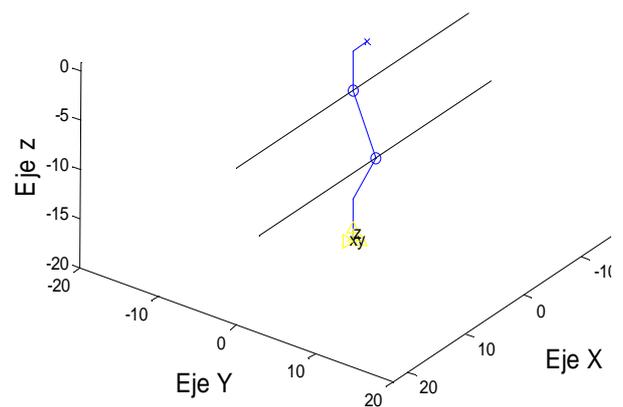
$$g(\theta) = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & -15 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (63)$$

Se parte de la siguiente ecuación, según el POE:

$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) \quad (64)$$

Al igual que en ejemplo anterior, se pasa $g(0)$ al lado izquierdo de la ecuación, dejando en el derecho sólo los *twists*.

$$g(\theta) \cdot g(0)^T = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} \quad (65)$$



(a)



Figura 21: Representación X-Y-Z Inicial

Se aplica a ambos lado un punto p contenido en el segundo eje, y según la propiedad de los *screws* “No afección del giro sobre el propio eje”, el punto no se ve afectado por el giro sobre el eje en el que está situado, por lo que el sistema se reduce a uno con un solo giro similar al apartado 4.2.1. De esta forma se pasa $g(0)$ al lado izquierdo y se aplica un punto $p=r_2$ a ambos lados de la ecuación quedando el siguiente sistema.

$$p = r_2 = \begin{pmatrix} 2.5 \\ 2.9 \\ -10 \end{pmatrix} \wedge \quad (66)$$

$$g(\theta) \cdot g(0)^T \cdot p = e^{\xi_1 \wedge \theta_1} \cdot p$$

Con estos datos se obtiene k , necesaria para aplicar PK1.

$$k = e^{\xi_1 \wedge \theta_1} \cdot p \rightarrow PK1 \Rightarrow \quad (67)$$

$$k = g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 2.5 \\ 7.9 \\ -6 \end{pmatrix}$$

Aplicando PK1 (11,29) se obtiene el siguiente valor para el primer giro.

$$u = \begin{pmatrix} 0 \\ 2.9 \\ -6 \end{pmatrix} \wedge v = \begin{pmatrix} 0 \\ 7.9 \\ -2 \end{pmatrix} \Rightarrow \theta_1 = 49.9972^\circ \quad (68)$$

En la siguiente figura puede observarse la representación de este giro:

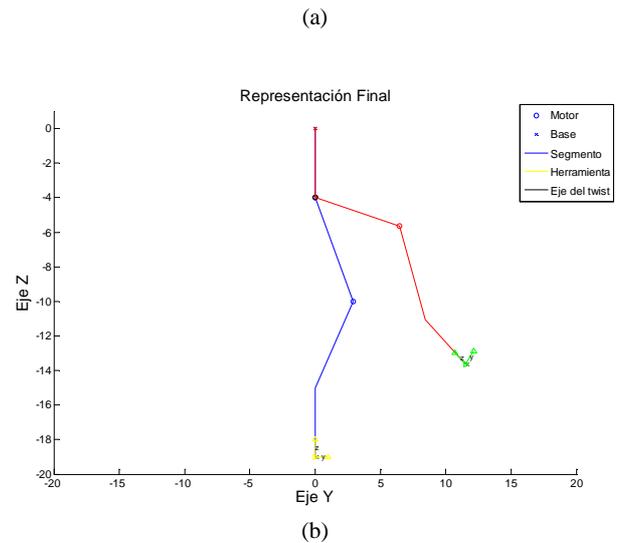
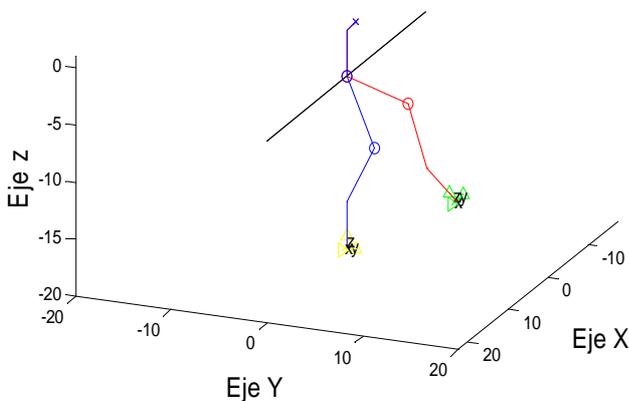


Figura 22: Representación del giro 1

La matriz de transformación para el *twist* 1 obtenida mediante (7,32) es la siguiente:

$$e^{\omega \wedge \theta_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.6428 & -0.766 & -3.0641 \\ 0 & 0.766 & 0.6428 & -1.4287 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (62)$$

Una vez obtenida $e^{\omega \wedge \theta_1}$ se opera para obtener el segundo giro. En este caso el valor de p es la posición del pie. Se pasa al lado izquierdo de la ecuación $e^{\omega \wedge \theta_1}$ y $g(0)$ y se aplica a ambos lados el punto p para obtener el valor de k .

$$p = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix} \quad (69)$$

$$e^{\xi_1 \wedge \theta_1} \cdot g(\theta) \cdot g(0)^T \cdot p = e^{\xi_2 \wedge \theta_2} \cdot p$$

$$k = e^{\xi_1 \wedge \theta_1} \cdot p \rightarrow PK1$$

$$k = e^{\xi_1 \wedge \theta_1} \cdot g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 2.5 \\ -5.212 \\ -14.9011 \end{pmatrix}$$

Con estos datos se puede aplicar PK1 (11,29) para obtener el valor de θ_2 y la matriz de transformación del *twist* 2 para este giro.

$$u = \begin{pmatrix} 0 \\ -2.9 \\ -9 \end{pmatrix} \wedge v = \begin{pmatrix} 0 \\ -8.112 \\ -4.9011 \end{pmatrix} \Rightarrow \theta_2 = -41.0003 \quad (70)$$

$$e^{\omega \wedge \theta_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.7547 & 0.6561 & 7.272 \\ 0 & -0.6561 & 0.7547 & -0.5504 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La representación de este segundo giro se puede observar en la figura 33.

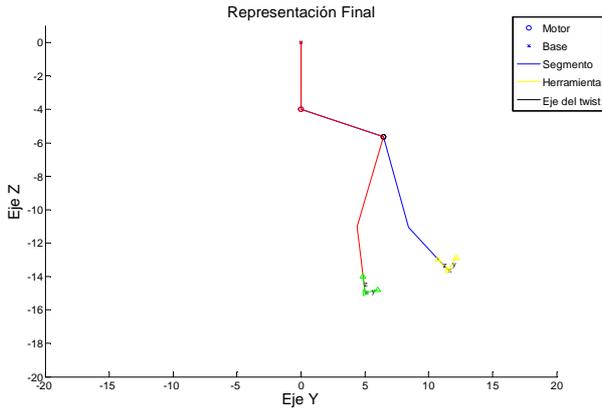


Figura 23: Representación del giro 1

Aplicando POE (12) se obtiene la $g(\theta)$ alcanzada:

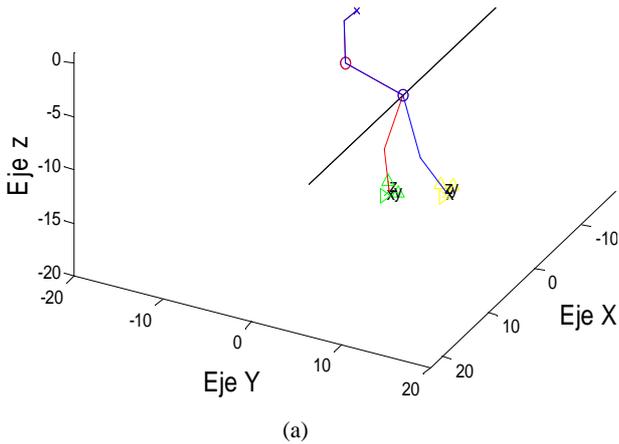
$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) = \begin{pmatrix} 0.9962 & 0 & 0.0872 & 2.5 \\ 0 & 1 & 0 & 5.0034 \\ -0.0872 & 0 & 0.9962 & -14.9783 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (71)$$

Se puede observar que no se alcanzan las especificaciones del problema, siendo el error obtenido el siguiente, expresado en centímetros el error absoluto y en porcentaje el error relativo:

$$\text{Error Absoluto} = 0.022 \quad (72)$$

$$\text{Error relativo} = 0.3432$$

El error relativo se ha obtenido dividiendo el error absoluto entre el desplazamiento realizado y multiplicado por 100. Se ha obtenido un error absoluto de 0.022 cm. y un error relativo del 0.3432%. Este error es prácticamente insignificante, ya que la precisión de los motores del robot es mucho menor. El tiempo de cómputo ha sido de 0.1406 segundos, un tiempo muy eficaz para el cómputo en tiempo real. En la figura 31 se compara el resultado obtenido y la posición alcanzada en el sistema real.



(b)

Figura 24: Representación final del sistema.

No se alcanzan los resultados deseados debido a que el POE relaciona la orientación y la traslación del pie. El robot no dispone de ningún motor en la herramienta para orientarla, por lo que no puede modificar la orientación de la misma sin variar la traslación. Ésto conlleva que debe conocerse la orientación del pie después de realizar los giros sobre sendos motores antes de calcular los ángulos que deben girar. En el apartado 4.2.5 se propone un algoritmo recursivo para solucionar el problema.

4.2.4 Movimiento hacia atrás de la pierna del robot con dos motores a través del plano Y-Z

Esta apartado complementa al anterior estudiando el caso del movimiento inverso al definido en el apartado 4.2.1. para observar el error obtenido. Los datos de los *twists* son similares al apartado anterior. El problema planteado consiste en encontrar los ángulos que deben girar los *twists* para alcanzar $g(\theta)$.

$$g(\theta) = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (73)$$

Se parte de la siguiente ecuación, según el POE:

$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) \quad (74)$$

Se pasa $g(0)$ al lado izquierdo de la ecuación, dejando en el derecho sólo los *twists*.

$$g(\theta) \cdot g(0)^T = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} \quad (75)$$

El punto p que se debe aplicar es el mismo que en el apartado anterior, $p=r_2$, necesario para simplificar el problema aplicando la fórmula ().

$$p = r_2 = \begin{pmatrix} 2.5 \\ 2.9 \\ -10 \end{pmatrix} \quad (76)$$

$$g(\theta) \cdot g(0)^T \cdot p = e^{\xi_1 \wedge \theta_1} \cdot p$$

El nuevo valor de k obtenido es el siguiente:

$$k = e^{\xi_1 \wedge \theta_1} \cdot p \rightarrow PK1 \quad (77)$$

$$k = g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 2.5 \\ -0.1 \\ -8 \end{pmatrix}$$

Aplicando PK1 () se obtiene el siguiente valor para el twist 1..

$$u = \begin{pmatrix} 0 \\ 2.9 \\ -6 \end{pmatrix} \wedge v = \begin{pmatrix} 0 \\ -0.1 \\ -4 \end{pmatrix} \Rightarrow \theta_1 = 27.2281^\circ \quad (78)$$

La matriz de transformación para este twist, obtenida mediante () es:

$$e^{\omega \wedge \theta_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.8892 & 0.4575 & 1.8301 \\ 0 & -0.4575 & 0.8892 & -0.4432 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (79)$$

El resultado de aplicar dicho giro sobre el twist 1 se puede observar en la figura 25. Para el twist 2, el valor de p es la posición del pie. Se pasa al lado izquierdo de la ecuación $e^{\omega \wedge \theta_1}$ y $g(0)$ y se aplica a ambos lados el punto p para obtener el valor de k.

$$p = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix} \quad (80)$$

$$e^{\xi_1 \wedge \theta_1} \cdot g(\theta) \cdot g(0)^T \cdot p = e^{\xi_2 \wedge \theta_2} \cdot p$$

$$k = e^{\xi_1 \wedge \theta_1} \cdot p \rightarrow PK1$$

$$k = e^{\xi_1 \wedge \theta_1} \cdot g(\theta) \cdot g(0)^T \cdot p = \begin{pmatrix} 2.5 \\ 3.2804 \\ -16.9321 \end{pmatrix}$$

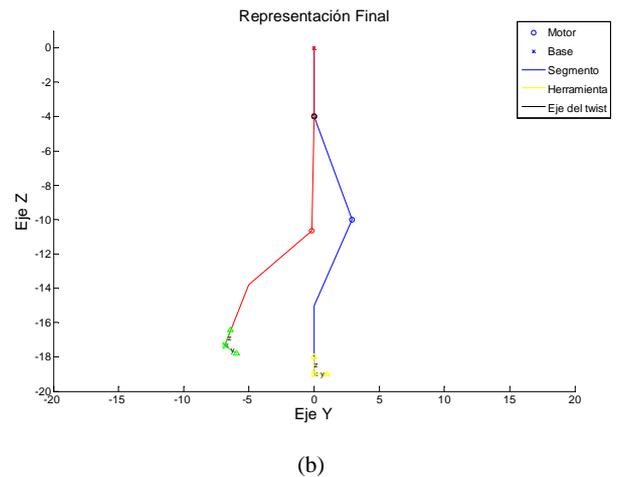
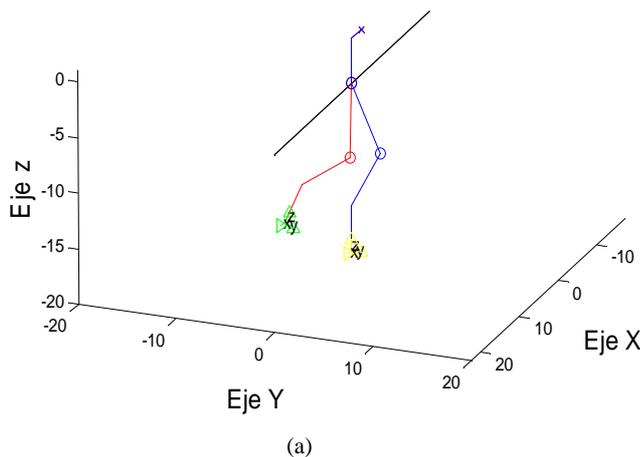


Figura 25: Representación del giro 1

Con datos obtenidos en (80) se aplica PK1 (11) para obtener el valor de θ_2 y la matriz de transformación del twist 2 para este giro.

$$u = \begin{pmatrix} 0 \\ 2.9 \\ -9 \end{pmatrix} \wedge v = \begin{pmatrix} 0 \\ 0.3804 \\ -6.9321 \end{pmatrix} \Rightarrow \theta_2 = 21.0008 \quad (81)$$

$$e^{\omega \wedge \theta_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9336 & -0.3584 & -3.3912 \\ 0 & 0.3584 & 0.9336 & -1.7036 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En la figura 26 se representa este segundo giro. Aplicando POE (12) se obtiene la posición y orientación de la herramienta.

$$g(\theta) e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} \cdot g(0) = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 0.9941 & 0.1085 & -4.0257 \\ 0 & -0.1085 & 0.9941 & -19.2943 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (82)$$

Se puede observar que no se alcanzan las especificaciones del problema, siendo el error obtenido el siguiente, expresado en centímetros y porcentaje:

$$\text{Error Absoluto} = 2.5132 \quad (83)$$

$$\text{Error relativo} = 69.7025$$

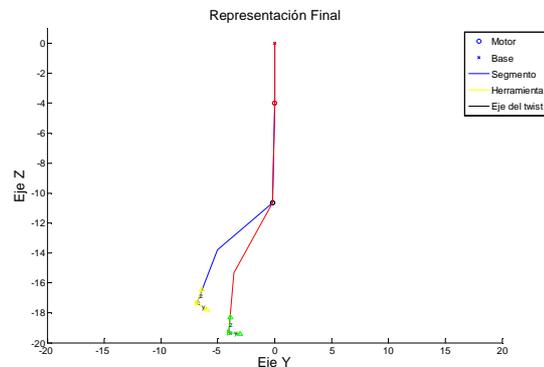


Figura 26: Representación del giro 2 sobre Y-Z

Se ha obtenido un error Absoluto de 2.5132 cm y un error relativo del 69.7025%. Este error es extremadamente alto. En la siguiente figura se compara el resultado obtenido y la posición alcanzada en el sistema real.

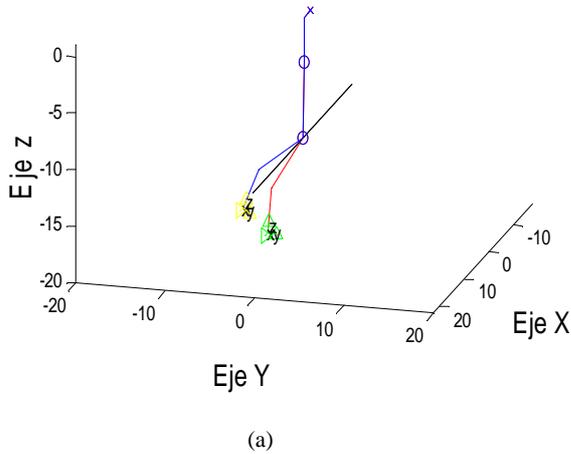


Figura 27: Representación final

4.2.5 Movimiento hacia delante de la pierna del robot con dos motores a través del plano Y-Z usando un algoritmo recursivo

En el apartado anterior produce un error en la traslación debido a que POE relaciona ésta con la orientación del pie, y dicha orientación se desconoce antes de realizar los movimientos. Sin embargo, se puede observar que el resultado del POE aporta una orientación más aproximada que la establecida inicialmente, por lo cual se ha implementado el POE de forma recursiva, en el que la matriz de transformación final utilizada en cada iteración está formada por la matriz de orientación obtenida en el paso anterior y el vector de traslación que se desea obtener.

$$g(\theta)_i = \begin{pmatrix} R_{i-1} & T \\ 0 & 1 \end{pmatrix} \quad (84)$$

El algoritmo implementado es el siguiente:

```
function [MTF] = POE2Recurso(MTI, (85)
MTF, PE2, VectorTwist1, VectorTwist2,
```

iteracion)

Este algoritmo devuelve una $g(\theta)$ con la orientación correcta para poder aplicar POE. Esta incluida en la función PK1POE2 y para utilizarla hay que definir la variable *Recurso* como 0. Utiliza como parámetros $g(0)$ y $g(\theta)$ (MTI y MTF), el punto del eje del *twist 2* (PE2), los vectores que definen ambos *twists* (VectorTwist1, VectorTwist2) y una variable de control (*iteración*) para evitar entrar en un bucle infinito.

Se ha elegido la misma posición del apartado 4.2.3 para intentar corregir el error producido mediante el algoritmo recursivo. De esta forma se obtiene el siguiente valor para θ_1 y para su matriz de transformación:

$$\theta_1 = 49.8071^\circ \quad (86)$$

$$e^{\omega_1 \theta_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.6454 & -0.7639 & -3.0555 \\ 0 & 0.7639 & 0.6454 & -1.4185 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En la siguiente figura se representa el giro y sus proyecciones sobre X-Z e Y-Z.

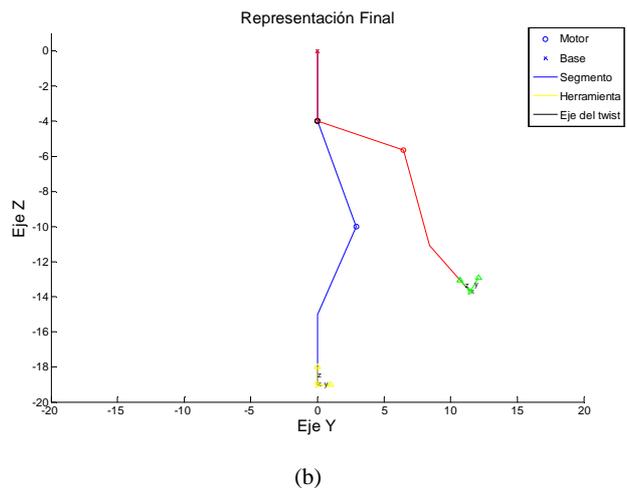
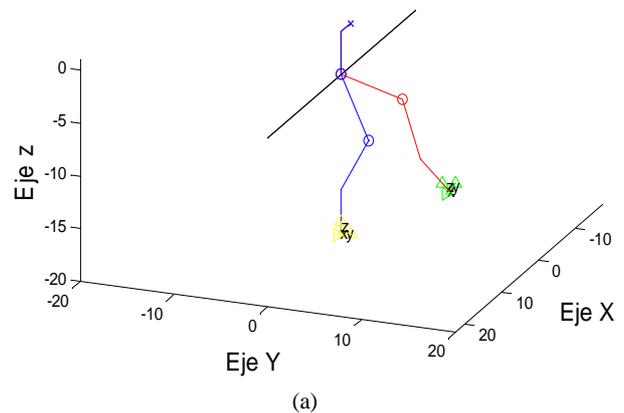


Figura 28: Representación del giro 1

De igual forma, para el *twist 2*:

$$\theta_2 = -40.7974^\circ \quad (87)$$

$$e^{\omega^{\wedge_2} \theta_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.757 & 0.6534 & 7.2385 \\ 0 & -0.6534 & 0.757 & -0.5349 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Se puede observar este giro mediante su proyección sobre el plano Y-Z en la figura 29:

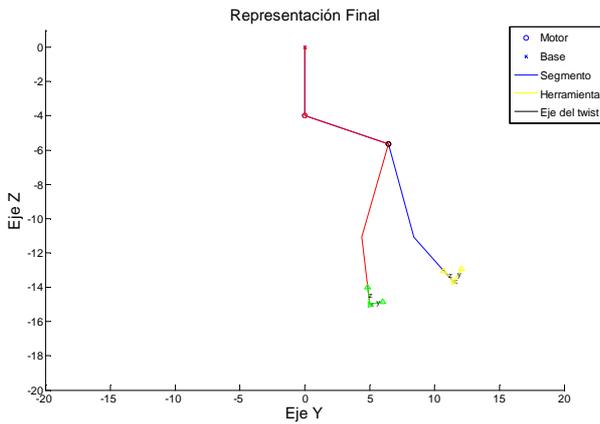
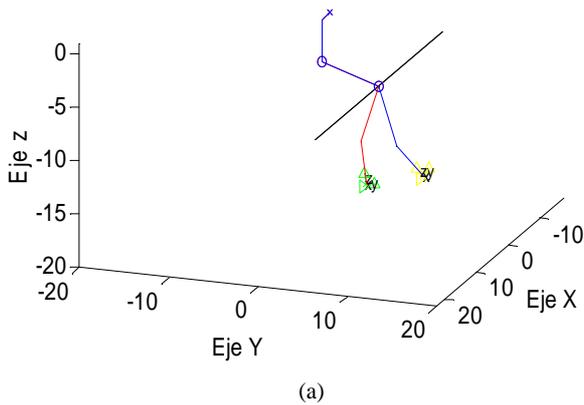


Figura 29: Representación del giro 2

Aplicando POE (12) se comprueban los resultados.

$$g(\theta) = e^{\xi_1^{\wedge} \theta_1} \cdot e^{\xi_2^{\wedge} \theta_2} g(0) = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 0.9877 & 0.1566 & 5 \\ 0 & 0.1566 & 0.9877 & -15 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (88)$$

Se verifica que se alcanzan los resultados pretendidos. El tiempo computacional ha sido de 0.25 segundos, lo cual sigue siendo un resultado muy bueno. En la figura 30 se observa la posición alcanzada en el simulador y la alcanzada por el robot.



(a)



(b)

Figura 30: Posición final alcanzada

4.2.6 Movimiento hacia atrás de la pierna del robot con dos motores a través del plano Y-Z usando un algoritmo recursivo

En este apartado se va a estudiar si el algoritmo recursivo soluciona el error obtenido en el apartado 4.2.4. Los datos obtenidos para el primer giro aplicando dicho algoritmo recursivo son:

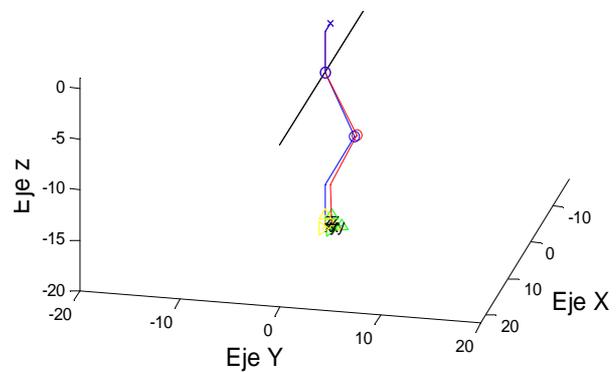
$$\theta_1 = 2.7964^\circ \quad (89)$$

$$e^{\omega^{\wedge_1} \theta_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9988 & -0.0488 & -0.1951 \\ 0 & 0.0488 & 0.9988 & -0.0048 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En la figura 31 se puede observar este primer giro sobre los planos X-Y-Z (a) y su proyección sobre los planos Y-Z (b). Los datos obtenidos para el twist 2 son los siguientes:

$$\theta_2 = -25.8223^\circ \quad (90)$$

$$e^{\omega^{\wedge_2} \theta_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9001 & 0.4356 & 4.6454 \\ 0 & -0.4356 & 0.9001 & 0.2647 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



(a)

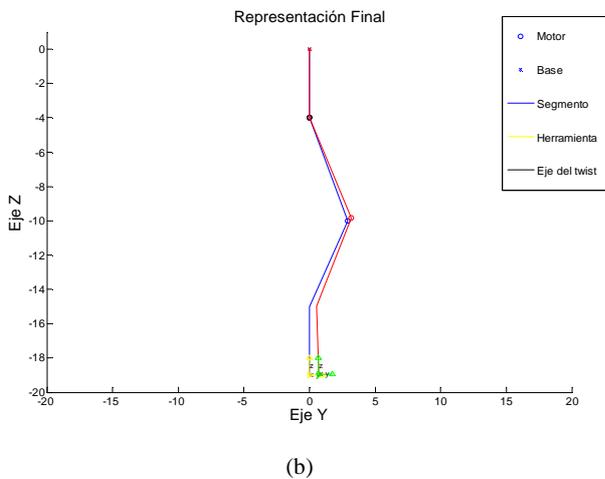


Figura 31: Representación del giro 1

La representación de este segundo giro puede observarse en la figura 32.

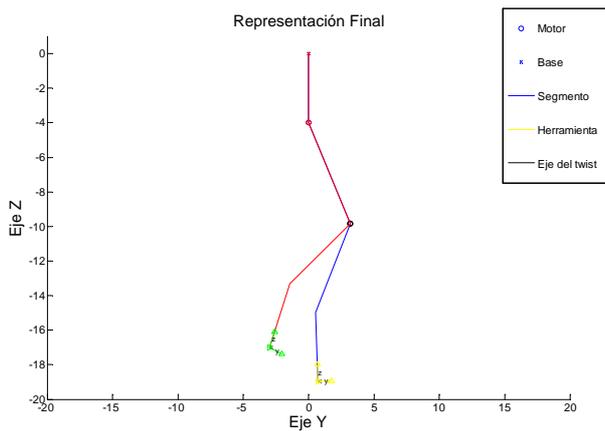


Figura 32: Representación del giro 2

Se comprueban los resultados mediante el POE (12):

$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} g(0) = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 0.9203 & 0.3911 & -3 \\ 0 & -0.3911 & 0.9203 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (91)$$

Se verifica que se alcanzan los resultados deseados. En la figura 33 se puede observar el segundo giro mediante su representación en los planos X-Y-Z y su comparación con la posición alcanzada en el sistema real.

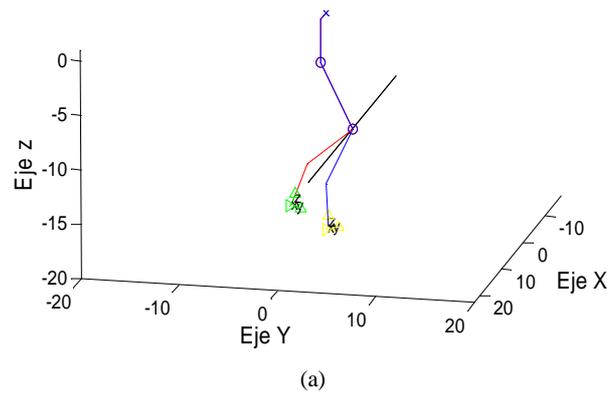


Figura 33: representación final

4.3 Movimiento de la pierna del robot a través de tres ejes

En este apartado se resuelve el problema de la orientación y traslación de la herramienta en el plano Y-Z, es decir, el problema de caminar hacia delante y hacia atrás.

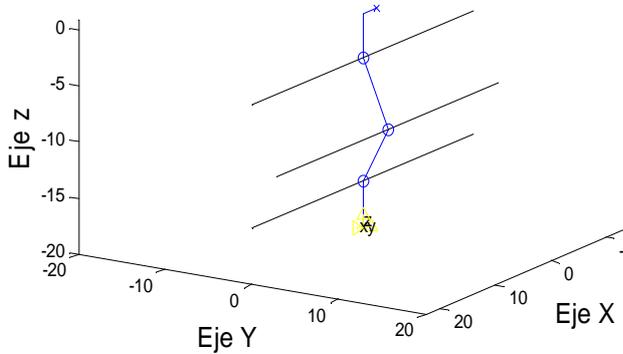
Al sistema planteado en el apartado 4.2.3. se le añade un tercer motor dedicado a la orientación del pie. Dicho motor no está situado en el pie, por lo que al orientar éste se provoca también una traslación en el mismo. En este proyecto se busca que el pie mantenga en todo momento la orientación respecto al suelo, por lo que el *twist* 4 estará siempre situado 4 cm. encima del pie. De esta forma, se pueden realizar todos los cálculos suponiendo que el pie está situada en el *twist* 4, trasladando ésta a 4 cm. sobre la posición de destino deseada. Como el tercer motor debe corregir el giro provocado por los motores anteriores, que tienen ejes de giro paralelos al de este motor, se puede considerar un GDL virtual y para calcular el giro que debe girar basta con calcular la suma de los grados girados por los motores anteriores pero en sentido contrario.

$$\theta_3 = -(\theta_1 + \theta_2) \quad (92)$$

Con este fin se ha implementado la función PK1POE3, que utiliza los mismos parámetros que (46) y que hace uso del algoritmo recursivo (85).

```
function PK1POE3(VTI, VTF, ejes, MTI, (93)
MTF, Restricciones, Recursivo)
```

En la siguiente figura se representan los ejes de giro en el simulador y en el robot.



(a)



(b)

Figura 34: Representación inicial

En la figura 35 se observan las proyecciones sobre los ejes X-Z e Y-Z. Se parte de la siguiente ecuación, según el POE (12):

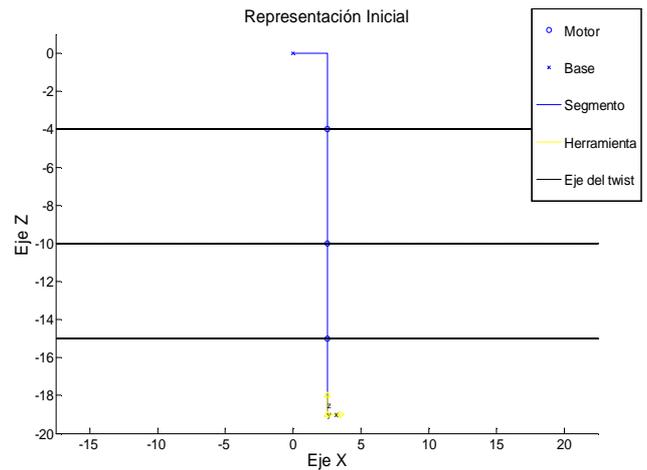
$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} \cdot e^{\xi_3 \wedge \theta_3} \cdot g(0) \quad (94)$$

Como se ha explicado anteriormente, el *twist* 3 puede considerarse un GDL virtual, ya que no afecta a la traslación del pie, y de esta forma se elimina del problema inicial. Así, se queda la siguiente ecuación.

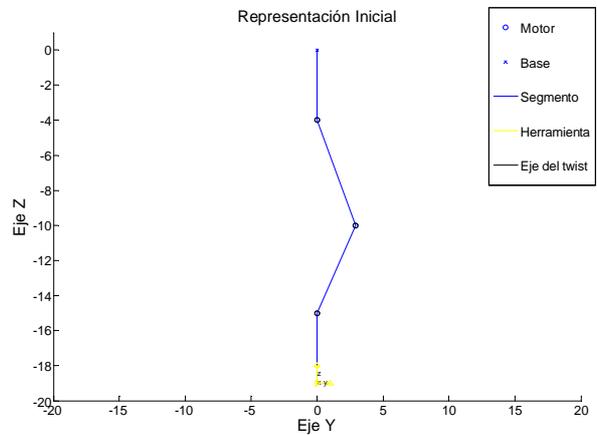
$$g(\theta) = e^{\xi_1 \wedge \theta_1} \cdot e^{\xi_2 \wedge \theta_2} \cdot g(0) \quad (95)$$

El problema planteado consiste en encontrar los ángulos que deben girar los *twists* para alcanzar $g(\theta)$.

$$g(\theta) = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (96)$$



(a)



(b)

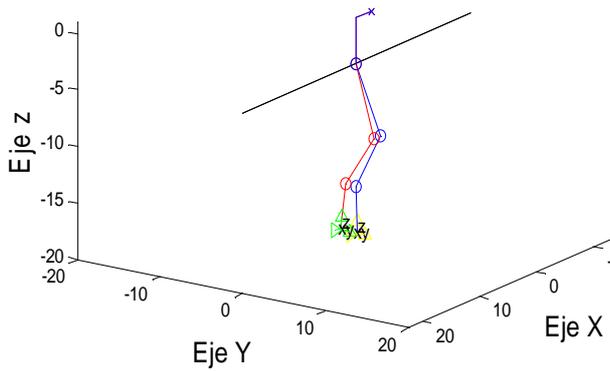
Figura 35: Proyecciones sobre los X-Z e Y-Z iniciales

Para resolver el problema se va a utilizar el algoritmo recursivo (89) desarrollado en el apartado 4.2.5. Tras aplicar dicho algoritmo recursivo se obtienen los siguientes resultados para el *twist* 1:

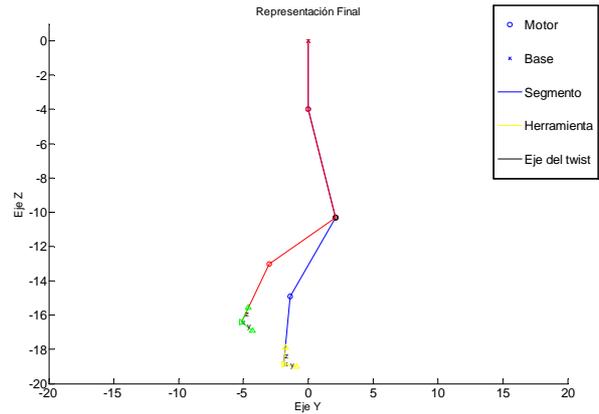
$$\theta_1 = -7.2474^\circ \quad (97)$$

$$e^{\omega_1 \theta_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.992 & 0.1262 & 0.5046 \\ 0 & -0.1262 & 0.992 & -0.032 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Este giro puede observarse en la siguiente figura, donde se representan los planos X-Y-Z y su proyección sobre Y-Z.



(a)



(b)

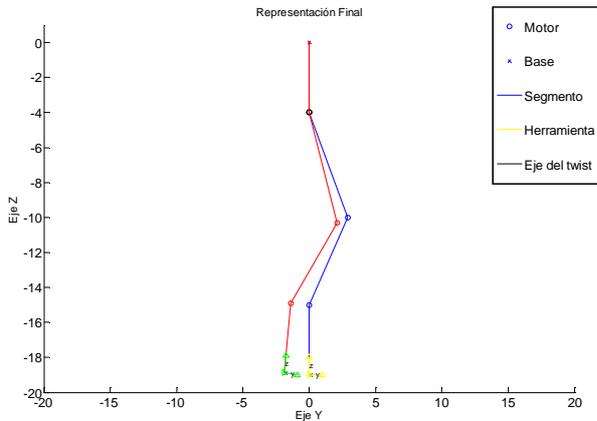
Figura 37: Representación del giro 2

El tercer giro debe corregir los realizados por los motores anteriores (92), obteniéndose la siguiente $g(\theta)$:

$$\theta_3 = -(\theta_1 + \theta_2) = 32.2382^\circ \quad (99)$$

$$e^{\omega^{\wedge}_3 \theta_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.8458 & -0.5334 & 4.4962 \\ 0 & 0.5334 & 0.8458 & 0.2889 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En la siguiente figura se representa el tercer giro:



(b)

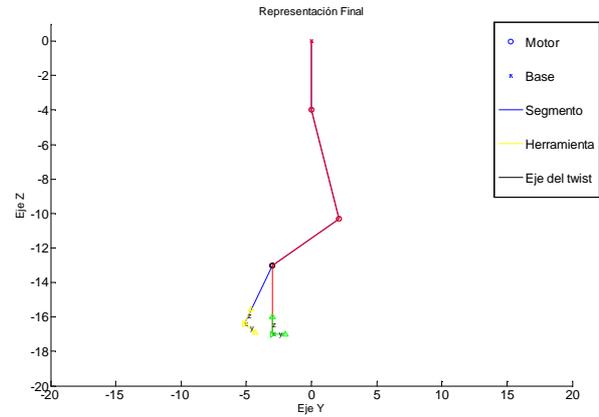
Figura 36: Representación del giro 1

Para el *twist* 2 se obtienen los siguientes datos:

$$\theta_2 = -24.9908^\circ \quad (98)$$

$$e^{\omega^{\wedge}_2 \theta_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.9064 & 0.4225 & 4.4962 \\ 0 & -0.4225 & 0.9064 & 0.2889 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

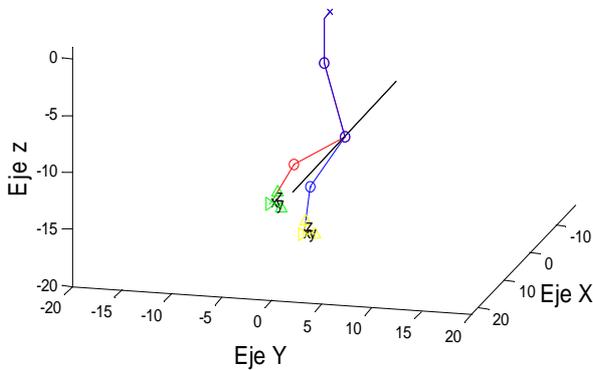
En la siguiente figura se muestra el giro realizado:

**Figura 38: Representación del giro 3**

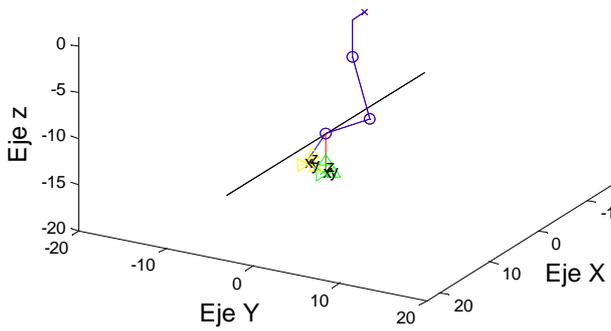
Aplicando POE para el problema cinemático directo, se obtiene la posición y orientación del pie:

$$g(\theta) = e^{\xi_1^{\wedge} \theta_1} \cdot e^{\xi_2^{\wedge} \theta_2} \cdot e^{\xi_3^{\wedge} \theta_3} g(0) = \begin{pmatrix} 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (100)$$

Se puede observar que se alcanzan los objetivos buscados. El tiempo de cómputo empleado es de 0.2344 segundos, que sigue siendo muy eficiente para ser implementado en un sistema en tiempo real. En la figura 39 se observa el resultado final y el sistema real.



(a)



(a)



(b)

Figura 39: Posición final del sistema

5. Simulación de un paso

En este apartado se va a simular un paso completo de la pierna derecha del robot humanóide. Para ello se ha implementado la siguiente función en Matlab.

```
function POE3Iterativo(Posiciones, (101)
Restricciones)
```

Esta función define las condiciones iniciales del robot, ya que se supone que siempre se parte de la posición inicial del robot. Utiliza como parámetros *Posiciones* y *Restricciones*. La primera de estas variables es una matriz $n \times 3$ con las posiciones a las que se desea trasladar secuencialmente el robot, donde n es el número de movimientos que se van a realizar. La otra variable, *Restricciones*, es un valor booleano que indica si se desean usar las restricciones reales de los motores del robot. Tras construir las estructuras necesarias, esta función hace uso de la función *PK1POE3*.

```
function [Estado] = (102)
PK1POE3Iterativo(Estado,
VTF, Restricciones)
```

Esta función adapta el algoritmo (93) para hacerlo iterativo. La variable *Estado* es una matriz 9×4 , donde las 5 primeras filas almacenan la posición de los motores. La matriz 4×4 restante representa el elemento perteneciente a $SE(3)$ que define la

posición y orientación del pie. *VTF* indica la siguiente posición a la que se desea trasladar el pie y *Restricciones* es igual que en (93). Ha sido necesario implementar una nueva función *DibujarRobot* para representar el movimiento sobre los planos X - Y - X . Utiliza como parámetros las posiciones de los motores inicial y final (*Ejes* y *Ejes2*), la matriz de transformación del pie inicial y final (*Matriz* y *Matriz2*), los motores sobre los que se realizan los giros (*Motores* y *Motores2*) el rango de representación (*Rango*) y una variable de texto *Titulo*, que indica el título que debe tener la gráfica.

```
function DibujarRobot(Ejes, Ejes2, (103)
Titulo, Matriz, Matriz2, Motores,
Motores2, Rango)
```

La variable *Estado* tiene la siguiente estructura, donde r son vectores 1×4 que definen la posición de los motores y g es una matriz 4×4 con la posición y orientación del pie en un momento dado.

$$\text{Estado} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ g \end{pmatrix} \quad (104)$$

Se desea dar un paso de 2 cm. de longitud elevando el pie 2 cm. La secuencia de posiciones h que se van a usar como posiciones intermedias es la siguiente:

$$h_1 = \begin{pmatrix} 2.5 \\ 0 \\ -17 \end{pmatrix} \wedge h_2 = \begin{pmatrix} 2.5 \\ 2 \\ -17 \end{pmatrix} \wedge h_3 = \begin{pmatrix} 2.5 \\ 2 \\ -19 \end{pmatrix} \wedge h_4 = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix} \wedge \quad (105)$$

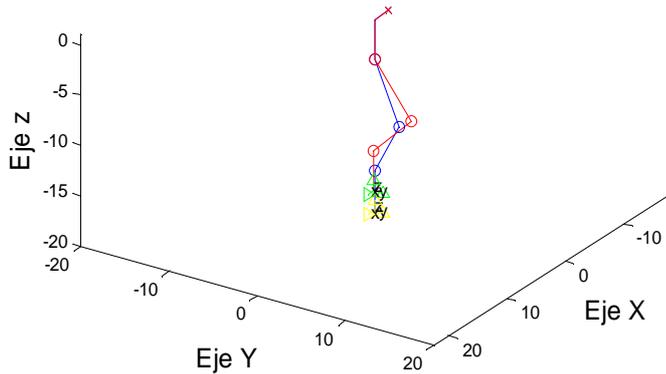
$$h_5 = \begin{pmatrix} 2.5 \\ -2 \\ -19 \end{pmatrix} \wedge h_6 = \begin{pmatrix} 2.5 \\ -2 \\ -17 \end{pmatrix} \wedge h_7 = \begin{pmatrix} 2.5 \\ -2 \\ -19 \end{pmatrix} \wedge h_8 = \begin{pmatrix} 2.5 \\ 0 \\ -19 \end{pmatrix}$$

El primer movimiento consiste en levantar el pie verticalmente 2 cm. Para ello, se llama a la función () utilizando h_1 como posición de destino. Tras aplicar el algoritmo recursivo, nos da el siguiente conjunto de ángulos para el movimiento hacia h_1 .

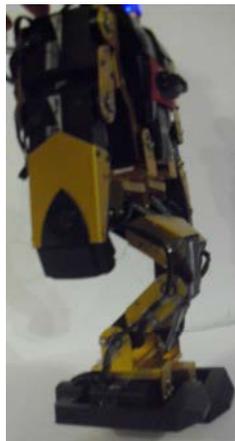
$$\begin{aligned} \theta_1 &= 14.1219^\circ & (106) \\ \theta_2 &= -31.7245^\circ \\ \theta_3 &= 17.6026^\circ \end{aligned}$$

En la figura 40 se representa este primer movimiento y la posición alcanzada por el robot. La función (102) devuelve el estado del sistema tras éste movimiento.

$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 4.2763 & -9.1111 & 1 \\ 2.5 & 0 & -13 & 1 \\ 2.5 & 0 & -17 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (107)$$



(a)



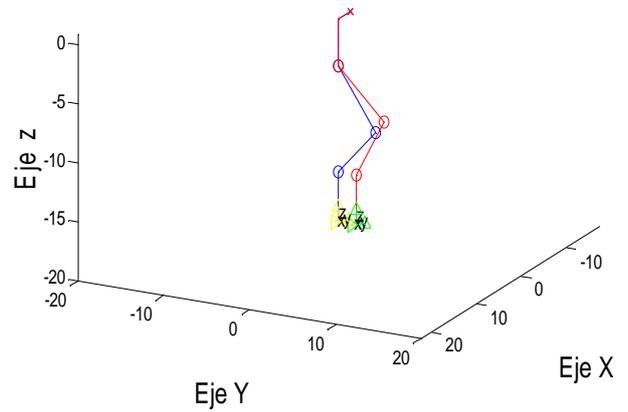
(b)

Figura 40: Posición h_1

Este estado se usará en la siguiente iteración de (102) como parámetro de entrada. El segundo movimiento consiste en trasladar el pie 2 cm. en dirección Y. Así, y utilizando h_2 como posición de destino, se obtiene este conjunto de ángulos para el movimiento hacia h_2 .

$$\begin{aligned} \theta_1 &= 11.2355^\circ \\ \theta_2 &= 2.9820^\circ \\ \theta_3 &= -14.2174^\circ \end{aligned} \quad (108)$$

En la figura siguiente se representa este segundo movimiento y la posición alcanzada por el robot.



(a)



(b)

Figura 41: Posición h_2

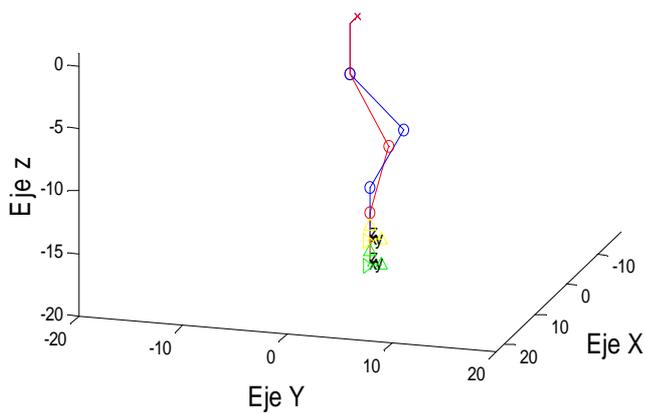
El estado del sistema tras este movimiento es el siguiente:

$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 5.1902 & -8.18 & 1 \\ 2.5 & 2 & -13 & 1 \\ 2.5 & 2 & -17 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (109)$$

El siguiente movimiento consiste en apoyar el pie en el suelo. Para ello se usa h_3 como posición de destino, obteniéndose el siguiente valor para los ángulos.

$$\begin{aligned} \theta_1 &= -16.7058^\circ \\ \theta_2 &= 32.3834^\circ \\ \theta_3 &= -15.6776^\circ \end{aligned} \quad (110)$$

En la figura se representa este movimiento y la posición alcanzada por el robot.



(a)



(b)

Figura 42: Posición h_3

El estado del sistema tras realizar este movimiento es el siguiente:

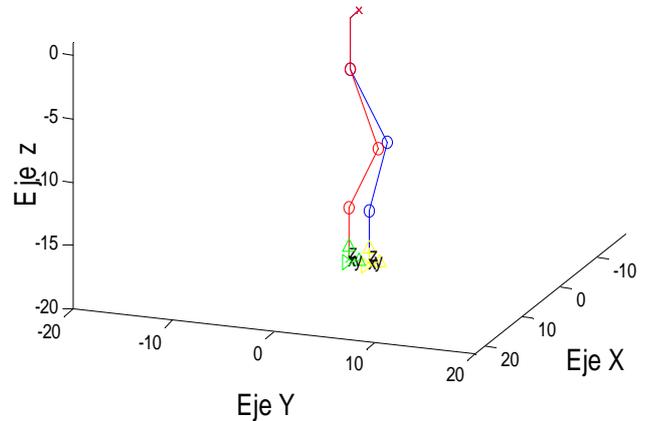
$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 3.7696 & -9.4955 & 1 \\ 2.5 & 2.0006 & -14.9983 & 1 \\ 2.5 & 2.0006 & -18.9983 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 2.0006 \\ 0 & 0 & 1 & -18.9983 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (111)$$

Se puede observar un ligero error en los resultados, pero es tan pequeño que puede despreciarse.

Una vez apoyado el pie, hay que trasladarlo hacia atrás por el suelo para mover el cuerpo hacia delante. Este movimiento se ha dividido en dos, primero hasta la posición inicial para después moverlo hasta 2 cm. en dirección $-Y$. La siguiente posición debe ser entonces h_4 . Los valores de los ángulos conseguidos son los siguientes:

$$\begin{aligned} \theta_1 &= -8.6517 & (112) \\ \theta_2 &= -3.6407^\circ \\ \theta_3 &= 12.2924^\circ \end{aligned}$$

En la figura 43 se representa este movimiento y la posición alcanzada por el robot.



(a)



(b)

Figura 43: Posición h_4

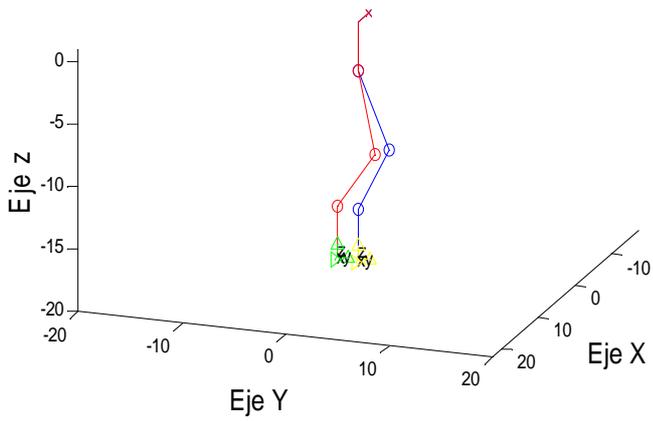
El estado del sistema tras realizar este movimiento es el siguiente:

$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 2.9 & -10 & 1 \\ 2.5 & 0 & -15 & 1 \\ 2.5 & 0 & -19 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -19 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (113)$$

Se comprueba que la posición de los motores es la correcta. Para terminar de desplazar el pie hacia atrás se establece la siguiente posición a trasladar, h_5 , obteniendo los ángulos siguientes:

$$\begin{aligned} \theta_1 &= -11.9784^\circ & (114) \\ \theta_2 &= 3.6755^\circ \\ \theta_3 &= 8.3029^\circ \end{aligned}$$

En la figura siguiente se representa este movimiento y la posición alcanzada por el robot.



(a)



(b)

Figura 44: Posición h_5

La función nos aporta los siguientes resultados:

$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 1.5916 & -10.4712 & 1 \\ 2.5 & -2 & 15.0001 & 1 \\ 2.5 & -2 & -19.0001 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -19 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (115)$$

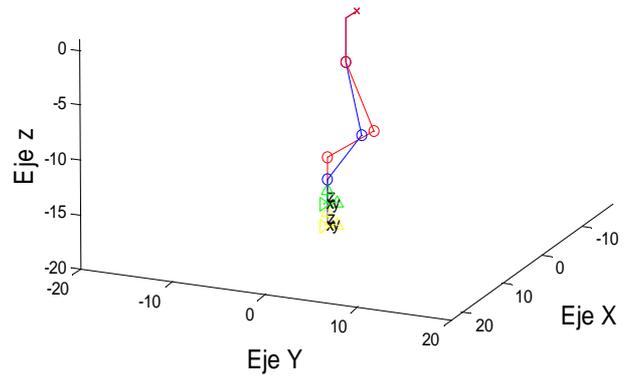
Una vez retrasado el pie se eleva para volver a la posición original. Se establece la posición h_6 como objetivo, obteniéndose los ángulos siguientes:

$$\theta_1 = 12.2783^\circ \quad (116)$$

$$\theta_2 = -32.4182^\circ$$

$$\theta_3 = 20.1399^\circ$$

En la figura 45 se representa este movimiento.



(a)



(b)

Figura 45: Posición h_6

A continuación se adelanta el pie hasta situarlo sobre la posición inicial. La función nos aporta los siguientes resultados:

$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 2.9314 & -9.9847 & 1 \\ 2.5 & -2 & 13.0001 & 1 \\ 2.5 & -2 & -17.0001 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (117)$$

La siguiente posición, h_7 , mover el pie hacia delante hasta la altura de la cintura. Los valores de los ángulos conseguidos son:

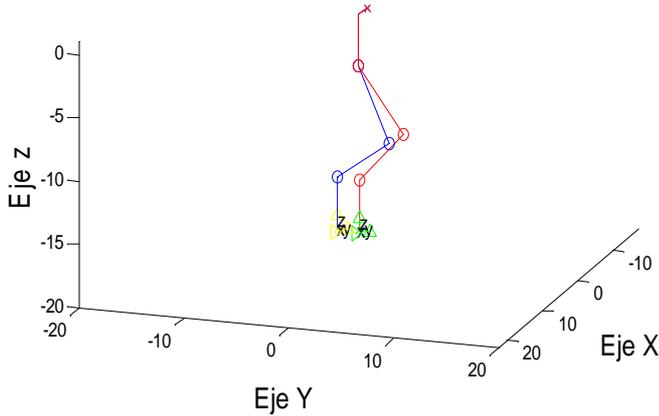
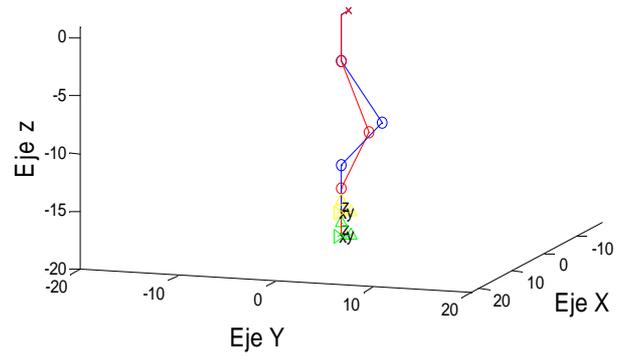
$$\theta_1 = 13.8222^\circ \quad (118)$$

$$\theta_2 = -2.982^\circ$$

$$\theta_3 = -10.8402^\circ$$

El movimiento realizado puede verse en la figura 46. La función nos aporta los siguientes resultados:

$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 4.2763 & -9.1111 & 1 \\ 2.5 & 0 & 13.0001 & 1 \\ 2.5 & 0 & -17.0001 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -17 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (119)$$



(a)



(b)

Figura 46: Posición h_7

El último movimiento, h_8 , consiste en bajar el pie hasta alcanzar la posición original. Los ángulos obtenidos son los mostrados en (120).

$$\begin{aligned} \theta_1 &= 13.8222^\circ \\ \theta_2 &= -2.982^\circ \\ \theta_3 &= -10.8402^\circ \end{aligned} \quad (120)$$

El movimiento realizado puede verse en la figura siguiente:



(b)

Figura 47: Posición h_8

La función nos aporta los siguientes resultados:

$$\text{Estado} = \begin{pmatrix} 2.5 & 0 & 0 & 1 \\ 2.5 & 0 & -4 & 1 \\ 2.5 & 2.9 & -10 & 1 \\ 2.5 & 0 & -15 & 1 \\ 2.5 & 0 & -19 & 1 \\ 1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -19 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (121)$$

En total, se han utilizado 0.8906 segundos para el cómputo de los ángulos necesarios para alcanzar las posiciones intermedias. Teniendo en cuenta que se han realizado 8 cálculos de posiciones, el resultado es excelente para ser implementado en un sistema en tiempo real.

6. Conclusiones y Trabajos Futuros

Como se ha demostrado en este proyecto, la combinación de Álgebra de Lie y POE lleva a una herramienta idónea para la formulación de la cinemática, tanto directa como inversa, de un

robot humanoíde. El movimiento queda definido simplemente por la posición y orientación del pie antes y después del giro, sin que existan singularidades que restrinjan el método. Hay que destacar los tiempos de cómputo obtenidos. Para el movimiento lateral, el simulador empleó 0.1406 segundos en resolver la cinemática inversa del movimiento, mientras que para la simulación del paso hacia delante, que consta de 8 posiciones, empleó 0.8906 segundos. Estos resultados indican que la metodología seguida es muy apropiada para una posible implementación en el sistema real. Esto se debe a la simplicidad de las operaciones matemáticas a realizar, mucho menos costosas computacionalmente que otras formulaciones. Otra interesante ventaja del simulador desarrollado es que se ha conseguido un toolbox para Matlab y por lo tanto dicho software se puede utilizar y modificar fácilmente.

El problema de éste método es que para simplificar la ecuación inicial planteada por POE y poder aplicar Paden-Kahan, el sistema debe tener unas determinadas especificaciones. En ningún caso el número de GDL puede ser mayor de 5 para una extremidad. Y esto sólo en el caso en que coincidan los ejes de giro en sentido X e Y de la cadera y el tobillo, permitiendo utilizar Paden-Kahan Dos. En el caso del Robonova-I no se produce dicha circunstancia, por lo que se pueden resolver simultáneamente 3 GDL como máximo.

Como futuros trabajos se plantea el extender el simulador a las dos piernas y adaptar el problema para el caso en que la base del sistema es el pie y la herramienta el punto central del eje que une ambas piernas. Esto es necesario para cargar el peso en una determinada pierna y así poder mover la otra. Una vez ampliado el simulador, y con un cálculo de trayectorias adecuado, se puede usar controlar el movimiento de las piernas del robot Robonova-I.

7. Referencias

- [1] Robot humanoide pasa lista y regaña en aula japonesa. reparacionlcd.com, 2009. http://www.reparacionlcd.com/noticias/Robot_humanoide_pasa_lista_y_rega%C3%B1a_en_aula_japonesa--1238. Consultado el 30/11/2012.
- [2] El robot recepcionista. elsingular.com, 2006. <http://elsingular.com/2006/04/09/el-robot-recepcionista/>. Consultado el 30/11/2012.
- [3] Robots para asistencia a personas. aishoutrends.blogspot.com.es, 2008. <http://aishoutrends.blogspot.com.es/2008/11/robots-para-asistencia-personas.html>. Consultado el 30/11/2012.
- [4] Wakamaru, el robot recepcionista de alquiler. zonasilicia.blogspot.com.es, 2007. <http://zonasilicia.blogspot.com.es/2007/06/wakamaru-el-robot-recepcionista-de.html>. Consultado el 30/11/2012.
- [5] Los escolares de Tokio tienen un robot guardian. elsingular.com, 2006. <http://elsingular.com/2006/04/20/les-escolares-de-tokio-tienen-un-robot-guardian/>. Consultado el 30/11/2012.
- [6] Japón diseña un robot humanoide socorrista. bajoelagua.com, 2012. <http://www.bajoelagua.com/mundo-submarino/noticias/2012-09-12/japon-disena-robot-humanoide-socorrista-0854.html>. Consultado el 30/11/2012.
- [7] Construyen en EEUU androides que podrían ayudar en rescates y desastres naturales. biobiochile.cl, 2012. <http://tecnologia.biobiochile.cl/notas/2012/09/06/construyen-en-eeuu-androides-que-podrian-ayudar-en-rescates-y-desastres-naturales.shtml>. Consultado el 30/11/2012.
- [8] Riba II, el robot humanoide sanitario para el traslado de pacientes. opinion.com.bo, 2011. <http://www.opinion.com.bo/opinion/articulos/2011/1108/noticias.php?id=31103>. Consultado el 30/11/2012.
- [9] kokoro-dreams.co.jp, 2012. Actroid-DER Series. http://www.kokoro-dreams.co.jp/english/rt_rent/actroid.html. Consultado el 30/11/2012.
- [10] HRP-4C. es.wikipedia.org, 2009. <http://es.wikipedia.org/wiki/HRP-4C>. Consultado el 30/11/2012.
- [11] El éxito en el desarrollo de robots con apariencia humana y comportamiento de rendimiento. aist.go.jp, 2009. http://www.aist.go.jp/aist_j/press_release/pr2009/pr20090316/pr20090316.html. Consultado el 30/11/2012.
- [12] *Actroid robot greets Japan World Expo visitors a bit too naturally*, Robot Actroid recibe a los visitantes en la Expo Mundial de Japón con naturalidad. engadget.com, 2005. <http://www.engadget.com/2005/03/10/actroid-robot-greets-japan-world-expo-visitors-a-bit-too/>. Consultado el 30/11/2012.
- [13] *HRP-4C fashion model robot*, HRP-4C el robot modelo de moda. pinktentacle.com, 2009. <http://pinktentacle.com/2009/03/video-hrp-4c-fashion-model-robot/>. Consultado el 30/11/2012.
- [14] Ramírez Benavides, Kryscia Daviana, Locomoción de Robot Móvil. Escuela de Ciencias de la Computación e Informática. Universidad de Costa Rica. http://www.kramirez.net/Robotica/Material/Presentaciones/Loco_mocion.pdf. Consultado el 30/11/2012.
- [15] del Solar San Martín, Javier Ruiz. ¿Por qué los robots imitan a los seres vivos? Departamento de Ingeniería Eléctrica. Universidad de Chile. <http://graficas.explora.cl/otros/metro/metroformas/robot.html>. Consultado el 30/11/2012.
- [16] Página principal del proyecto ASIMO, Honda. <http://world.honda.com/ASIMO/>. Consultado el 30/11/2012.
- [17] Página principal del proyecto Partner Robot, Toyota. http://www.toyota-global.com/innovation/partner_robot/. Consultado el 30/11/2012.
- [18] Especificaciones del Robot HOAP-3, Fujitsu. http://home.comcast.net/~jtechsc/HOAP-3_Spec_Sheet.pdf. Consultado el 30/11/2012.
- [19] Página principal del proyecto NAO, Aaldebaran Robotics. <http://www.aldebaran-robotics.com/en/>
- [20] Kato, Ichiro. 2007. Development of Waseda Robot. Desarrollo del robot Waseda. http://www.humanoid.waseda.ac.jp/booklet/kato_2.html. Consultado el 30/11/2012.
- [21] *Perception-Action Integrated Humanoid Robot : H6 & H7*. Percepción-Actuación integrada en robot humanoids: H6 y H7. Johou Systems Kougaku (JSK) Laboratory, The Mechano-Informatics Department. Tokyo University, Japón. <http://www.jsk.t.u-tokyo.ac.jp/research/h6/>. Consultado el 30/11/2012.

[22] *Development of a Humanoid Robot*. Desarrollo de un robot humanoide. Design of Intelligent Machines Laboratory, Tohoku University. Japón. <http://www.space.mech.tohoku.ac.jp/research/saika3/saika3-e.html>

[23] El robot humanoide de la Universidad Departamento de Automática, Ingeniería Electrónica e Informática Industrial. Universidad Carlos III. Madrid. España. http://www.uc3m.es/portal/page/portal/actualidad_cientifica/actualidad/reportajes/archivo_reportajes/robot_humanoide_uc3m. Consultado el 30/11/2012.

[24] Página principal del proyecto REEM-B, Pal Robotics. <http://pal-robotics.com/robots/reem-b>. Consultado el 30/11/2012.

[25] Página principal del proyecto OpenHRP3. <http://www.openrtp.jp/openhrp3/en/index.html>. Consultado el 30/11/2012.

[26] T. Reichenbach, 2009. *A dynamic simulator for humanoid robots*. Un imulador diámico para robots. *humanoids. Artificial Life and Robotics journal.*

[27] Página principal de Webot. <http://www.cyberbotics.com/>. Consultado el 30/11/2012.

[28] Página principal de Adams. <http://www.mscsoftware.com/Products/CAE-Tools/Adams.aspx>. Consultado el 30/11/2012.

[29] Página principal de RoboWorks. http://www.newtonium.com/public_html/Products/RoboWorks/RoboWorks.htm. Consultado el 30/11/2012.

[30] G. A. Medrano-Cerda, H. Dallali, M. Brown, N. G. Tsagarakis, D. G. Caldwell, 2011. Modelling and Simulation of the Locomotion of Humanoid Robots. Modelado y simulación de robots humanoids. *Istituto Italiano di Tecnologia (IIT)*, Genova, Italia y The University of Manchester, Reino Unido.

[31] J.M. Pardos Gotor. 2005. Algoritmos de Geometría Diferencial para la Locomoción y Navegación Bipedas de Robots Humanoides. Aplicación al robot RH0. Tesis Doctoral Universidad Carlos III Madrid. España.

[32] Muhammad A. Ali, H. Andy Park, y C. S. George Lee 2010. *Closed-Form Inverse Kinematic Joint Solution for Humanoid Robots*. Solución Cerrada para la Cinemática Inversa de un Robot Humanóide. *The International Journal of Humanoid Robotics*, Vol. 9, No. 3 (2012)

[33] Matlab, el lenguaje de cálculo técnico. mathworks.es, 2012. <http://www.mathworks.es/products/matlab/>. Consultado el 30/11/2012.

Apéndice A: Descripción del robot Robonova-I

Datos del robot

- Dimensiones: 310 x 180 x 90mm.
- Peso con batería de serie: aprox. 1.3 kg

Sensores

En estos momentos no hay conectado ningún sensor, pero anteriormente se han usado un sonar y sensores de fin de carrera en manos y pies.

Actuadores

Dispone de 16 servomotores HSR-8498MB que se usan como las articulaciones del robot.

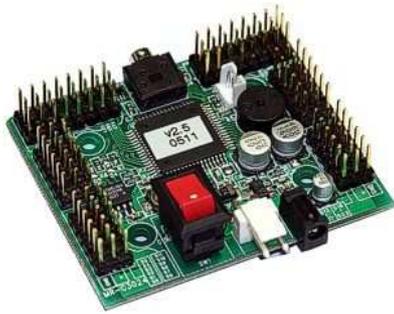


Las especificaciones de dichos motores son las siguientes:

- Interface : Protocolo HMI, PWM
- Tensión de trabajo: 6.0V
- Velocidad máxima: 0.20seg/60° a 6.0V
- Torque: 10kg.cm (138.87oz.in) a 6.0V
- Ángulo: Max 180°
- Peso: 55g (1.94oz)
- Dimensiones: 40 x 20 x 47mm (1.57 x 0.78 x 1.85 in)
- Pulso de control Neutral : 1500 μ Seg , 0~180° : \pm 1100~1900/ μ Seg
- Ciclo de pulsos: 12~26mSec (Normal : 21msec)

Hardware:

El robot Robonova 1 está montado sobre un circuito de control MR-C3024 controlado por un microcontrolador Atmel ATmega 128 que cuenta entre con 40 puertos de entrada y salida digitales, puerto serie, bus I2C y 8 entradas analógicas. Con esto se pueden controlar dispositivos de todas clases como servos, sensores de distancia, giróscopos, displays LCD, sensores de infrarrojos, etc. Además la placa cuenta con un altavoz para generar tonos de diferentes frecuencias y un conector para un led que se puede gobernar a voluntad. Dispone de 64 KB de memoria para los programas, que permiten que una vez que se han descargado, el robot sea completamente autónomo y pueda ejecutar los movimientos sin necesidad de estar conectado al ordenador.



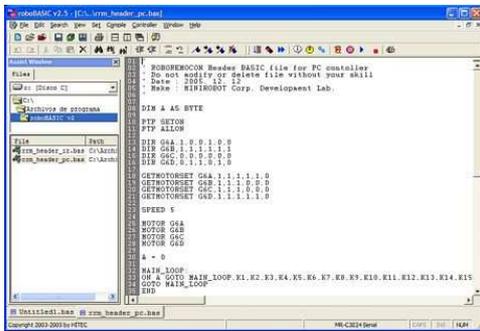
Datos técnicos de la placa de control

- dimensiones aprox.: 61 x 50,5mm
- Controlador: Atmel ATMega128
- tensión de uso: 6 - 8 Voltios.
- I/O Ports: 40 Puertas de entrada/salida digitales, de los cuales 24 como puertas de servos y 8 como transformador A/D
- puerto serie RS232 y bus I2C
- emisor de señales
- memoria adicional: 64K *8 EEPROM

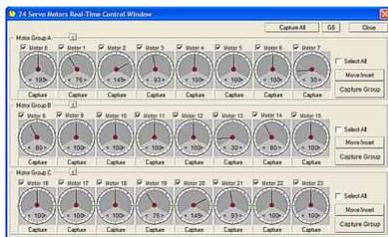
Software:

El fabricante aporta las siguientes herramientas para programar el robot:

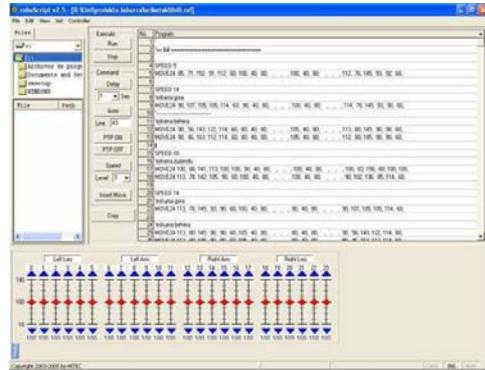
- Robobasic: Es un entorno de programación basado en un lenguaje basado BASIC y diseñado específicamente para el control de los controladores de la serie MR-C para gestión de Robots.



Además, incluye una herramienta para el control de los servos en tiempo real:



- Roboscript: Esta herramienta permite incluir una secuencia de movimientos de forma intuitiva, sin tener que programar nada.



Apéndice B: Funciones implementadas en Matlab

En este proyecto se ha implementado una toolbox para Matlab con las siguientes funciones:

Función	Descripción
DibujarInicial	Dibuja el sistema inicial en 3 dimensiones y las proyecciones sobre los ejes X-Z e Y-Z.
DibujarMov	Representa el movimiento a través de un eje tanto en los planos X-Y-Z como en sus proyecciones sobre X-Y e Y-Z.
Dibujar3d	Dibuja las representaciones del sistema sobre los planos X-Y-Z.
DibujarEje3d	Dibuja un conjunto de ejes adicionales a la representación sobre los planos X-Y-Z.
Representar	Dibuja la proyección el sistema sobre un plano.
DibujarEje	Dibuja un eje en una representación sobre un plano.
PK1	Resuelve el problema de la traslación del pie del robot Robonova-I a través de un eje aplicando PK1.
PK1conPOE	Resuelve el problema de la traslación del pie del robot Robonova-I a través de un eje aplicando PK1 y POE.
PK1POE2	Resuelve el problema de la traslación y orientación del pie del robot Robonova-I a través de dos ejes aplicando PK1 y POE.
POE2Recursivo	Calcula la matriz de transformación final de la herramienta mediante un algoritmo recursivo basado en PK1 y POE.
PK1POE3	Resuelve el problema de la traslación y orientación del pie del robot Robonova-I a través de dos ejes aplicando PK1 y POE.
POE3Iterativo	Resuelve el problema del movimiento en

	un plano a través de una serie de posiciones intermedias.
PK1POE3Iterativo	Resuelve el problema de la traslación y orientación del pie del robot Robonova-I partiendo de un estado distinto al inicial.

Para el desarrollo de este toolbox, se han utilizado las siguientes funciones del toolbox ROBOMAN [31].

Función	Descripción
revolutetwist	Construye ξ^{\wedge} a partir de ω y r .
padenkanone	Calcula PK1 a partir de ξ^{\wedge} , p y k .
twistexp	Calcula la exponencial de un <i>twist</i> .
rptohomogeneous	Construye un elemento de SE(3) a partir de su posición y orientación.

Abstract: The study of kinematic modelling of humanoid robots constitutes a challenge, mainly due to the number of degrees of freedom of these systems. The complexity of the problem is proportional to the number of joints and the number of engines. This paper shows a simulator related to the motion of one leg of one humanoid robot. For that purpose a kinematic model has been studied and implemented.. The simulator is based on the mathematical tools: Paden-Kahan One and Product of Exponentials. In particular, the robot Robonova-I has been used for modelling purposes. This robot has five degrees of freedom in each leg. With the aim of showing the operation of the simulator proposed, the cases of leg movements in one, two and three axes, as well as the simulation of a full step are tried.

Resumen: El estudio de la cinemática de un robot humanoide supone un reto, debido a la cantidad de grados de libertad del sistema. La complejidad del problema es proporcional a la cantidad de articulaciones, aumentando exponencialmente conforme aumenta el número de motores. En este trabajo, se presenta un simulador de la cinemática de un robot que ayudará a implementar los modos de caminar del mismo. Se basa en las herramientas matemáticas de Paden-Kahan Uno y el Producto de Exponenciales para resolver el problema de la cinemática de una pierna. Como caso práctico, se ha utilizado el robot Robonova-I, que dispone de cinco grados de libertad en cada pierna. Para mostrar el buen funcionamiento de esta herramienta, se analizarán los casos de los movimientos de la pierna del robots en uno, dos y tres ejes, así como la simulación de un paso completo.

