



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO  
NACIONAL**

Unidad Zacatenco

Departamento de Control Automático

*Caminado parametrizado del humanoide Robonova I*

**TESIS**

**QUE PARA OBTENER EL TÍTULO DE**

**MAESTRO EN CIENCIAS**

**PRESENTA**

**Jessica Jazmin Maldonado Ramos**

**Directores de Tesis**

**Dr. Juan Manuel Ibarra Zannatha**

**Dr. Alejandro Justo Malo Tamayo**

Ciudad de México

Diciembre 2016

# Resumen

En esta tesis se presenta el trabajo realizado para la rehabilitación y actualización del robot humanoide Robonova-I. Se desarrolló el modelo geométrico directo de dicho robot usando la metodología de Khalil-Kleinfinger, el cuál es un método extendido de Denavit-Hartenberg utilizado para modelar robots compuestos por cadenas cinemáticas arborescentes por medio de 6 parámetros. El modelo geométrico inverso del Robonova-I fue obtenido usando el método de Paul, con el cual se obtienen las ecuaciones para conocer el valor de las variables articulares de las piernas y brazos una vez que se conoce la pose del robot. Asimismo, se actualizó el hardware de control usando una Raspberry Pi 2, creando a su vez, un entorno de simulación nuevo para el Robonova-I y tarjetas de control de servomotores adicionales. Por último, se desarrolla un caminado parametrizado usando la Raspberry Pi para crear las trayectorias en cada uno de los ejes y convertirlas en ángulos para los motores en cada fase del caminado y lograr una trayectoria estable donde el robot se mantiene de pie todo el tiempo.

# Abstract

This thesis presents the work done for the rehabilitation and update of the Robonova-I humanoid robot. The direct geometric model was developed using the Khalil-Kleinfinger methodology, an extended Denavit-Hartenber method, used to model robots composed of tree-like kinematic chains by means of 6 parameters. The inverse geometric model was obtained using the method of Paul, with this method we obtain the equations for the articular variables for legs and arms, knowing the pose of the robot. The control hardware was updated using a Raspberry Pi 2, creating a new simulation environment for the Robonova-I and additional servomotors control cards. Finally, a parametrized walk is developed using Raspberry Pi to create the trajectories in each of the axes and turn them into angles for the motors in each phase of the walk and achieve a stable trajectory where the robot stands all the time.

# Agradecimientos

- A mis padres, por ser un ejemplo a seguir, por enseñarme a nunca rendirme, por apoyarme en todas mis metas, por brindarme todo su amor durante esta travesía.
- A mis hermanos por estar a mi lado siempre. Armando gracias por ser mi apoyo, por creer en mi y regañarme para que creyera en mi cuando era necesario. Ramón gracias por enseñarme a siempre dar lo mejor de mi y que las cosas siempre se pueden resolver.
- A mi pareja por darme palabras de animo, estar a mi lado en momentos difíciles y siempre hacerme sonreír.
- A mis amigos, Miguel, Jorge, Andy, Adry, mis mejores amigos gracias por escucharme en mis momentos difíciles, y creer en mi.
- Al CONACYT por el apoyo proporcionado durante la Maestría.
- Al departamento de Control Automático en el CINVESTAV, por darme la oportunidad de aprender de los mejores.



- A mis amigos y compañeros de la maestría, gracias por el apoyo durante este viaje.
- A mis asesores Dr. Juan Manuel Ibarra y Dr. Alejandro Malo por confiar y presionarme a ser mejor como investigadora, gracias por la paciencia que mostraron conmigo.
- A la UVM campus Tuxtla, mi universidad, gracias por darme mis primeras alas y darme las herramientas que necesite para esta etapa de mi vida

# Índice general

<b>Índice de tablas</b>	<b>VI</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Estado del Arte . . . . .	2
1.1.1. Caminado humano . . . . .	3
1.1.2. Robots Humanoides . . . . .	4
1.2. Planteamiento del problema . . . . .	6
1.3. Propósito del trabajo de tesis . . . . .	6
<b>2. Modelos</b>	<b>9</b>
2.1. Modelo Geométrico Directo . . . . .	9
2.1.1. Método Denavit-Hartenberg . . . . .	11
2.1.2. Método Khalil-Kleinfinger . . . . .	12
2.1.3. Modelado Robonova I . . . . .	13
2.2. Métodos de solución del modelo geométrico inverso . . . . .	21
2.2.1. Método de Paul . . . . .	22

# ÍNDICE GENERAL

2.2.2. Método Paul: Brazos Robonova I . . . . .	23
2.2.3. Método Paul: Piernas Robonova I . . . . .	27
2.2.3.1. Ejemplo: Ecuaciones MGI Piernas . . . . .	29
2.3. Jacobiano . . . . .	31
2.3.1. Derivación del Jacobiano . . . . .	31
2.3.1.1. Velocidad angular . . . . .	32
2.3.1.2. Velocidad lineal . . . . .	33
2.3.2. Singularidades . . . . .	35
2.3.3. Singularidades: brazo del Robonova . . . . .	36
2.3.4. Singularidades: pierna del Robonova . . . . .	39
<b>3. Hardware y Software</b>	<b>44</b>
3.1. Hardware . . . . .	44
3.1.1. Servomotor . . . . .	44
3.1.1.1. Interfaz de control HMI . . . . .	47
3.1.1.2. Interfaz Serial . . . . .	49
3.1.1.3. Tarjeta HMI . . . . .	52
3.1.2. Tarjeta de Control . . . . .	53
3.2. Software . . . . .	54
3.2.1. Simulación en SimMechanics . . . . .	54
3.2.2. Software Raspberry Pi . . . . .	56
<b>4. Caminado parametrizado</b>	<b>59</b>
4.1. Caminado humano . . . . .	59

# ÍNDICE GENERAL

4.2. Generación de Trayectoria en espacio de trabajo . . . . .	62
4.2.1. Plano sagital . . . . .	65
4.2.2. Plano Frontal . . . . .	68
4.2.3. Inicio y Final del caminado . . . . .	69
<b>5. Resultados y Conclusiones</b>	<b>71</b>
5.1. Resultados . . . . .	71
5.2. Conclusiones . . . . .	79
5.3. Trabajo Futuro . . . . .	81
<b>A. Marco Teórico</b>	<b>82</b>
A.1. Modelado . . . . .	82
<b>B. Programas</b>	<b>88</b>
B.1. Mathematica . . . . .	88
B.1.1. Transformaciones Homogeneas . . . . .	88
B.1.2. Gráfica . . . . .	89
B.1.3. Modelo Geométrico Directo completo del Robonova-I . . . . .	91
B.1.4. Modelo Geométrico Inverso para brazos . . . . .	94
B.2. Matlab . . . . .	95
B.2.1. Modelo Geométrico Inverso para piernas . . . . .	95
B.3. Python . . . . .	97
B.3.1. Módulo spline . . . . .	97
B.3.2. Módulo Check Sum . . . . .	98

## ÍNDICE GENERAL

B.3.3. Módulo Posición . . . . .	98
B.3.4. Módulo Caminado Parametrizado . . . . .	99
<b>C. Soluciones de ecuaciones de la tabla 2.5</b>	<b>108</b>
C.1. Tipo 1 . . . . .	108
C.2. Tipo 2 . . . . .	110
C.3. Tipo 3 . . . . .	110
C.4. Tipo 4 . . . . .	111
C.5. Tipo 5 . . . . .	112
<b>Bibliografía</b>	<b>114</b>

# Índice de tablas

2.1. Tabla de longitudes del Robonova . . . . .	16
2.2. Parámetros Cabeza . . . . .	18
2.3. Parámetros K-K lado izquierdo . . . . .	18
2.4. Parámetros K-K lado derecho . . . . .	19
2.5. Tipos de ecuaciones buscadas en el método Paul . . . . .	25
3.1. Características electromecánicas del servo HSR-8498HB . . . . .	46
3.2. Características del Motor DC en el servo HSR-8498HB . . . . .	46
3.3. Características del tren de engranes en el servo HSR-8498HB . . . . .	46
3.4. Características del tren de engranes en el servo HSR-8498HB . . . . .	46
3.5. Comandos seriales para EEPROM y Memoria RAM . . . . .	51
3.6. Comandos seriales . . . . .	51
3.7. Componentes de la tarjeta HMI . . . . .	52
3.8. Tabla Comparativa entre Raspberry Pi y MR C3024 . . . . .	53

# Índice de figuras

1.1. Ilustración del libro De Motu animalium de Giovanni A. Borelli [1] . . . .	4
1.2. WABOT-1(1973) [2] . . . . .	4
1.3. ASIMO - Honda . . . . .	5
2.1. Asignación de referenciales usando Denavit-Hartenberg [3] . . . . .	11
2.2. Asignación de referenciales usando Khalil-Kleinfinger [4] . . . . .	13
2.3. Robonova I CINVESTAV . . . . .	15
2.4. Longitudes de los eslabones Robonova I . . . . .	16
2.5. Articulaciones en el Robonova . . . . .	17
2.6. Referenciales Robonova . . . . .	17
2.7. Pierna Robonova I usando los ángulos de 2.33 . . . . .	30
2.8. Singularidad de la ecuación 2.70 mostrada en el brazo . . . . .	37
2.9. Singularidad de la ecuación 2.71 mostrada en el brazo . . . . .	37
2.10. Singularidad de la ecuación 2.72 mostrada en el brazo . . . . .	38
2.11. Singularidad de la ecuación 2.75 mostrada en la pierna . . . . .	41
2.12. Singularidad de la ecuación 2.77 mostrada en la pierna . . . . .	41
2.13. Singularidad de la ecuación 2.79 mostrada en la pierna . . . . .	42
2.14. Singularidad de la ecuación 2.80 mostrada en la pierna . . . . .	43
2.15. Singularidad de la ecuación 2.82 mostrada en la pierna . . . . .	43
3.1. Servomotor HSR-8498HB . . . . .	45
3.2. Esquemático procesador ATMega en servo HSR-8498HB . . . . .	48
3.3. Forma del comando serial . . . . .	49
3.4. Tarjeta HMI . . . . .	52
3.5. Vista Frontal del Estuche y Raspberry Pi . . . . .	54
3.6. Modelo Robonova-I en software CAD . . . . .	55
3.7. Simulación del Robonova-I en SimMechanics <sup>TM</sup> . . . . .	56
3.8. S.O. Raspberry Pi . . . . .	58
4.1. Planos anatómicos en un ser humano [5] . . . . .	60
4.2. Fase del caminado humano [6] . . . . .	62
4.3. Función $y = \sin(x)$ usando la interpolación segmentada cúbica . . . . .	65
4.4. Caminado de humanoide en el plano sagital [7] . . . . .	67
4.5. Caminado de humanoide en el plano frontal [7] . . . . .	69

## ÍNDICE DE FIGURAS

4.6. Pose inicial y final del robot humanoide . . . . .	70
5.1. GUI para control de caminado parametrizado del robot Robonova-I . . .	72
5.2. Trayectoria en Eje $X$ . . . . .	73
5.3. Trayectoria en Eje $Y$ . . . . .	74
5.4. Trayectoria en Eje $Z$ . . . . .	74
5.5. Trayectorias de las Articulaciones en la pierna de apoyo . . . . .	75
5.6. Trayectorias de las Articulaciones en la pierna flotante . . . . .	75
5.7. Caminado Físico VS Caminado Simulado . . . . .	76
5.8. Trayectoria del Centro de Masa en el Simulador . . . . .	78
5.9. Posición deseada vs medida . . . . .	79
A.1. Ejemplo de Junta Rotacional . . . . .	82
A.2. Ejemplo de Junta Prismática . . . . .	83



# Capítulo 1

## Introducción

Gracias a los avances tecnológicos de este siglo, se ha logrado un desarrollo gigantesco en materia de robótica, creando modelos más complejos y con más prestaciones, capaces de realizar diversas tareas y tomar decisiones cada vez más parecidas a un ser humano.

Durante este *boom* tecnológico se crea una rama de la robótica conocida como robótica humanoide. Para el desarrollo de aplicaciones y la introducción de esta tecnología a diferentes lugares, se crea a humanoides de formato comercial tipo educacional de bajas prestaciones; en esta clasificación se encuentra el *Robonova I*, el tema de esta tesis.

El *Robonova I* ha sido objeto de estudio desde el 2007 en el Laboratorio de Visión Artificial y Robótica del Departamento de Control. La tesis de maestría del Dr. Rafael Cisneros Limón [8], aborda el primer trabajo en el departamento sobre el *Robonova I*. En el trabajo del Dr. Cisneros se muestra el modelo geométrico directo, una aproximación del modelo geométrico inverso y da una propuesta de patrón de caminado. En

## CAPÍTULO 1. INTRODUCCIÓN

este trabajo, se retoma la tesis de maestría mencionada anteriormente con cambios muy específicos.

El cambio más importante a recalcar en este documento es la nueva tarjeta controladora que reemplaza a la obsoleta MR-C3024. Las implicaciones del cambio de esta tarjeta se perciben en el *software*, en la mejora de los modelos geométricos, y en el desarrollo de un caminado estable usando la teoría de caminado parametrizado.

Durante la realización de la tesis se encontraron problemas relacionados con el control de los servomotores, la conexión entre tarjeta-computadora, y otros. Los problemas fueron resueltos y su solución en el desarrollo de la tesis. Todos estos inconvenientes son normalmente generados por el cambio de la tarjeta de control; por ejemplo, el problema de control de los servomotores en forma serial se arregla con la introducción de la tarjeta *HMI* para lograr un control bidireccional.

### **1.1. Estado del Arte**

”...Un robot humanoide es un robot diseñado para asimilar el cuerpo y los movimientos de un ser humano...” [9]. Para entender esta definición primero se considerará los orígenes del estudio del caminado humano y la transición a robots humanoides.

### 1.1.1. Caminado humano

El andar de los humanos es la forma en que se lleva a cabo el desplazamiento utilizando los miembros del cuerpo humano. Este andar, llamado caminado bípedo, es una característica de los homínidos y las aves; los chimpancés, por ejemplo, pueden desenvolverse caminando sobre dos patas aunque no por un excesivo periodo de tiempo, esto se debe a que no tienen la anatomía preparada para el caminado bípedo.

Estudiar el caminar humano es uno de los objetivos de la biomecánica. La biomecánica es una disciplina científica que tiene por objeto el estudio de las estructuras de carácter mecánico que existen en los seres vivos, fundamentalmente del cuerpo humano. El interés en los patrones del movimiento humano y animal, se remonta a los tiempos pre-históricos donde se dibujaba los sistemas de locomoción humana y animal en las cavernas. Se le atribuyen a Aristóteles y Platón ser los fundadores de la Biomecánica al escribir el primer trabajo sobre los segmentos corporales, movimientos y desplazamientos de los animales [10].

Durante el renacimiento, en 1543 Leonardo da Vinci publica *De humani corporis fabrica* (Sobre la estructura del cuerpo humano [11]), una obra con mas de 300 dibujos que remarcan la estructura de los músculos y huesos del cuerpo humano. En 1680 se publica el libro *De Motu Animalium* de Giovanni Alfonso Borelli [12], donde usa las leyes mecánicas descubiertas por Galileo Galilei en el sistema oseo muscular.

Pasarían mas de 200 años y avances en matemáticas y física, para continuar los trabajos en biomecánica y avanzar al tema que se presenta en esta tesis, caminado de un robot humanoide.

## CAPÍTULO 1. INTRODUCCIÓN

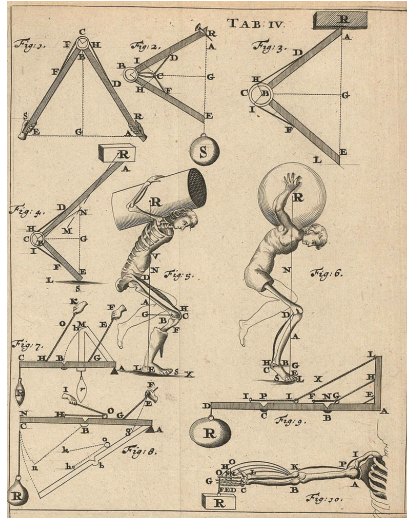


Figura 1.1: Ilustración del libro De Motu animalium de Giovanni A. Borelli [1]

### 1.1.2. Robots Humanoides

Según Shuuji Kajita [13], el primer robot humanoide fue el creado por Ichiro Kato de la Universidad de Waseda en 1973, conocido como *WABOT-1* (ver Figura 1.2). *WABOT-1* se podía reconocer objetos usando visión, entender el lenguaje hablado, manipular objetos con 2 manos y caminar en 2 piernas.

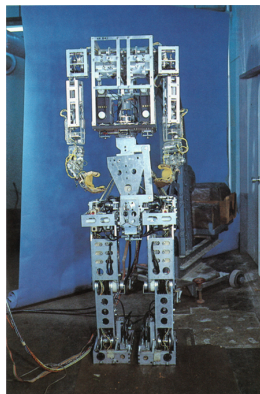


Figura 1.2: WABOT-1(1973) [2]

En 1996, se muestra el primer humanoide en tener un caminado estable en 2 pier-

## CAPÍTULO 1. INTRODUCCIÓN

nas conocido como P2 de la empresa Honda. En el año 2000, Honda presenta al mundo a ASIMO (*Advanced Step in Innovative MObility*), un robot de 1.20 metros de altura y 45 kg de peso, ASIMO es capaz de subir y bajar escaleras con gran estabilidad.



Figura 1.3: ASIMO - Honda

Del año 2000 en adelante, varias compañías buscaron lograr el mismo éxito que logro Honda con el ASIMO. Sony en el 2000 presenta a SDR-3X, un robot compacto de 50 cm de alto capaz de caminar en 2 piernas, equilibrarse en un solo pie y patear una pelota. En 2003 la empresa AIST presenta el HRP-2, un robot humanoide que incorpora una cámara 3D que le permite rodear obstáculos. En 2005, Sony presentó a Qrio, un robot humanoide a capaz de correr a una velocidad de 23 cm por segundo. En 2009, AIST presenta el HRP-C4, un robot capaz de “cantar” y bailar.

En 2011, Honda presenta al nuevo ASIMO, capaz de correr, saltar, abrir frascos y sostener y llenar un vaso de agua. En ese mismo año, Boston Dynamics desarrolla PETMAN un robot bípedo derivado del proyecto BigDog de la misma empresa en el 2005. PETMAN es el primer robot bípedo capaz de soportar empujones mientras camina sin perder el equilibrio. En 2013 se presenta Atlas, un robot bípedo avanzado creado por Boston Dynamics. Atlas puede caminar por todo tipo de terreno, sortear obstáculos y soportar fuertes impactos mientras mantienen el equilibrio sobre una

pierna.

### **1.2. Planteamiento del problema**

Desde el 2009, el Laboratorio de Visión Artificial y Robótica del Departamento de Control Automático ha participado en competencias internacionales de robots humanoides, en su mayoría de formato comercial como el Robonova y Bioloid [14], hasta llegar a el Nao y el DarwinOP.

Al pasar de los años, se ha vuelto evidente la deteriorada condición del Robonova I, el hardware y software que tenía el robot originalmente esta descontinuado. El robonova actualmente se considera un robot de bajas prestaciones en comparación con el Nao o el DarwinOP , por lo que se necesitaba mejorar el rendimiento del robot para mejorar su competitividad

Por lo que se ha mencionado anteriormente, el robot requiere que sea remplazada su tarjeta controladora original por un hardware mas reciente y de mejores prestaciones.

Al mismo tiempo se necesita que sea actualizado por completo, incluyendo modelos cinemáticos y un nuevo caminado.

### **1.3. Propósito del trabajo de tesis**

El objetivo de la tesis es desarrollar un caminado parametrizado y estable en robot tipo educacional de bajas prestaciones conocido como Robonova I. Para cumplir el propósito de la tesis, se tuvieron que desarrollar modelos, hardware y software, y pa-

## CAPÍTULO 1. INTRODUCCIÓN

trones de control actualizados además de cumplir con cualquier tarea que se necesite en el futuro.

Para lograr el objetivo propuesto se tiene que desarrollar diferentes métodos y resultados para obtener información necesaria para lograr la meta deseada. Algunos resultados derivados de la tesis que se encuentran en los siguientes capítulos serían:

- Capítulo 2: En este capítulo se da una descripción detallada del modelo geométrico directo, inverso y del jacobiano. Se explica a detalle como fueron obtenidos, se muestran figuras de referencia y se dan ejemplos para cada modelo.
- Capítulo 3: En este capítulo se describe la tarjeta de control Raspberry Pi, los servomotores HSR-8498HB y la tarjeta que ayuda a controlar estos servomotores en forma bidireccional. También se presentan todos los programas y modelos 3D que fueron desarrollados
- Capítulo 4: En este capítulo se describe que es el caminado, las fases del ciclo de caminado, y se explica la teoría detrás del caminado parametrizado. Se hace énfasis en los parámetros necesarios para el tipo de marcha propuesto en la tesis y se explica el uso de la interpolación cúbica segmentada para crear trayectorias de caminado.
- Capítulo 5: Se explican los resultados obtenidos tanto con la simulación como con el robot físico. También se expresan las conclusiones que fueron obtenidas de esta tesis.
- Apendice A: Se encuentra el marco teórico de esta tesis

## CAPÍTULO 1. INTRODUCCIÓN

- Apendice B: Se presentan los programas desarrollados en conjunto con el Dr. Alejandro Malo, para el modelo geométrico directo, inverso y caminado parametrizado.
- Apendice C: Se explica el marco teórico para la solución de ecuaciones necesarias para el modelo geométrico inverso.



# Capítulo 2

## Modelos

Los modelos son una representación de un sistema físico, usados para analizar, explicar o definir el sistema mencionado anteriormente. Los métodos usados para estos modelos son variados y dependen del sistema o procesos que se desean modelar. En robótica, el sistema que se desea modelar es un robot, en nuestro caso, un robot humanoide. Para poder analizar, visualizar y finalmente controlar a un robot humanoide se busca un modelo matemático que tenga como finalidad proporcionarnos las posiciones, velocidad y aceleraciones necesarias para cumplir una función específica, en el caso de este trabajo, la función específica sería el caminado del robot. Si se requiere marco teórico se puede encontrar en el Apéndice A.1.

### 2.1. Modelo Geométrico Directo

La cinemática es la rama de la mecánica que estudia el movimiento de un cuerpo, o un sistema de los cuerpos, sin consideración a su masa o las fuerzas que actúan

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

sobre él, dentro de esta ciencia se estudian principalmente posiciones, velocidades y aceleraciones.

El modelo geométrico, es un modelo matemático que relaciona el espacio de trabajo con el espacio articular. El robot se controla a través del control de las articulaciones (espacio articular) y realiza tareas que se definen en el espacio de trabajo. Para el modelo geométrico lo que nos interesa son las posiciones de cada elemento dentro del robot. [15]

El modelo geométrico directo tiene como finalidad determinar la posición y orientación en la que se encuentra el efector final con respecto al referencial base. Citando a [3] "...es un conjunto de relaciones que definen la posición final del robot como una función de coordenadas que depende del ángulo que forman entre ellas y las distancias que existen entre cada una de las articulaciones".

$${}^0T_n = {}^0A_1(q_1) {}^1A_2(q_2) \dots {}^{n-1}A_n(q_n) \quad (2.1)$$

Donde

$${}^{i-1}A_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}O_i \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

Donde  ${}^{i-1}A_i$  es una matriz de transformación homogénea que define la posición y orientación de  $x_i, y_i, z_i$  con respecto a  $x_{i-1}, y_{i-1}, z_{i-1}$ . Y donde  ${}^0T_n$  representa la matriz de transformación de eslabones sucesivos desde el referencial base  ${}^0R$  hasta el efector final  $R_f$ . Para la matriz de transformación desde el efector final al referencial base,

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

matemáticamente la ecuación sería

$${}^nT_0 = ({}^0T_n)^{-1} = {}^nA_{n-1}(q_n) \dots {}^2A_1(q_2) {}^1A_0(q_1) \quad (2.3)$$

### 2.1.1. Método Denavit-Hartenberg

Una convención común usada para definir los referenciales en un robot es la de Denavit-Hartenberg [16]. En esta convención cada transformación homogénea  $A_i$  es representada como el producto de transformaciones básicas

$$A_i = Rot(x, \alpha_i) Trans(x, d_i) Rot(z, \theta_i) Trans(z, r_i) \quad (2.4)$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & d_i \\ \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i & -r_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i & r_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

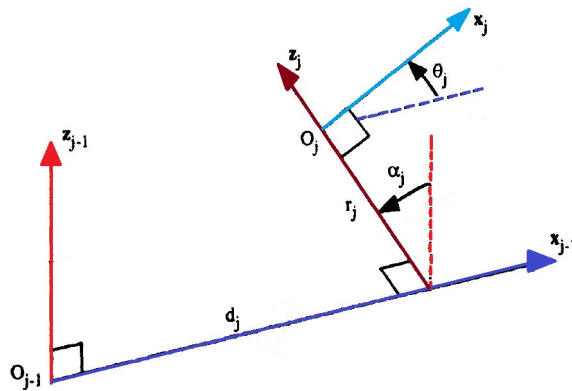


Figura 2.1: Asignación de referenciales usando Denavit-Hartenberg [3]

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Donde

- $d_i$  es la distancia de  $z_{i-1}$  a  $z_i$  medida sobre el eje  $x_{i-1}$
- $\alpha_i$  es el ángulo de  $z_{i-1}$  a  $z_i$  medido sobre el eje  $x_{i-1}$
- $r_i$  es la distancia de  $x_{i-1}$  a  $x_i$  medida sobre el eje  $z_i$
- $\theta_i$  es el ángulo de  $x_{i-1}$  a  $x_i$  medida sobre el eje  $z_i$

### 2.1.2. Método Khalil-Kleinfinger

Otro método de notación para la descripción de la cinemática de un robots es el creado por Khalil-Kleinginfer [17] (K-K), el cual es una extensión del conocido como Denavit-Hartenberg (D-H).

A pesar del que el método de D-H es el más usado para describir un modelo geométrico tiene una deficiencia cuando se aplica a eslabones que tienen mas de 2 articulaciones, lo cual puede llevar a ambigüedades, que es precisamente en lo que trabaja el método de K-K. El método de K-K en robots arborecentes, como los humanoides, se usa en caso de tener más de 2 articulaciones conectadas al mismo eslabón.

Cuando articulación se describe usando 6 parámetros, donde los primeros 4 parámetros ( $Rot(x, \alpha_j)Trans(x, d_j)Rot(z, \theta_j)Trans(z, r_j)$ ) especifican la orientación de  $j$  con respecto a  $j - 1$ , y los otros 2 parámetros ( $Rot(z, \beta_j)Trans(z, b_j)$ ) denotando el desplazamiento de la guía para la articulación base. Quedando de la siguiente manera

$${}^{j-1}T_j = Rot(z, \beta_j)Trans(z, b_j)Rot(x, \alpha_j)Trans(x, d_j)Rot(z, \theta_j)Trans(z, r_j) \quad (2.6)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Donde

- $\beta_j$  es el ángulo entre  $X_{j-1}$  y  $X_{(j-1)^*}$  sobre el eje  $Z_{j-1}$
- $b_j$  es la distancia entre  $O_{j-1}$  y  $O_{(j-1)^*}$
- $\alpha_j$  es el ángulo entre  $Z_{j-1}$  y  $Z_j$  sobre el eje  $X_{(j-1)^*}$
- $d_j$  es la distancia entre  $O_{(j-1)^*}$  y  $Z_j$
- $\theta_j$  ángulo entre  $X_{(j-1)^*}$  y  $X_j$  sobre el eje  $Z_j$
- $r_j$  es la distancia entre  $O_j$  y  $X_{(j-1)^*}$

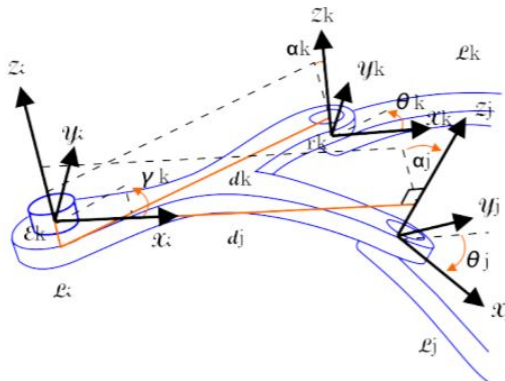


Figura 2.2: Asignación de referenciales usando Khalil-Kleinfinger [4]

### 2.1.3. Modelado Robonova I

En este trabajo de tesis, se utiliza el método de Khalil-Kleinfinger para modelar al robot humanoide Robonova 1 como un robot arborescente. Robonova 2.3 es un humanoide comercial nivel hobby fabricado por la empresa Hi-Tec, el robot cuenta con 16 servomotores digitales HSR8498HB diseñados únicamente para el Robonova unidos entre ellos por piezas de aluminio, formando una estructura de un robot humanoide

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

de 30 Cm de alto. Los servomotores tienen como características la lectura de posición, corriente, cambio de velocidad del servo, entre otros. Los servomotores están organizados como sigue

- 5 motores en la pierna derecha
- 5 motores en la pierna izquierda
- 3 motores en el brazo derecho
- 3 motores en el brazo izquierdo

A parte de lo mencionado anteriormente, se le han agregado al Robonova I, 2 motores en la cabeza los cuales realizan el movimiento de la cámara.

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

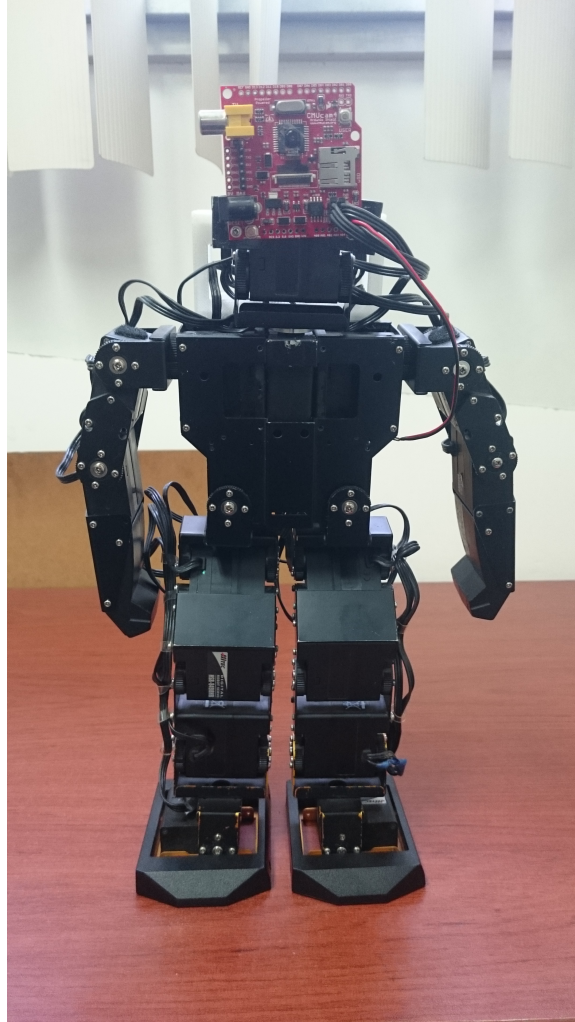


Figura 2.3: Robonova I CINVESTAV

Para empezar el modelo geométrico, necesitamos mediciones exactas de cada eslabón del Robonova I. Todas las longitudes necesarias se pueden obtener de la siguiente tabla 2.1, se han omitido algunos eslabones, gracias a la propiedad de simetría que tiene el Robonova I.

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Eslabón	Dimensión (mm)
$l_1$	80
$l_2$	73
$l_3$	65.5
$l_4$	48
$l_5$	89.8
$l_6$	28
$l_7$	40
$l_8$	58.4
$l_9$	48
$l_{10}$	40
$l_{11}$	12

Tabla 2.1: Tabla de longitudes del Robonova

La tabla hace referencia a la siguiente Figura 2.4

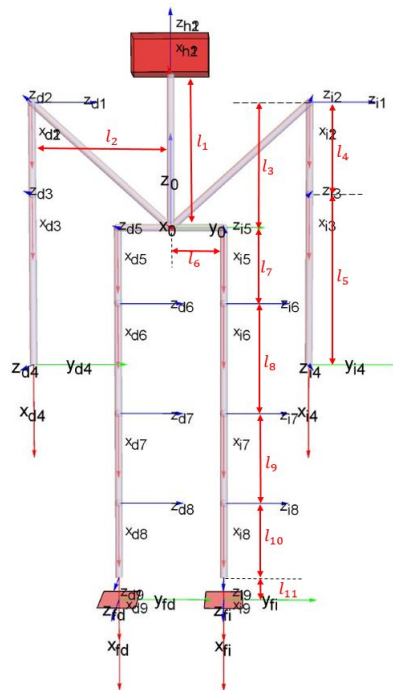


Figura 2.4: Longitudes de los eslabones Robonova I

Una vez definidas las longitudes de cada eslabón, se procede con la asignación de referenciales en cada articulación del robot humanoide, iniciando con el referencial base  ${}^0R$  ubicado en el torso. Estos referenciales se pueden observar en la Figura 2.6



## CAPÍTULO 2. MODELOS GEOMÉTRICOS

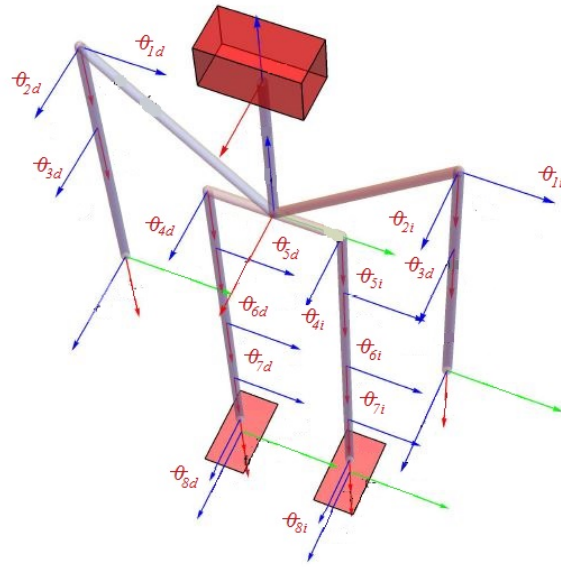


Figura 2.5: Articulaciones en el Robonova

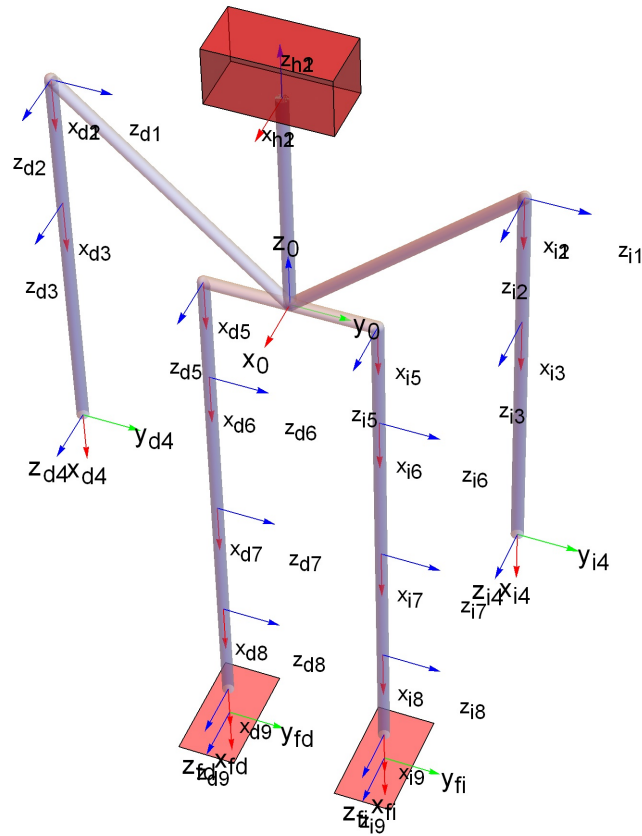


Figura 2.6: Referenciales Robonova

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

La tabla de parámetros de los referenciales de la figura 2.6 quedaría de la siguiente forma

- Parámetros Khalil-Kleinfinger: Cabeza

Eslabón	$\beta_i$	$b_i$	$\alpha_i$	$d_i$	$\theta_i$	$r_i$
$1_{head}$	0	0	0	0	$\theta_{1h}$	$l_1$
$2_{head}$	0	0	$90^\circ$	0	$\theta_{1h} + 90^\circ$	0

Tabla 2.2: Parámetros Cabeza

- Parámetros Khalil-Kleinfinger: lado izquierdo

Brazo Izquierdo						
Eslabón	$\beta_i$	$b_i$	$\alpha_i$	$d_i$	$\theta_i$	$r_i$
$1_i$	0	$l_3$	$-90^\circ$	0	$\theta_{1i} + 90^\circ$	$-l_2$
$2_i$	0	0	$90^\circ$	0	$\theta_{2i}$	0
$3_i$	0	0	0	$l_4$	$\theta_{3i}$	0
$4_i$	0	0	0	$l_5$	0	0
Pierna Izquierda						
Eslabón	$\beta_i$	$b_i$	$\alpha_i$	$d_i$	$\theta_i$	$r_i$
$5_i$	$90^\circ$	0	$90^\circ$	$l_6$	$\theta_{4i} - 90^\circ$	0
$6_i$	0	0	$-90^\circ$	$l_7$	$\theta_{5i}$	0
$7_i$	0	0	0	$l_8$	$\theta_{6i}$	0
$8_i$	0	0	0	$l_9$	$\theta_{7i}$	0
$9_i$	0	0	$90^\circ$	$l_{10}$	$\theta_{8i}$	0
$10_{fi}$	0	0	0	$l_{11}$	0	0

Tabla 2.3: Parámetros K-K lado izquierdo

- Parámetros Khalil-Kleinfinger: Lado Derecho

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Brazo Derecho						
Eslabón	$\beta_i$	$b_i$	$\alpha_i$	$d_i$	$\theta_i$	$r_i$
$1_d$	0	$l_3$	$-90^\circ$	0	$\theta_{1d} + 90^\circ$	$l_2$
$2_d$	0	0	$90^\circ$	0	$\theta_{2d}$	0
$3_d$	0	0	0	$l_4$	$\theta_{3d}$	0
$4_d$	0	0	0	$l_5$	0	0
Pierna Derecha						
Eslabón	$\beta_i$	$b_i$	$\alpha_i$	$d_i$	$\theta_i$	$r_i$
$5_d$	$90^\circ$	0	$90^\circ$	$-l_6$	$\theta_{4i} - 90^\circ$	0
$6_d$	0	0	$-90^\circ$	$l_7$	$\theta_{5i}$	0
$7_d$	0	0	0	$l_8$	$\theta_{6i}$	0
$8_d$	0	0	0	$l_9$	$\theta_{7i}$	0
$9_d$	0	0	$90^\circ$	$l_{10}$	$\theta_{8i}$	0
$10_{fd}$	0	0	0	$l_{11}$	0	0

Tabla 2.4: Parámetros K-K lado derecho

Se puede observar que entre la tabla 2.4 y la tabla 2.3 la única diferencia reside en el cambio de signo de los parámetros  $l_2$  y  $l_6$ , esto también se observa en las tablas de parámetros para las piernas. Esto se debe a la forma en que se colocaron los referenciales en cada articulación, buscando que gracias a la simetría del robot, se pudieran obtener modelos para el lado derecho como para el lado izquierdo con solo un cambio de signo, y sin cambios en otros parámetros. Debido a esto, en siguientes modelos del robot se usara la variable *signo* para determinar si se habla del lado derecho o izquierdo.

La matriz resultante desde el referencial base  ${}^0R$  al efector final del brazo  ${}^{fb}R$  sería

$${}^0T_{fb} = \begin{bmatrix} -c(\theta_2 + \theta_3)s(\theta_1) & s(\theta_1)s(\theta_2 + \theta_3) & c(\theta_1) & -s(\theta_1)(l_4c(\theta_2) + l_5c(\theta_2 + \theta_3)) \\ s(\theta_2 + \theta_3) & c(\theta_2 + \theta_3) & 0 & l_2\textit{signo} + l_4s(\theta_2) + l_5s(\theta_2 + \theta_3) \\ -c(\theta_1)c(\theta_2 + \theta_3) & c(\theta_1)s(\theta_2 + \theta_3) & -s(\theta_1) & l_3 - c(\theta_1)(l_4c(\theta_2) + l_5c(\theta_2 + \theta_3)) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

La matriz resultante desde el referencial base  ${}^0R$  al efector final de la pierna  ${}^{fp}R$  sería

$${}^0T_{fp} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \quad (2.8)$$

Donde

$$a_{11} = -c(\theta_8)s(\theta_5 + \theta_6 + \theta_7)$$

$$a_{12} = c(\theta_5 + \theta_6 + \theta_7)c(\theta_8)s(\theta_4) + c(\theta_4)s(\theta_8)$$

$$a_{13} = -c(\theta_4)c(\theta_5 + \theta_6 + \theta_7)c(\theta_8) + s(\theta_4)s(\theta_8)$$

$$a_{14} = 0$$

$$a_{21} = s(\theta_8)s(\theta_5 + \theta_6 + \theta_7)$$

$$a_{22} = c(\theta_4)c(\theta_8) - c(\theta_5 + \theta_6 + \theta_7)s(\theta_8)s(\theta_4)$$

$$a_{23} = s(\theta_4)c(\theta_8) + c(\theta_5 + \theta_6 + \theta_7)s(\theta_8)c(\theta_4)$$

$$a_{24} = 0$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

$$a_{31} = c(\theta_5 + \theta_6 + \theta_7)$$

$$a_{32} = s(\theta_4)s(\theta_5 + \theta_6 + \theta_7)$$

$$a_{33} = -c(\theta_4)s(\theta_5 + \theta_6 + \theta_7)$$

$$a_{34} = 0$$

$$a_{41} = -l_8s(\theta_5) - (l_9s(\theta_5 + \theta_6) + (l_{10} + l_{11}c(\theta_8))s(\theta_5 + \theta_6 + \theta_7))$$

$$a_{42} = l_6\text{signo} + (l_7 + l_8c(\theta_5) + l_9c(\theta_5 + \theta_6) + l_{10}c(\theta_5 + \theta_6 + \theta_7) + l_{11}c(\theta_5 + \theta_6 + \theta_7)c(\theta_8))s(\theta_4) \\ + l_{11}c(\theta_4)s(\theta_8)$$

$$a_{43} = -c(\theta_4)(l_7 + l_8c(\theta_5) + l_9c(\theta_5 + \theta_6) + l_{10}c(\theta_5 + \theta_6 + \theta_7) + l_{11}c(\theta_5 + \theta_6 + \theta_7)c(\theta_8)) \\ + l_{11}s(\theta_4)s(\theta_8)$$

## 2.2. Métodos de solución del modelo geométrico inverso

Se le conoce como problema geométrico inverso, cuando se desea conocer las variables articulares que llevan al efector final a una posición deseada. Este problema tiene múltiples soluciones y puede ser muy complejo dependiendo de la geometría del robot. El modelo geométrico inverso (MGI) nos sirve para presentar posibles soluciones a lo mencionado anteriormente. Específicamente para el Robonova I, el método usado es el propuesto por Paul [18] en 1981. Se escoge este método debido a las siguientes observaciones en las tablas 2.4,2.3.

- Dentro de los parámetros de distancia se encuentran muchos ceros.

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

- Dentro de los parámetros de ángulos se encuentran muchos ángulos de  $90^\circ$

Estas observaciones nos ayudan a escoger el método de Paul como el mas efectivo para el problema de modelo geométrico inverso y así obtener analíticamente todos los resultados posibles.

### 2.2.1. Método de Paul

Considerando un robot cuya matriz tiene la siguiente forma

$${}^0A_n = {}^0A_1(q_1) {}^1A_2(q_2) \dots {}^{n-1}A_n(q_n) \quad (2.9)$$

Sea la posición deseada

$${}^0U = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Donde el  $\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$  es el vector de posición y  $\begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix}$  la matriz de rotación deseada

El MGI se obtiene resolviendo la ecuación

$${}^0U = {}^0A_n = {}^0A_1(q_1) {}^1A_2(q_2) \dots {}^{n-1}A_n(q_n) \quad (2.11)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Donde  ${}^0U$  es la pose deseada. La solución de la ecuación 2.11 se puede encontrar con el método de Paul. El método propone mover la variable articular  $q_i$  al lado izquierdo una por una pre multiplicando la ecuación por  ${}^{j-1}A_j$  variando  $j = 1 \dots n - 1$  y resolviendo articulación por articulación

$${}^0U = {}^0T_1 {}^1T_2 {}^2T_3$$

$${}^1T_0 {}^0U = {}^1T_2 {}^2T_3$$

$${}^2T_1 {}^1U = {}^2T_3$$

### 2.2.2. Método Paul: Brazos Robonova I

Se desea conocer el modelo geométrico inverso de los brazos del Robonova I, es decir se desea conocer las variables articulares  $\theta_1, \theta_2, \theta_3$  que llevan al efector final a una posición deseada  $T_p(x_f, y_f, z_f)$ . Se tiene la siguiente matriz desde la base  ${}^0R$  al efector final

$${}^0U = {}^0 T_1(\theta_1) {}^1T_2(\theta_2) {}^2T_3(\theta_3) T_p(x_f, y_f, z_f) \quad (2.12)$$

Expandimos los términos de 2.12 en transformaciones básicas

$${}^0U = T_z(l_1) R_x(-\Pi/2) R_z(\theta_1) R_z(\Pi/2) \cdot T_z(l_2 \text{ signo}) R_x(\Pi/2) R_z(\theta_2) T_x(l_4) R_z(\theta_3) T_p(l_5, 0, 0) \quad (2.13)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Usamos el método de Paul que se discutió antes y obtener expresiones de cada variable articular.

$$t_1 = R_z(\theta_2)T_x(l_4)R_z(\theta_3)T_p(l_5, 0, 0)$$

$$U_1 = R_x(-\Pi/2)T_z(-l_2 \text{signo})R_z(-\Pi/2)R_z(-\theta_1)R_x(\Pi/2)T_z(-l_1)^0 U$$

$$U_1 = t_1$$

En forma de matriz seria

$$\begin{bmatrix} -s_z c\theta_1 - s_x s\theta_1 & -n_z c\theta_1 - n_x s\theta_1 & -a_z c\theta_1 - a_x s\theta_1 & (l_1 - P_z)c\theta_1 - P_x s\theta_1 \\ s_y & n_y & a_y & P_y - l_2 \text{signo} \\ s_x c\theta_1 - s_z s\theta_1 & n_x c\theta_1 - n_z s\theta_1 & a_x c\theta_1 - a_z s\theta_1 & P_x c\theta_1 + (l_1 - P_z)s\theta_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c(\theta_2 + \theta_3) & -s(\theta_2 + \theta_3) & 0 & l_4 c\theta_2 + l_5 c(\theta_2 + \theta_3) \\ s(\theta_2 + \theta_3) & c(\theta_2 + \theta_3) & 0 & l_4 s\theta_2 + l_5 s(\theta_2 + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

Para resolver la ecuación 2.12, analizamos la matriz 2.14 en búsqueda de ciertas ecuaciones como las mostradas en la Tabla 2.5 cuya solución se encuentra en el apéndice B.



## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Tipo 1	$XS\theta_i + YC\theta_i = Z$
Tipo 2	$X_1S\theta_i + Y_1C\theta_i = Z_1$ $X_2S\theta_i + Y_2C\theta_i = Z_2$
Tipo 3	$WS\theta_j = XC\theta_i + YS\theta_i + Z_1$ $WC\theta_j = XS\theta_i - YC\theta_i + Z_2$
Tipo 4	$W_1C\theta_j + W_2S\theta_j = XC\theta_i + YS\theta_i + Z_1$ $W_1S\theta_j - W_2C\theta_j = XS\theta_i - YC\theta_i + Z_2$
Tipo 5	$XC\theta_i + YC(\theta_i + \theta_j) = Z_1$ $XS\theta_i + YS(\theta_i + \theta_j) = Z_2$

Tabla 2.5: Tipos de ecuaciones buscadas en el método Paul

Revisando la matriz 2.14 se observan que las siguientes ecuaciones cumplen con el criterio antes propuesto

$$l_4c\theta_2 + l_5c(\theta_2 + \theta_3) = (l_1 - P_z)c\theta_1 - P_x s\theta_1 \quad (2.15)$$

$$l_4s\theta_2 + l_5s(\theta_2 + \theta_3) = P_y - l_2 \text{signo} \quad (2.16)$$

$$P_x c\theta_1 + (l_1 - P_z)s\theta_1 = 0 \quad (2.17)$$

De 2.17 podemos obtener  $\theta_1$

$$P_x c\theta_1 = (-l_1 + P_z)s\theta_1 \quad (2.18)$$

$$\frac{s\theta_1}{c\theta_1} = \frac{P_x}{-l_1 + P_z} \quad (2.19)$$

$$\theta_1 = \tan^{-1} \left( \frac{P_x}{-l_1 + P_z} \right) \quad (2.20)$$

De la ecuación 2.15 y la ecuación 2.16, se pueden obtener las ecuaciones para resolver los ángulos  $\theta_2$  y  $\theta_3$ . Estas ecuaciones se relacionan con las ecuaciones de tipo 5 de la tabla 2.5 y su solución se encuentra en el apéndice B. Usando el apéndice B, la

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

ecuación tipo 5 sería

$$XC\theta_i + YC(\theta_i + \theta_j) = Z_1$$

$$XS\theta_i + YS(\theta_i + \theta_j) = Z_2$$

Haciendo referencia a las ecuaciones 2.15 y 2.16, se tiene que

$$Z_1 = (l_1 - P_z)c\theta_1 - P_x s\theta_1$$

$$Z_2 = P_y - l_2 \text{signo}$$

$$X = l_4$$

$$Y = l_5$$

Por lo que, se tiene que  $\theta_3$  sería

$$\theta_3 = \tan^{-1} \left( \frac{\sqrt{1 - \left( \frac{Z_1^2 + Z_2^2 - X^2 - Y^2}{2XY} \right)^2}}{\frac{Z_1^2 + Z_2^2 - X^2 - Y^2}{2XY}} \right) \quad (2.21)$$

Para resolver  $\theta_2$ , se necesitan definir las siguientes variables

$$B1 = X + YC\theta_3 \quad (2.22)$$

$$B2 = YS\theta_3 \quad (2.23)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Una vez teniendo esas variables  $\theta_2$  sería

$$\theta_2 = \tan^{-1} \left( \frac{\frac{B_1 Z_2 - B_2 Z_1}{B_1^2 + B_2^2}}{\frac{B_1 Z_1 + B_2 Z_2}{B_1^2 + B_2^2}} \right) \quad (2.24)$$

Por lo que las ecuaciones del modelo geométrico inverso para los brazos, sería

$$\theta_1 = \frac{P_x}{-l_1 + P_z} \quad (2.25)$$

$$\theta_2 = \tan^{-1} \left( \frac{\frac{B_1 Z_2 - B_2 Z_1}{B_1^2 + B_2^2}}{\frac{B_1 Z_1 + B_2 Z_2}{B_1^2 + B_2^2}} \right) \quad (2.26)$$

$$\theta_3 = \tan^{-1} \left( \frac{\sqrt{1 - \left( \frac{Z_1^2 + Z_2^2 - X^2 - Y^2}{2XY} \right)^2}}{\frac{Z_1^2 + Z_2^2 - X^2 - Y^2}{2XY}} \right) \quad (2.27)$$

### 2.2.3. Método Paul: Piernas Robonova I

Usando el mismo método que se uso anteriormente, las ecuaciones del MGI para las piernas, serían

$$\theta_4 = \tan^{-1} \left( \frac{P_y - l_6 \text{signo} - l_{11} s_y}{-(P_z - l_{11} s_z)} \right) \quad (2.28)$$

$$\theta_5 = \tan^{-1} \left( \frac{\frac{B_1 R_2 - B_2 R_1}{B_1^2 + B_2^2}}{\frac{B_1 R_1 + B_2 R_2}{B_1^2 + B_2^2}} \right) \quad (2.29)$$

$$\theta_6 = \tan^{-1} \left( \frac{\sqrt{1 - \left( \frac{R_1^2 + R_2^2 - X^2 - Y^2}{2XY} \right)^2}}{\frac{R_1^2 + R_2^2 - X^2 - Y^2}{2XY}} \right) \quad (2.30)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

$$\theta_7 = \tan^{-1} \left( \frac{\sqrt{1 - \left( \frac{R_3^2 - l_7^2 - l_9^2}{2l_7l_9} \right)^2}}{\frac{R_3^2 - l_7^2 - l_9^2}{2l_7l_9}} \right) \quad (2.31)$$

$$\theta_8 = \tan^{-1} \left( \frac{s_y \cos \theta_6 + s_z \cos \theta_6}{n_y \cos \theta_6 + n_z \cos \theta_6} \right) \quad (2.32)$$

Donde

$$X = l_8$$

$$Y = l_9$$

$$R_1 = -l_7 + c\theta_4(-P_z + l_{11}s_z + l_7s_zc\theta_8 - l_7n_zs\theta_8) + (P_y - l_6signo - l_{11}s_y - l_7s_yc\theta_8 + l_7n_ys\theta_8)s\theta_4$$

$$R_2 = -P_1 + l_{11}s_x + l_7s_xc\theta_8 - l_7n_xs\theta_8$$

$$B_1 = X + Yc\theta_6$$

$$B_2 = Ys\theta_6$$

$$\begin{aligned} R_3 = & l_7^2 + l_8^2 + P_x^2 + P_y^2 + P_z^2 - 2l_6P_ysigno + l_6^2signo^2 - 2l_{11}P_xs_x + l_{11}^2s_x^2 - 2l_{11}P_ys_y \\ & + 2l_{11}l_6signos_y + l_{11}^2s_y^2 - 2l_{11}P_zs_z + l_{11}^2s_z^2 + 2(P_z - l_{11}s_z)c\theta_4(l_7 + l_8c\theta_5) - 2l_7P_ys\theta_4 + \\ & 2l_7l_6signos\theta_4 + 2l_{11}l_7s_ys\theta_4 + 2l_8c\theta_5(l_7 + (-P_y + l_6signo + l_{11}s_y)s\theta_4) + 2l_8P_xs\theta_5 \\ & - 2l_{11}l_8s_xs\theta_5 \end{aligned}$$

### 2.2.3.1. Ejemplo: Ecuaciones MGI Piernas

Comprobaremos las ecuaciones que se obtuvieron resolviendo el Modelo Geométrico Inverso. Para la comprobación, primero usaremos el modelo geométrico directo para obtener la matriz resultante desde el referencial base al efector final de la pierna, usaremos ángulos arbitrarios tales como

$$\theta_4 = 0,1745 \text{ rad} \quad (2.33)$$

$$\theta_5 = 0,3490 \text{ rad} \quad (2.34)$$

$$\theta_6 = -0,5235 \text{ rad} \quad (2.35)$$

$$\theta_7 = 0,6981 \text{ rad} \quad (2.36)$$

$$\theta_8 = -0,1745 \text{ rad} \quad (2.37)$$

Usando estos ángulos, la matriz resultante de la base al efector final en la pierna es la siguiente

$$\begin{bmatrix} -0,4924 & -0,0868 & 0,866025 & -3,75477 \\ -0,0229 & 0,99596 & 0,0868 & 5,8424 \\ -0,87 & 0,0229 & -0,4924 & -18,4545 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.38)$$

La pierna se puede observar en la figura 2.7

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

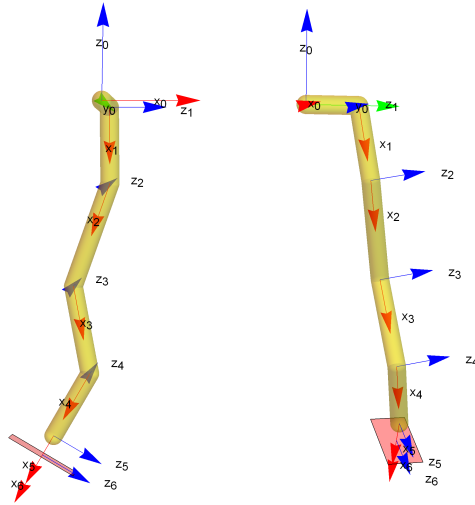


Figura 2.7: Pierna Robonova I usando los ángulos de 2.33

Usando las ecuaciones del MGI, y el programa mostrado en el apéndice B, se obtienen los siguientes resultados.

$$\theta_4 = 0,174533 \text{ rad} \quad (2.39)$$

$$\theta_5 = 0,349066 \text{ rad} \quad (2.40)$$

$$\theta_6 = -0,523599 \text{ rad} \quad (2.41)$$

$$\theta_7 = 0,698132 \text{ rad} \quad (2.42)$$

$$\theta_8 = -0,174533 \text{ rad} \quad (2.43)$$

Y así comprobamos las ecuaciones del MGI como correctas.

## 2.3. Jacobiano

La matriz jacobiana es una matriz formada por las derivadas parciales de primer orden de una función [19]. En robótica, la matriz jacobiana relaciona las velocidades de la articulación con velocidades cartesianas. Las aplicaciones del jacobiano son extensas, por ejemplo, sirve en la determinación de singularidades, en la derivación de las ecuaciones dinámicas del robot, y otras. [20] [14]

La ecuación queda como sigue

$$\dot{X} = J(q)\dot{q} \quad (2.44)$$

Donde  $J(q)$  es la matriz jacobiana. Siendo  $q$  la variable articular que llamamos  $\theta$

### 2.3.1. Derivación del Jacobiano

Consideremos una cadena con  $n$  eslabones y denotamos a  $\theta$  como variable articular.

$${}^0T_n(\theta) = \begin{bmatrix} {}^0R_n(\theta) & {}^0O_n(\theta) \\ 0 & 1 \end{bmatrix} \quad (2.45)$$

Siendo  ${}^0T_n$  la transformación entre el efector final y el referencial base. La velocidad del efector final se puede describir mediante una componente lineal  ${}^0v_n$  y una angular  ${}^0w_n$ , todo respecto al referencial base como se muestra

$$\begin{bmatrix} {}^0v_n \\ {}^0w_n \end{bmatrix} = {}^0J_n\dot{\theta} \quad (2.46)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

donde  ${}^0J_n$  representa el jacobiano entre los referenciales mencionados anteriormente y se expresa como

$${}^0J_n = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \quad (2.47)$$

Es decir, la descomposición en 2 submatrices, una la velocidad lineal y otra la velocidad angular [21].

### 2.3.1.1. Velocidad angular

Las velocidades angulares, pueden sumarse vectorialmente siempre que esten expresadas en el mismo referencial. Ya que solo tenemos articulaciones rotacionales en el robot humanoide, solo se mostrará la velocidad angular en una articulación rotacional.

La velocidad angular del eslabón  $i$  expresada en el referencial  $i - 1$  está dada por

$${}^{i-1}w_i = \dot{\theta}_i ({}^{i-1}z) = \dot{\theta}_i k \quad (2.48)$$

Donde

$$k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (2.49)$$

Entonces la velocidad angular del efector final con respecto al referencial base se hace de la siguiente manera

$${}^0w_n = \sum_{i=1}^n \dot{\theta}_i [{}^{i-1}z] \quad (2.50)$$



## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Donde  ${}^{i-1}z = {}^{i-1}R_0k$ . Sustituyendo quedaría la matriz

$$J_w = \begin{bmatrix} {}^0z & \dots & {}^{n-1}z \end{bmatrix} \quad (2.51)$$

### 2.3.1.2. Velocidad lineal

La velocidad lineal del efector final esta dada por  ${}^0\dot{o}_n$

$${}^0\dot{o}_n = \sum_{i=1}^n \frac{\partial {}^0o_n}{\partial \theta_i} \dot{\theta}_i \quad (2.52)$$

Así que la  $i$ -ésima columna de  $J_v$  sería

$$J_v = \frac{\partial {}^0o_n}{\partial \theta_i} \quad (2.53)$$

Para el caso de la articulación rotacional, se tiene que

$$\frac{\partial {}^0o_n}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} [{}^0R_i{}^i o_n + {}^0R_{i-1}{}^{i-1} o_i] \quad (2.54)$$

$$= \frac{\partial}{\partial \theta_i} {}^0R_i{}^i o_n + {}^0R_{i-1} \frac{\partial}{\partial \theta_i} {}^{i-1} o_i \quad (2.55)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Derivando la ecuación 2.55 y dado que el movimiento del eslabón  $i$  es una rotación alrededor del eje  $z_{i-1}$  se tiene que

$${}^0R_{i-1} \frac{\partial}{\partial \theta_i} {}^{i-1}o_i = {}^0R_{i-1} S(k\dot{\theta}_i) {}^{i-1}o_i \quad (2.56)$$

$$= {}^0R_{i-1} S(k\dot{\theta}_i) ({}^0R_{i-1})^T ({}^0R_{i-1}) {}^{i-1}o_i \quad (2.57)$$

$$= S({}^0R_{i-1} k\dot{\theta}_i) ({}^0R_{i-1}) {}^{i-1}o_i \quad (2.58)$$

$$= \dot{\theta}_i S({}^0z_{i-1}) ({}^0R_{i-1}) {}^{i-1}o_i \quad (2.59)$$

Lo mismo pasaría con el primer término de la ecuación 2.55, la ecuación final sería

$$= \dot{\theta}_i S({}^0z_{i-1}) [{}^0R_i {}^i o_n + {}^0R_{i-1} {}^{i-1}o_i] \quad (2.60)$$

$$= \dot{\theta}_i S({}^0z_{i-1}) ({}^0o_n - {}^0o_{i-1}) \quad (2.61)$$

$$= \dot{\theta}_i {}^0z_{i-1} \times ({}^0o_n - {}^0o_{i-1}) \quad (2.62)$$

Donde  $R$  es la matriz rotacional de un referencial a otro y  $S$  una matriz anti simétrica (vease referencia [22]). Por lo que

$$J_v = z_{i-1} \times (o_n - o_{i-1}) \quad (2.63)$$

Ahora el jacobiano para una articulación rotacional sería

$$J_i = \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} \quad (2.64)$$

### 2.3.2. Singularidades

El jacobiano, una matriz de  $6 \times n$  llamada  $J(\theta)$  define un mapeo entre el vector de velocidades articulares y la velocidad del efector final [23] [24]

$$\begin{bmatrix} v_f \\ w_f \end{bmatrix} = J(\theta)\dot{\theta} \quad (2.65)$$

Lo anterior implica que las velocidades del efector final son combinaciones lineales de la matriz del jacobiano. Ya se había mencionado anteriormente que es el rango de una matriz, en el caso de que la matriz jacobiana tenga rango de  $J = 6$ , el efector final puede tener cualquier velocidad. Este rango, por razones obvias no es constante y depende de las configuraciones que tengan las articulaciones.

Las configuraciones en las cuales la matriz jacobiana  $J(\theta)$  tiene un rango inferior al máximo valor del rango de la matriz se le conocen como singularidades y estas representan

- Ciertas direcciones de movimiento que pueden ser inalcanzables
- Pueden corresponder a puntos donde se alcanza la frontera del espacio de trabajo
- Ciertas configuraciones donde no existe una solución única al problema del modelo geométrico inverso

Para conocer las singularidades, se dice que una configuración de  $\theta$  es singular si y solo si

$$\text{Det } J(\theta) = 0 \quad (2.66)$$

### 2.3.3. Singularidades: brazo del Robonova

En la ecuación 2.67 se muestra el jacobiano del brazo desde el referencial base al efector final

$${}^0J_{fb} = \begin{bmatrix} -c\theta_1 [l_4c\theta_2 + l_5c(\theta_2 + \theta_3)] & s\theta_1 [l_4s\theta_2 + l_5s(\theta_2 + \theta_3)] & l_5s\theta_1s(\theta_2 + \theta_3) \\ 0 & l_4c\theta_2 + l_5c(\theta_2 + \theta_3) & l_5c(\theta_2 + \theta_3) \\ s\theta_1 [l_4c\theta_2 + l_5c(\theta_2 + \theta_3)] & c\theta_1 [l_4s\theta_2 + l_5s(\theta_2 + \theta_3)] & l_5c\theta_1s(\theta_2 + \theta_3) \\ 0 & c\theta_1 & c\theta_1 \\ 1 & 0 & 0 \\ 0 & -s\theta_1 & -s\theta_1 \end{bmatrix} \quad (2.67)$$

Se puede observar que la matriz  ${}^0J_{fb}$  no es de rango pleno, es de rango  $6 \times 3$ , debido a eso usaremos los menores para sacar el determinante de la matriz. En álgebra lineal, un menor de una matriz A es el determinante de alguna submatriz, obtenido de A mediante la eliminación de una o más de sus filas o columnas. [25]. Se obtienen los siguientes determinantes

$$s\theta_1(l_4c\theta_2 + l_5c(\theta_2 + \theta_3))s\theta_3 \quad (2.68)$$

$$c\theta_1(l_4c\theta_2 + l_5c(\theta_2 + \theta_3))s\theta_3 \quad (2.69)$$

La primera singularidad que se encuentra lo da la ecuación 2.70

$$l_4c\theta_2 + l_5c(\theta_2 + \theta_3) \quad (2.70)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

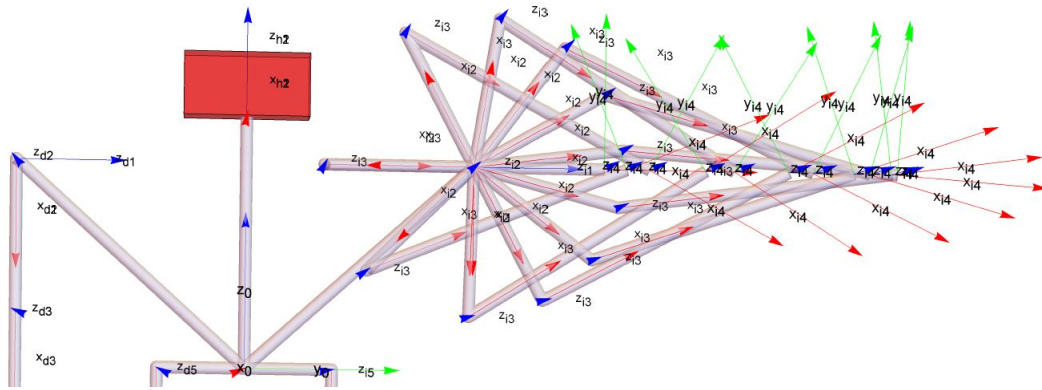


Figura 2.8: Singularidad de la ecuación 2.70 mostrada en el brazo

Como se observa en la Figura 2.8, la singularidad se muestra cuando el efector final queda en el mismo eje de movimiento que el ángulo  $\theta_1$ , es decir, no importa que ángulo tenga  $\theta_1$ , el efector final siempre tendrá la misma posición.

La siguiente singularidad esta dada por la ecuación 2.71

$$s\theta_3 = 0 \rightarrow \theta_3 = 0, 180 \quad (2.71)$$

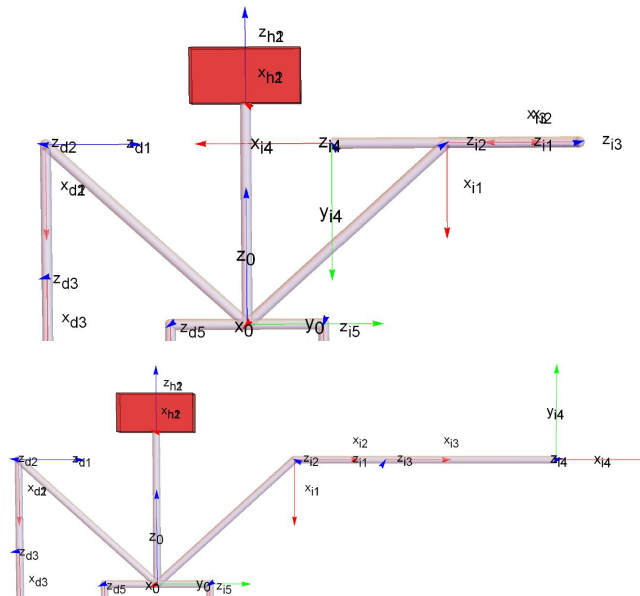


Figura 2.9: Singularidad de la ecuación 2.71 mostrada en el brazo

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Donde la Figura 2.9 corresponde al brazo completamente extendido o retraído.

Otra singularidad se da por la ecuación 2.72

$$s\theta_1 = 0 \rightarrow \theta_1 = 0, 180 \quad (2.72)$$

$$c\theta_1 = 0 \rightarrow \theta_1 = 90, -90 \quad (2.73)$$

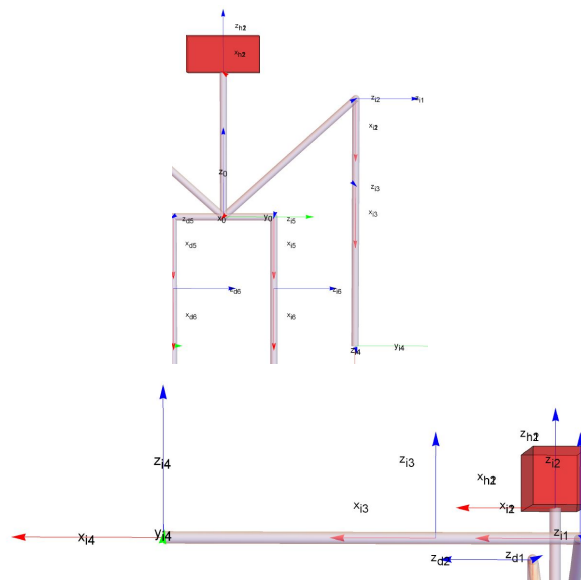


Figura 2.10: Singularidad de la ecuación 2.72 mostrada en el brazo

Donde la Figura 2.10, muestra a  $\theta_1$  a  $0^\circ$  y a  $90^\circ$  que es donde la matriz jacobiana pierde rango, lo cual lo vuelve una singularidad.

### 2.3.4. Singularidades: pierna del Robonova

En la ecuación 2.74 se muestra el jacobiano de la pierna desde el referencial base al efector final.

$${}^0J_{fp} = \begin{bmatrix} 0 & b_2 & b_3 & b_4 & 0 \\ b_6 & b_7 & b_8 & b_9 & 0 \\ b_{11} & b_{12} & b_{13} & b_{14} & 0 \\ 1 & 0 & 0 & 0 & c(\theta_5 + \theta_6 + \theta_7) \\ 0 & c(\theta_4) & c(\theta_4) & c(\theta_4) & s(\theta_4)s(\theta_5 + \theta_6 + \theta_7) \\ 0 & s(\theta_4) & s(\theta_4) & s(\theta_4) & -c(\theta_4)s(\theta_5 + \theta_6 + \theta_7) \end{bmatrix} \quad (2.74)$$

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

Donde

$$b_2 = -l_8 c \theta_5 - l_9 c(\theta_5 + \theta_6) - l_{10} c(\theta_5 + \theta_6 + \theta_7)$$

$$b_3 = -l_9 c(\theta_5 + \theta_6) - l_{10} c(\theta_5 + \theta_6 + \theta_7)$$

$$b_4 = -l_{10} c(\theta_5 + \theta_6 + \theta_7)$$

$$b_6 = c \theta_4 (l_7 + l_8 c \theta_5 + l_9 c(\theta_5 + \theta_6) + l_{10} c(\theta_5 + \theta_6 + \theta_7))$$

$$b_7 = -s \theta_4 (l_8 s \theta_5 + l_9 s(\theta_5 + \theta_6) + l_{10} s(\theta_5 + \theta_6 + \theta_7)) b_8 = -s \theta_4 (l_9 s(\theta_5 + \theta_6) + l_{10} s(\theta_5 + \theta_6 + \theta_7))$$

$$b_9 = -s \theta_4 (l_{10} s(\theta_5 + \theta_6 + \theta_7))$$

$$b_{11} = s \theta_4 (l_7 + l_8 c \theta_5 + l_9 c(\theta_5 + \theta_6) + l_{10} c(\theta_5 + \theta_6 + \theta_7))$$

$$b_{12} = c \theta_4 (l_8 s \theta_5 + l_9 s(\theta_5 + \theta_6) + l_{10} s(\theta_5 + \theta_6 + \theta_7))$$

$$b_{13} = c \theta_4 (l_9 s(\theta_5 + \theta_6) + l_{10} s(\theta_5 + \theta_6 + \theta_7))$$

$$b_{14} = c \theta_4 (l_{10} s(\theta_5 + \theta_6 + \theta_7))$$

Usando el mismo método que se uso para el jacobiano de los brazos, usamos los menores para obtener el determinante, y las ecuaciones obtenidas son las siguientes

$$c \theta_4 \rightarrow \theta_4 = 90, -90 \quad (2.75)$$

$$s \theta_4 \rightarrow \theta_4 = 0, 180 \quad (2.76)$$

$$c \theta_8 \rightarrow \theta_8 = 90, -90 \quad (2.77)$$

$$s \theta_8 \rightarrow \theta_8 = 0, 180 \quad (2.78)$$

$$s \theta_6 \rightarrow \theta_6 = 0, 180 \quad (2.79)$$



## CAPÍTULO 2. MODELOS GEOMÉTRICOS

$$s(\theta_5 + \theta_6 + \theta_7) \rightarrow \theta_5 + \theta_6 + \theta_7 = 0, 180 \quad (2.80)$$

$$c(\theta_5 + \theta_6 + \theta_7) \rightarrow \theta_5 + \theta_6 + \theta_7 = 90, -90 \quad (2.81)$$

$$l_7 + l_8 c\theta_5 + l_9 c(\theta_5 + \theta_6) + l_{10} c(\theta_5 + \theta_6 + \theta_7) \quad (2.82)$$

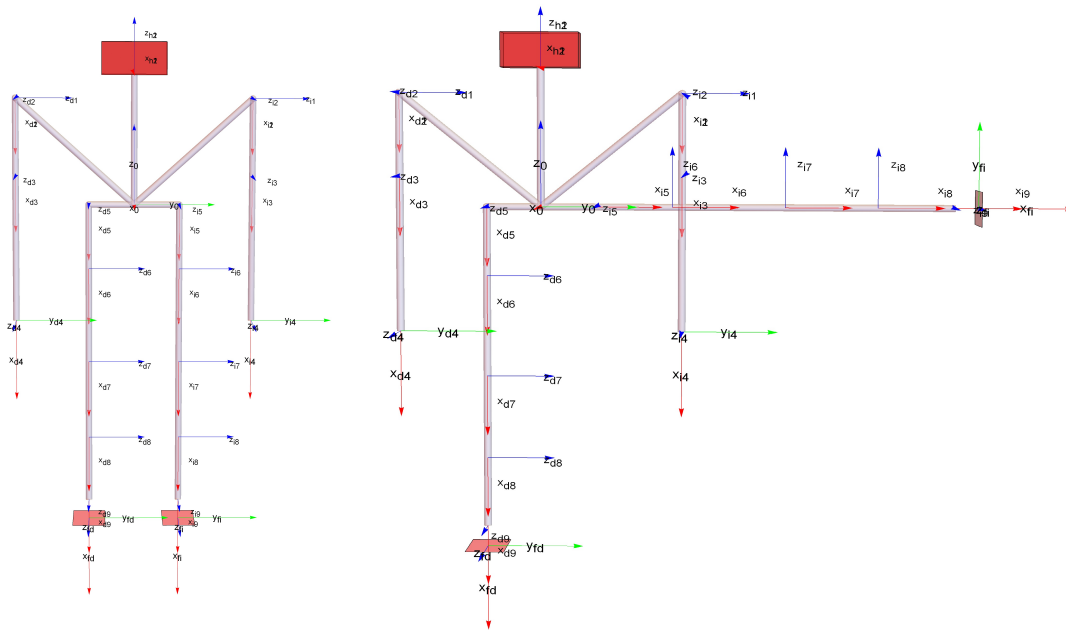


Figura 2.11: Singularidad de la ecuación 2.75 mostrada en la pierna

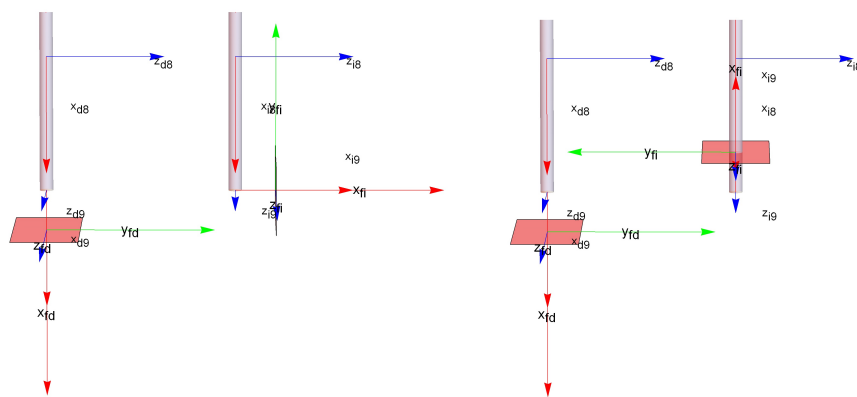


Figura 2.12: Singularidad de la ecuación 2.77 mostrada en la pierna

CAPÍTULO 2. MODELOS GEOMÉTRICOS

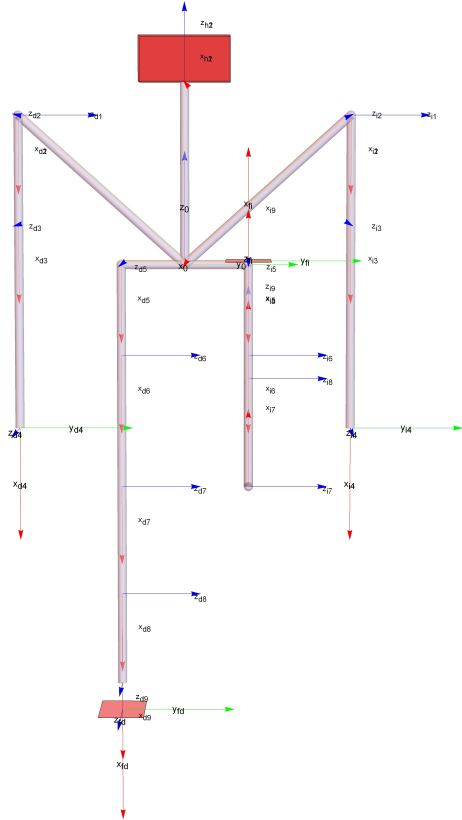


Figura 2.13: Singularidad de la ecuación 2.79 mostrada en la pierna

## CAPÍTULO 2. MODELOS GEOMÉTRICOS

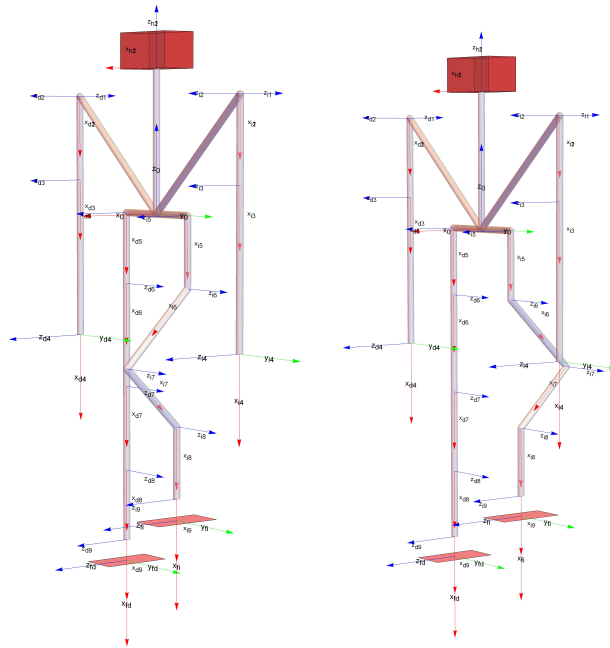


Figura 2.14: Singularidad de la ecuación 2.80 mostrada en la pierna

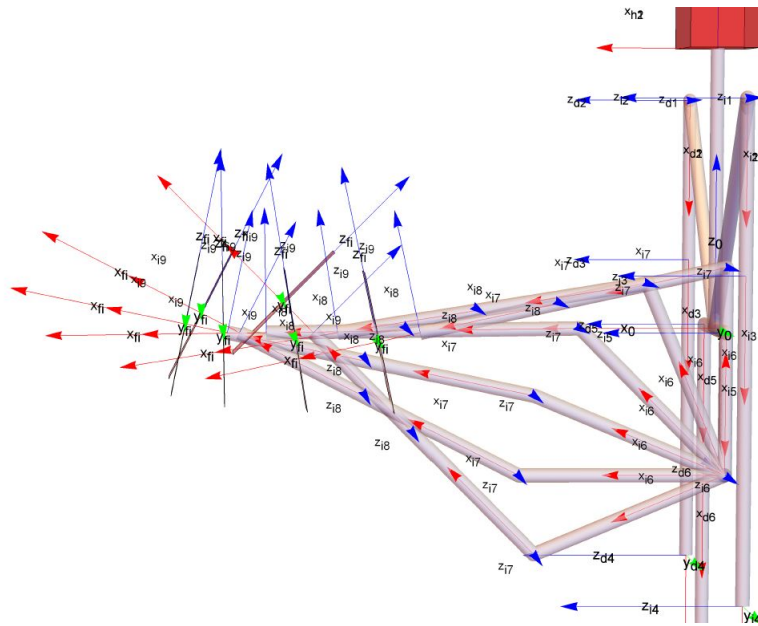


Figura 2.15: Singularidad de la ecuación 2.82 mostrada en la pierna

# Capítulo 3

## Hardware y Software

### 3.1. Hardware

La palabra *hardware* se refiere a las partes físicas tangibles de un sistema informático, es decir, sus componentes electrónicos, eléctricos, electromecánicos y mecánicos [26]. La Real Academia Española lo define como "...Conjunto de los componentes que integran la parte material de una computadora".

#### 3.1.1. Servomotor

Un servomotor, normalmente llamado servo, es un dispositivo que además de generar movimiento rotacional posee la capacidad de mantener fija y estable una posición angular determinada dentro de su rango de operación. Está conformado por un motor, un reductor de velocidad y un circuito de control por retroalimentación [27].

Los servomotores usados en el Robonova I, son servos digitales HSR-8498HB ,como en

## CAPÍTULO 3. HARDWARE Y SOFTWARE

la figura 3.1, de la empresa Hi-TEC creados especialmente para este robot. Los servos digitales incluyen un controlador digital de alta frecuencia que permite una precisión y resolución mas alta, además de tener un mejor par estático haciendo que los servos digitales sean mas pequeños y con un mayor par que un servo analógico.



Figura 3.1: Servomotor HSR-8498HB

Otra diferencia entre un servo digital a uno analógico es que los servos digitales son programables, es decir, se puede especificar la dirección de rotación, la velocidad y otras características importantes. El Robonova I tiene en total 18 servos digitales HSR-8498HB con las siguientes características [28]

- Características electromecánicas

## CAPÍTULO 3. HARDWARE Y SOFTWARE

<b>Voltaje</b>	6.0 a 7.4 V
<b>Rango de temperatura</b>	-20°C a 60°C
<b><i>Stall Torque</i></b>	7.4 Kg cm a 9.0 Kg cm
<b><i>Stall Current</i></b>	1200 mA a 1480 mA
<b><i> Holding torque</i></b>	6.7 Kg cm a 8.1 Kg cm
<b>Rango angular</b>	190°
<b>Velocidad máxima [29]</b>	364.5 °/ segundo

Tabla 3.1: Características electromecánicas del servo HSR-8498HB

- Características del Motor DC

<b>Corriente usada</b>	DC
<b>Cubierta</b>	Redonda
<b>Campo magnético</b>	Imán permanente de neodyum
<b>Rotor</b>	3 polos
<b>Resistencia de armadura</b>	4.4 Ω
<b>Escobillas</b>	Metal brush

Tabla 3.2: Características del Motor DC en el servo HSR-8498HB

- Características del tren de engranes

<b>Número de engranes</b>	8
<b>Número de ejes</b>	3 (2 intermedios y 1 de salida)
<b>Relación de engranaje</b>	345:1
<b>Material</b>	Resina de Karbonite

Tabla 3.3: Características del tren de engranes en el servo HSR-8498HB

- Características del procesador Atmel ATmega8

2.7 V a 5.5 V
8K memoria flash reprogramable
1024 bytes RAM
512 bytes EEPROM
UART
10 bit ADC
4 MHz clock

Tabla 3.4: Características del tren de engranes en el servo HSR-8498HB

### 3.1.1.1. Interfaz de control HMI

El control es diferente a versiones previas de servos digitales de Hi-TEC, el control para los servos del Robonova I es bidireccional, es decir, puede transmitir y recibir información por el mismo canal gracias a un colector abierto controlado por un transistor NPN.

El control bidireccional también nos permite conectar varios servos en un mismo bus, usando un *pull-up* externo. Como se observa en la Figura 3.2, la interfaz de control se conecta a los puertos del procesador D0 y D1 los cuales corresponden a las terminales *Tx* y *Rx* del *UART (Universal Asynchronous Receiver Transmitter)*, el cual maneja el protocolo de comunicación serial entre los dispositivos y el procesador [30]. También se ve en la figura 3.2, el puerto B0/ICP se encuentra también conectado a la terminal *Rx* para que se pueda medir el ancho de pulso (PWM). La modulación por ancho de pulso, conocida como PWM, es una técnica en la que se modifica el ciclo de trabajo de una señal periódica. Usado generalmente para controlar la cantidad de energía que se envía a un dispositivo electrónico. En un servomotor, se utiliza el PWM para modificar su posición de acuerdo al ancho del pulso enviado cada cierto periodo de tiempo. El servo HSR-8489HB tiene un multi-protocolo de control, es decir, se tiene la habilidad de trabajar en 3 formas diferentes.

- Pulso estándar compatible con PWM: el servo responde a pulsos en el rango de 550 a 2450 microsegundos ( $-95^\circ$  a  $95^\circ$ ). El centro del servo se localiza a un pulso de 1500 microsegundos ( $0^\circ$ ), y se puede obtener la ecuación 3.1 que relaciona el

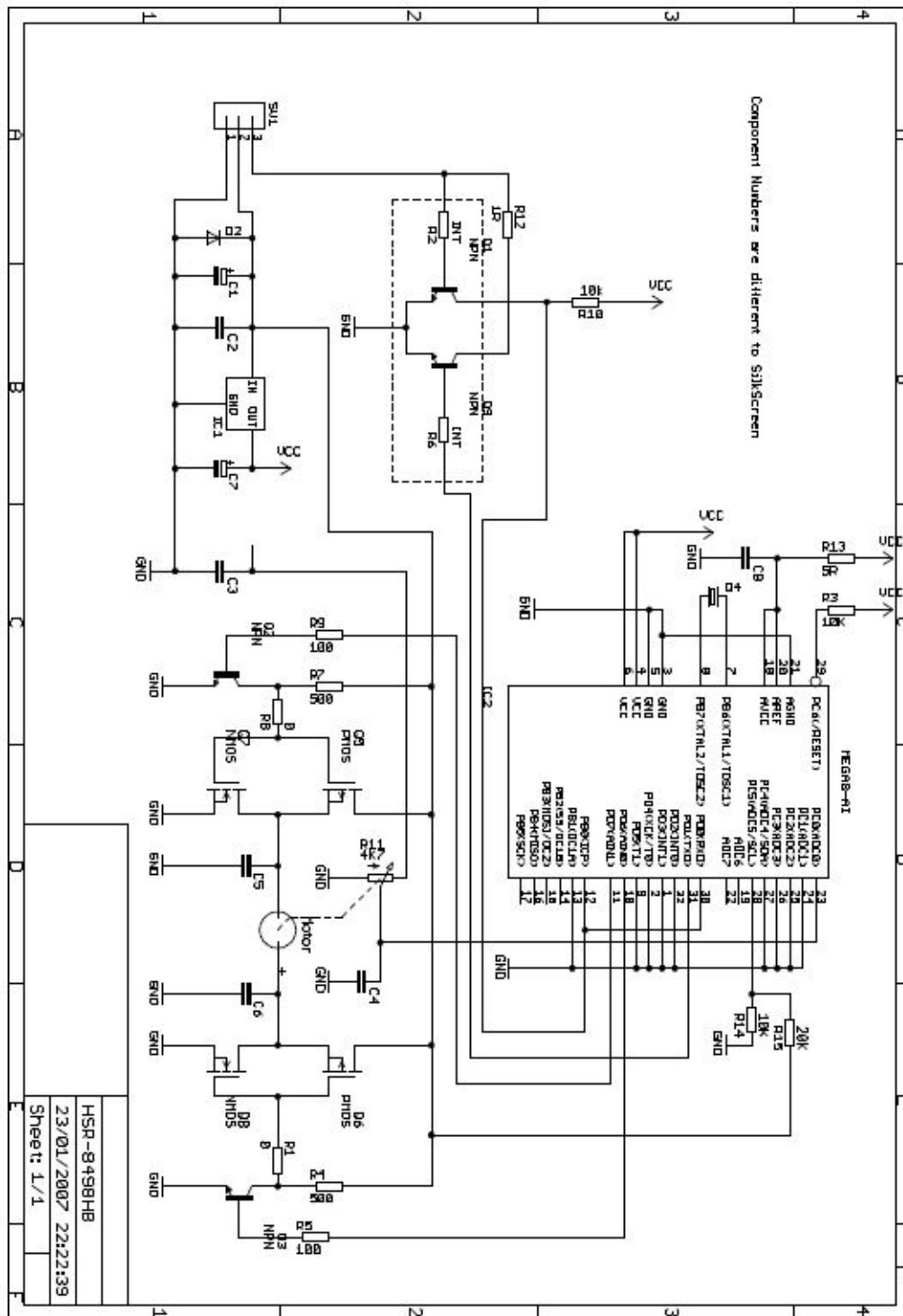


Figura 3.2: Esquemático procesador ATmega en servo HSR-8498HB



## CAPÍTULO 3. HARDWARE Y SOFTWARE

pulso con el ángulo del servo.

$$Angulo(grados) = \frac{Pulso(microsegundos) - 1500}{10} \quad (3.1)$$

- Pulso Extendido, donde se agrega una retroalimentación de la posición y cambio de parámetros. En el modo de pulso extendido, el servo también responde a pulsos de 50 microsegundos a 200 microsegundos. El pulso a 50 microsegundos sirve para la retroalimentación de la posición deshabilitando al servo. Los siguientes pulsos sirven para el cambio de parámetros como la ganancia-P (100 microsegundos), ganancia-D (150 microsegundos) y la banda muerta (200 microsegundos).
- Interfaz serial bi-direccional a 19200 bits por segundo, 8 bits por caracter, 2 bits de stop y sin paridad. Cabe recordar que la interfaz serial es con colector abierto usando un *pull-up* externo.

### 3.1.1.2. Interfaz Serial

Para usar la interfaz serial del servomotor [31], como se dijo anteriormente, la velocidad seria de 19200 bits por segundo, enviando 8 bits por caracter, sin paridad y teniendo 2 bits como stop. Lo anterior se puede ver reflejado en la siguiente figura

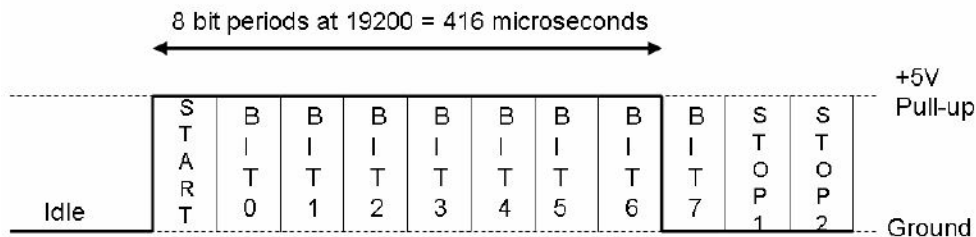


Figura 3.3: Forma del comando serial

## CAPÍTULO 3. HARDWARE Y SOFTWARE

Como se observa en la figura 3.3, cada comando tiene 7 bytes y se usan los últimos 2 bytes como respuesta del servo. Cada byte se explica a continuación

- Byte 1: *Header*
- Byte 2: Comando
- Byte 3: Parámetro 1
- Byte 4: Parámetro 2
- Byte 5: *Checksum*
- Byte 6: Respuesta 1
- Byte 7: Respuesta 2

El Byte 1, es el byte inicial y siempre tendrá el mismo valor en hexadecimal,  $0x80$ . Para el byte 6 y 7 se necesita enviar  $0x00$  para recibir respuesta del servo. Para el byte 5, se utiliza la ecuación 3.2

$$Cs = 256 - (0x80 + Comando + Parametro1 + Parametro2) \bmod 256 \quad (3.2)$$

Donde  $Cs$  es *checksum*. A continuación se muestra en la tabla 3.6 los comandos y parámetros que pueden ser usados en los servomotores a través de la interfaz serial, en la tabla 3.5 se observan los comandos para accesar y escribir tanto en la memoria RAM como en el EEPROM, para conocer las direcciones de la EEPROM y de la memoria RAM se encuentra en la referencia [32]

### CAPÍTULO 3. HARDWARE Y SOFTWARE

<b>Comando</b>	<b>P1</b>	<b>P2</b>	<b>Return 1</b>	<b>Return 2</b>	<b>Descripción</b>
0xE1	addr	0x00	datos	0x03	Leer EEPROM
0xE2	addr	datos	0x03	0x03	Escribir en el EEPROM
0xE3	addr	0x00	datos	0x03	Leer Memoria
0xE4	addr	datos	0x03	0x03	Escribir en la memoria

Tabla 3.5: Comandos seriales para EEPROM y Memoria RAM

<b>Comando</b>	<b>P1</b>	<b>P2</b>	<b>Return 1</b>	<b>Return 2</b>	<b>Descripción</b>
0...0x7F (id)	pos high	pos low	0x00	0x00	Envió de posición al servo deseado
0xE5	0x00	0x00	pos high	pos low	Leer posición de los servos en el bus
0xE6	pos high	pos low	0x00	0x00	Enviar posición a todos los servos en el bus
0xE7	0x00	0x00	versión	id	Leer id y versión de un servo.
0xE8	0x00	0x00	Ancho de pulso	Voltaje	Leer el ancho de pulso y voltaje de los servos en el bus
0xE9	0...0x7F (id)	velocidad	pos high	pos low	Fijar velocidad a un servo específico y leer su posición
0xEA	0x00	1/2/3	0x03	0x06	Seleccionar el parámetro a controlar.
0xEB	0x00	1/0	0x03	0x06	Fijar inicio / final para los servos en el bus
0xEF	0x00	0x00	0x03	0x06	Desactivar los servomotores

Tabla 3.6: Comandos seriales

## CAPÍTULO 3. HARDWARE Y SOFTWARE

Para conocer el parámetro de *pos high* y *pos low*, se transforma el ángulo deseado a pulsos, estos pulsos a un número hexadecimal y se divide este número hexadecimal en mas significativo es decir *pos high* y menos significativo es decir *pos low*.

### 3.1.1.3. Tarjeta HMI

La comunicación del HMI esta basada en una comunicación serial bidireccional en un solo canal. Para lograr esta comunicación [33], se tiene que desarrollar una interfaz lógica. Esta interfaz lógica se realiza en la tarjeta HMI, figura 3.4, con los transistores MOSFET permitiendo que los servos se conecten en un solo bus para enviar y recibir instrucciones.

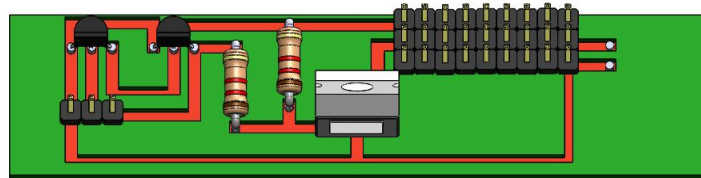


Figura 3.4: Tarjeta HMI

Cuando ninguno de los servos esta enviando información, el bus se corta usando una las resistencias, además solo un servo puede enviar datos al mismo tiempo. En la tabla 3.7 , se muestran los componentes de la tarjeta HMI

2 Resistencias de 1 kOhm
Regulador de Voltaje LM7805
2 MOSFET 2N7000
Header 3 Pines

Tabla 3.7: Componentes de la tarjeta HMI

### 3.1.2. Tarjeta de Control

La tarjeta de control original para el Robonova I es conocida como MR C3024 con un procesador Atmel ATMEGA128 8bit RISC, se observa en la Tabla 3.8 las ventajas de cambiar la tarjeta original a la Raspberry Pi 2 [34]

	Raspberry Pi 2	Atmel ATMEGA128
Memoria	1 GB RAM	64KB flash type
Velocidad	900MHz	400kHz
SO	Basado en Linux	ninguno

Tabla 3.8: Tabla Comparativa entre Raspberry Pi y MR C3024

Una de las ventajas, no mostrada en la tabla 3.8, tiene que ver con la disponibilidad de la tarjeta de control, mientras que la Raspberry Pi 2 esta disponible y es actualizable, la tarjeta MR C3024 esta descontinuada y no es posible encontrarla de forma comercial fácilmente.

El Sistema operativo original que tiene la Raspberry Pi es Raspbian. Para la instalación de sistemas operativos la Raspberry Pi usa un programa conocido como NOOBS. *New Out of Box Software* (NOOBS) [35] es un programa que selecciona e instala diferentes sistemas operativos en la tarjeta de forma fácil y rápida desde una memoria SD. Se puede instalar en la tarjeta desde un sistema operativo basado en Linux hasta Windows 10.

Las tarjetas (Raspberry Pi y HMI) se guardan en un estuche colocado en la parte posterior del torso del Robonova, figura 3.5

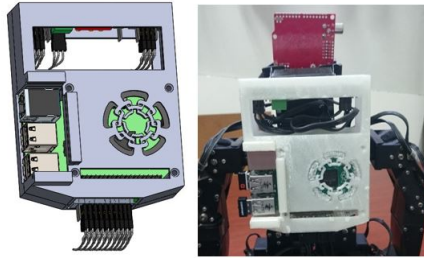


Figura 3.5: Vista Frontal del Estuche y Raspberry PI

## 3.2. Software

Se conoce como software al "...conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación..." [36].

### 3.2.1. Simulación en SimMechanics

Una simulación es una técnica usada para conducir experimentos sobre un modelo de un sistema real para conducir experimentos y conocer como podría reaccionar el sistema real [37]. Para el robot humanoide Robonova-I se usa un modelo creado en un software CAD en tercera dimensión, ver figura 3.6.

El software CAD [38] (diseño asistido por computadora) permite modelar piezas y ensamblajes para después extraer planos técnicos necesarios para producción. Las piezas se construyen virtualmente tomando en cuenta el material, dimensiones y pesos de cada pieza. Para el robot humanoide también se toma en cuenta el movimiento de la pieza, el software CAD permite crear ensamblajes, una pieza de varias piezas, con el

## CAPÍTULO 3. HARDWARE Y SOFTWARE

mismo movimiento que tendría la articulación en el robot físico.

Una vez que se tiene el modelo completo del robot humanoide Robonova-I en el software CAD, es posible gracias al software calcular el centro de masa del robot humanoide y los momentos de inercia necesarios para el calculo de modelos dinámicos mas precisos.

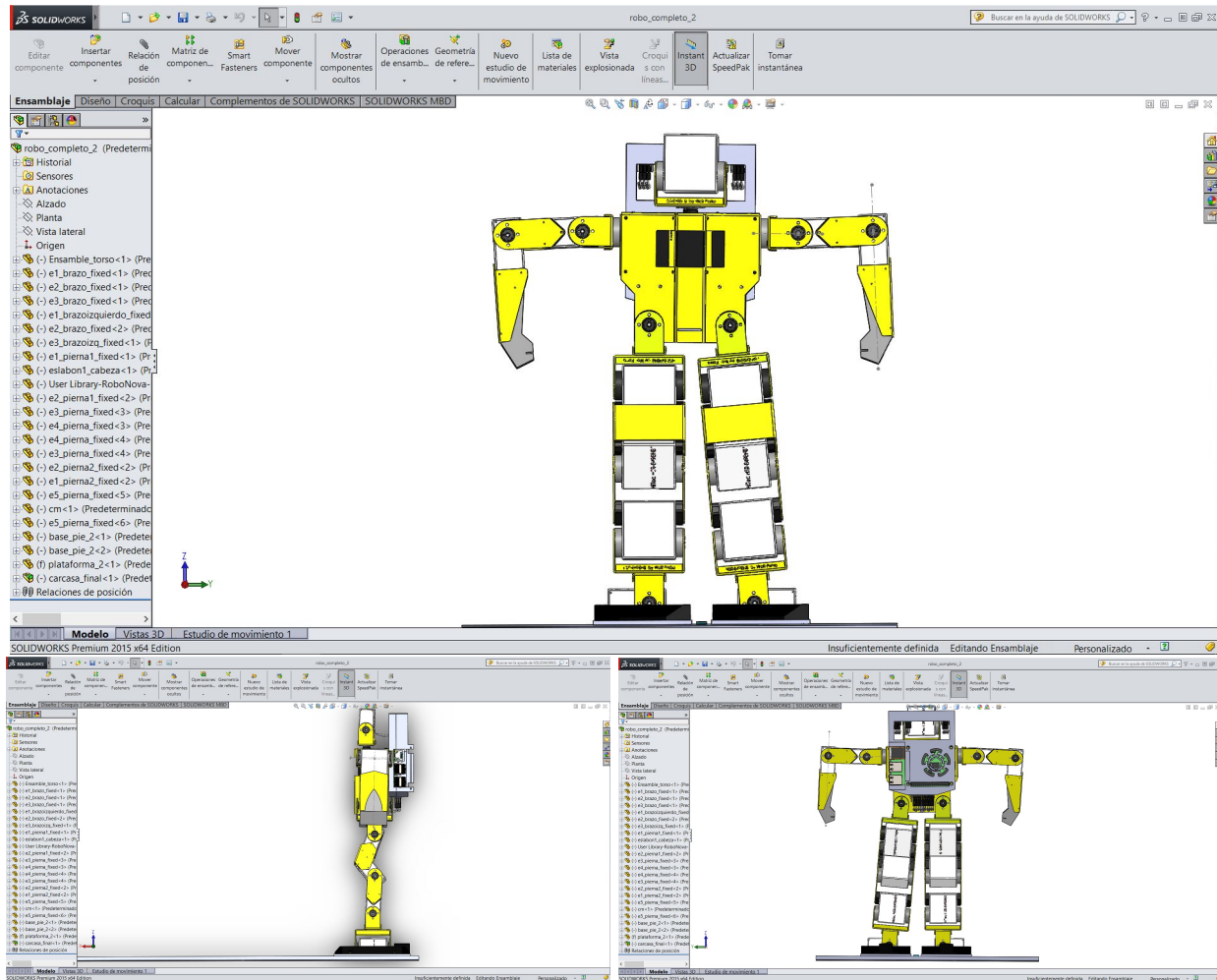


Figura 3.6: Modelo Robonova-I en software CAD

A pesar de las grandes ventajas que tiene el software CAD, el simulador incluido es poco flexible para pruebas avanzadas . Por lo anterior se necesita migrar a otro software que nos permita simular un mejor ambiente para el robot humanoide sin perder las

## CAPÍTULO 3. HARDWARE Y SOFTWARE

características de cada pieza proporcionadas por el software CAD. El simulador que se usa para el Robonova-I, ver figura 3.7, es SimMechanics™ [39], el cual proporciona un entorno de simulación para sistemas mecánicos 3D usando bloques como sensores, articulaciones y diferentes elementos.

SimMechanics™ puede importar diseños CAD, generar automáticamente una animación 3D que permita visualizar la dinámica del sistema y ayudar a desarrollar sistemas de control en coordinación con los programas de MATLAB® y Simulink®.

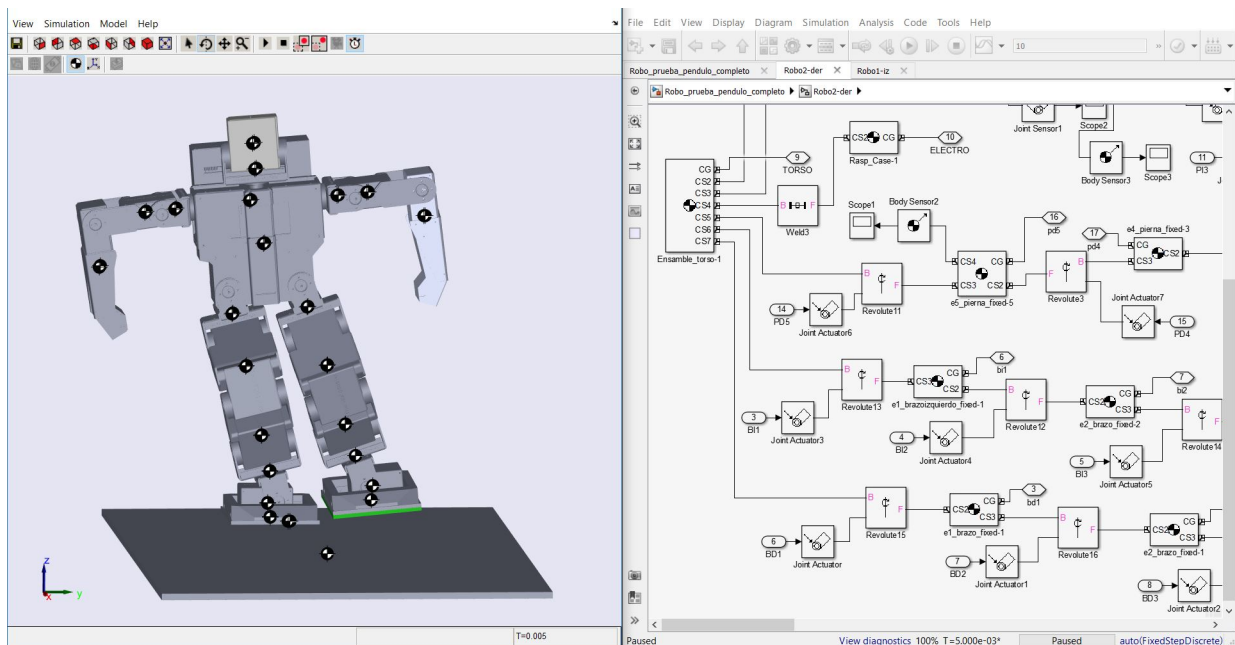


Figura 3.7: Simulación del Robonova-I en SimMechanics™

### 3.2.2. Software Raspberry Pi

El sistema operativo que se utilizó en la Raspberry Pi es el conocido como Raspbian, basado en el S.O de Linux, Debian. El control del Robonova-I usando la tarjeta Raspberry Pi, se puede realizar de diferentes formas



## CAPÍTULO 3. HARDWARE Y SOFTWARE

- Procesando los controles en la computadora y usando la Raspberry Pi solo para comunicar los protocolos seriales
- Procesando los controles directamente en la Raspberry Pi

Para el primer punto, se utiliza una conexión entre el Matlab<sup>TM</sup> de una computadora y la tarjeta Raspberry Pi. Para la conexión con Matlab<sup>TM</sup>, se necesita que se cambie el sistema operativo por uno proporcionado por Matlab<sup>TM</sup>, el procedimiento se puede encontrar en la webpage de ayuda de Matlab<sup>TM</sup>. Una vez instalado el nuevo sistema operativo, una versión de Debian 7.8 (wheezy), se descarga el firmware para habilitar una conexión a través de WIFI, y con esta configuración se puede establecer una conexión automática entre Matlab<sup>TM</sup> y Raspberry Pi. Se usa un programa de Matlab que envía el código hexadecimal a los servos con el ángulo deseado usando el protocolo de comunicación serial de la tarjeta Raspberry Pi.

Para el segundo punto, se programa el control del Robonova-I dentro de la tarjeta Raspberry Pi. El programa ha utilizar se conoce como Python, Python [40] es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

El programa utilizado se encuentra en el apéndice B, además se muestra en el capítulo de resultados el caminado parametrizado realizado por el Robonova-I usando el programa de Python. Este programa se puede iniciar usando la terminal de una computadora o entrando directamente al sistema operativo de la tarjeta. Se usa la tecnología WI-FI para iniciar e indicar al programa la cantidad de pasos que se desea que

## CAPÍTULO 3. HARDWARE Y SOFTWARE

el Robonova-I realice, con esta información el programa procesa, dentro de la misma tarjeta, las trayectorias y variables articulares que se necesitan para el caminado parametrizado.

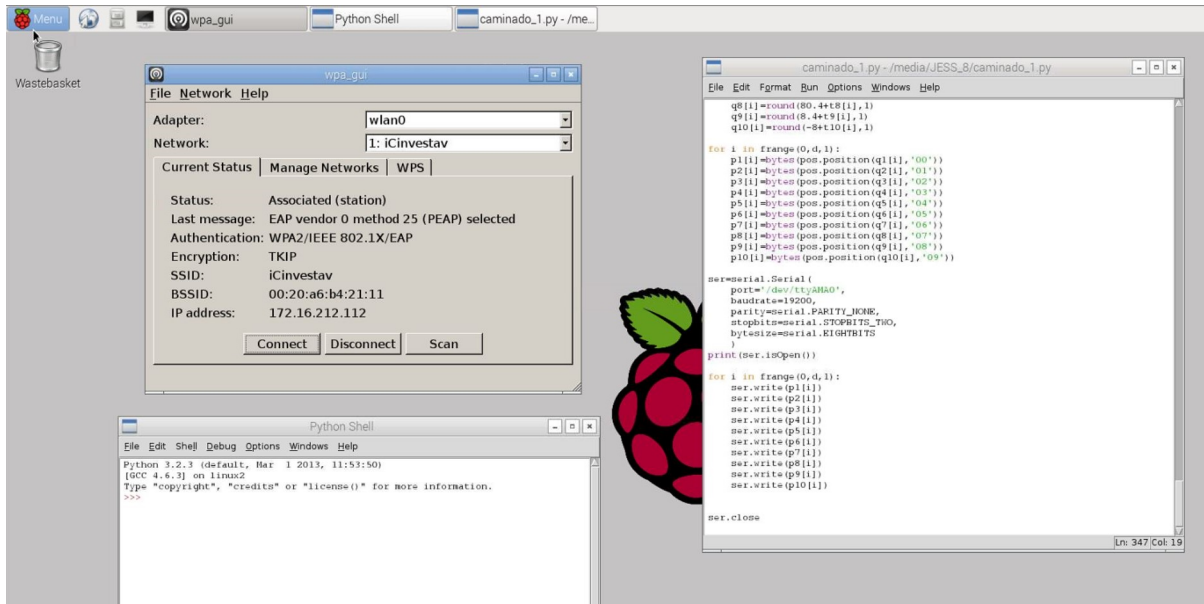


Figura 3.8: S.O. Raspberry Pi

# Capítulo 4

## Caminado parametrizado

El acto de caminar en un robot humanoide consiste en generar trayectorias articulares que definan una secuencia de posiciones para cada articulación de modo que los movimientos que se produzcan en las piernas del robot generen un desplazamiento de todo el cuerpo a lo largo de una trayectoria previamente definida.

Usaremos la definición de robot bípedo para el robot humanoide Robonova I. Un robot bípedo es aquel robot caminante cuya locomoción se efectúa únicamente con las piernas. El modelo de locomoción bípeda más conocido es el del ser humano.

### 4.1. Caminado humano

Como se explico en el estado del arte de esta tesis, el estudio del caminado humano es campo de la biomecánica. El caminado que realiza el ser humano se define como la forma en que se lleva a cabo el desplazamiento utilizando los miembros del cuerpo humano, lo anterior implica establecer un referencial con respecto a los planos

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

anatómicos del cuerpo humano.

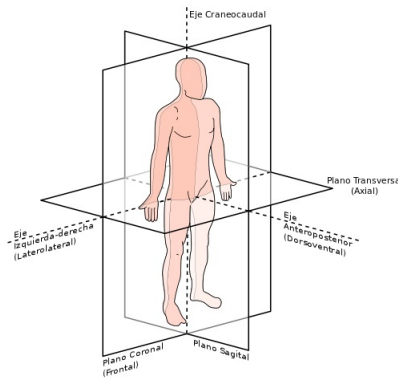


Figura 4.1: Planos anatómicos en un ser humano [5]

Al observar como se realiza el caminado se reconocen 2 unidades funcionales [41]

- Unidad Pasajera: comprende la cabeza, cuello, brazos, tronco y pelvis. La unidad pasajera es transportada por la unidad locomotora.
- Unidad Locomotora: comprende la pelvis y las extremidades inferiores. La unidad locomotora es responsable de portar el peso y de llevar a cabo el movimiento de caminado

Como se menciona anteriormente, la pelvis se usa tanto para la unidad pasajera como la locomotora, siendo la conexión entre ambas unidades. La unidad locomotora se encarga de proporcionar estabilidad estructural al cuerpo y brindar movilidad transfiriendo el peso del cuerpo de una pierna a otra, además esta unidad se encarga de impulsar la pierna liberada hacia adelante para hacerse cargo del peso del cuerpo cuando éste caiga en sentido del movimiento y volver a iniciar todo de nuevo. Este ciclo de caminado.<sup>es</sup> repetido por ambas piernas a velocidades variables y en diferentes terrenos.

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

Durante el ciclo de caminado se reconocen 2 fases representadas por el pie. Si el pie esta apoyado en el suelo se le considera fase de soporte, si el pie esta levantado en el aire se le considera fase de transferencia.

A la pierna que se encuentra en la fase de soporte se le denomina *pierna de apoyo*, mientras que la pierna que esta en la fase de transferencia o fase de balanceo se le denomina *pierna flotante*.

Un ciclo de caminado se divide en 4 fases. [42]

- Doble soporte inicial: Esta fase es el inicio del ciclo de caminado, su objetivo es absorber el impacto inicial del pie y transferir el peso del robot a la pierna de apoyo. Figura 4.2 a
- Soporte simple: En esta fase, el centro de masa del cuerpo esta soportado en una sola pierna, la pierna de apoyo, de tal forma que la cadena cinemática va desde la pierna de apoyo a la pierna flotante. Esta fase mantiene el movimiento del caminado sobre la pierna de apoyo y ayuda a mantener la estabilidad del tronco. Figura 4.2 b
- Doble soporte terminal: En esta fase se traspassa el peso de la pierna de apoyo a la pierna flotante para pasar a la fase de transferencia sin perder estabilidad en la postura. Figura 4.2 c
- Transferencia: En esta fase, se hace el intercambio entre la pierna de apoyo y la pierna flotante. En esta fase, la pierna que estaba de apoyo durante el soporte simple se vuelve la pierna flotante y viceversa. Figura 4.2 d

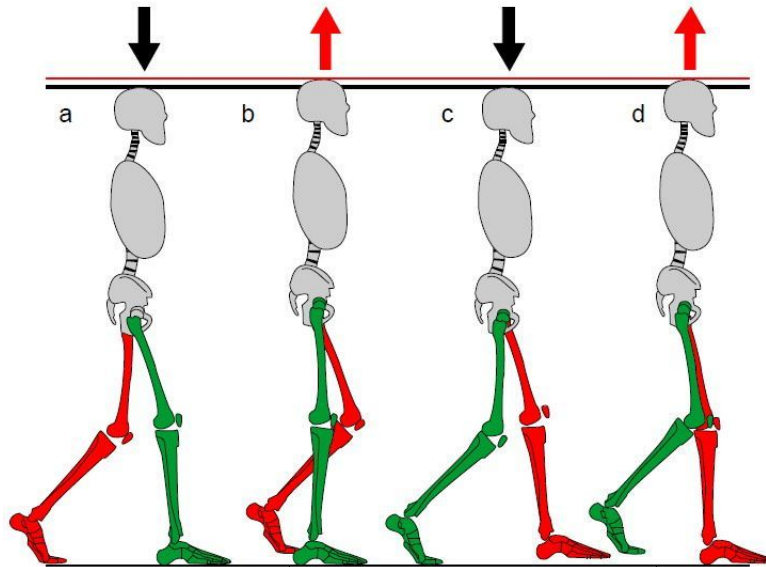


Figura 4.2: Fase del caminado humano [6]

## 4.2. Generación de Trayectoria en espacio de trabajo

El caminado parametrizado [43] permite generar distintas trayectorias de caminado basadas en parámetros deseados en una caminata. Los patrones son descritos a nivel del espacio de trabajo y usando la cinemática inversa se transforman al espacio articular. La ventaja principal de esta forma de generación de trayectoria es lo simple y fácil de implementar en los robots humanoides y como ventaja añadida de este caminado es el control de velocidades y aceleraciones.

Para desarrollar el caminado deseado se necesita especificar la trayectoria para la pierna flotante y la pierna de apoyo, esto se logra especificando los parámetros que definen características específicas de la caminata deseada. Estos parámetros definen

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

altura del pie flotante con respecto al suelo, rizo de la cadera, tamaño del paso, entre otros.

Los parámetros son puntos críticos en la trayectoria de cada pierna, de tal manera que a partir de estos puntos se puede encontrar una ecuación general a pedazos que interpole esos valores y genere una trayectoria suave. Esta ecuación debe ser continua y cuya derivada exista, y debe ser suave para no tener cambios bruscos que afecten a los motores.

Considérese la siguiente ecuación

$$x(t) = a + bt + ct^2 + dt^3 \quad (4.1)$$

Donde la ecuación 4.1 es una ecuación polinomial de tercer orden. Los coeficientes son  $a, b, c$  y  $d$  y  $t \in [0, T]$ . Primero evaluamos en el intervalo de tiempo

$$x(0) = x_0 = a \quad (4.2)$$

$$x(T) = x_f = a + bT + cT^2 + dT^3 \quad (4.3)$$

Ahora evaluamos las derivadas en el mismo intervalo de tiempo

$$\dot{x}(0) = \dot{x}_0 = v_0 = b \quad (4.4)$$

$$\dot{x}(T) = \dot{x}_f = v_f = b + 2cT + 3dT^2 \quad (4.5)$$

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

El sistema de ecuaciones quedaría

$$x_0 = a \quad (4.6)$$

$$x_f = a + bT + cT^2 + dT^3 \quad (4.7)$$

$$v_0 = b \quad (4.8)$$

$$v_f = b + 2cT + 3dT^2 \quad (4.9)$$

Teniendo los valores de los coeficientes  $a$  y  $b$ , nos disponemos a encontrar  $d$ . Despejando  $c$  tenemos la ecuación 4.10

$$c = \frac{v_f - v_0 - 3dT^2}{2T} \quad (4.10)$$

Sustituyendo la ecuación 4.10 en 4.7 tenemos que

$$x_f = x_0 + v_0T + T^2 \left( \frac{v_f - v_0 - 3dT^2}{2T} \right) + dT^3 \quad (4.11)$$

$$x_f = x_0 + v_0T + T \left( \frac{v_f - v_0 - 3dT^2}{2} \right) + dT^3 \quad (4.12)$$

$$2x_f = 2x_0 + 2v_0T + v_fT - v_0T - 3dT^3 + 2dT^3 \quad (4.13)$$

$$-d = \frac{2(x_f - x_0) - T(v_0 + v_f)}{T^3} \quad (4.14)$$

$$d = \frac{2(x_0 - x_f) + T(v_0 + v_f)}{T^3} \quad (4.15)$$

Sustituyendo  $d$  en la ecuación 4.10 y simplificando tenemos que

$$c = \frac{3(x_f - x_0) - T(2v_0 + v_f)}{T^2} \quad (4.16)$$



## CAPÍTULO 4. CAMINADO PARAMETRIZADO

Sustituyendo los coeficientes en la ecuación original

$$x(t) = x_0 + v_0 t + \left( \frac{3(x_f - x_0) - T(2v_0 + v_f)}{T^2} \right) t^2 + \left( \frac{2(x_0 - x_f) + T(v_0 + v_f)}{T^3} \right) t^3 \quad (4.17)$$

La ecuación 4.17 es una interpolación segmentada cúbica también conocida como *cubic spline interpolation* o simplemente *spline*, la cual genera una trayectoria del punto  $x_0$  a  $x_f$  suave dentro del intervalo  $t \in [0, T]$ . El término *spline* define funciones que son utilizadas en interpolación de datos o suavizado de curvas.

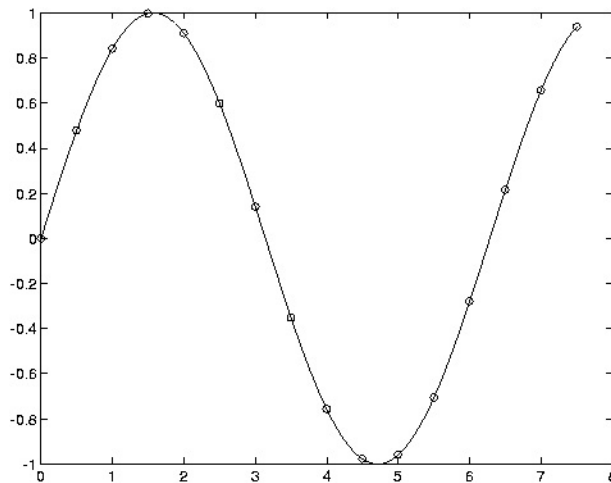


Figura 4.3: Función  $y = \sin(x)$  usando la interpolación segmentada cúbica

### 4.2.1. Plano sagital

Haciendo un breve repaso, recordemos que el ciclo de caminado se divide en 4 fases, de las cuales el doble soporte inicial, soporte simple y doble soporte terminal son usando la misma pierna de apoyo para luego durante la fase de transferencia cambiar la pierna de apoyo y volver a iniciar el ciclo de caminado.

Para explicar los parámetros necesarios y como se planea la trayectoria en cada punto

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

del caminado dividiremos la caminata en planos. En el plano sagital (ver figura 4.1), el robot se mueve en el eje  $X$  y en el eje  $Z$  (ver figura 2.6).

Analizando el plano sagital, figura 4.4, el primer parámetro sería [8]

$$t_p = \frac{1}{2}T_c \quad (4.18)$$

Siendo  $T_c$  la duración del ciclo de caminado completo, y  $t_p$  el intervalo de tiempo donde ocurre el paso. El siguiente parámetro sería

$$D_p = 2L_p \quad (4.19)$$

Siendo  $D_p$  la distancia del paso, siendo  $L_p$  la longitud en que se ubica el pie flotante por detrás del pie de apoyo. Una vez que se tiene el tiempo del caminado y la longitud de cada paso, los parámetros que siguen es la altura máxima del pie flotante con respecto al suelo  $h_p$  y la altura máxima de la cadera, conocida como rizo de cadera  $h_b$ . Finalmente, el parámetro  $h$  corresponde a la altura mínima del cuerpo durante el caminado.

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

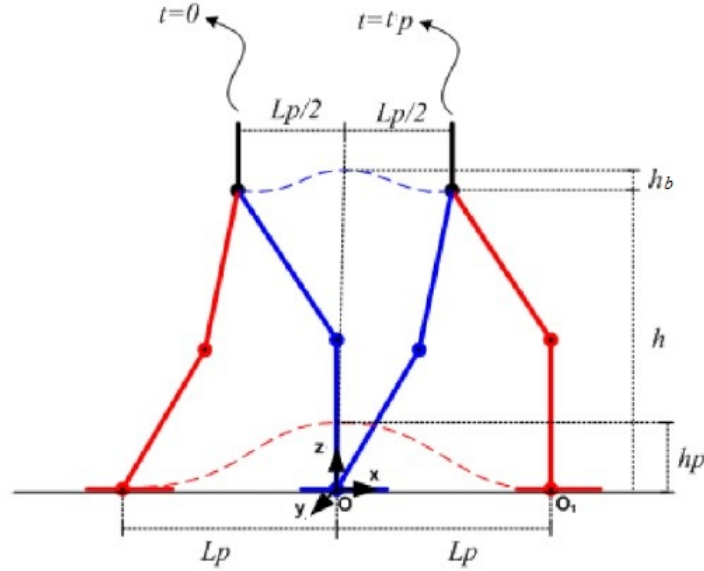


Figura 4.4: Caminado de humanoide en el plano sagital [7]

Ahora solo falta considerar el tiempo para la fase de doble soporte  $t_{ds}$ , para que se reparta de manera equitativa al inicio y al final del paso, se considera la mitad del tiempo de doble soporte. Además, se observa que esta fase afecta mas a la pierna flotante, de tal forma que para la trayectoria de la pierna de apoyo, el  $t_{ds}$  no es relevante. Utilizando los parámetros que se definieron anteriormente y la ecuación 4.17 (ISC) se puede describir la trayectoria tanto de la pierna flotante  $(X_f, Z_f)$  como de la pierna de apoyo  $(X_b, Z_b)$ , Se considerará también la siguiente notación  $ISC(x_0, x_f, t_0, t_f, v_0, v_f)$  para el *spline* siendo  $t_0$  como tiempo inicial,  $t_f$  como tiempo final y considerando la velocidad inicial y final como cero.

- Trayectoria pierna flotante

$$X_f = \begin{cases} -L_p & \text{si } 0 \leq t \leq t_{ds}/2 \\ ISC(t - (t_{ds}/2), t_p - t_{ds}, -L_p, L_p, 0, 0) & \text{si } t_{ds}/2 \leq t \leq t_p - (t_{ds}/2) \\ L_p & \text{si } t_p - (t_{ds}/2) \leq t \leq t_p \end{cases} \quad (4.20)$$

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

$$Z_f = \begin{cases} 0 & \text{si } 0 \leq t \leq t_{ds}/2 \\ ISC(t - (t_{ds}/2), (t_p - t_{ds})/2, 0, h_p, 0, 0) & \text{si } t_{ds}/2 \leq t \leq t_p/2 \\ ISC(t - (t_p/2), (t_p - t_{ds})/2, h_p, 0, 0, 0) & \text{si } t_p/2 \leq t \leq t_p - (t_{ds}/2) \\ 0 & \text{si } t_p - (t_{ds}/2) \leq t \leq t_p \end{cases} \quad (4.21)$$

- Trayectoria pierna de apoyo

$$X_b = \begin{cases} ISC(t, t_p, -L_p/2, L_p/2, 0, 0) & \text{si } 0 \leq t \leq t_p \end{cases} \quad (4.22)$$

$$Z_b = \begin{cases} ISC(t, t_p/2, h, h + h_b, 0, 0) & \text{si } 0 \leq t \leq t_p/2 \\ ISC(t - t_p/2, t_p/2, h + h_b, h, 0, 0) & \text{si } t_p/2 \leq t \leq t_p \end{cases} \quad (4.23)$$

### 4.2.2. Plano Frontal

En el plano frontal, se analiza como traspasar el peso del robot de una pierna a otra. Los parámetros que definen la trayectoria de traspaso son  $d_b$  y  $d_p$  donde

- $d_b$  es el sobrepaso que presenta el cuerpo del robot con respecto a la pierna de apoyo.
- $d_p$  distancia de separación entre los pies
- $l_c$  distancia desde el referencial base al primer eslabón

Los parámetros se pueden observar en la figura 4.5

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

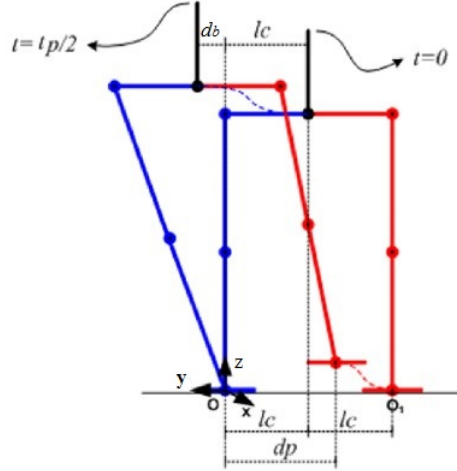


Figura 4.5: Caminado de humanoide en el plano frontal [7]

- Trayectoria pierna de apoyo

$$Y_b = \begin{cases} ISC(t, t_p/2, -l_c sig, d_b sig, 0, 0) & \text{si } 0 \leq t \leq t_p/2 \\ ISC(t - t_p/2, t_p/2, d_b sig, -l_c sig, 0, 0) & \text{si } t_p/2 \leq t \leq t_p \end{cases} \quad (4.24)$$

- Trayectoria pierna flotante

$$Y_f = \begin{cases} -2l_c sig & \text{si } 0 \leq t \leq t_{ds}/2 \\ ISC(t - (t_{ds}/2), (t_p - t_{ds})/2, -2l_c sig, -d_p sig, 0, 0) & \text{si } t_{ds}/2 \leq t \leq t_p/2 \\ ISC(t - (t_p/2), (t_p - t_{ds})/2, -d_p sig, -2l_c sig, 0, 0) & \text{si } t_p/2 \leq t \leq t_p - (t_{ds}/2) \\ -2l_c sig & \text{si } t_p - (t_{ds}/2) \leq t \leq t_p \end{cases} \quad (4.25)$$

### 4.2.3. Inicio y Final del caminado

En las trayectorias que se muestran anteriormente se asume que el robot ya ha dado un medio paso y ha ocurrido un movimiento inicial. Para ello, se necesita que el robot tenga un paso inicial, avanzando de la posición cero al medio paso y que tenga

## CAPÍTULO 4. CAMINADO PARAMETRIZADO

un paso final, avanzando del medio paso a la posición cero. Se define la posición cero como ambos pies a la misma distancia sin separación en el eje  $X$ , como se observa en la figura

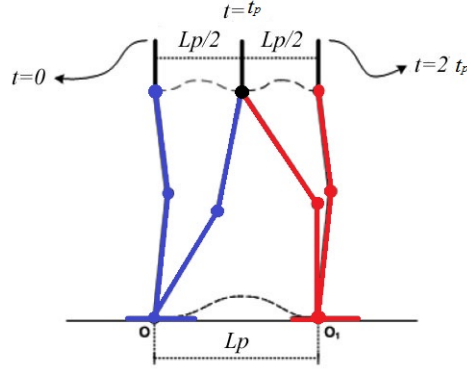


Figura 4.6: Pose inicial y final del robot humanoide

- Trayectoria inicial pierna de apoyo

$$X_{bi} = \begin{cases} ISC(t, t_p, 0, L_p/2, 0, 0) & \text{si } 0 \leq t \leq t_p \end{cases} \quad (4.26)$$

- Trayectoria final pierna de apoyo

$$X_{bf} = \begin{cases} ISC(t, t_p, -L_p/2, 0, 0, 0) & \text{si } 0 \leq t \leq t_p \end{cases} \quad (4.27)$$

- Trayectoria inicial pierna flotante

$$X_{fi} = \begin{cases} 0 & \text{si } 0 \leq t \leq t_{ds}/2 \\ ISC(t - (t_{ds}/2), t_p - t_{ds}, 0, L_p, 0, 0) & \text{si } t_{ds}/2 \leq t \leq t_p - (t_{ds}/2) \\ L_p & \text{si } t_p - (t_{ds}/2) \leq t \leq t_p \end{cases} \quad (4.28)$$

- Trayectoria final pierna flotante

$$X_{ff} = \begin{cases} -L_p & \text{si } 0 \leq t \leq t_{ds}/2 \\ ISC(t - (t_{ds}/2), t_p - t_{ds}, -L_p, 0, 0, 0) & \text{si } t_{ds}/2 \leq t \leq t_p - (t_{ds}/2) \\ 0 & \text{si } t_p - (t_{ds}/2) \leq t \leq t_p \end{cases} \quad (4.29)$$

# Capítulo 5

## Resultados y Conclusiones

En esta sección se presentaran los resultados obtenidos en el simulador del Robonova-I y el robot físico. Se mostrarán gráficas de trayectorias generadas por los programas mostrados en el Apendice B y validación del caminado parametrizado corroborando los modelos presentados al inicio de esta tesis.

### 5.1. Resultados

Con el fin de analizar el caminado parametrizado explicado anteriormente y por consecuencia validar los modelos geométricos mostrados en el capítulo 2. Los parámetros del caminado parametrizado quedan definidos como sigue

- Número de pasos : 2
- Tiempo de cada paso : 1 segundo
- Tiempo de doble soporte : 0.2 segundos es decir 20 % del tiempo que dura el paso

## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

- Incremento del tiempo : 0.025
- Longitud del paso: 3 cm
- Altura del cuerpo : 17 cm (altura mínima del cuerpo)
- Rizo de cadera: 0.5 cm
- Altura máxima del pie flotante con respecto al suelo: 2 cm
- Distancia entre los pies a la mitad del paso: 5 cm
- Sobrepaso que tiene el cuerpo con respecto al pie de apoyo: 0.2 cm

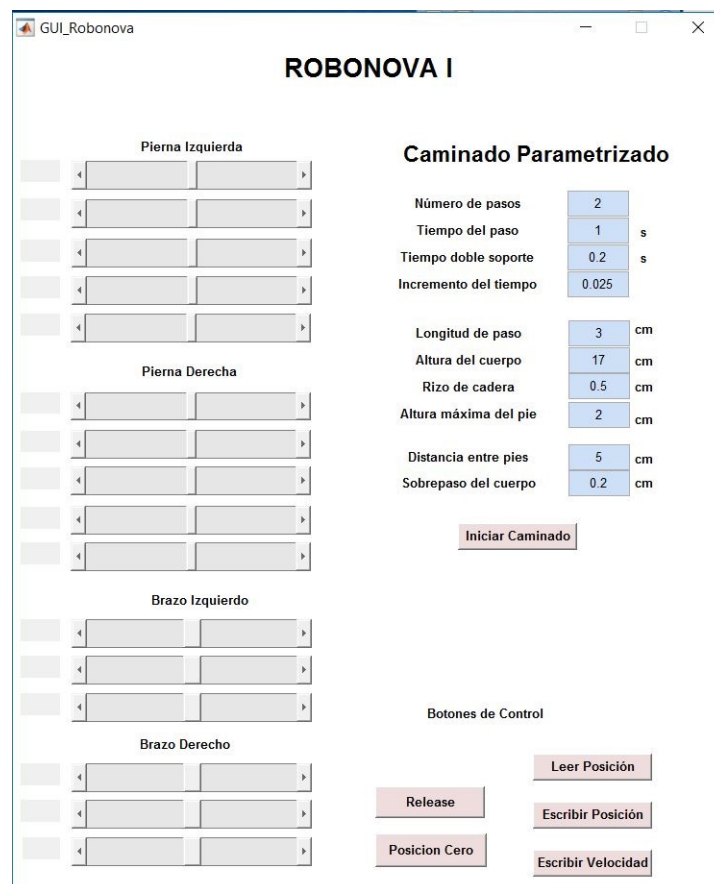


Figura 5.1: GUI para control de caminado parametrizado del robot Robonova-I



## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

Todos los parámetros son introducidos en la GUI diseñada para el control del robot humanoide Robonova-I, como se observa en la figura 5.1 se tienen todos los parámetros antes mencionados, además de controlar cada articulación del robot por separado.

Los parámetros generan trayectorias en los 3 ejes tanto para la pierna de apoyo como para la pierna flotante. Se observa en la figura 5.2, la gráfica de la trayectoria en el eje X, es decir el eje de avance del robot humanoide donde podemos identificar la longitud del paso de 3 cm en el avance de la pierna de apoyo.

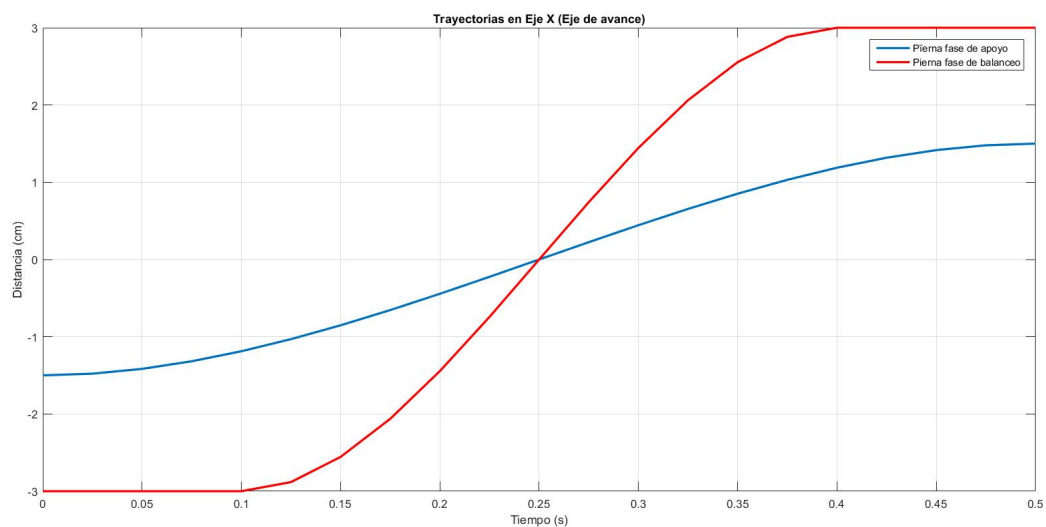


Figura 5.2: Trayectoria en Eje X

En la figura 5.3, se muestra la gráfica de la trayectoria en el eje Y, es decir el eje del movimiento lateral del robot humanoide donde se identifica claramente la longitud entre los pies.

## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

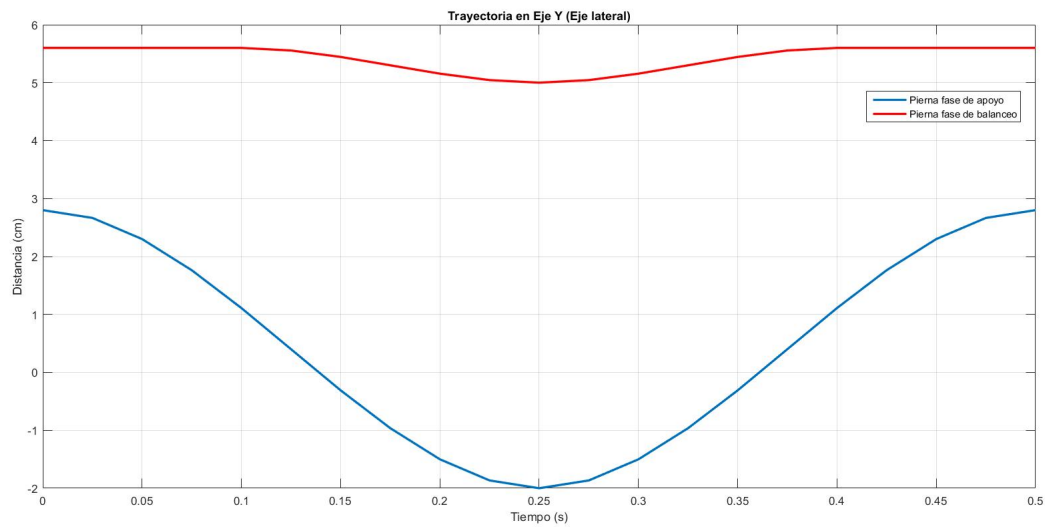


Figura 5.3: Trayectoria en Eje Y

En la figura 5.4, se observa la gráfica de la trayectoria en el eje Z, es decir el eje del movimiento vertical del robot humanoide donde se distingue los parametros de altura máxima del pie flotante y altura mínima del cuerpo incluido el rizo de cadera.

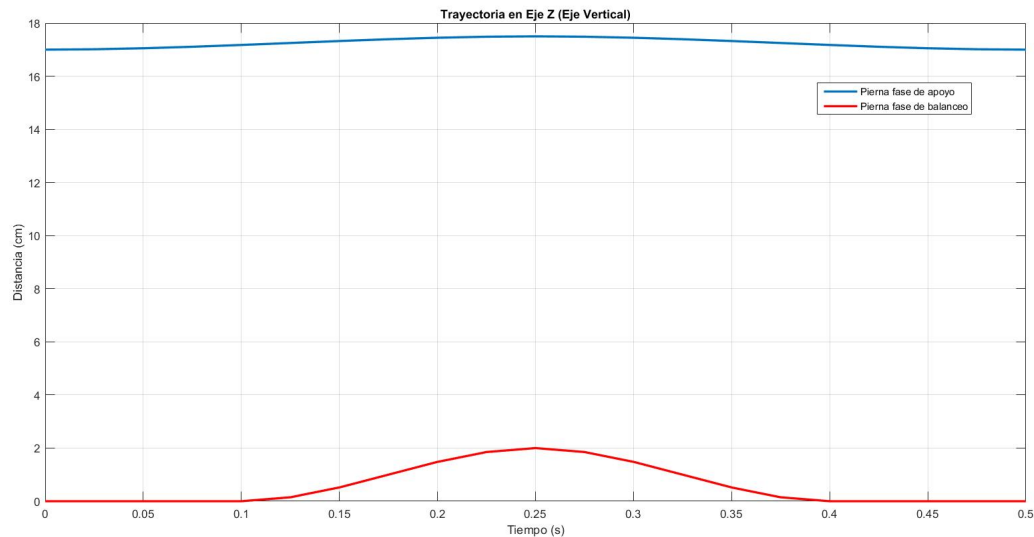


Figura 5.4: Trayectoria en Eje Z

Usando las trayectorias en el espacio de trabajo del robot humanoide, utilizamos el

## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

modelo geométrico inverso para convertir esos caminos a trayectorias articulares y poderlas usar en el movimiento del robot. Estas trayectorias articulares son expresadas en las figuras 5.5 y 5.6 , los recorridos mostradas son suaves gracias a la interpolación segmentada cúbica. Estas trayectorias suaves mejoran el rendimiento del servomotor, logrando una mayor precisión ya que no se tienen cambios abruptos en la trayectoria.

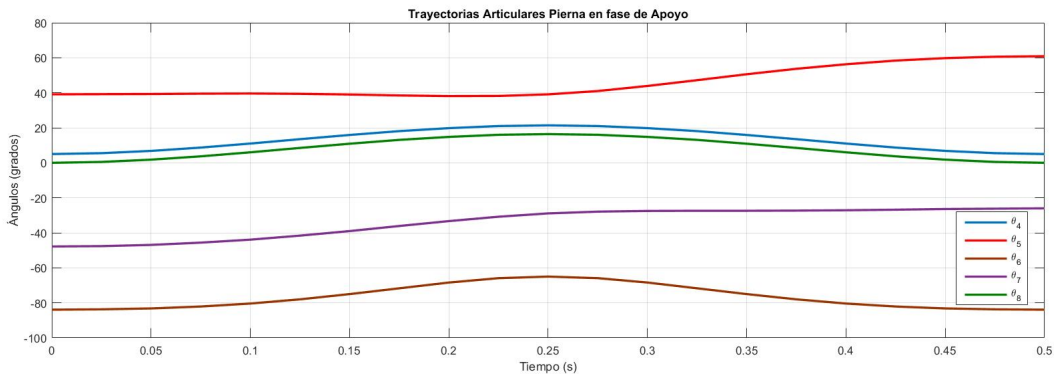


Figura 5.5: Trayectorias de las Articulaciones en la pierna de apoyo

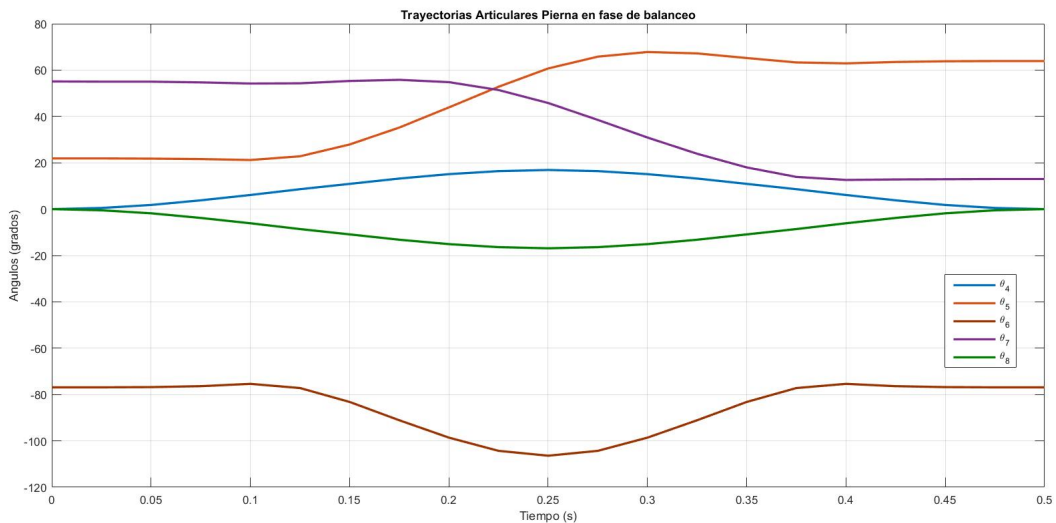


Figura 5.6: Trayectorias de las Articulaciones en la pierna flotante

CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

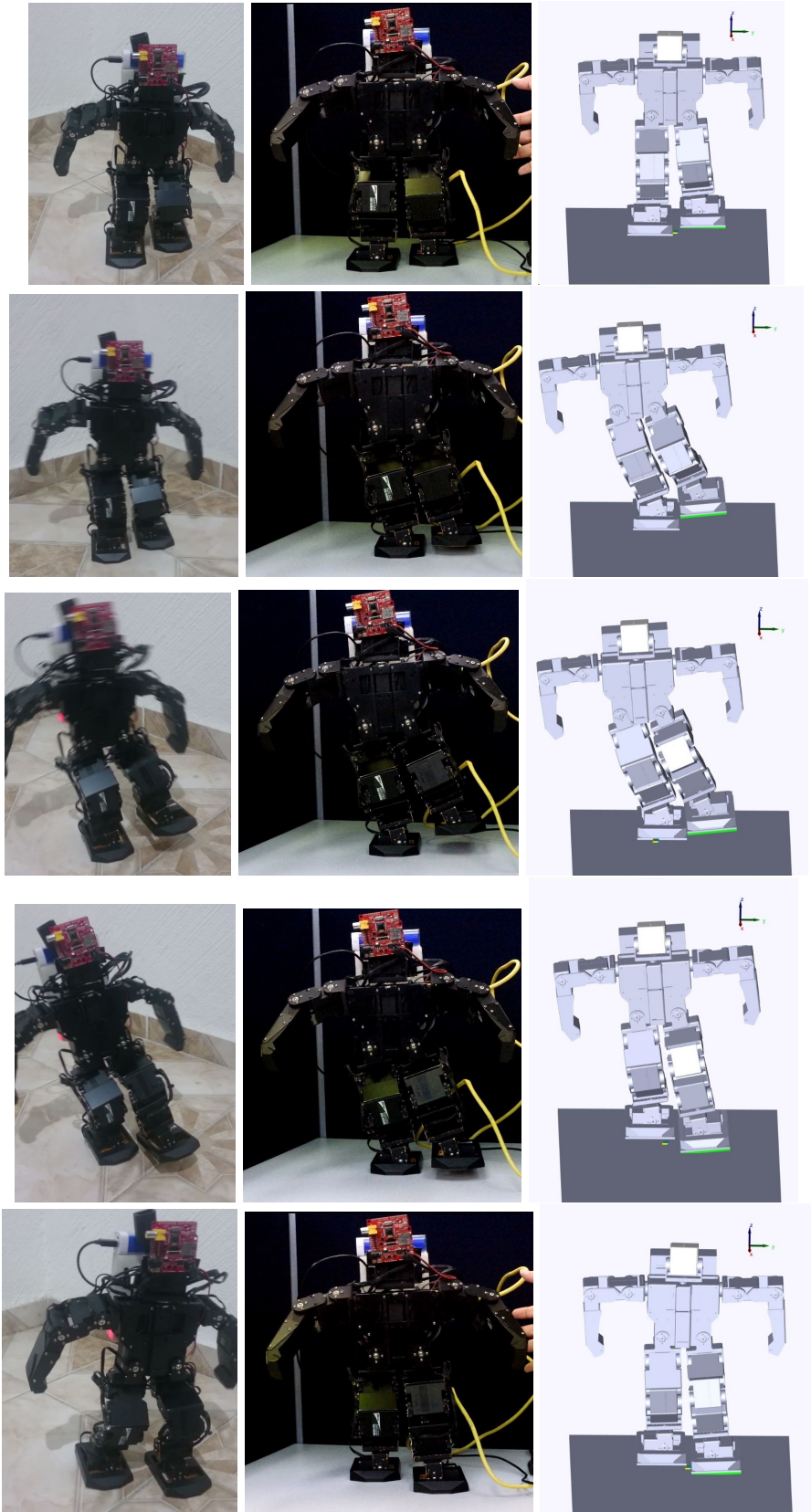


Figura 5.7: Caminado Físico VS Caminado Simulado

## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

En la figura 5.7, se puede analizar el caminado en el simulador al igual que el caminado en el robot físico usando Matlab y usando Python. En los caminados se observa el funcionamiento del caminado parametrizado, el cual es pasar el centro de masa del robot a la pierna en la fase de apoyo mientras que la pierna que queda libre, entra en la fase de balanceo avanzando una cierta distancia para luego entrar a la fase de doble soporte y cambiar la pierna que actuaba como apoyo siendo ahora la pierna que queda libre y la pierna que estaba en fase de balanceo se vuelve la pierna de apoyo, comenzando el ciclo de nuevo. La figura 5.7 nos permite verificar el caminado parametrizado para el robot físico, además de permitirnos verificar los modelos con los que se han trabajado hasta este momento en la tesis

Se verifica durante el caminado que el robot se mantiene estable durante el caminado manteniendo la postura del tronco, moviendo siempre el centro de masa al polígono de soporte que forma el pie que está apoyado en el suelo.

Se comprobó la utilización del simulador de SimMechanics como una forma de conocer el comportamiento del robot humanoide en forma ideal, y de cómo reaccionan las articulaciones y el centro de masa de forma ideal. En el simulador, se puede calcular el centro de masa, ver figura 5.8, en donde se observa el movimiento en los 3 ejes y la relación que tienen con el movimiento del robot humanoide y las trayectorias creadas. Se visualiza como el movimiento en el eje  $Z$ , que sería la altura del Robonova-I se mantiene casi constante con pequeños movimientos que pertenecen al rizo de la cadera, en el eje  $X$  se observa como avanza el robot y finalmente en el eje  $Y$  se puede ver como se mueve el centro de masa a un lado para ubicarse en la pierna que está en fase de apoyo.

## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

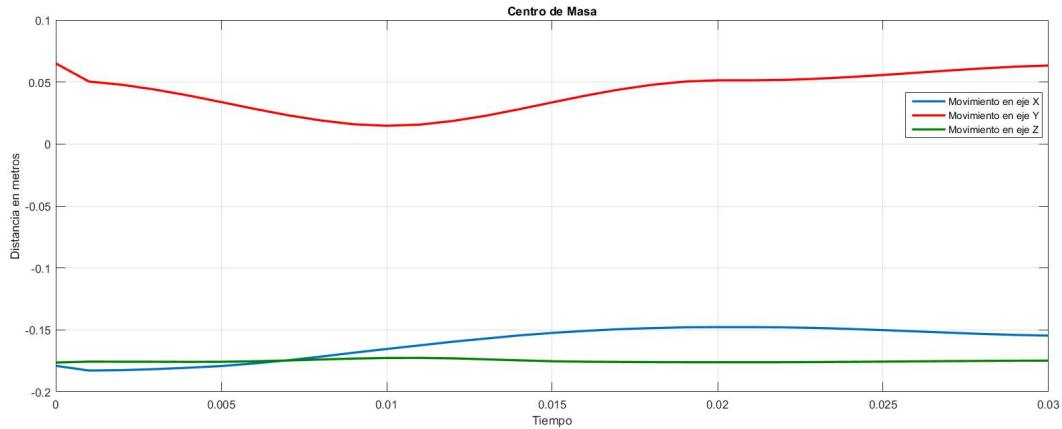


Figura 5.8: Trayectoria del Centro de Masa en el Simulador

También se midió la posición de los motores durante el caminado, y se observa en las siguientes gráficas, figura 5.9, la posición deseada contra la posición medida.

Las diferencias en el seguimiento de trayectoria en los motores, se deben a 2 factores:

- El tiempo de uso que tienen los motores
- Y si ese motor, es el que soporta todo el peso del robot durante la fase de soporte.

Pero en general, los errores son mínimos y para ser un sistema cuyo control es en lazo abierto, estos errores se pueden minimizar usando realimentación sensorial.

## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

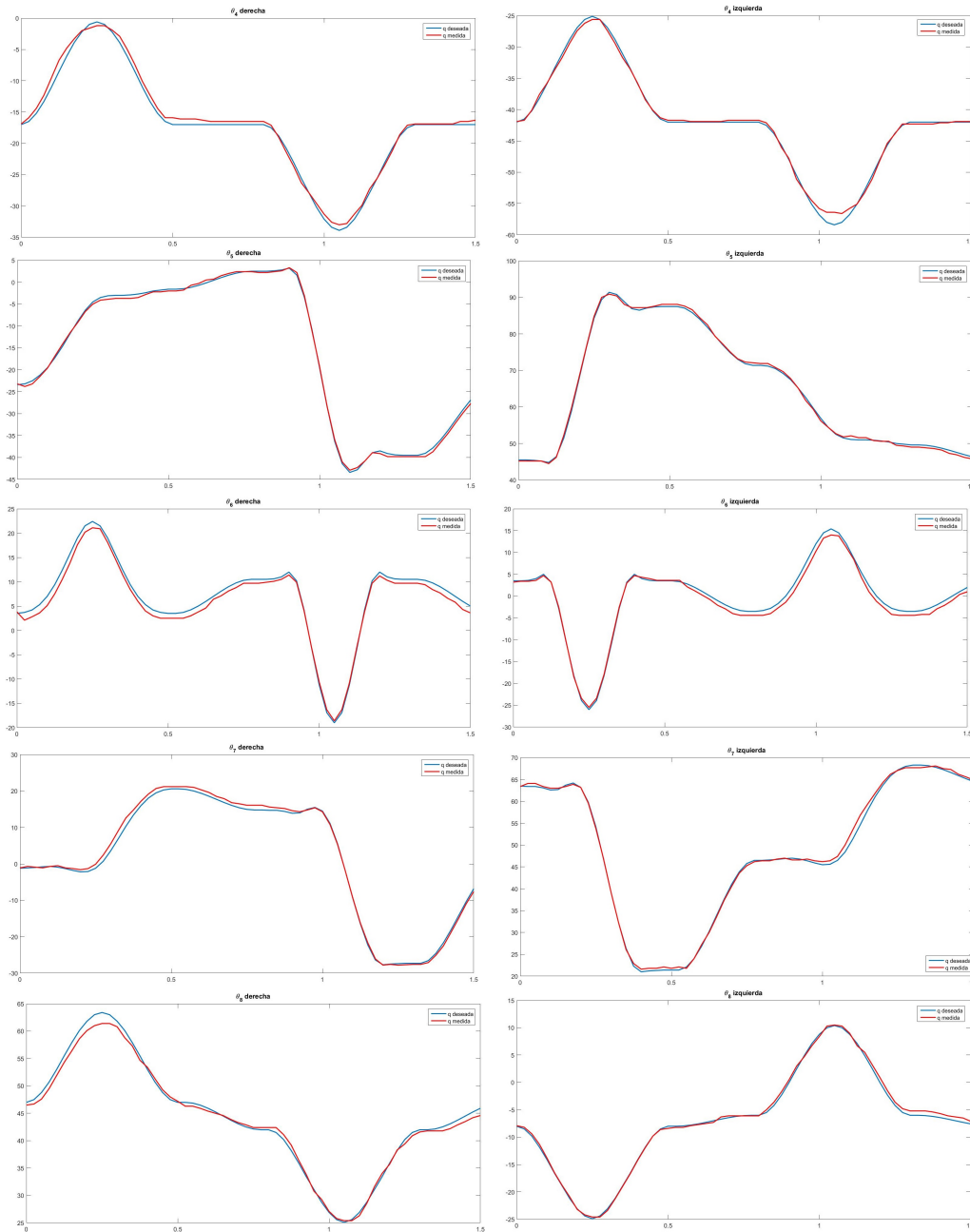


Figura 5.9: Posición deseada vs medida

## 5.2. Conclusiones

Durante la realización de la tesis, se han manejado conceptos y técnicas tanto nuevas como antiguas. El modelado del robot no se hace bajo el método convencional de

## CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

Denavit-Hartenberg, en lugar de eso se maneja una nueva técnica que a la larga facilita el trabajo y el análisis del modelado geométrico directo e inverso.

Se logra desarrollar un análisis analítico para resolver el modelo geométrico inverso y obtener las ecuaciones de la solución, podemos programar estas soluciones y así optimizar la obtención de las variables articulares.

Pero la mayor aportación de esta tesis, es el desarrollo de nuevo hardware y software. El Robonova-I se consideraba un robot obsoleto, esto se debía principalmente a la existencia de mejores robots humanoides con más prestaciones, y la utilización de una tarjeta de control básica y que es difícil de conseguir.

Aun así, el Robonova-I puede ser de utilidad con los sensores y tarjeta adecuadas. Siendo un robot con una estructura simple, funcionaría como plataforma para experimentar con diferentes sensores, leyes de control y tipos de caminado.

La tarjeta de control Raspberry Pi que ahora esta en el Robonova-I, puede ser reemplazada fácilmente, tiene su propio sistema operativo y el poder computacional para lograr diferentes técnicas de caminado e implementar algoritmos visión. Con lo anterior no solo se consiguió recuperar el funcionamiento del robot, se logró mejorarlo y añadirle más prestaciones de las que tenía originalmente.

A pesar de que queda trabajo por realizar en este robot humanoide se consiguió el objetivo principal, además de proporcionar una base sólida para futuros trabajos en materia de visión y control.



## 5.3. Trabajo Futuro

Se proponen los siguientes puntos como trabajo futuro a desarrollar para el robot humanoide Robonova-I

- Agregar una unidad de medición inercial (IMU) y sensores de fuerza en la planta de los pies del Robonova-I para desarrollar un control en lazo cerrado
- Instalar y desarrollar el sistema de visión para el Robonova-I
- Mejorar la conexión WiFi entre la PC y Raspberry Pi
- Mejorar el controlador usando un dispositivo móvil

# Apéndice A

## Marco Teórico

### A.1. Modelado

Un robot [15] [22] se puede considerar como elementos conectados mediante articulaciones, una secuencia de  $n + 1$  eslabones y  $n$  articulaciones. Cada eslabón se define como rígido, y cada articulación se define ideal. Las articulaciones se dividen en rotacionales (R) A.1 y prismáticas (P) A.2, y representa la conexión entre 2 eslabones.

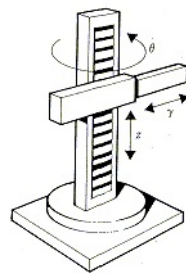


Figura A.1: Ejemplo de Junta Rotacional

## APÉNDICE A. MARCO TEÓRICO

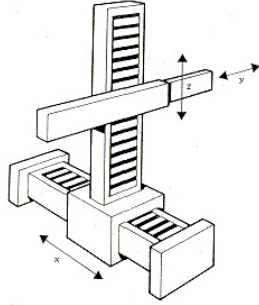


Figura A.2: Ejemplo de Junta Prismática

Los eslabones se numeran empezando con el eslabón 0, este eslabón se considera la base, en el caso del Robonova I, sería el torso.

Una vez que se tiene el eslabón base se define las variables que siguen del robot, a cada articulación  $i$  se le asigna la variable  $q_i$  y también un referencial. Para cada articulación rotacional, el eje  $z_i$  se fija en el eje de rotación de la articulación que une los eslabones  $i$  y  $i - 1$  y para una articulación prismática el eje  $z_i$  se fija en la dirección de la traslación. El eje  $x_{i-1}$  se establece perpendicular al eje  $z_{i-1}$  y  $z_i$ . [22]

Conociendo la cantidad de articulaciones que se tienen se conocen los grados de libertad del robot. Un robot con 6 grados de libertad, generalmente, tiene 3 grados de libertad para posición y 3 grados de libertad para orientación.

Para obtener la localización de un punto del robot, se necesita conocer el sistema de coordenadas al cual esta referenciado. Cuando el sistema de coordenadas esta establecido se puede ubicar cualquier punto con un vector de posición.

$${}^i P = \begin{bmatrix} {}^i P_x \\ {}^i P_y \\ {}^i P_z \end{bmatrix} \quad (\text{A.1})$$

## APÉNDICE A. MARCO TEÓRICO

Para la orientación de un cuerpo en un sistema de coordenadas de 3 dimensiones, el sistema de coordenadas del cuerpo  $i(x_i, y_i, z_i)$  se proyecta en el sistema de coordenadas  $i-1(x_{i-1}, y_{i-1}, z_{i-1})$  lo que resulta en la matriz de rotación siguiente

$${}^{i-1}R_i = \begin{bmatrix} x_i \cdot x_{i-1} & y_i \cdot x_{i-1} & z_i \cdot x_{i-1} \\ x_i \cdot y_{i-1} & y_i \cdot y_{i-1} & z_i \cdot y_{i-1} \\ x_i \cdot z_{i-1} & y_i \cdot z_{i-1} & z_i \cdot z_{i-1} \end{bmatrix} \quad (\text{A.2})$$

Donde la notación es

- $R$  es la matriz de rotación
- $i$  es el referencial rotado con respecto a  $i-1$
- $i-1$  es el referencial original

Una vez que se sabe como representar la posición y orientación de un objeto, se establece la definición de un movimiento en un cuerpo rígido rígido. El movimiento de un cuerpo rígido es aquel que se compone de una traslación y una rotación de ese punto en un referencial base, como se observa en la ecuación A.3

$${}^0p = {}^0R_1 {}^1p + {}^0d_1 \quad (\text{A.3})$$

Donde  ${}^0p$  es el vector base,  ${}^0R_1$  es la rotación final del cuerpo proyectado al referencial base,  ${}^1p$  es el vector final de posición del cuerpo y  ${}^0d_1$  es la traslación final del cuerpo proyectado en el referencial base. Si se tienen 2 movimientos rígidos, por ejemplo, la

## APÉNDICE A. MARCO TEÓRICO

ecuación sería

$${}^0p = {}^0R_1{}^1p + {}^0d_1 \quad (\text{A.4})$$

$${}^1p = {}^1R_2{}^2p + {}^1d_2 \quad (\text{A.5})$$

Substituyendo la ecuación A.5 en A.4 quedaría

$${}^0p = {}^0R_1{}^1R_2 + {}^0R_1{}^1d_2 + {}^0d_1 \quad (\text{A.6})$$

Simplificando quedaría

$${}^0p = {}^0R_2{}^2p + {}^0d_2 \quad (\text{A.7})$$

Donde  ${}^0R_2 = {}^0R_1{}^1R_2$  y  ${}^0d_2 = {}^0R_1{}^1d_2 + {}^0d_1$ . Esta ecuación puede ser representada mejor en forma de matriz

$$\begin{bmatrix} {}^0R_1 & {}^0d_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1R_2 & {}^1d_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0R_1{}^1R_2 & {}^0R_1{}^1d_2 + {}^0d_1 \\ 0 & 1 \end{bmatrix} \quad (\text{A.8})$$

Y se puede reescribir la matriz de la siguiente forma

$${}^0A_2 = \begin{bmatrix} {}^0R_2 & {}^0d_2 \\ 0 & 1 \end{bmatrix} \quad (\text{A.9})$$

## APÉNDICE A. MARCO TEÓRICO

Generalizando la matriz quedaría de la siguiente forma

$${}^0A_i = \begin{bmatrix} {}^0R_i & {}^0d_i \\ 0^{1 \times 3} & 1 \end{bmatrix} \quad (\text{A.10})$$

A esta representación A.10 del movimiento de un cuerpo rígido se le conoce como transformación homogénea. Un conjunto básico de estas transformaciones es el siguiente

$$T(x, a) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.11})$$

$$T(y, a) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.12})$$

$$T(z, a) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.13})$$

Donde  $T(\text{eje}, \text{distancia})$  es la traslación una distancia  $a$  alrededor del eje seleccionado,

## APÉNDICE A. MARCO TEÓRICO

y sea  $R(\text{eje}, \text{angulo})$  la rotación un ángulo  $\theta$  alrededor del eje seleccionado.

$$R(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.14})$$

$$R(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.15})$$

$$R(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.16})$$

# Apéndice B

## Programas

### B.1. Mathematica

#### B.1.1. Transformaciones Homogeneas

(\*Autores: Ing. Jessica J. Maldonado

Dr. Alejandro Malo

Institución: CINVESTAV

Fecha última de modificación: Marzo 2016\*)

(\*Esta parte del programa, es para inicializar las variables usadas a lo largo de los demas programas a usar. Aqui se encuentran las transformaciones homogéneas básicas para la rotación en X y en Z, la traslación en X, Y, Z\*)

```
(*Matriz Base*)
R0 = IdentityMatrix[4];
(*Rotación en X*)
Rx[alpha_] := {{1, 0, 0, 0}, {0, Cos[alpha], -Sin[alpha], 0},
{0, Sin[alpha], Cos[alpha], 0}, {0, 0, 0, 1}};
(*Traslación en X*)
Tx[b_] := {{1, 0, 0, b}, {0, 1, 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}};
(*Rotación en Z*)
Rz[theta_] := {{Cos[theta], -Sin[theta], 0, 0}, {Sin[theta],
Cos[theta], 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}};
(*Traslación en Z*)
```



## APÉNDICE B. PROGRAMAS

```
Tz[r_] := {{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, r}, {0, 0, 0, 1}};
(*Traslación en Y*)
Ty[y_] := {{1, 0, 0, 0}, {0, 1, 0, y}, {0, 0, 1, 0}, {0, 0, 0, 1}};
(*Vector de Poción*)
TP[x_, y_,
z_] := {{1, 0, 0, x}, {0, 1, 0, y}, {0, 0, 1, z}, {0, 0, 0, 1}}
```

(\*En esta parte se inicializa las transformaciones de eslabón a eslabón, ya sea usando 4 parámetros (D-H) o usando 6 Parámetros (K-K) \*)

```
T4[alfa_, d_, theta_, r_] := Rx[alfa].Tx[d].Rz[theta].Tz[r];
T6[gama_, b_, alfa_, d_, theta_, r_] :=
Rz[gama].Tz[b].Rx[alfa].Tx[d].Rz[theta].Tz[r];
```

(\*Aquí se tienen funciones para separar la matriz en secciones, la parte de rotación y la parte de traslación. Además de cada parte de la matriz de rotación\*)

```
A[M_] := Take[M.{{1, 0, 0}, {0, 1, 0}, {0, 0, 1}, {0, 0, 0}}, 3]
v3[M_, v_] := Take[M.v, 3]
s[M_] := v3[M, {1, 0, 0, 0}]
n[M_] := v3[M, {0, 1, 0, 0}]
a[M_] := v3[M, {0, 0, 1, 0}]
p[M_] := v3[M, {0, 0, 0, 1}]
```

### B.1.2. Gráfica

(\*Autores: Ing. Jessica J. Maldonado  
Dr. Alejandro Malo  
Institución: CINVESTAV  
Fecha última de modificación: Marzo 2016\*)

(\*Funciones para graficar usadas mas adelante\*)

```
(*Gráfica de ejes X-Z con etiquetas*)
RefT[T_, eti_, inc_: 0] := {RGBColor[1, 0, 0], Arrowheads[.01],
Arrow[{p[T], p[T] + (3.5 + inc) s[T]}], RGBColor[0, 0, 1],
Arrow[{p[T], p[T] + 3.5 a[T]}], RGBColor[0, 0, 0],
Text[Subscript["x", eti], p[T] + (2.5 + inc) s[T] + {0, 1, 1},
BaseStyle -> {FontSize -> 8}],
```

## APÉNDICE B. PROGRAMAS

```
Text[Subscript["z", eti], p[T] + 2.5 a[T] + {1, 1, 0},  
BaseStyle -> {FontSize -> 8}]}
```

```
(*Gráfica de ejes X-Z-Y con etiquetas*)  
Ref3[T_, eti_: "0"] := {RGBColor[1, 0, 0], Arrowheads[.01],  
Arrow[{p[T], p[T] + 5 s[T]}, RGBColor[0, 1, 0],  
Arrow[{p[T], p[T] + 5 n[T]}, RGBColor[0, 0, 1],  
Arrow[{p[T], p[T] + 5 a[T]}, RGBColor[0, 0, 0],  
Text[Subscript["x", eti], p[T] + 2.5 s[T],  
BaseStyle -> {FontSize -> 10}],  
Text[Subscript["y", eti], p[T] + 2.5 n[T],  
BaseStyle -> {FontSize -> 10}],  
Text[Subscript["z", eti], p[T] + 2.5 a[T],  
BaseStyle -> {FontSize -> 10}]}
```

(\*Módulo para graficar TODO el modelo geométrico directo, se ingresan los ángulos en los que se quiere cada articulación en el módulo del modelo geométrico directo y los resultados de ese módulo se utilizan como entras para el modulo de gráficas y así se grafica todo el modelo geométrico directo\*)

```
Grafica[M_] :=  
Module[{h1, h2, d1, d2, d3, d4, d5, d6, d7, d8, d9, i1, i2, i3, i4,  
i5, i6, i7, i8, i9, fd, fi, tp1, tp2, tp3, tp4, tp5, tp6, tp7, tp8,  
h3, h4},  
(*Variables para las articulaciones de la cabeza*)  
h1 = M[[1, 1]];  
h2 = M[[2, 1]];  
d1 = M[[5, 1]];  
d2 = M[[6, 1]];  
d3 = M[[7, 1]];  
d4 = M[[8, 1]];  
d5 = M[[13, 1]];  
d6 = M[[14, 1]];  
d7 = M[[15, 1]];  
d8 = M[[16, 1]];  
d9 = M[[17, 1]];  
i1 = M[[9, 1]];  
i2 = M[[10, 1]];  
i3 = M[[11, 1]];  
i4 = M[[12, 1]];  
i5 = M[[18, 1]];  
i6 = M[[19, 1]];  
i7 = M[[20, 1]];
```

## APÉNDICE B. PROGRAMAS

```
i8 = M[[21, 1]];
i9 = M[[22, 1]];
fd = M[[23, 1]];
fi = M[[24, 1]];
tp1 = M[[25, 1]];
tp2 = M[[26, 1]];
tp3 = M[[27, 1]];
tp4 = M[[28, 1]];
tp5 = M[[29, 1]];
tp6 = M[[30, 1]];
tp7 = M[[31, 1]];
tp8 = M[[32, 1]];
h3 = M[[3, 1]];
h4 = M[[4, 1]];

Graphics3D[{Ref3[R0], RefT[h1, "h1"], RefT[h2, "h2"],
RefT[d1, "d1"], RefT[d2, "d2"], RefT[d3, "d3"], Ref3[d4, "d4"],
RefT[i1, "i1"], RefT[i2, "i2"], RefT[i3, "i3"], Ref3[i4, "i4"],
RefT[d5, "d5"], RefT[d6, "d6"], RefT[d7, "d7"], RefT[d8, "d8"],
RefT[d9, "d9"],
RefT[i5, "i5"], RefT[i6, "i6"], RefT[i7, "i7"], RefT[i8, "i8"],
RefT[i9, "i9"],
Ref3[fd, "fd"], Ref3[fi, "fi"], CapForm["Butt"],
White, Opacity[0.5],
Tube[{p[R0], p[h1], p[h2]}, 0.2],
White, Opacity[0.5],
Tube[{p[R0], p[d1], p[d2], p[d3], p[d4]}, 0.2],
Tube[{p[R0], p[i1], p[i2], p[i3], p[i4]}, 0.2],
White, Opacity[0.5],
Tube[{p[R0], p[d5], p[d6], p[d7], p[d8], p[d9]}, 0.2],
Tube[{p[R0], p[i5], p[i6], p[i7], p[i8], p[i9]}, 0.2],
Red, Opacity[0.5],
Polygon[{p[tp1], p[tp2], p[tp3], p[tp4], p[tp1]}],
Polygon[{p[tp5], p[tp6], p[tp7], p[tp8], p[tp5]}],
Cuboid[{p[h3], p[h4]}], Boxed -> False,
ViewVertical -> {0, 0, 1}]]
```

### B.1.3. Modelo Geométrico Directo completo del Robonova-

## I

## APÉNDICE B. PROGRAMAS

```
(*Autores: Ing. Jessica J. Maldonado  
Dr. Alejandro Malo  
Institución: CINVESTAV  
Fecha última de modificación: Marzo 2016*)
```

```
(*Modulo de Geometria directa,  
se ingresan los ángulos deseados para lugar  
enviar los resultados a gráficar.*)  
lufb = 8.98;  
lufp = 1.2;  
MGD[{thetah1_, thetah2_, thetali_, theta2i_, theta3i_, theta4i_,  
theta5i_, theta6i_, theta7i_, theta8i_, thetald_, theta2d_,  
theta3d_, theta4d_, theta5d_, theta6d_, theta7d_, theta8d_}] :=
```

```
(*thetah1 y thetah2 son los ángulos relacionados con la cabeza*)  
(*1i,2i,3i ángulos para el brazo izquierdo*)  
(*4i,5i,6i,7i,8i ángulos para la pierna izquierda*)  
(*1d,2d,3d ángulos para el brazo izquierdo*)  
(*4d,5d,6d,7d,8d ángulos para la pierna izquierda*)
```

```
Module[{lu1, lu2, lu3, lu4, lufb, lu5, lu6, lu7, lu8, lu9, lu10,  
lufp, lu11, alpha1},  
lu1 = 3.25;  
lu2 = 5.3;  
lu3 = 2;  
lu4 = 4.8;  
lufb = 8.98;  
lu5 = 3.3;  
lu6 = 2.8;  
lu7 = 4;  
lu8 = 5.84;  
lu9 = 4.8;  
lu10 = 4;  
lufp = 1.2;  
lu11 = 1.45;  
alpha1 = 90 Degree;
```

```
R0 = IdentityMatrix[4];
```

```
(*De la base a la cabeza*)
```

```
TH1 = R0.T4[0, 0, thetah1, lu11 + lu1 + lu5];  
TH2 = TH1.T4[alpha1, 0, thetah2 + alpha1, 0];  
TH3 = TH2.TP[0, -1, -2];
```

## APÉNDICE B. PROGRAMAS

```
TH4 = TH2.TP[2, 1, 2];
```

```
(*De la base al efector final del brazo derecho*)
```

```
BD1 = R0.T6[0, lu1 + lu5, -alpha1, 0,  
theta1d + alpha1, -(lu2 + lu3)];  
BD2 = BD1.T4[alpha1, 0, theta2d, 0];  
BD3 = BD2.T4[0, lu4, theta3d, 0];  
BD4 = BD3.T4[0, 0, 0, 0].TP[lufb, 0, 0];
```

```
(*De la base al efector final del brazo izquierdo*)
```

```
BI1 = R0.T6[0, lu1 + lu5, -alpha1, 0,  
theta1i + alpha1, (lu2 + lu3)];  
BI2 = BI1.T4[alpha1, 0, theta2i, 0];  
BI3 = BI2.T4[0, lu4, theta3i, 0];  
BI4 = BI3.T4[0, 0, 0, 0].TP[lufb, 0, 0];
```

```
(*De la base al efector final de la pierna derecha*)
```

```
D1 = R0.T6[alpha1, 0, alpha1, -(lu6), theta4d - alpha1, 0];  
D2 = D1.T4[-alpha1, lu7, theta5d, 0];  
D3 = D2.T4[0, lu8, theta6d, 0];  
D4 = D3.T4[0, lu9, theta7d, 0];  
D5 = D4.T4[alpha1, lu10, theta8d, 0];  
TP1 = D5.TP[lufp, 1, -2];  
TP2 = D5.TP[lufp, 1, 2];  
TP3 = D5.TP[lufp, -1, 2];  
TP4 = D5.TP[lufp, -1, -2];
```

```
(*De la base al efector final de la pierna izquierda*)
```

```
I1 = R0.T6[alpha1, 0, alpha1, (lu6), theta4i - alpha1, 0];  
I2 = I1.T4[-alpha1, lu7, theta5i, 0];  
I3 = I2.T4[0, lu8, theta6i, 0];  
I4 = I3.T4[0, lu9, theta7i, 0];  
I5 = I4.T4[alpha1, lu10, theta8i, 0];  
TP5 = I5.TP[lufp, 1, -2];  
TP6 = I5.TP[lufp, 1, 2];  
TP7 = I5.TP[lufp, -1, 2];  
TP8 = I5.TP[lufp, -1, -2];  
Rfd = D5.TP[lufp, 0, 0];  
Rfi = I5.TP[lufp, 0, 0];
```

```
Return[{{TH1}, {TH1}, {TH3}, {TH4}, {BD1}, {BD2}, {BD3}, {BD4}, \
```

## APÉNDICE B. PROGRAMAS

```
{BI1}, {BI2}, {BI3}, {BI4}, {D1}, {D2}, {D3}, {D4}, {D5}, {I1}, {I2}, \
{I3}, {I4}, {I5}, {Rfd}, {Rfi}, {TP1}, {TP2}, {TP3}, {TP4}, {TP5}, \
{TP6}, {TP7}, {TP8}}]]
```

### B.1.4. Modelo Geométrico Inverso para brazos

```
(*Autores: Ing. Jessica J. Maldonado
Dr. Alejandro Malo
Institución: CINVESTAV
Fecha última de modificación: Junio 2016*)
```

```
(*Modulo para modelo geométrico inverso para brazos,
la entrada del módulo es la matriz de posición deseada*)
MGIB[M_, {signo_}] :=
Module[{sx, sy, sz, nx, ny, nz, ax, ay, az, px, py, pz, lu1, lu2,
lu3, lu4, lu5, lufb, stheta1, ctheta1, theta1, stheta3, ctheta3,
theta3, stheta2, ctheta2, theta2, x1, x2, B1, B2, R1, R2, theta3N,
B1N, B2N, stheta2N, ctheta2N, theta2N},
lu1 = 3.25;
lu2 = 5.3;
lu3 = 2;
lu4 = 4.8;
lufb = 8.98;
lu5 = 3.3;
sx = M[[1, 1]];
sy = M[[2, 1]];
sz = M[[3, 1]];
nx = M[[1, 2]];
ny = M[[2, 2]];
nz = M[[3, 2]];
ax = M[[1, 3]];
ay = M[[2, 3]];
az = M[[3, 3]];
px = M[[1, 4]];
py = M[[2, 4]];
pz = M[[3, 4]];

stheta1 = px;
ctheta1 = -(lu1 + lu5) + pz;
theta1 = ArcTan[-ctheta1, -stheta1];

ctheta3 = ((lu1 + lu5)^2 + px^2 + py^2 - 2 (lu1 + lu5) pz + pz^2 -
```

## APÉNDICE B. PROGRAMAS

```
2 (lu2 + lu3) py signo + (lu2 + lu3)^2 signo^2 - (lu4^2 +
lufb^2))/(2 lu4 lufb);
Print["ct3 ", ctheta3];
stheta3 = Sqrt[1 - (ctheta3^2)];
theta3 = ArcTan[ctheta3, stheta3];
theta3N = ArcTan[ctheta3, -stheta3];

x1 = lu4;
x2 = lufb;

B1 = x1 + x2*Cos[theta3];
B2 = x2*Sin[theta3];
R1 = ((lu1 + lu5) - pz) Cos[theta1] - px Sin[theta1];
R2 = py - (lu2 + lu3) signo;

stheta2 = (B1*R2 - B2*R1)/(B1^2 + B2^2);
ctheta2 = (B1*R1 + B2*R2)/(B1^2 + B2^2);
theta2 = ArcTan[ctheta2, stheta2];

B1N = x1 + x2*Cos[theta3N];
B2N = x2*Sin[theta3N];

stheta2N = (B1N*R2 - B2N*R1)/(B1N^2 + B2N^2);
ctheta2N = (B1N*R1 + B2N*R2)/(B1N^2 + B2N^2);
theta2N = ArcTan[ctheta2N, stheta2N];

(*Diferentes resultados, como por ejemplo codo afuera
o codo dentro*)

Return[Re[{{theta1, theta2, theta3}, {theta1, theta2N,
theta3}, {theta1, theta2N, theta3N}, {theta1, theta2,
theta3N}}]]]
```

## B.2. Matlab

### B.2.1. Modelo Geométrico Inverso para piernas

\*Autores: Ing. Jessica J. Maldonado  
Dr. Alejandro Malo

## APÉNDICE B. PROGRAMAS

Institución: CINVESTAV

Fecha última de modificación: Agosto 2016\*

\*Función para usar en matlab. La función necesita como variables las posiciones finales de X,Y,Z y el signo dice si es pierna izquierda o derecha\*

```
function angulos=MGIP(PX,PY,PZ,signo)
CP=[0 0 1 PX; 0 1 0 PY; -1 0 0 -PZ; 0 0 0 1];

l1 = 1.2;
l2 = 4;
l3 = 4.8;
l4 = 5.84;
l6 = 2.8;
sx = CP(1,1);
sy = CP(2,1);
sz = CP(3,1);
nx = CP(1,2);
ny = CP(2,2);
nz = CP(3,2);
ax = CP(1,3);
ay = CP(2,3);
az = CP(3,3);
px = CP(1,4);
py = CP(2,4);
pz = CP(3,4);

st6 = py-l6*signo-l1*sy;
ct6 = pz-l1*sz;
t6 = atan2d(st6,-ct6);

st10 = sy*cosd(t6)+sz*sind(t6);
ct10 = ny*cosd(t6)+nz*sind(t6);
t10 = atan2d(st10,ct10);

X = l4;
Y = l3;
R1 = -l2+cosd(t6)*(-pz+l1*sz+l2*sz*cosd(t10)-l2*nz*sind(t10))+
(py-l6*signo-l1*sy-l2*sy*cosd(t10)+l2*ny*sind(t10))*sind(t6);
R2 = -px+l1*sx+l2*sx*cosd(t10)-l2*nx*sind(t10);

ct8 = (R1^2+R2^2-X^2-Y^2)/(2*X*Y);
st8 = sqrt(1-ct8^2);
t8 = atan2d(st8,ct8);
```



## APÉNDICE B. PROGRAMAS

```
B1 = X+Y*cosd(t8);
B2 = Y*sind(t8);

st7=(B1*R2-B2*R1)/(B1^2+B2^2);
ct7=(B1*R1+B2*R2)/(B1^2+B2^2);
t7=atan2d(st7,ct7);

ct9 =(l2^2+l4^2+px^2+py^2+pz^2-2*l6*py*signo+l6^2*signo^2-
2*l1*px*sx+l1^2*sx^2-2*l1*py*sy+2*l1*l6*signo*sy+l1^2*sy^2-
2*l1*pz*sz+l1^2*sz^2+2*(pz-l1*sz)*cosd(t6)*(l2+l4*cosd(t7))-
2*l2*py*sind(t6)+2*l2*l6*signo*sind(t6)+2*l1*l2*sy*sind(t6)+
2*l4*cosd(t7)*(l2+(-py+l6*signo+l1*sy)*sind(t6))+
2*l4*px*sind(t7)-2*l1*l4*sx*sind(t7)-l2^2-l3^2)/(2*l2*l3);
st9 = sqrt(1-ct9^2);
t9 = atan2d(-st9,ct9);

angulos=[t6 t7 t8 t9 t10];
end

*La salida son los ángulos para alcanzar la posición deseada
con la rodilla viendo hacia enfrente*
```

## B.3. Python

### B.3.1. Módulo spline

```
#Autores: Ing. Jessica J. Maldonado
#Institución: CINVESTAV
#Fecha última de modificación: Octubre 2016*

#Programa para generar un spline, el modelo necesita de las variables
#posición inicial, final, tiempo inicial, tiempo final, velocidad inicial
#y velocidad final

import math
def cubic(x0,xf,t0,tf,v0,vf):

X=x0+v0*t0+((3*(xf-x0)-tf*(2*v0+vf))/tf**2)*t0**2+((2*(x0-xf)
```

## APÉNDICE B. PROGRAMAS

```
+tf*(v0+vf))/tf**3)*t0**3
```

```
return X
```

### B.3.2. Módulo Check Sum

```
#Autores: Ing. Jessica J. Maldonado
```

```
#Institución: CINVESTAV
```

```
#Fecha última de modificación: Octubre 2016*
```

```
#Programa para calcular el check sum de los comandos seriales
```

```
import operator
```

```
def csum(C,P1,P2):
```

```
SumaP=128+C+P1+P2
```

```
Ck3=operator.mod((256-operator.mod(SumaP,256)),256)
```

```
return (hex(Ck3))
```

### B.3.3. Módulo Posición

```
#Autores: Ing. Jessica J. Maldonado
```

```
#Institución: CINVESTAV
```

```
#Fecha última de modificación: Octubre 2016*
```

```
#Programa para transformar los ángulos dados a comandos
```

```
#seriales de posición.
```

```
import cksum
```

```
import string
```

```
def position(ang,ID):
```

```
p=int(ang*10+1500)
```

```
y=hex(p)
```

```
C=int(ID,16)
```

```
P1=int(y[2],16)
```

```
a=''.join([y[3],y[4]])
```

```
P2=int(a,16)
```

```
ck=cksum.csum(C,P1,P2)
```

```
B=[int('80',16),C,P1,P2,int(ck,16),0,0]
```

```
x1=hex(int('80',16))
```

## APÉNDICE B. PROGRAMAS

```
x2=hex(C)
x3=hex(P1)
x4=hex(P2)
x5=hex(int(ck,16))
x6=hex(0)
x7=hex(0)
return B
```

### B.3.4. Módulo Caminado Parametrizado

```
#Autores: Ing. Jessica J. Maldonado
# Dr. Alejandro Malo
#Institución: CINVESTAV
#Fecha última de modificación: Octubre 2016*

#Programa de caminado parametrizado, se dan los parámetros
#para el caminado, la cantidad de pasos a dar, y el programa
#crea las trayectorias necesarias en el espacio de trabajo, las
#pasa al espacio articular y luego las traduce en comandos
#seriales de posición para los motores.

import math
import modelo
import cubico
import cksum
import pos
import string

def frange(start, stop, step):
    i=start
    while i<stop:
        yield i
        i += step

t=1          #tiempo de paso
T=t/2
Lp=3        #Distancia en X de paso
tds=0.2     #tiempo de doble soporte
h=17       #Altura del cuerpo
hb=0.5     #Rizo de cadera
hp=2       #altura de pie flotante
v0=0
```

## APÉNDICE B. PROGRAMAS

```
vf=0
db=2
dp=5          #distancia entre los pies
lds=2.8
sig=-1
dt=0.025     #variación de tiempo

X=[None]*21
Z=[None]*21
Y=[None]*21
Xfloat=[None]*21
Zfloat=[None]*21
Yfloat=[None]*21
pc=[[None] * 5 for i in range(21)]
pf=[[None] * 5 for i in range(21)]

tp1=[None]*21; tp2=[None]*21; tp3=[None]*21; tp4=[None]*21; tp5=[None]*21;
tp6=[None]*21; tp7=[None]*21; tp8=[None]*21; tp9=[None]*21; tp10=[None]*21;

tq1=[None]*21; tq2=[None]*21; tq3=[None]*21; tq4=[None]*21; tq5=[None]*21;
tq6=[None]*21; tq7=[None]*21; tq8=[None]*21; tq9=[None]*21; tq10=[None]*21;

M1=[None]*11; M2=[None]*11; M3=[None]*11; M4=[None]*11; M5=[None]*11;
M6=[None]*11; M7=[None]*11; M8=[None]*11; M9=[None]*11; M10=[None]*11;

N1=[None]*11; N2=[None]*11; N3=[None]*11; N4=[None]*11; N5=[None]*11;
N6=[None]*11; N7=[None]*11; N8=[None]*11; N9=[None]*11; N10=[None]*11;

tm1=[None]*32; tm2=[None]*32; tm3=[None]*32; tm4=[None]*32; tm5=[None]*32;
tm6=[None]*32; tm7=[None]*32; tm8=[None]*32; tm9=[None]*32; tm10=[None]*32;

tn1=[None]*32; tn2=[None]*32; tn3=[None]*32; tn4=[None]*32; tn5=[None]*32;
tn6=[None]*32; tn7=[None]*32; tn8=[None]*32; tn9=[None]*32; tn10=[None]*32;

"%--Movimiento del cuerpo en X de 0 a T--%"

j=0
for i in frange(0,T+dt,dt):
    tiempo=round(i,3)
    X[j]=round(cubico.cubic(-Lp/2,Lp/2,tiempo,T,0,0),3)
    j=j+1

"%--Movimiento del cuerpo en Z de 0 a T--%"
```

## APÉNDICE B. PROGRAMAS

```
j=0
for i in frange(0, (T+dt)/2, dt):
    tiempo=round(i, 3)
    Z[j]=round(cubico.cubic(h, h+hb, tiempo, T/2, 0, 0), 3)
    j=j+1

for i in frange(T/2, (T+dt), dt):
    tiempo=round(i, 3)
    Z[j-1]=round(cubico.cubic(h+hb, h, tiempo-(T/2), T/2, 0, 0), 3)
    j=j+1

"%--Movimiento del pie flotante en X--%"

j=0
for i in frange(0, (tds/2)+dt, dt):
    tiempo=round(i, 3)
    Xfloat[j]=-Lp
    j=j+1

for i in frange(tds/2, (T-(tds/2))+dt, dt):
    tiempo=round(i, 3)
    Xfloat[j-1]=round(cubico.cubic(-Lp, Lp, tiempo-(tds/2), T-tds, 0, 0), 3)
    j=j+1

for i in frange((T-(tds/2)), T+dt, dt):
    tiempo=round(i, 3)
    Xfloat[j-2]=Lp
    j=j+1

"%--Movimiento del pie flotante en Z--%"

j=0
for i in frange(0, (tds/2)+dt, dt):
    Zfloat[j]=0
    j=j+1

for i in frange((tds/2), (T/2)+dt, dt):
    tiempo=round(i, 3)
    Zfloat[j-1]=round(cubico.cubic(0, hp, tiempo-(tds/2), (T-tds)/2, 0, 0), 3)
    j=j+1

for i in frange(T/2, (T-(tds/2))+dt, dt):
    tiempo=round(i, 3)
    Zfloat[j-3]=round(cubico.cubic(hp, 0, tiempo-(T/2), (T-tds)/2, 0, 0), 3)
    j=j+1
```

## APÉNDICE B. PROGRAMAS

```
for i in frange(T-(tds/2),T+dt,dt):
Zfloat[j-4]=0
j=j+1

"%---Movimiento del cuerpo en Y---%"

j=0
for i in frange(0,(T+dt)/2,dt):
tiempo=round(i,3)
Y[j]=round(cubico.cubic(-lds*sig,db*sig,tiempo,T/2,0,0),3)
j=j+1

for i in frange(T/2,(T+dt),dt):
tiempo=round(i,3)
Y[j-1]=round(cubico.cubic(db*sig,-lds*sig,tiempo-(T/2),T/2,0,0),3)
j=j+1

"%---Movimiento del pie flotante en Y----%"

j=0
for i in frange(0,(tds/2)+dt,dt):
Yfloat[j]=-2*lds*sig
j=j+1

for i in frange((tds/2),(T/2)+dt,dt):
tiempo=round(i,3)
Yfloat[j-1]=round(cubico.cubic(-2*sig*lds,-dp*sig,tiempo-(tds/2),(T-tds)/2,0,0),3)
j=j+1

for i in frange(T/2,(T-(tds/2))+dt,dt):
tiempo=round(i,3)
Yfloat[j-3]=round(cubico.cubic(-dp*sig,-2*sig*lds,tiempo-(T/2),(T-tds)/2,0,0),3)
j=j+1

for i in frange(T-(tds/2),T+dt,dt):
Yfloat[j-4]=-2*lds*sig
j=j+1

"---Calculo de variables articulares---"

for i in frange(0,21,1):
pc[i]=modelo.MGIP(-X[i],Y[i],Z[i],1)

for i in frange(0,21,1):
```

## APÉNDICE B. PROGRAMAS

```
pf[i]=modelo.MGIP(-Xfloat[i],-Yfloat[i]+Y[i],h-Zfloat[i],-1)
```

```
"---%Paso pierna derecha en fase de apoyo%---"
```

```
for i in frange(0,21,1):  
tp1[i]=-pc[i][0]+5  
tp2[i]=-pc[i][3]+3  
tp3[i]=-pc[i][2]  
tp4[i]=pc[i][1]  
tp5[i]=pc[i][4]  
tp6[i]=pf[i][4]  
tp7[i]=-pf[i][3]  
tp8[i]=-pf[i][2]  
tp9[i]=-pf[i][1]  
tp10[i]=pf[i][0]
```

```
"---%Paso pierna izquierda en fase de apoyo%---"
```

```
for i in frange(0,21,1):  
tq1[i]=-pf[i][4]  
tq2[i]=-pf[i][1]  
tq3[i]=-pf[i][2]  
tq4[i]=pf[i][3]  
tq5[i]=pf[i][0]  
tq6[i]=pc[i][0]  
tq7[i]=-pc[i][1]  
tq8[i]=-pc[i][2]  
tq9[i]=-pc[i][3]+2  
tq10[i]=pc[i][4]+2
```

```
"---%Fase de doble soporte%---"
```

```
df=0.05  
j=0  
for i in frange(0,T,df):  
tiempo=round(i,3)  
M1[j]=round(cubico.cubic(tp1[20],tq1[0],tiempo,T,0,0),3)  
M2[j]=round(cubico.cubic(tp2[20],tq2[0],tiempo,T,0,0),3)  
M3[j]=round(cubico.cubic(tp3[20],tq3[0],tiempo,T,0,0),3)  
M4[j]=round(cubico.cubic(tp4[20],tq4[0],tiempo,T,0,0),3)  
M5[j]=round(cubico.cubic(tp5[20],tq5[0],tiempo,T,0,0),3)  
M6[j]=round(cubico.cubic(tp6[20],tq6[0],tiempo,T,0,0),3)  
M7[j]=round(cubico.cubic(tp7[20],tq7[0],tiempo,T,0,0),3)  
M8[j]=round(cubico.cubic(tp8[20],tq8[0],tiempo,T,0,0),3)  
M9[j]=round(cubico.cubic(tp9[20],tq9[0],tiempo,T,0,0),3)
```

## APÉNDICE B. PROGRAMAS

```
M10[j]=round(cubico.cubic(tp10[20],tq10[0],tiempo,T,0,0),3)
N1[j]=round(cubico.cubic(tq1[20],tp1[0],tiempo,T,0,0),3)
N2[j]=round(cubico.cubic(tq2[20],tp2[0],tiempo,T,0,0),3)
N3[j]=round(cubico.cubic(tq3[20],tp3[0],tiempo,T,0,0),3)
N4[j]=round(cubico.cubic(tq4[20],tp4[0],tiempo,T,0,0),3)
N5[j]=round(cubico.cubic(tq5[20],tp5[0],tiempo,T,0,0),3)
N6[j]=round(cubico.cubic(tq6[20],tp6[0],tiempo,T,0,0),3)
N7[j]=round(cubico.cubic(tq7[20],tp7[0],tiempo,T,0,0),3)
N8[j]=round(cubico.cubic(tq8[20],tp8[0],tiempo,T,0,0),3)
N9[j]=round(cubico.cubic(tq9[20],tp9[0],tiempo,T,0,0),3)
N10[j]=round(cubico.cubic(tq10[20],tp10[0],tiempo,T,0,0),3)
j=j+1
```

```
for i in frange(0,21,1):
```

```
tm1[i]=tp1[i]
tm2[i]=tp2[i]
tm3[i]=tp3[i]
tm4[i]=tp4[i]
tm5[i]=tp5[i]
tm6[i]=tp6[i]
tm7[i]=tp7[i]
tm8[i]=tp8[i]
tm9[i]=tp9[i]
tm10[i]=tp10[i]
tn1[i]=tq1[i]
tn2[i]=tq2[i]
tn3[i]=tq3[i]
tn4[i]=tq4[i]
tn5[i]=tq5[i]
tn6[i]=tq6[i]
tn7[i]=tq7[i]
tn8[i]=tq8[i]
tn9[i]=tq9[i]
tn10[i]=tq10[i]
```

```
for i in frange(21,32,1):
```

```
tm1[i]=M1[i-21]
tm2[i]=M2[i-21]
tm3[i]=M3[i-21]
tm4[i]=M4[i-21]
tm5[i]=M5[i-21]
tm6[i]=M6[i-21]
tm7[i]=M7[i-21]
tm8[i]=M8[i-21]
tm9[i]=M9[i-21]
```



## APÉNDICE B. PROGRAMAS

```
tm10[i]=M10[i-21]
tn1[i]=N1[i-21]
tn2[i]=N2[i-21]
tn3[i]=N3[i-21]
tn4[i]=N4[i-21]
tn5[i]=N5[i-21]
tn6[i]=N6[i-21]
tn7[i]=N7[i-21]
tn8[i]=N8[i-21]
tn9[i]=N9[i-21]
tn10[i]=N10[i-21]

"---%Numero de pasos%---"

n=2
f2=32
d=n*f2

t1=[None]*n*f2; t2=[None]*n*f2; t3=[None]*n*f2; t4=[None]*n*f2;
t5=[None]*n*f2;
t6=[None]*n*f2; t7=[None]*n*f2; t8=[None]*n*f2; t9=[None]*n*f2;
t10=[None]*n*f2;

q1=[None]*n*f2; q2=[None]*n*f2; q3=[None]*n*f2; q4=[None]*n*f2;
q5=[None]*n*f2;
q6=[None]*n*f2; q7=[None]*n*f2; q8=[None]*n*f2; q9=[None]*n*f2;
q10=[None]*n*f2;

p1=[[None] * 7 for i in range(d)]
p2=[[None] * 7 for i in range(d)]
p3=[[None] * 7 for i in range(d)]
p4=[[None] * 7 for i in range(d)]
p5=[[None] * 7 for i in range(d)]
p6=[[None] * 7 for i in range(d)]
p7=[[None] * 7 for i in range(d)]
p8=[[None] * 7 for i in range(d)]
p9=[[None] * 7 for i in range(d)]
p10=[[None] * 7 for i in range(d)]

P1=[[None] * 7 for i in range(d)]
P2=[[None] * 7 for i in range(d)]
P3=[[None] * 7 for i in range(d)]
P4=[[None] * 7 for i in range(d)]
P5=[[None] * 7 for i in range(d)]
P6=[[None] * 7 for i in range(d)]
```

## APÉNDICE B. PROGRAMAS

```
P7=[[None] * 7 for i in range(d)]
P8=[[None] * 7 for i in range(d)]
P9=[[None] * 7 for i in range(d)]
P10=[[None] * 7 for i in range(d)]

for i in frange (0,n+1,1):
    if i%2==0:
        for j in frange(f2*i-f2,f2*i,1):
            t1[j]=tn1[j-(f2*i)]
            t2[j]=tn2[j-(f2*i)]
            t3[j]=tn3[j-(f2*i)]
            t4[j]=tn4[j-(f2*i)]
            t5[j]=tn5[j-(f2*i)]
            t6[j]=tn6[j-(f2*i)]
            t7[j]=tn7[j-(f2*i)]
            t8[j]=tn8[j-(f2*i)]
            t9[j]=tn9[j-(f2*i)]
            t10[j]=tn10[j-(f2*i)]
        else:
            for j in frange(f2*i-f2,f2*i,1):
                t1[j]=tm1[j-(f2*i)]
                t2[j]=tm2[j-(f2*i)]
                t3[j]=tm3[j-(f2*i)]
                t4[j]=tm4[j-(f2*i)]
                t5[j]=tm5[j-(f2*i)]
                t6[j]=tm6[j-(f2*i)]
                t7[j]=tm7[j-(f2*i)]
                t8[j]=tm8[j-(f2*i)]
                t9[j]=tm9[j-(f2*i)]
                t10[j]=tm10[j-(f2*i)]

for i in frange (0,d,1):
    q1[i]=round(42+t1[i],1)
    q2[i]=round(-40.3+t2[i],1)
    q3[i]=round(87.4+t3[i],1)
    q4[i]=round(24.4+t4[i],1)
    q5[i]=round(-17+t5[i],1)
    q6[i]=round(-83.3+t6[i],1)
    q7[i]=round(23.6+t7[i],1)
    q8[i]=round(80.4+t8[i],1)
    q9[i]=round(8.4+t9[i],1)
    q10[i]=round(-8+t10[i],1)

for i in frange(0,d,1):
    p1[i]=bytes(pos.position(q1[i],'00'))
```

## APÉNDICE B. PROGRAMAS

```
p2[i]=bytes(pos.position(q2[i],'01'))
p3[i]=bytes(pos.position(q3[i],'02'))
p4[i]=bytes(pos.position(q4[i],'03'))
p5[i]=bytes(pos.position(q5[i],'04'))
p6[i]=bytes(pos.position(q6[i],'05'))
p7[i]=bytes(pos.position(q7[i],'06'))
p8[i]=bytes(pos.position(q8[i],'07'))
p9[i]=bytes(pos.position(q9[i],'08'))
p10[i]=bytes(pos.position(q10[i],'09'))
```

```
ser=serial.Serial(
port='/dev/ttyAMA0',
baudrate=19200,
parity=serial.PARITY_NONE,
stopbits=serial.STOPBITS_TWO,
bytesize=serial.EIGHTBITS
)
print(ser.isOpen())
```

```
for i in frange(0,d,1):
ser.write(p1[i])
ser.write(p2[i])
ser.write(p3[i])
ser.write(p4[i])
ser.write(p5[i])
ser.write(p6[i])
ser.write(p7[i])
ser.write(p8[i])
ser.write(p9[i])
ser.write(p10[i])
```

```
ser.close
```

# Apéndice C

## Soluciones de ecuaciones de la tabla

### 2.5

#### C.1. Tipo 1

Ecuación tipo 1

$$XS\theta_i + YC\theta_i = Z$$

Se pueden considerar 4 tipos de casos para la solución de las ecuación tipo 1 [17] [21]

- Si  $X = 0$  y  $Y \neq 0$

Se puede reescribir la ecuación como

$$C\theta_i = \frac{Z}{Y}$$

## APÉNDICE C. SOLUCIONES DE ECUACIÓN

y  $\theta_i$  sería

$$\theta_i = \arctan \left( \frac{\pm \sqrt{1 - (C\theta_i)^2}}{C\theta_i} \right) \quad (\text{C.1})$$

- Si  $Y = 0$  y  $X \neq 0$

Se reescribe la ecuación como

$$S\theta_i = \frac{Z}{X}$$

y  $\theta_i$  sería

$$\theta_i = \arctan \left( \frac{S\theta_i}{\pm \sqrt{1 - (S\theta_i)^2}} \right) \quad (\text{C.2})$$

- Si  $X$  y  $Y$  son diferentes de cero y  $Z = 0$

Se reescribe la ecuación y  $\theta_i$  sería

$$\theta_i = \arctan \left( \frac{-Y}{X} \right) \quad (\text{C.3})$$

- Si  $X, Y$  y  $Z$  no son cero

Reescribiendo la ecuación tipo 1 y elevando al cuadrado tenemos

$$Y^2 C^2 \theta_i = Y^2 (1 - S^2 \theta_i) = Z^2 - 2ZX S\theta_i + X^2 S^2 \theta_i$$

Y las ecuaciones para  $S\theta_i$  y  $C\theta_i$  quedarían

$$S\theta_i = \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \quad (\text{C.4})$$

$$C\theta_i = \frac{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \quad (\text{C.5})$$

## APÉNDICE C. SOLUCIONES DE ECUACIÓN

Siendo  $\varepsilon = \pm 1$ , las ecuaciones anteriores funcionan mientras  $X^2 + Y^2 \leq Z^2$

### C.2. Tipo 2

Ecuación tipo 2

$$X_1 S\theta_i + Y_1 C\theta_i = Z_1$$

$$X_2 S\theta_i + Y_2 C\theta_i = Z_2$$

Bajo la condición de  $X_1 Y_2 - X_2 Y_1 \neq 0$ , se tiene que

$$S\theta_i = \frac{Z_1 Y_2 - Z_2 Y_1}{X_1 Y_2 - X_2 Y_1} \quad (\text{C.6})$$

$$C\theta_i = \frac{Z_2 X_1 - Z_1 X_2}{X_1 Y_2 - X_2 Y_1} \quad (\text{C.7})$$

$$\theta_i = \arctan\left(\frac{S\theta_i}{C\theta_i}\right) \quad (\text{C.8})$$

### C.3. Tipo 3

Ecuación tipo 3

$$WS\theta_j = XC\theta_i + YS\theta_i + Z_1$$

$$WC\theta_j = XS\theta_i - YC\theta_i + Z_2$$

## APÉNDICE C. SOLUCIONES DE ECUACIÓN

Elevando al cuadrado las ecuaciones anteriores y sumandolas tenemos que

$$B_1 S\theta_i + B_2 C\theta_i = B_3$$

Donde

$$B_1 = 2(Z_1 Y + Z_2 X)$$

$$B_2 = 2(Z_1 X - Z_2 Y)$$

$$B_3 = W^2 - X^2 - Y^2 - Z_1^2 - Z_2^2$$

Una vez conociendo  $\theta_i$ , la cual se resuelve como una ecuación de tipo 1, se puede obtener fácilmente  $\theta_j$  resolviéndola como una ecuación de tipo 2.

### C.4. Tipo 4

Ecuación tipo 4

$$W_1 C\theta_j + W_2 S\theta_j = X C\theta_i + Y S\theta_i + Z_1$$

$$W_1 S\theta_j - W_2 C\theta_j = X S\theta_i - Y C\theta_i + Z_2$$

Estas ecuaciones son la forma generalizada de las ecuaciones presentadas de tipo 3, así que la forma de solucionar las ecuaciones tipo 4 es la presentada anteriormente

con un cambio

$$B3 = W_1^2 + W_2^2 - X^2 - Y^2 - Z_1^2 - Z_2^2$$

## C.5. Tipo 5

Ecuación de tipo 5

$$XC\theta_i + YC(\theta_i + \theta_j) = Z_1$$

$$XS\theta_i + YS(\theta_i + \theta_j) = Z_2$$

Al elevar las ecuaciones al cuadrado y sumarlas entre ellas, se obtiene

$$C\theta_j = \frac{Z_1^2 + Z_2^2 - X^2 - Y^2}{2XY} \quad (\text{C.9})$$

Por lo que

$$\theta_j = \arctan \left( \frac{\pm \sqrt{1 - (C\theta_j)^2}}{C\theta_j} \right) \quad (\text{C.10})$$

Resolviendo  $\theta_j$ , las ecuaciones de tipo 5 quedan como sigue

$$S\theta_i = \frac{B_1 Z_2 - B_2 Z_1}{B_1^2 + B_2^2} \quad (\text{C.11})$$

$$C\theta_i = \frac{B_1 Z_1 + B_2 Z_2}{B_1^2 + B_2^2} \quad (\text{C.12})$$



## APÉNDICE C. SOLUCIONES DE ECUACIÓN

Donde  $B_1 = X + YC\theta_j$  y  $B_2 = YS\theta_j$ , quedando  $\theta_i$

$$\theta_i = \arctan \left( \frac{S\theta_i}{C\theta_i} \right) \quad (\text{C.13})$$

# Bibliografía

- [1] H. U. IC6 B6447 680db, Houghton Library, “De motu animalium, 1685.” [Online]. Available: [https://es.wikipedia.org/wiki/Giovanni\\_Alfonso\\_Borelli#/media/File:Houghton\\_IC6\\_B6447\\_680db\\_-\\_De\\_motu\\_animalium,\\_TAB\\_IV.jpg](https://es.wikipedia.org/wiki/Giovanni_Alfonso_Borelli#/media/File:Houghton_IC6_B6447_680db_-_De_motu_animalium,_TAB_IV.jpg)
- [2] S. of Science and E. of Waseda University, “Wabot (waseda robot).” [Online]. Available: [http://www.humanoid.waseda.ac.jp/booklet/kato\\_2.html](http://www.humanoid.waseda.ac.jp/booklet/kato_2.html)
- [3] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*, paper ed. London, United Kingdom: Hermes Penton Ltd, 2002.
- [4] J. L. M. V. M. D. O. FÁTIMA H. RODRIGUES, EVERSON B. SIQUEIRA, “An alternative methodology for modeling the kinematics of bifurcated robotic system,” 2013.
- [5] D. Edoarado, “File:anatpos.png by radiólogo and own work., cc by-sa 3.0.” [Online]. Available: <https://commons.wikimedia.org/w/index.php?curid=17275160>
- [6] J. A. A. Zapata, “Un método para generar la carga dinámica del paso humano.” [Online]. Available: <http://estructurando.net/2015/01/26/un-metodo-para-generar-la-carga-dinamica-del-paso-humano/>

## BIBLIOGRAFÍA

- [7] A. O. O. P. V. B. A. M. T. J. I. Z. A. Enríquez Cobo, R.S. Núñez Cruz, “Desarrollo de un robot humanoide. parte 2: Visión, modelado y control, resultados experimentales,” *Memorias del XVII Congreso Mexicano de Robótica 2015*, 2015.
- [8] R. C. Limón, “Estrategias de modelado cinemático y simulación en robots humanos,” Master’s thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2009.
- [9] l. e. l. Wikipedia, “Andadura(humanos).” [Online]. Available: [https://es.wikipedia.org/wiki/Andadura\\_\(humanos\)](https://es.wikipedia.org/wiki/Andadura_(humanos))
- [10] T. a. i. p. R. W. D. y. S. J. A. Aristóteles, *De motu animalium*. Oxford, Clarendon Press, 1912.
- [11] R. K. y Garry Tee, *Understanding Human Motion: A Historic Review, Computational Imaging and Vision*. Springer Link, 2008.
- [12] B. G. A., *De Motu Animalium*. (On the Movement of Animals, translated from Latin to English by P. Maquet, Springer, Berlin, 1989), 1680.
- [13] K. H. K. Y. Shuuji Kajita, Hirohisa Hirukawa, *Introduction to Humanoid Robotics*, springer tracts in advanced robotics 101 ed. Ohmsha, Ltd 2005, 2005.
- [14] A. A. O. Olvera, “Control dinámico de la marcha de un robot humanoide,” Master’s thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2012.
- [15] J. J. Craig, *Robótica*, 3rd ed. México: Prentice Hall, Inc., 2006.

## BIBLIOGRAFÍA

- [16] J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanism based on matrices,” *Journal of Applied Mechanism*, vol. 22, pp. 215–221, 1955.
- [17] W. Khalil and J. Kleinfinger, “A new geometric notation for open and closed-loop robots,” 1986.
- [18] P. R.C.P., “Robot manipulators: mathematics, programming and control,” *The MIT Press*, 1981.
- [19] l. e. l. Wikipedia, “Jacobiano.” [Online]. Available: <https://es.wikipedia.org/wiki/Jacobiano>
- [20] W. D.E., “Resolved motion rate control of manipulators and human prostheses,” *IEEE Trans, on Man Machine Systems*, pp. 47–53, 1969.
- [21] G. B. and R. M., “Modèles des robots-manipulateurs ; application à leur commande,” *Cepadues Edition*, 1984.
- [22] M. W. S. S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*. John Wiley and Sons, 2006.
- [23] B. P., “Contribution à la modélisation géométrique des robots-manipulateurs; application à la conception assistée par ordinateur,” *Thèse d’Etat, USTL*, 1986.
- [24] W. P., “A classification of manipulator geometries based on singularity avoidance ability,” *Proc. Int. Conf on Advanced Robotics*, pp. 649–654, 1993.
- [25] l. e. l. Wikipedia, “Menor (álgebra lineal).” [Online]. Available: [https://es.wikipedia.org/wiki/Menor\\_\(álgebra\\_lineal\)](https://es.wikipedia.org/wiki/Menor_(álgebra_lineal))

## BIBLIOGRAFÍA

[26] —, “Hardware.” [Online]. Available: <https://es.wikipedia.org/wiki/Hardware>

[27] —, “Servomotor.” [Online]. Available: <https://es.wikipedia.org/wiki/Servomotor>

[28] L. Jun Hee, “General specification of hsr-8498hb digital robot servo.” [Online]. Available: <http://www.robotstorehk.com/motors/doc/hsr-8498hb.pdf>

[29] R. C. Limón, *Actuadores del robot humanoide Robonova-I*, Departamento de Control Automático del CINVESTAV-IPN, 2009.

[30] R. Ibbotson, “Hitec hsr-8498hb digital servo operation and interface.” [Online]. Available: <https://robosavvy.com/Builders/i-Bot/HSR8498HB%20Servo.pdf>

[31] Latchup, “Hitec robot servos.” [Online]. Available: [http://www.staff.uni-bayreuth.de/~bt150361/cmt2007/geraete/hitec\\_hsr/HSR\\_Serial\\_communication.pdf](http://www.staff.uni-bayreuth.de/~bt150361/cmt2007/geraete/hitec_hsr/HSR_Serial_communication.pdf)

[32] Hitec, “Hitec robot servos: Hsr-5498sg,8498hb,5980sg,5990tg,” 2007.

[33] J. P. Q. Edgar Simó Serra, “Slave architecture for the robonova mr-c3024 using the hmi protocol. iri technical report.” [Online]. Available: <http://www.iri.upc.edu/files/scidoc/1186-Slave-architecture-for-the-Robonova-MR-C3024-using-the-HMI-protocol-IRI-Technical.pdf>

[34] R. P. Foundation, “Raspberry pi 2 model b.” [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

[35] R. Pi, “Noobs (new out of box software).” [Online]. Available: <https://github.com/raspberrypi/noobs/blob/master/README.md>

## BIBLIOGRAFÍA

- [36] I. Std, “IEEE software engineering standard: Glossary of software engineering terminology,” *IEEE Computer Society Press*, 1993.
- [37] J. D. Shannon, Robert; Johannes, “Systems simulation: the art and science,” *IEEE Transactions on Systems, Man and Cybernetics*, pp. 723–724, 1976.
- [38] D. S. S. Corporation, “¿por que elegir solidworks?” [Online]. Available: <http://www.solidworks.es/sw/3d-software-solid-modeling.htm>
- [39] I. The MathWorks, “Simscape multibody.” [Online]. Available: <https://www.mathworks.com/products/simmechanics/>
- [40] l. e. l. Wikipedia, “Python.” [Online]. Available: <https://es.wikipedia.org/wiki/Python>
- [41] L. C. J. W. DeLany, *APLICACIÓN CLÍNICA DE LAS TÉCNICAS NEUROMUSCULARES. Extremidades inferiores*. Paidotribo, 2007.
- [42] D. Wollherr, “Design and control aspects of humanoid walking robots,” Ph.D. dissertation, Technische Universität München, 2005.
- [43] O. L. V. G. J. L. G. E. E. Trelles, “Planificación de caminata para un robot bípedo,” *Technical report*, 2006.