

**Universidad Nacional de Cuyo**

**Facultad de Ingeniería**

**Ingeniería en Mecatrónica**

# **Teleoperación de robot hexápodo**

**Proyecto Final de Estudios**

**Año 2020**

**Director:**

**Ing. Eric Sánchez**

**Alumnos:**

**Gonzalo Romero Román**

**José Andrés Elustondo**

## **Resumen**

Los robots móviles son cada vez más utilizados en entornos domésticos e industriales. Los hexápodos son robots móviles con 6 patas que permiten un movimiento omnidireccional y autónomo. Su flexibilidad a distintos entornos hace que sean muy estudiados. Sin embargo, un mejor entendimiento del funcionamiento de estos robots puede producir una mejora de desempeño en diversas aplicaciones donde podrían ser útiles. En el presente proyecto se implementa la teleoperación de un robot hexápedo de 18 articulaciones, detallando los modelos matemáticos utilizados, así como el hardware y el software usados para el desarrollo de la aplicación. Se especifican los componentes del robot, la arquitectura de control aplicada, el código utilizado, las comunicaciones, como también la fabricación y el ensamblado. Se genera, además, un entorno de simulación para la evaluación de los algoritmos previo a su utilización en el robot real. Una vez construido el hexápedo se evalúa el correcto funcionamiento de los desarrollos realizados. Por último se expresan los resultados obtenidos.

# Índice

<b>1. Tabla de variables</b>	<b>5</b>
<b>2. Introducción</b>	<b>6</b>
<b>3. Objetivos</b>	<b>7</b>
<b>4. Antecedentes históricos</b>	<b>8</b>
<b>5. Estructura y fundamentos del robot</b>	<b>9</b>
5.1. Disposición de patas . . . . .	9
5.2. Tipos de pasos . . . . .	13
<b>6. Cinemática</b>	<b>14</b>
6.1. Cinemática directa . . . . .	14
6.2. Cinemática inversa . . . . .	15
6.3. Jacobiano y singularidades . . . . .	22
6.4. Rotación sobre el eje central . . . . .	23
6.5. Traslación . . . . .	24
6.6. Estabilidad . . . . .	26
<b>7. Dinámica</b>	<b>28</b>
7.1. Análisis de la fuerza de contacto . . . . .	34
7.2. Estimación de torques de los actuadores . . . . .	35
<b>8. Selección del hardware</b>	<b>38</b>
8.1. Placa controladora . . . . .	39
8.2. Selección de actuadores . . . . .	39
8.2.1. Servomotores MG996R . . . . .	40
8.2.2. Servomotores TD-8120MG . . . . .	41
8.2.3. Placa controladora PCA9685 . . . . .	41
8.2.4. Baterías 18650 . . . . .	42
8.3. Selección de sensores . . . . .	42
8.3.1. Potenciómetros . . . . .	42
8.3.2. Sensores no implementados . . . . .	43
<b>9. Diseño mecánico</b>	<b>43</b>
9.1. Base . . . . .	44
9.2. Extremidades . . . . .	46
9.2.1. Eslabón 1: Coxa . . . . .	47
9.2.2. Eslabón 2: Fémur . . . . .	48
9.2.3. Eslabón 3: Tibia . . . . .	49
<b>10. Arquitectura de control</b>	<b>50</b>
10.1. Planificador de trayectorias del robot . . . . .	50
10.2. Planificador de trayectorias de las patas . . . . .	51

<b>11.Simulación del robot</b>	<b>53</b>
<b>12.Programación del robot</b>	<b>59</b>
<b>13.Comunicaciones</b>	<b>63</b>
<b>14.Fabricación, ensamblaje y puesta en marcha</b>	<b>65</b>
14.1. Fabricación . . . . .	65
14.2. Ensamblaje . . . . .	66
14.2.1. Verificación de motores . . . . .	66
14.2.2. Vinculación de base, eslabones y motores . . . . .	66
14.2.3. Definición de los offsets . . . . .	67
14.3. Puesta en marcha . . . . .	67
14.3.1. Pruebas en vacío . . . . .	68
14.3.2. Pruebas en el suelo . . . . .	68
14.4. Iteraciones . . . . .	69
14.4.1. Base . . . . .	69
14.4.2. Patas . . . . .	71
<b>15.Resultados</b>	<b>72</b>
15.1. Orientación y posición estáticas . . . . .	72
15.1.1. Rotación: Roll, Pitch & Yaw . . . . .	72
15.1.2. Posición: X, Y & Z . . . . .	74
15.2. Desplazamientos del robot . . . . .	75
15.2.1. Traslación . . . . .	75
15.2.2. Rotación . . . . .	75
<b>16.Recursos implicados</b>	<b>78</b>
<b>17.Conclusión</b>	<b>78</b>
<b>18.Trabajo futuro</b>	<b>80</b>

## 1. Tabla de variables

Lista de variables		
Símb.	Significado	Unid.
$\theta_1$	Variable articular que une la base con la coxa	rad
$\theta_2$	Variable articular que une la coxa con el fémur	rad
$\theta_3$	Variable articular que une el fémur con la tibia	rad
$l_0$	Longitud del centro del hexápedo a la primera articulación	mm
$l_1$	Longitud de la coxa	mm
$l_2$	Longitud del fémur	mm
$l_3$	Longitud de la tibia	mm
$l_{c1}$	Longitud del centro de la primera articulación al centro de masa del eslabón 1	mm
$l_{c2}$	Longitud del centro de la segunda articulación al centro de masa del eslabón 2	mm
$l_{c3}$	Longitud del centro de la tercera articulación al centro de masa del eslabón 3	mm
$m_1$	Masa de la coxa	kg
$m_2$	Masa del fémur	kg
$m_3$	Masa de la tibia	kg
$m_b$	Masa de la base (sin sensores, baterías ni cables)	kg
$m_{bt}$	Masa total de la base (con sensores, baterías y cables)	kg
$m_{sb}$	Masa del Hexápedo sin baterías	kg
$m_t$	Masa Total del Hexápedo	kg
$I_j$	Tensor de inercia del eslabón j	$\text{kg}\cdot\text{m}^2$
$g$	Aceleración de la gravedad	$\text{m}/\text{s}^2$
$g_x$	Aceleración de la gravedad en x de la base	$\text{m}/\text{s}^2$
$g_y$	Aceleración de la gravedad en y de la base	$\text{m}/\text{s}^2$
$g_z$	Aceleración de la gravedad en z de la base	$\text{m}/\text{s}^2$
$\tau_{1i}$	Torque motor de la articulación 1 de la i-ésima pata	N.m
$\tau_{2i}$	Torque motor de la articulación 2 de la i-ésima pata	N.m
$\tau_{3i}$	Torque motor de la articulación 3 de la i-ésima pata	N.m
$M$	Matriz de inercia	$\text{N}\cdot\text{m.s}^2$
$C$	Matriz de Coriolis y efectos centrífugos	$\text{N}\cdot\text{m.s}$
$F$	Matriz de fricción	N.m
$G$	Matriz de gravedad	N.m
$f$	Vector de fuerzas y momentos	N, N.m
$\tau_f$	Torques debidos a las fuerzas en el efector final	N.m
$\tau_{fa}$	Torques debidos a la fricción en los actuadores	N.m
$b_j$	Coeficientes de fricción viscosa de la articulación j de la pata	$\text{N}\cdot\text{m.s}$
$T_{cj}$	Fricción de Coulomb de la articulación j de la pata	N.m
$dx_i$	Desplazamiento en x de la pata i	m
$dy_i$	Desplazamiento en y de la pata i	m
$sm$	Margen de estabilidad	m
$di$	Distancias perpendiculares hasta el lado i del polígono de soporte	m
$Si$	Área i del polígono de soporte	$\text{m}^2$
$Li$	Longitud del lado i del polígono de soporte	m
$Pi$	Coordenada del punto de contacto de la pierna i	m
$PCM$	Coordenada de la proyección del centro de masa sobre el polígono de contacto	m
$\bar{O}$	Vector desde el origen del sistema de referencia fijo al suelo al de la plataforma	m
$d\alpha$	Incremento de ángulo deseado durante la rotación	rad
$dp$	Desplazamiento necesario para obtener la rotación $d\alpha$	m

Lista de variables		
Símb.	Significado	Unid.
$l_x$	Distancia en x del efecto final respecto al sistema de referencia de la pata	m
$\beta$	Factor de tarea	-
$L$	Desplazamiento que debe realizar el robot en un ciclo de desplazamiento	m
$dX$	Proyección de L en x sobre el sistema fijo al suelo	m
$dY$	Proyección de L en y sobre el sistema fijo al suelo	m
$F_x$	Fuerza en x el efecto final con respecto a los ejes del efecto final	N
$F_y$	Fuerza en y el efecto final con respecto a los ejes del efecto final	N
$F_z$	Fuerza en z el efecto final con respecto a los ejes del efecto final	N
$W$	Peso del robot	N
$A$	Matriz de posiciones de los puntos de contacto del robot	-
$\mu$	Coeficiente de fricción estático	-
$u$	Ley de control	N.m
$K_v$	Matriz de ganancias derivativas	-
$K_p$	Matriz de ganancias proporcionales	-
$K_i$	Matriz de ganancias integrales	-
$e$	Error entre variables de consigna y medidas de posición angular	rad

Cuadro 1: Tabla de variables

## 2. Introducción

El puntapié inicial de este proyecto fue la necesidad de generar un robot que se adapte a terrenos de alta complejidad, que al poseer una morfología determinada le brinde la característica única de poder portar y trasladar una carga sobre terrenos irregulares.

A lo largo de la carrera se estudiaron numerosos conceptos relacionados con la Mecánica, la Electrónica, el Control, la Informática y otros campos anexos. La elaboración de este proyecto de gran complejidad, constata la culminación de una etapa y nos abre las puertas a desafíos futuros.

Analizando diferentes tipos de robots dotados de patas, los hexápodos han sido aquellos que cumplen con las características mencionadas anteriormente, siendo ampliamente utilizados en la exploración de lugares remotos y ambientes hostiles como pueden ser plantas nucleares, fondo oceánico, donde se realizan operaciones de exploración, rescate y búsqueda. También este tipo de robot puede ser utilizado en la industria forestal, transporte de carga y a su vez como robots de servicio y entretenimiento.

Los robots móviles con patas permiten la locomoción en terrenos inaccesibles para otros tipos de robots, dado que no necesitan una superficie de contacto continua, pudiendo sobrepasar obstáculos comparables con el tamaño de sus piernas [1]. Sumado a esto poseen una mayor movilidad y adaptabilidad en terrenos irregulares no estructurados adoptando diferentes tipos de pasos y velocidades, así como también mantienen la estabilidad del sistema, siendo ésta esencial para sustentar el paso [2]. Sin embargo, la coordinación de las patas y el control es más complejo que aquellos basados en ruedas debido al número de variables que deben ser monitoreadas. Otras desventajas tienen que ver con el

costo de los componentes, la menor eficiencia energética, velocidad y el mayor costo computacional [3]. Si bien hay muchos tipos de robots móviles con patas, los hexápodos son los más utilizados debido a su robustez y a su equilibrio estático. Los robots de seis patas brindan una mayor fiabilidad en términos de autonomía, dado que pueden continuar con el paso cuando una de sus patas falla [4]. Su alta capacidad de carga y su habilidad para desplazarse en todas las direcciones, son las características que lo destacan frente a otros robots dotados de patas. Pueden ser utilizados en aplicaciones donde se requiera elevada movilidad, manipulación de un amplio espacio de trabajo, desplazamiento en terrenos irregulares y trabajo en ambientes con restricciones [5].

Los hexápodos han sido ampliamente estudiados. C. S. Surel [3] presentó una arquitectura de control jerárquica modelada en MapleSim<sup>1</sup>. Tedeschi et al. presentó la problemática de diseño en robots móviles de tipo hexápodo [1]. Mănoiu et al. [6] desarrolló una simulación dinámica y el control experimental usando Matlab<sup>2</sup> en hexápodos de base rectangular, al igual que en Thilderkvist et al. [7]. Mahfou et al. realizó el modelado dinámico y el control de operaciones de un hexápodo [8]. Sandoval-Castro et al. [9] hizo un análisis de la cinemática inversa del robot. Mientras que en Ghayour et al. [4] realizó el estudio de su cinemática directa. Wang et al. [2] presentó el análisis de la locomoción típica de un hexápodo simétrico. Raheem et al. [10] realizó un estudio de la estabilidad en pasos periódicos. Mientras que J. Mrva [11] diseñó el movimiento del robot en terrenos irregulares. Un análisis más detallado del comportamiento del robot es realizado por M. M. A. Hajiabadi [5]. En lo que respecta a la Universidad Nacional de Cuyo, el diseño y la construcción de un robot hexápodo fue realizado como proyecto final de estudios por J. V. Sanjurjo y F. S. Sesto [12].

En el presente informe se analizan los antecedentes del robot. Posteriormente se describe su estructura, así como su principio de funcionamiento. Luego se desarrolla un estudio de los modelos matemáticos que describen su comportamiento, se describe tanto la cinemática como dinámica del hexápodo. Se selecciona el hardware necesario para su construcción y se realiza el diseño del robot mediante un software de modelado 3D. Para verificar el correcto funcionamiento de éste, se procede a realizar la simulación utilizando Unity3D<sup>3</sup>. Una vez corroborado el correcto comportamiento, adquiridos los actuadores y demás piezas y habiendo fabricado la estructura del robot mediante impresión 3D, se procede con el ensamblado y la programación. Por último, se analizan los resultados obtenidos y se exponen las conclusiones alcanzadas.

### 3. Objetivos

- Investigar sobre los principios de funcionamiento, estructura y antecedentes de robots hexápodos de 18 articulaciones.

---

<sup>1</sup><https://www.maplesoft.com/products/maplesim/>

<sup>2</sup><https://www.mathworks.com/products/matlab.html>

<sup>3</sup><https://unity.com/>

- Desarrollar y validar modelos matemáticos que describan la cinemática y dinámica del robot con el fin de poder realizar los movimientos deseados.
- Realizar un diseño mecánico del robot mediante un software de modelado 3D que permita a éste desplazarse de manera efectiva y que valide el comportamiento simulado.
- Investigar y seleccionar los actuadores y sensores necesarios para realizar los movimientos deseados y para obtener información sobre el entorno del robot.
- Realizar una simulación con el fin de verificar los algoritmos de control y visualizar el comportamiento del robot en un entorno virtual.
- Investigar y comparar protocolos de comunicación para poder realizar la teleoperación del hexápedo. Seleccionar y aplicar aquél que se considere más apropiado para la operación del robot.
- Generar una interfaz que permita al operador comandar al robot y también otorgue la posibilidad de visualizar el comportamiento esperado de éste.
- Construir y evaluar el desempeño del robot utilizando el hardware seleccionado y los algoritmos programados.

#### **4. Antecedentes históricos**

La utilización de los robots hexápodos surge en la década de los 70. En 1972 se construyó uno de los primeros hexápodos en la Universidad de Roma [13] (Peternelly & Salinari), este permitía movimientos lentos y no permitía variar dichos movimientos con el terreno. En 1976, éste fue superado por el hexápedo Masha (Gurfinkel et al) que fue construido por la Universidad Estatal de Moscú [13], el robot tenía seis patas con tres grados de libertad cada una y permitía sobrepasar obstáculos usando sensores de proximidad y sensores de contacto en cada una de las patas. Otros ejemplos de hexápodos que valen la pena mencionar son “OSU Hexapod” (Universidad de Ohio) en 1977, “Six-Legged Hydraulic Walker” (Universidad de Carnegie-Mellon) en 1983, Odex 1 (Odetic Inc) en 1984, “NMIIIA” en 1985, entre otros.

En los últimos años se han producido avances importantes en informática y sistemas de control que han permitido que surjan nuevas configuraciones de robots con patas. En el año 2001 surgió RHex que consistía en un robot de seis patas con un grado de libertad cada una. En 2004 fue desarrollado ATHLETE que permitía desplazarse mediante ruedas en caso de que el terreno fuera liso y permitía que el robot caminase en caso de que el terreno fuese irregular. Otros ejemplos de hexápodos que se han realizado en los últimos años son AQUA, RiSE, COMET, Mantis, entre muchos otros. Cada uno de estos robots fue diseñado teniendo en mente aplicaciones distintas, por lo que sus diseños

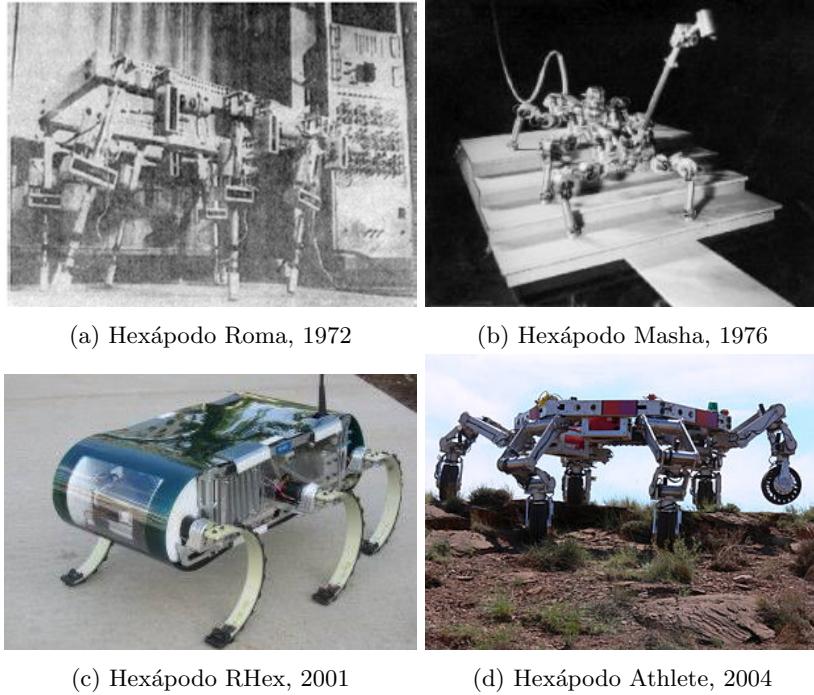


Figura 1: Algunos robots hexápodos.

varían notablemente. Algunos de los robots mencionados pueden observarse en la Figura 1.

Los robots hexápodos han sido ampliamente utilizados en la exploración lugares remotos y ambientes hostiles como puede ser plantas nucleares, el fondo del océano, otros planetas y en operaciones de rescate y búsqueda. Otras aplicaciones interesantes son inspección de puentes, inspección de tuberías o tanques, siendo un robot apto en áreas en que los drones u otros robots con patas no son ideales.

Diversas mejoras pueden realizarse en todos lo ámbitos para hacer que los movimientos del hexápodo sean más fluidos y precisos. Un mayor conocimiento de estos tipo de robots podría traer mejoras de desempeño en las aplicaciones mencionadas y en muchas otras donde este robot podría ser utilizado.

## 5. Estructura y fundamentos del robot

### 5.1. Disposición de patas

El robot posee seis patas que le permiten desplazarse en el terreno. Cada una de las patas está formada por tres eslabones unidos entre sí por los actuadores, cada pierna puede considerarse como un robot serie de 3 grados de libertad. Se

requiere por lo tanto tres actuadores por pata y 18 actuadores en total. Cada una de las piernas del hexápedo está conectado a la plataforma del robot y permiten el desplazamiento de ésta.

Se utiliza una plataforma circular en donde las piernas del robot se encuentran distribuidas a  $60^\circ$  sobre ésta. Tal como menciona en [5] el robot es axialmente simétrico. La acción de las patas permite que el robot se traslade y cambie su orientación, a su vez permite el movimiento en 6 GDL de la plataforma. La disposición de las patas puede observarse en la Figura 2.

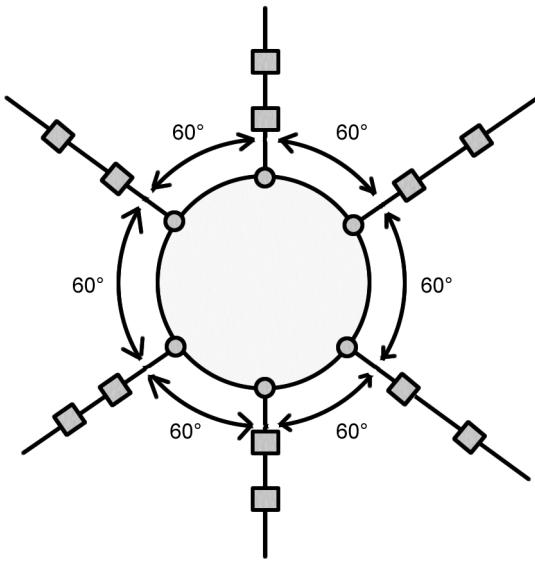


Figura 2: Disposición de las patas en el hexápedo.

Cada pierna está compuesta por tres eslabones: la coxa, el fémur y la tibia, tal como sugiere A. Hajiabadi [5]. Esto puede observarse en la Figura 3. Cada pata posee tres actuadores, los cuales se encuentran en las articulaciones 1, 2 y 3.

Para brindar una noción del alcance angular de cada una de las articulaciones en las patas, se establecieron los límites que se explican a continuación.

El primer actuador, como se observa en la Figura 4, se encuentra asociando la base del robot con la coxa de la pata. Se fijó un límite prudente de  $\pm 25^\circ$  con el objetivo de evitar las colisiones entre piernas aledañas del robot durante la rotación y traslación.

Respecto a  $\theta_2$  tomó valores entre  $\pm 90^\circ$  como puede observarse en la Figura 5. Estos valores de diseño fueron tomados a partir de las limitaciones de los actuadores que se encontraban en el mercado.

En el caso de  $\theta_3$  se optó por fijar un límite de  $160^\circ$  para la tercera articulación.

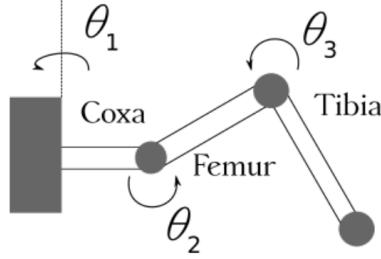


Figura 3: Diagrama esquemático de una pata del hexápedo.

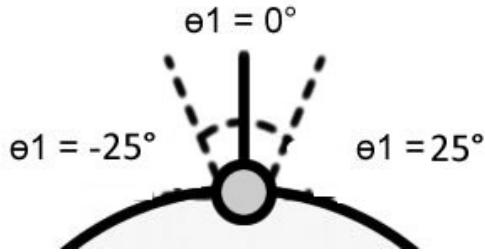


Figura 4: Rango de movimiento de la articulación 1.

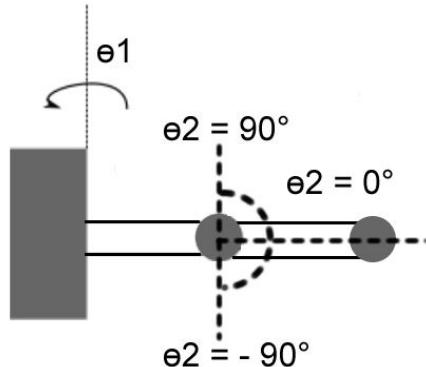


Figura 5: Rango de movimiento de la articulación 2.

Esto puede observarse en la Figura 6. Con esto se busca evitar colisiones entre los distintos eslabones del robot.

Un diagrama esquemático del hexápedo puede observarse en la Figura 7. En dicha imagen se detallan los centros de masa, momentos de inercia y longitudes a considerar. Siendo  $l_1, l_2, l_3$  las longitudes de los eslabones,  $l_{c1}, l_{c2}, l_{c3}$  las distancias hasta los centros de masa y  $m_1, m_2, m_3$  las masas de cada uno de

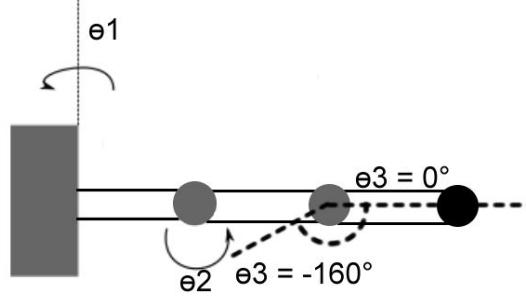


Figura 6: Rango de movimiento de la articulación 3.

ellos. Mientras que  $m_{bt}$  representa la masa total de la base del robot. Utilizando estos parámetros se obtienen, en secciones posteriores, las ecuaciones que rigen la cinemática y dinámica del hexápedo.

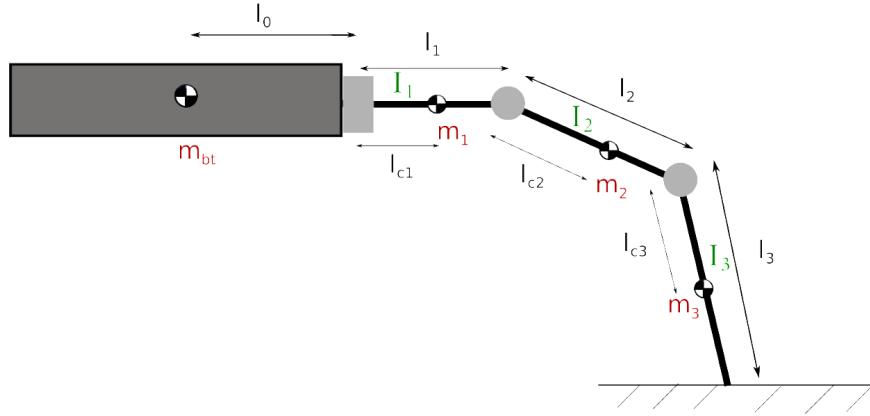


Figura 7: Diagrama esquemático de la base más una pierna del hexápedo.

Los valores numéricos de las variables pueden observarse en el Cuadro 2. Dichos valores serán justificados en la sección dedicada al diseño mecánico del robot.

Lista de variables	
Símbolo	Magnitud
$l_0$	101,15 mm
$l_1$	28,45 mm
$l_2$	76,2 mm
$l_3$	130,74 mm
$lc_1$	14,21 mm

Lista de variables	
Símbolo	Magnitud
$lc_2$	38,1 mm
$lc_3$	15,57 mm
$m_1$	0,133 kg
$m_2$	0,013 kg
$m_3$	0,083 kg
$m_b$	0,388 kg
$m_{bt}$	0,838 kg
$m_{sb}$	2,026 kg
$m_t$	2,239 kg
$I_j$	$[0,0,0; 0,0,0; 0,0,0]$ kg m <sup>2</sup>

Cuadro 2: Tabla de masas y longitudes

## 5.2. Tipos de pasos

Para poder producir el movimiento del robot móvil, es decir que este se traslade o rote sobre el terreno, cada pata debe seguir un ciclo de movimiento. Se considera que el robot móvil no pierde adherencia con el terreno por donde circula, es decir, no se producen resbalamientos de las patas. El ciclo de movimiento de cada pata está formado por dos partes: una fase de impulsión donde la pierna pasa de una posición inicial a una posición final por el aire y una fase de apoyo donde ésta pasa de una posición final a una posición inicial en contacto con el suelo [3][7]. La combinación de estos movimientos en cada una de las patas permite el desplazamiento del centro de masa del hexápedo. El comportamiento puede observarse en la Figura 8, siendo la línea azul el movimiento durante la fase de impulsión, mientras que la línea roja es el movimiento durante la fase de apoyo, cabe destacar que ambos movimientos son relativos a la base del robot.

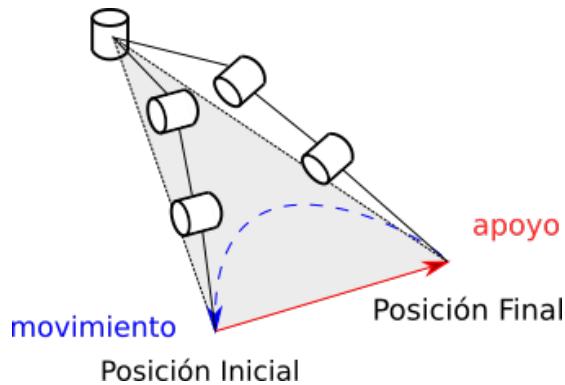


Figura 8: Fase de movimiento y fase de apoyo. Basado en [3].

Los tres tipos de movimientos típicos que realiza el hexápedo son el *wave*

*gait*, el *ripple gait* y el *tripod gait* [3] [11]. El *wave gait* consiste en que una sola pata se encuentre en la fase de impulsión mientras que las otras se encuentran en la fase de apoyo. Si bien es el más lento en cuanto a locomoción, es el más estable y suele ser utilizado en casos de terrenos muy irregulares. El *ripple gait* es más rápido que el *wave gait* ya que en este tipo de paso dos patas en posiciones opuestas se encuentran en la fase de movimiento desfasados  $180^\circ$  en cada instante de tiempo. En cambio, en el *tripod gait* tres patas se encuentran en la fase de movimiento y las otras tres en fase de apoyo, siendo éste el paso más rápido pero el más inestable dado que en cada momento se encontrarán solo tres patas tocando el suelo.

El orden de las patas y las fases de apoyo e impulsión influyen en gran medida en la velocidad y estabilidad del sistema, por lo que deben ser utilizados en función de las condiciones del terreno para asegurar en todo momento la estabilidad [11]. En el presente informe se implementa únicamente el *tripod gait*, se realiza el análisis de estabilidad solo de este tipo de paso. Sin embargo, la implementación del *ripple gait* y del *wave gait* puede ser realizada sin mayores dificultades teniendo como base los desarrollos realizados.

## 6. Cinemática

Existen dos formas de estudiar la cinemática del robot, la cinemática de la plataforma y la cinemática de cada pata individual. Esto es válido tanto para la cinemática inversa como directa. En esta sección se desarrollarán las relaciones matemáticas que existen entre las variables articulares y las posiciones y orientaciones del efector final.

Es importante remarcar que cuando se hable de efector final se referirá al extremo de la pata, si se está hablando de la cinemática de una pierna del robot y se referirá a la base del robot cuando se hable de cinemática de la plataforma.

Primero se hablará de la cinemática directa del robot, una vez desarrollada, se procederá a explicar la cinemática inversa.

### 6.1. Cinemática directa

La cinemática directa se utiliza para obtener el conjunto de ecuaciones que determinan la posición y orientación del efector final del robot en función de las variables articulares.

Primero se hace el análisis de la cinemática de una pierna del robot. Cada pata del robot puede considerarse como un robot serial de 3 grados de libertad. A partir de los sistemas de referencia mostrados en la Figura 9 se obtienen los parámetros de Denavit-Hartenberg. En el diagrama pueden verse los ángulos  $\theta_1$ ,  $\theta_2$  y  $\theta_3$ . Es necesario remarcar que en el valor de  $\theta_3$  que se muestra en la figura corresponde a un valor negativo,  $\theta_3$  será positivo cuando el ángulo sea tal de que el efector final se encuentre por encima del sistema de referencia 2 en la imagen. Los parámetros de Denavit-Hartenberg se muestran en la Figura 10.

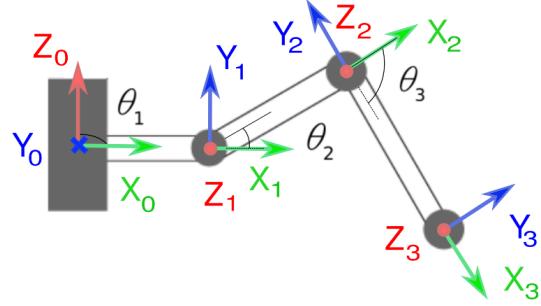


Figura 9: Sistemas de referencia utilizados para obtener los parámetros de Denavit-Hartenberg.

	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	0	$l_1$	$\pi/2$
2	$\theta_2$	0	$l_2$	0
3	$\theta_3$	0	$l_3$	0

Figura 10: Parámetros de Denavit-Hartenberg de una pata del hexápodo.

Estos parámetros fueron calculados para poder visualizar el espacio de trabajo de la pierna del robot. Para ello se utilizó el toolbox de robótica de Peter Corke<sup>4</sup>. Se hizo variar cada una de las variables articulares en valores discretos obteniendo las Figura 11 y Figura 12. Como vimos previamente se limitó  $\theta_1$  entre  $\pm 25^\circ$  y  $\theta_2$  entre  $\pm 90^\circ$ .

La cinemática directa de la plataforma consiste en determinar la posición y orientación de la base del robot en función de los valores de las variables articulares. Esta relación es útil para el control de posición y velocidad del efecto final [14], en este caso la plataforma del robot. Sin embargo, no existe una solución analítica general para la cinemática directa de hexápodos [5]. Si bien se puede calcular por métodos numéricos, como se hace en [4], esto excede los alcances del proyecto y no se ha realizado el análisis.

## 6.2. Cinemática inversa

La cinemática inversa se utiliza para obtener el valor de las variables articulares en función de la posición y orientación del efecto final. Como ya se mencionó anteriormente, dos tipos de análisis pueden realizarse. Uno consiste en determinar la cinemática inversa de cada pata individual y otra consiste en obtener la cinemática inversa del robot paralelo, es decir, de la plataforma del hexápodo [5]. Esta última permite generalizar el movimiento del robot para que pueda realizar trayectorias de mayor complejidad.

Se realiza primero el análisis de una pata individual del robot, siendo este

<sup>4</sup><https://petercorke.com/toolboxes/robotics-toolbox/>

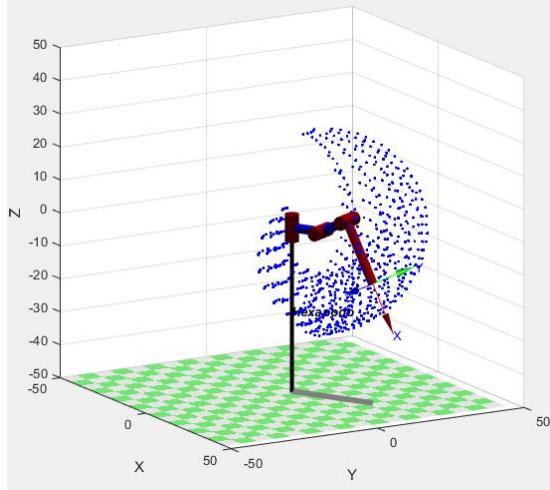


Figura 11: Espacio de trabajo de una pata del robot.

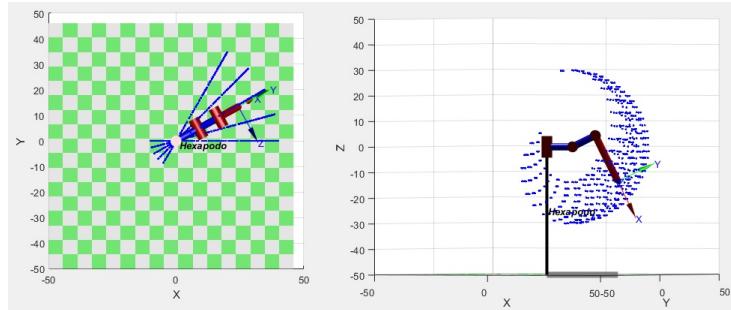


Figura 12: Espacio de trabajo de una pata del robot, plano XY e Z-XY.

el análisis de la cinemática inversa de un robot serie de 3GDL. Esta puede ser obtenida por métodos geométricos.

A partir de la Figura 13 podemos obtener las siguientes ecuaciones de las coordenadas cartesianas del extremo de la pata referidas a la segunda articulación.

$$x_0 = x - l_1 \cos \theta_1$$

$$y_0 = y - l_1 \sin \theta_1$$

Siendo:

$$\rho_0 = \sqrt{x_0^2 + y_0^2}$$

La primera variable articular puede obtenerse de la siguiente forma:

$$\theta_1 = \arctan\left(\frac{y_0}{x_0}\right)$$

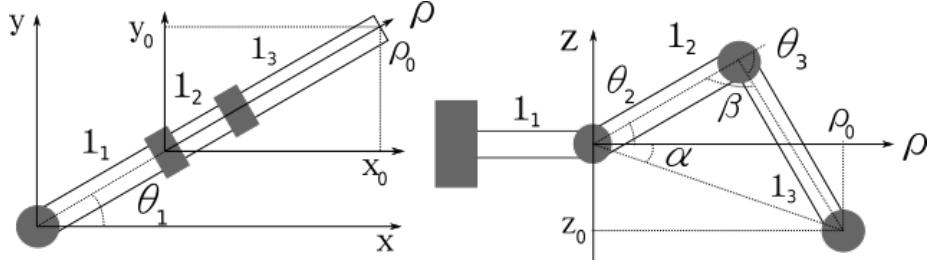


Figura 13: Cálculo de la cinemática inversa de la pata mediante el método geométrico.

Usando la ley del coseno pueden obtenerse las demás variables articulares:

$$\begin{aligned}\alpha &= \arctan\left(\frac{z_0}{\sqrt{x_0^2 + y_0^2}}\right) \\ \beta &= \pm \arccos\left(\frac{x_0^2 + y_0^2 + z_0^2 - l_2^2 - l_3^2}{-2l_2l_3}\right) \\ \theta_2 + \alpha &= \pm \arccos\left(\frac{-l_3^2 + l_2^2 + x_0^2 + y_0^2 + z_0^2}{2l_2\sqrt{x_0^2 + y_0^2 + z_0^2}}\right) \\ \theta_2 &= \pm \arccos\left(\frac{-l_3^2 + l_2^2 + x_0^2 + y_0^2 + z_0^2}{2l_2\sqrt{x_0^2 + y_0^2 + z_0^2}}\right) - \arctan\left(\frac{z_0}{\sqrt{x_0^2 + y_0^2}}\right)\end{aligned}$$

La tercera variable articular está dada por:

$$\begin{aligned}\theta_3 &= -(\pi - \beta) \\ \theta_3 &= \pm \arccos\left(\frac{x_0^2 + y_0^2 + z_0^2 - l_2^2 - l_3^2}{2l_2l_3}\right)\end{aligned}$$

Teniendo de esta forma dos soluciones: codo arriba y codo abajo. Dado que se pretende que en todo momento sea la misma cara de la base la que enfrente el suelo se considera como única solución el codo arriba. En caso de que el robot tuviese que funcionar con ambas caras enfrentando el suelo, ambas soluciones deberían ser consideradas. Por lo tanto las ecuaciones quedarían de la siguiente forma:

$$\begin{aligned}\theta_1 &= \arctan\left(\frac{y_0}{x_0}\right) \\ \theta_2 &= \arccos\left(\frac{-l_3^2 + l_2^2 + x_0^2 + y_0^2 + z_0^2}{2l_2\sqrt{x_0^2 + y_0^2 + z_0^2}}\right) + \arctan\left(\frac{z_0}{\sqrt{x_0^2 + y_0^2}}\right) \\ \theta_3 &= -\arccos\left(\frac{x_0^2 + y_0^2 + z_0^2 - l_2^2 - l_3^2}{2l_2l_3}\right)\end{aligned}$$

Es necesario además limitar el espacio de trabajo del robot restringiendo los valores de las variables articulares impidiendo colisiones con otros elementos.

Para la obtención de la cinemática inversa de la plataforma se define un conjunto de sistemas de coordenadas. Una tabla con las variables utilizadas para el desarrollo puede observarse en el Cuadro 3, mientras que el sistema de coordenadas de cada una de las patas y de la base puede observarse en la Figura 14.

Sistema de coordenadas		
Símbolo	Significado	Unidades
$x$	Eje x referido al origen de la base del hexápodo	m
$y$	Eje y referido al origen de la base del hexápodo	m
$z$	Eje z referido al origen de la base del hexápodo	m
$x_i$	Eje x referido al origen de la i-ésima pata	m
$y_i$	Eje y referido al origen de la i-ésima pata	m
$z_i$	Eje z referido al origen de la i-ésima pata	m
$\rho_i$	Eje Polar referido al origen de la i-ésima pata	m

Cuadro 3

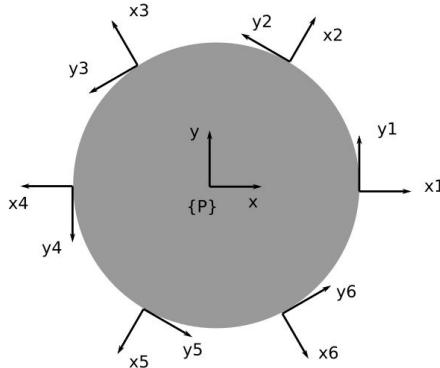


Figura 14: Sistemas coordenados de cada pata y de la base.

En algunas ocasiones es necesario que el robot mantenga una cierta orientación mientras camina o mientras se encuentre parado. Esto puede ocurrir en el caso de que el robot tenga que caminar en un terreno con orientación específica o deba mantener una configuración horizontal. Para este caso debe calcularse la cinemática inversa de la plataforma.

Este desarrollo se encuentra realizado en [5] pero se adaptó para tener en cuenta el sistema de referencia utilizado. Se define un sistema coordenado fijo al suelo definido como  $(X, Y, Z)$  así como un sistema coordenado local fijo al centro de la plataforma  $(x, y, z)$ . Se supone que en un principio ambos sistemas son colineales. Dado que todas las piernas son idénticas, la cinemática inversa es solucionada para una sola pierna y luego es generalizada para las demás.

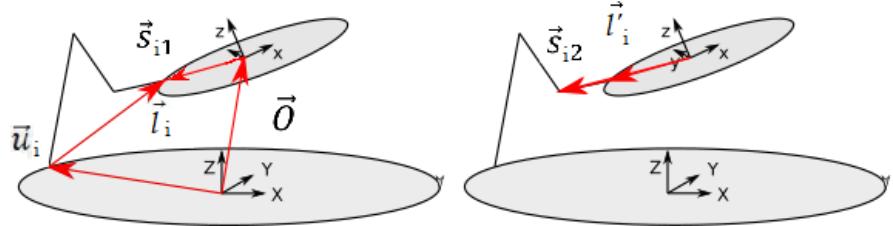


Figura 15: Cinemática inversa de la plataforma. Basado en [5].

A partir de la Figura 15 (izquierda) puede observarse que

$$\bar{l}_i = \bar{O} + R\bar{s}_{i1} - \bar{u}_i$$

donde  $\bar{O}$  es el vector posición desde el origen del sistema de referencia fijo al suelo al sistema de referencia fijo a la plataforma.  $\bar{s}_{i1}$  representa la posición de la articulación de la coxa con respecto al sistema de referencia fijo a la plataforma,  $\bar{u}_i$  es el punto de contacto de la pierna  $i$  con respecto al sistema de referencia fijo al suelo  $O$ ,  $\bar{l}_i$  es el vector que va desde el punto de contacto a la articulación de la cadera y  $R$  es la matriz de rotación de la plataforma con respecto al sistema coordenado fijo al suelo y puede ser calculada a partir de los ángulos de Euler (*roll, pitch* y *yaw*). A partir de  $\bar{l}_i$  puede obtenerse  $\theta_{1i}$ .

$$\theta'_{1i} = \arctan\left(\frac{l_{i,Y}}{l_{i,X}}\right)$$

Donde  $l_{i,X}$  y  $l_{i,Y}$  son las proyecciones del vector  $\bar{l}_i$  sobre los ejes X e Y del sistema coordenado fijo al suelo. A partir de  $\theta'_{1i}$  se puede obtener  $\theta_{1i}$  corrigiendo los ángulos a partir del desfasaje con respecto al centro de coordenadas fijo al centro de la plataforma.

La referencia del valor angular de la primera articulación de cada pata con respecto al origen de coordenadas del robot, se encuentra dado por la siguiente expresión:

$$\theta_{1i} = \theta'_{1i} - \frac{\pi}{3}(i-1)$$

Donde el subíndice  $i$  indica el número de pata.

El ángulo  $\theta_{1i}$  es igual  $\theta'_{1i}$  que puede observarse en la Figura 16.

El vector que va desde el centro de la plataforma a la posición de la segunda articulación  $\bar{s}_{i2}$  puede obtenerse a partir de la longitud del primer eslabón  $l_1$  y el vector que va desde dicho centro a la posición del primer eslabón  $\bar{s}_{i1}$  como se muestra en la Figura 15 (derecha).

$$\bar{s}_{i2} = \begin{pmatrix} s_{i1x} - l_{1i} \cos(\theta_{1i}) \\ s_{i1y} - l_{1i} \sin(\theta_{1i}) \\ s_{i1z} \end{pmatrix}$$

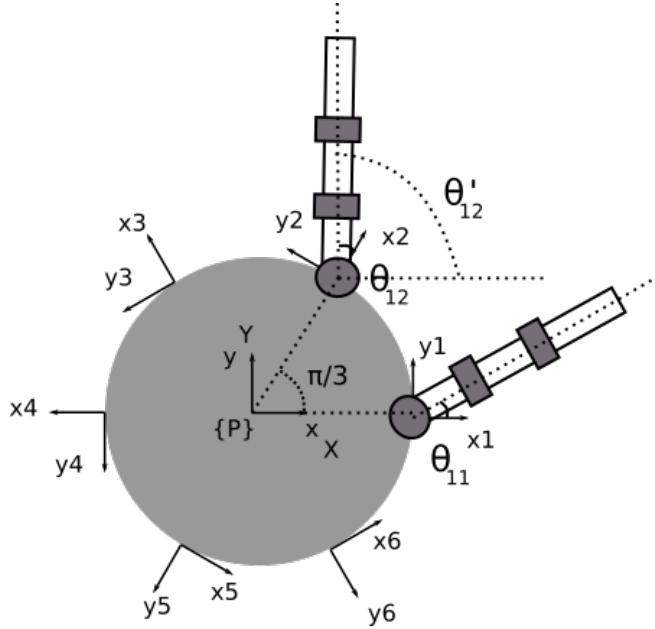


Figura 16: Esquema que muestra los ángulos del robot.

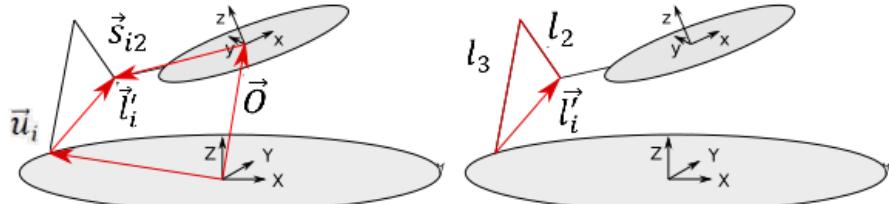


Figura 17: Cinemática inversa de la plataforma. Basado en [5].

El siguiente paso consiste en encontrar el vector que va desde el punto de contacto de la pierna  $i$  con el piso hasta la posición de la segunda articulación  $\bar{l}'_i$ . Esto puede observarse en la Figura 17 (izquierda). Entonces:

$$\bar{l}'_i = \bar{O} + R\bar{s}_{i2} - \bar{u}_i$$

A partir de la Figura 17 (derecha) se puede obtener el valor de las demás variables articulares siendo  $l_{2i}$  y  $l_{3i}$  las longitudes del fémur y la tibia del robot.

$$\cos(\lambda_i) = \frac{l_i'^2 + l_{3i}^2 - l_{2i}^2}{2l_i'l_{3i}}$$

$$\cos(\pi - \theta_{3i}) = \frac{l_{2i}^2 + l_{3i}^2 - l_i'^2}{2l_{2i}l_{3i}}$$

$$\cos(\theta_{2i} + \rho_i + \phi_i) = \frac{l_{2i}^2 + l_i'^2 - l_{3i}^2}{2l_{2i}l_i'}$$

Observando la Figura 18 se pueden obtener los valores de los ángulos que intervienen en el cálculo de la cinemática inversa del robot.

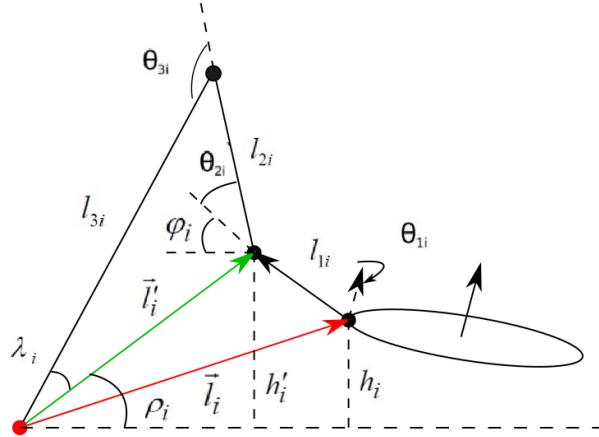


Figura 18: Cinemática inversa de la plataforma, cálculo de las variables articulares. Basado en [5].

Se obtienen las siguientes relaciones:

$$\rho_i = \arctan\left(\frac{h'_i}{\sqrt{l_{i,X}^2 + l_{i,Y}^2}}\right)$$

$$\phi_i = \arcsin\left(\frac{h'_i - h_i}{l_{1i}}\right)$$

Por lo tanto, las ecuaciones que corresponden a la segunda y tercera variable articular están dadas por:

$$\theta_{2i} = \arccos\left(\frac{l_{2i}^2 + l_i'^2 - l_{3i}^2}{2l_{2i}l_i'}\right) - (\rho_i + \phi_i)$$

$$\theta_{3i} = \pi - \arccos\left(\frac{l_{2i}^2 + l_{3i}^2 - l_i'^2}{2l_{2i}l_{3i}}\right)$$

Obteniendo de esta forma la cinemática inversa de la plataforma. Si bien el algoritmo es más complejo y de mayor costo computacional, permite controlar tanto la posición como la orientación de la plataforma permitiendo trayectorias más complejas. Las ecuaciones finales de la cinemática inversa por lo tanto serán

$$\theta_{1i} = \arctan\left(\frac{l_{i,Y}}{l_{i,X}}\right) - \frac{\pi}{3}$$

$$\theta_{2i} = \arccos\left(\frac{l_{2i}^2 + l_i'^2 - l_{3i}^2}{2l_{2i}l_i'}\right) - (\rho_i + \phi_i)$$

$$\theta_{3i} = \pi - \arccos\left(\frac{l_{2i}^2 + l_{3i}^2 - l_i'^2}{2l_{2i}l_{3i}}\right)$$

El procedimiento se describe a continuación. Primero se obtiene la matriz de rotación R basada en la orientación deseada de la plataforma. Luego se calcula los vectores posición en función de la posición en  $x$ ,  $y$  y  $z$  deseada de la plataforma. Luego se calcula en ángulo  $\theta_{1i}$ . Se obtiene los valores  $s_{i2}$  y  $\bar{l}_i$ . Se obtiene el valor de los ángulos auxiliares y por último se obtiene el valor de  $\theta_{2i}$  y  $\theta_{3i}$ .

### 6.3. Jacobiano y singularidades

El análisis de la matriz jacobiana y de las singularidades se realiza únicamente para una pata del robot. Se analiza solo una de ellas debido a que las 6 piernas poseen el mismo diseño.

Es importante conocer la relación entre las velocidades angulares de las coordenadas articulares y las velocidades de los extremos del robot. Esto permite a éste establecer qué velocidades debe imprimir a cada articulación para conseguir que el extremo desarrolle una trayectoria temporal concreta. A partir de la matriz jacobiana directa se puede obtener las velocidades del efecto final (extremo de la pata) a partir de las velocidades articulares.

La pata como robot serie, se considera un robot subactuado, dado que el número de articulaciones es menor a 6. Esto permite considerar que ciertos grados de libertad del efecto final no sean controlables, en el caso particular de la pata son los 3 grados de orientación. Por lo que el jacobiano sería de dimensiones 3x3, referidos a las velocidades  $\dot{x}$ ,  $\dot{y}$  e  $\dot{z}$ . Los fundamentos de esto pueden verse con mayor detalle en [15].

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

$$\begin{aligned} J(1,2) &= l_2 \cos(\theta_2 + \theta s_1)(l_3 + l_2 \cos(\theta_3)) \\ J(1,3) &= -l_3 \sin(\theta_2 + \theta_3) \cos(\theta_1) \\ J(2,1) &= \cos(\theta_1)(l_1 + l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ J(2,2) &= l_2 \cos(\theta_2 + \theta_3) \sin(\theta_1) \sin(\theta_3) - \sin(\theta_2 + \theta_3) \sin(\theta_1)(l_3 + l_2 \cos(\theta_3)) \\ J(2,3) &= -l_3 \sin(\theta_2 + \theta_3) \sin(\theta_1) \\ J(3,1) &= 0 \\ J(3,2) &= \cos(\theta_2 + \theta_3)(l_3 + l_2 \cos(\theta_3)) + l_2 \sin(\theta_2 + \theta_3) \sin(\theta_3) \end{aligned}$$

$$J(3,3) = l_3 \cos(\theta_2 + \theta_3)$$

Si la matriz no es singular entonces se puede invertir. En caso de que la matriz no sea invertible se presenta una singularidad. Cuando un mecanismo serie se encuentra en una configuración singular se pierde uno o más grados de libertad por lo que hay cierta dirección en el espacio cartesiano en la cual el extremo de dicho mecanismo no puede moverse o girar [16]. Esto se produce cuando el determinante de la matriz jacobiana se hace 0. Por lo tanto se plantea la ecuación general de dicho determinante y se iguala a cero para determinar en qué casos se anula.

$$\det(J) = \left| \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} \end{bmatrix} \right|$$

$$\begin{aligned} \det(J) = & -l_2 l_3 (l_1 \sin(\theta_3) - l_3 \sin(\theta_2) + l_2 \cos(\theta_2) \sin(\theta_3) \\ & + l_3 \cos(\theta_3)^2 \sin(\theta_2) + l_3 \cos(\theta_2) \cos(\theta_3) \sin(\theta_3)) \end{aligned}$$

Se obtuvo los valores de las singularidades a partir de las ecuaciones. Es importante remarcar que los puntos singulares no dependen de  $\theta_1$ . Usando Matlab se exploró el espacio de trabajo variando  $\theta_2$  y  $\theta_3$  desde  $-\pi$  hasta  $\pi$  con incrementos de ángulo pequeños y obteniendo el valor de del determinante en cada punto. Se obtuvo que el determinante se hace cero en todos los valores en los cuales  $\theta_3 = n \cdot \pi$  en donde  $n$  es un número entero, es decir, en todos los puntos en donde la segunda y tercera articulación se alinean.

$$\theta_3 = n \cdot \pi \wedge n \in Z \implies \det(J) = 0$$

Dado que la articulación 3 está limitada en un rango de  $\theta_{3máx} = 0^\circ$  a  $\theta_{3min} = -160^\circ$ . Se evitan las singularidades que se dan cuando  $\theta_3 = \pm 180^\circ$

Respecto a las singularidades tomando al hexápodo en su totalidad, algunos análisis han sido realizados [17]. Sin embargo, debido a la complejidad del análisis, esto excede el alcance del proyecto y por lo tanto no será detallado en el presente informe.

#### 6.4. Rotación sobre el eje central

El hexápodo puede rotar en un espacio reducido y es una de las ventajas de los robots móviles con patas. El desplazamiento que debe realizar la pierna para producir un incremento de ángulo  $d\alpha$  puede observarse en la Figura 19.

Sea  $d\alpha$  el incremento de ángulo,  $dp$  el desplazamiento con respecto al sistema de referencia de la pierna para generar dicho incremento, sea  $l_0$  el radio de la plataforma del robot y sea  $l_x$  la distancia en x del efecto final con respecto al sistema de referencia de la pata. Entonces puede obtenerse la siguiente relación

$$dp = (l_0 + l_x) \tan(d\alpha)$$

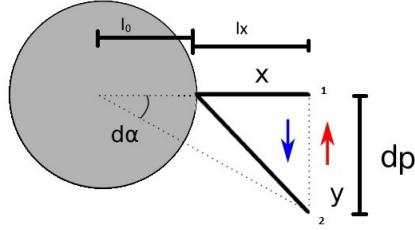


Figura 19: Rotación del hexápodo.

Los incrementos en x e y de cada pierna con respecto a su sistema de referencia pueden obtenerse como:

$$dx_i = 0$$

$$dy_i = dp$$

La rotación se producirá al desplazar tres de las piernas una distancia  $dp$  en el aire, mientras que las tres piernas en contacto con el suelo se desplazan la misma distancia en sentido contrario, luego se repite el ciclo cambiando la fase de cada terna de patas. Esto se explica con mayor detalle en secciones posteriores.

## 6.5. Traslación

En el presente proyecto se utiliza el *tripod gait* en el cual en todo momento son tres patas las que se encuentran en fase de apoyo y tres las que se encuentran en fase de impulsión.

Como ya se mencionó, en un ciclo de desplazamiento cada pierna debe realizar una fase de desplazamiento y una fase de impulsión. Una vez finalizado dicho ciclo cada una de las patas se encuentra en la misma posición y orientación espacial con respecto al sistema de referencia fijo a la plataforma pero se ha desplazado con respecto al sistema de referencia fijo al suelo. Dos tipos de enfoque pueden ser utilizados cuando el robot se desplaza en una trayectoria curva [3]. El primero consiste en combinar el movimiento de traslación en línea recta con el movimiento de rotación. El objetivo es mantener en todo momento el vector que va desde el centro de la plataforma a un punto frontal arbitrario del hexápodo paralelo a la trayectoria. Es decir cuando el robot deba seguir una trayectoria recta, éste se desplazará normalmente, pero al haber una curva, éste deberá cesar su traslación, rotar sobre su eje, cesar su rotación y luego trasladarse. Este tipo de enfoque podría ser útil en el caso de que el robot posea algún tipo de sensor que deba ubicarse en dirección del movimiento.

El enfoque a implementar consiste en que el robot se mueva omnidireccionalmente en función de la trayectoria requerida manteniendo su orientación inicial. Este enfoque permite una mayor velocidad de desplazamiento dado que el robot no debe detenerse cada vez que se presente una curva.

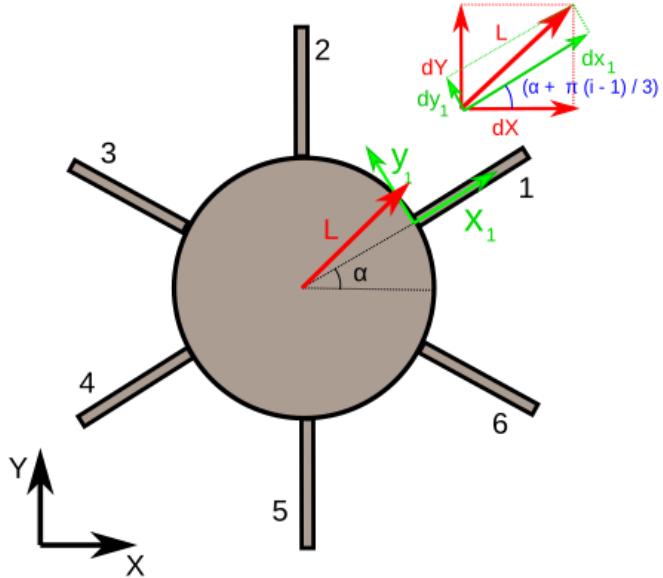


Figura 20: Obtención de las ecuaciones de traslación del hexápodo.

Se supone que se desea desplazar el robot desde una posición inicial  $P_i$  a un posición final  $P_f$ . Para realizar dicho desplazamiento se requiere discretizar la trayectoria en porciones de recta que pueda realizar el robot en un paso. En cada porción es necesario desplazar el centro geométrico del robot una distancia de paso  $L$ , dicha distancia se toma como referencia al sistema de referencia fijo al suelo y su valor está limitado al diseño mecánico del robot, de manera que no se produzcan colisiones entre las patas. Las ecuaciones que describen la traslación del robot pueden obtenerse a partir de la Figura 20.

Como se mencionó anteriormente, se desea desplazar el centro de masa del robot una distancia  $L$  con respecto al sistema de referencia fijo al suelo  $XY$ . Sea  $dX$  y  $dY$  la proyección del desplazamiento  $L$  sobre los ejes del sistema fijo al suelo, sea  $\alpha$  la rotación del hexápodo con respecto al mismo sistema de referencia y sea  $i$  el número de pata que varía entre 1 y 6, entonces el desplazamiento que debe realizar cada pata con respecto a sus sistemas de referencia está dado por las siguientes ecuaciones.

$$dx_i = dX \cos(\alpha + \frac{\pi}{3}(i-1)) + dY \sin(\alpha + \frac{\pi}{3}(i-1))$$

$$dy_i = dY \cos(\alpha + \frac{\pi}{3}(i-1)) - dX \sin(\alpha + \frac{\pi}{3}(i-1))$$

Considerando que dentro de cada ciclo de trabajo, ocurre una fase de apoyo y una fase de impulsión en cada terna de patas. Los desplazamientos que deben

realizarse en dichas fases corresponden a la mitad de las magnitudes calculadas previamente.

## 6.6. Estabilidad

Cuando se trabaja la estabilidad de un robot móvil con patas, se deben tener en cuenta dos aspectos fundamentales, la estabilidad estática y la estabilidad dinámica.

En términos estáticos, si el robot no requiere de una fuerza externa para balancearse, es considerado como estáticamente estable [18]. La proyección vertical del centro de gravedad debe estar en todo momento dentro del patrón de soporte de las patas en contacto con el suelo. Se hace un análisis similar al detallado en [3] [10]. Para el análisis de la estabilidad estática del hexápodo se utiliza el margen de estabilidad que corresponde a la mínima distancia desde la proyección del centro de masa y el polígono de soporte [18] [10]. El robot será más estable mientras mayor sea el margen de estabilidad y será inestable en caso de que el margen de estabilidad sea negativo.

Se considera que el robot se desplaza utilizando un paso tripoidal, por lo que hay tres piernas tocando el suelo. Para el siguiente análisis, se supone que son las patas 1, 3 y 5 las que se encuentran en fase de apoyo, siendo  $P1$ ,  $P3$  y  $P5$  las posiciones de contacto con el suelo. Esto puede observarse en la Figura 21. En dicha imagen se muestran las tres piernas en contacto con el suelo así como el centro de masa del robot.  $S1$ ,  $S2$  y  $S3$  son las áreas que forman los triángulos que unen el centro de masa con los puntos de apoyo.  $L1$ ,  $L2$  y  $L3$  son las longitudes de los lados del triángulo.

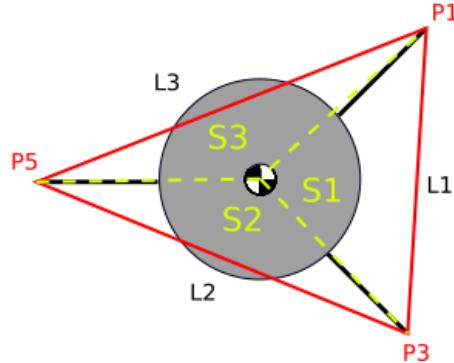


Figura 21: Patrón de soporte del robot y proyección del centro de masa. Basado en [10].

Sea  $P1 = (X1, Y1)$  y  $P3 = (X3, Y3)$  y sea  $PCM = (XCM, YCM)$  la coordenada del centro de masa, entonces el área  $S1$  está dado por,

$$S1 = \frac{1}{2}[(X1 - XCM)(Y3 - YCM) - (X3 - XCM)(Y1 - YCM)]$$

Y  $L1$  estará dado por

$$L1 = \sqrt{(X3 - X1)^2 + (Y3 - Y1)^2}$$

Las distancias perpendiculares hasta cada uno de los lados pueden observarse en la Figura 22 y son calculados para obtener el margen de estabilidad.

$$d1 = 2 \frac{S1}{|L1|}$$

El margen de estabilidad  $sm$  estará dado por

$$sm = \min(d1, d2, d3)$$

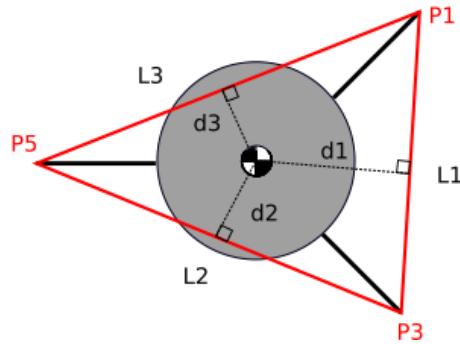


Figura 22: Diagrama para obtener el margen de estabilidad. Basado en [10].

El sistema será estáticamente estable si la suma de las áreas  $S1$ ,  $S2$  y  $S3$  es igual al área del polígono de soporte, esto sucede cuando la proyección del centro de masa se encuentra dentro del polígono de soporte.

En caso de que el centro de masa se encuentre fuera del polígono de soporte, el robot caerá a menos que sea dinámicamente estable. En caso de serlo, el robot estará balanceado durante el movimiento debido a la inercia de éste y será estáticamente inestable cuando cese el movimiento [19]. Es decir que el robot se mantendrá estable debido a las fuerzas que actúan durante el movimiento independientemente de la posición del centro de masa proyectado sobre el polígono de soporte.

Los planeadores de movimientos basados en el criterio de estabilidad estática tienen su velocidad de locomoción limitada, debido a que el centro de gravedad debe encontrarse dentro del área de soporte en todo momento [20]. En caso de que se tuviese en cuenta los efectos de la aceleración en el movimiento del robot, se podría mantener la estabilidad dinámicamente, mejorando las velocidades de locomoción. Una solución consistiría en restringir la aceleración del centro de masa de robot de manera tal de que el punto de momento cero (ZMP) se encuentre en todo momento dentro del polígono de soporte del robot durante el movimiento de éste [3]. El ZMP se define como el punto en el piso en el

cual la suma de momentos de las fuerzas de inercia y de gravedad no tienen componentes sobre el eje vertical.

Utilizando las ecuaciones obtenidas en [20], el robot será dinámicamente estable en el caso de que las coordenadas  $(x_{zmp}, y_{zmp})$  se encuentren en todo momento dentro del polígono de soporte, siendo

$$x_{zmp} = x_m - \frac{z_m \ddot{x}_m}{\ddot{z}_m + g} \quad (1)$$

$$y_{zmp} = y_m - \frac{z_m \ddot{y}_m}{\ddot{z}_m + g} \quad (2)$$

Donde  $x_{zmp}$  y  $y_{zmp}$  la posición de ZMP en  $x$  y en  $y$  respectivamente.  $x_m$ ,  $y_m$  y  $z_m$  son las posiciones del centro de masa del robot en sus respectivos ejes,  $\ddot{x}_m$ ,  $\ddot{y}_m$  y  $\ddot{z}_m$  sus aceleraciones y  $g$  es la aceleración de la gravedad. Puede observarse a partir de las ecuaciones que el punto de momento cero coincidirá con el centro de masa en caso de que el robot se encuentre a velocidad constante.

Por lo tanto es necesario generar un perfil de aceleración del robot que permita en todo momento que el ZMP se encuentre sobre el polígono. Esto puede observarse en la Figura 23.

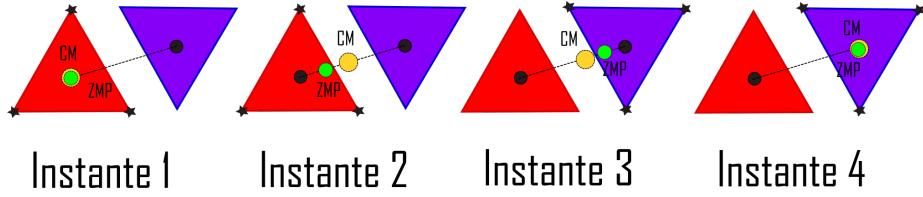


Figura 23: Estabilidad dinámica usando el criterio ZMP. Basado en [20].

Se pretende llevar al robot desde una posición inicial a una posición final. El robot se encuentra en reposo en el instante 1, tres patas del robot se encuentran en contacto con el suelo. En el instante 2 el robot comienza a moverse, pero aunque el centro de masa se encuentre fuera del polígono de soporte, el ZMP se encuentra dentro de éste. En el instante 3, se apoyan las otras tres piernas y se cambian las condiciones de aceleración de manera tal que nuevamente el ZMP se encuentre dentro del polígono. En el instante 4 se llega a la posición final. Se puede apreciar que en ningún momento de la trayectoria el robot es dinámicamente inestable.

En este proyecto se utilizó dentro del entorno de simulación el criterio de estabilidad estática para corroborar el correcto funcionamiento del sistema. Sin embargo, debido a que no se utilizarán los sensores adecuados, no se realizará el cálculo de margen de estabilidad en el robot real.

## 7. Dinámica

El diagrama de la pierna del robot utilizado para calcular la dinámica de ésta puede observarse en la Figura 24. En dicha figura pueden observarse los

ejes de cada articulación asignados en función de la convención de Denavit-Hartenberg. Cada eslabón tiene una masa y una longitud asociada, además cada eslabón posee una matriz de inercias relacionada con su distribución de masas. Se consideran matrices de inercia diagonales, por lo que se desprecian los posibles productos de inercia de los eslabones. Las longitudes  $l_{ci}$  corresponden a las distancias de cada articulación al centro de masa de cada eslabón.

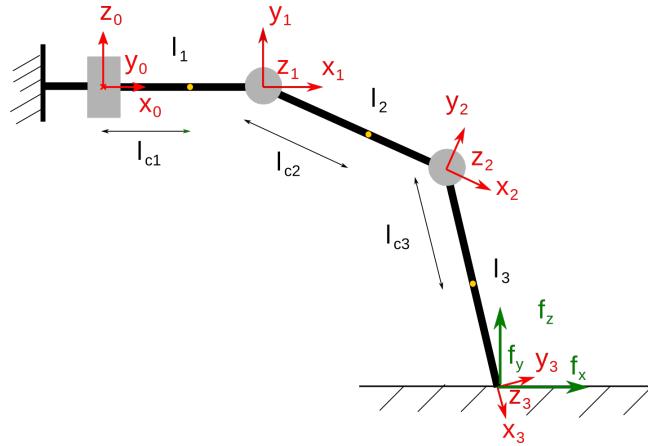


Figura 24: Diagrama para el cálculo de la dinámica de una pierna del robot.

La dinámica de la pierna del hexápodo está dado por la siguiente ecuación [15]

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + F(\dot{\theta}) + G(\theta) + J^T f$$

Donde  $\theta, \dot{\theta}$  y  $\ddot{\theta}$  son el vector de coordenadas generalizadas, de velocidades y de aceleraciones respectivamente.  $M$  es la matriz de inercia,  $C$  la matriz de Coriolis y de efectos centrífugos,  $F$  es la matriz de fricción,  $G$  es la matriz de gravedad,  $J$  es el jacobiano del manipulador y  $f$  es el vector que describe la fuerza aplicada en el efecto final.

Para obtener las matrices que describen el comportamiento dinámico de una pierna del robot, se puede utilizar el método de Newton-Euler.

La formulación de Newton-Euler parte del equilibrio de fuerza y pares:

$$\sum F_i = m_i \cdot \dot{v}_i$$

$$\sum T_i = I_i \cdot \dot{\omega}_i + \omega_i \times (I_i \cdot \omega_i)$$

Siendo:

- $F_i$ : Fuerzas ejercidas sobre el eslabón i.
- $T_i$ : Pares ejercidas sobre eslabón i.

- $m_i$ : Masa de eslabón i.
- $I_i$ : Tensor de inercia del eslabón i en torno a su centro de masas.
- $\dot{v}_i$ : Aceleración lineal del centro de masas de la articulación i.
- $\dot{\omega}_i$  y  $\omega_i$ : Velocidad y aceleración angular de la articulación i.

Trabajando las 2 ecuaciones dinámicas presentadas en la parte superior, de manera recursiva; podemos obtener la posición y sus respectivas derivadas en el tiempo del eslabón  $i$  referidos a la base del robot a partir de los correspondientes del eslabón  $i-1$  [21]. Con este procedimiento, comenzando con el eslabón inicial, la coxa, se alcanza el último eslabón de la cadena, la tibia. A partir de estas variables se obtienen las fuerzas y torques ejercidos sobre el eslabón, referidos a la base del robot, a partir de los respectivos al eslabón  $i+1$ , realizando el recorrido contrario, desde la tibia, hasta la coxa.

Los pasos a realizar para la obtención del modelo dinámico son:

1. Asignar a cada eslabón un sistema de referencia de acuerdo con las normas de D-H.
2. Establecer las condiciones iniciales.
3. Obtener las matrices de rotación y sus inversas.

Para  $i = 1 \dots n$ ,

4. Obtener la velocidad angular del sistema i.
5. Obtener la aceleración angular del sistema i.
6. Obtener la aceleración lineal del sistema i.
7. Obtener la aceleración lineal del centro de gravedad del eslabón i.

Para  $i = n \dots 1$ ,

8. Obtener la fuerza ejercida sobre el eslabón i.
9. Obtener el par ejercido sobre el eslabón i.
10. Obtener la fuerza o par aplicado a la articulación i

El grado de complejidad de este algoritmo depende directamente del número de grados de libertad.

Para poder corroborar la dinámica calculada se utilizó el toolbox de Peter Corke.

La matriz de inercia está dada por:

$$\begin{aligned}
M_{11} = & I_{xx2}/2 + I_{xx3}/2 + I_{yy1} + I_{yy2}/2 + I_{yy3}/2 - (I_{xx3}\cos(2\theta_2 + 2\theta_3))/2 + \\
& (I_{yy3}\cos(2\theta_2 + 2\theta_3))/2 + l_1^2 m_2 + l_1^2 m_3 + (l_2^2 m_3)/2 + l_{c1}^2 m_1 + \\
& (l_{c2}^2 m_2)/2 + (l_{c3}^2 m_3)/2 - (I_{xx2}\cos(2\theta_2))/2 + (I_{yy2}\cos(2\theta_2))/2 + \\
& (l_2^2 m_3 \cos(2\theta_2))/2 + (l_{c2}^2 m_2 \cos(2\theta_2))/2 + (l_{c3}^2 m_3 \cos(2\theta_2 + 2\theta_3))/2 \\
& + 2l_1 l_{c3} m_3 \cos(\theta_2 + \theta_3) + 2l_1 l_2 m_3 \cos(\theta_2) + 2l_1 l_{c2} m_2 \cos(\theta_2) \\
& + l_2 l_{c3} m_3 \cos(\theta_3) + l_2 l_{c3} m_3 \cos(2\theta_2 + \theta_3)
\end{aligned}$$

$$M_{12} = M_{13} = M_{21} = M_{31} = 0$$

$$M_{22} = m_3 l_2^2 + 2m_3 \cos(\theta_3) l_2 l_{c3} + m_2 l_{c2}^2 + m_3 l_{c3}^2 + I_{zz2} + I_{zz3}$$

$$M_{23} = M_{32} = m_3 l_{c3}^2 + l_2 m_3 \cos(\theta_3) l_{c3} + I_{zz3}$$

$$M_{33} = m_3 l_{c3}^2 + I_{zz3}$$

La matriz de Coriolis y efectos centrífugos está dada por:

$$\begin{aligned}
C_{11} = & -(\dot{\theta}_2(m_3 \sin(2\theta_2) l_2^2 + 2m_3 \sin(2\theta_2 + \theta_3) l_2 l_{c3} + \\
& 2l_1 m_3 \sin(\theta_2) l_2 + m_2 \sin(2\theta_2) l_{c2}^2 + 2l_1 m_2 \sin(\theta_2) l_{c2} + \\
& m_3 \sin(2\theta_2 + 2\theta_3) l_{c3}^2 + 2l_1 m_3 \sin(\theta_2 + \theta_3) l_{c3})) / 2 - \\
& (l_{c3} m_3 \dot{\theta}_3 (l_{c3} \sin(2\theta_2 + 2\theta_3) + 2l_1 \sin(\theta_2 + \theta_3) + \\
& l_2 \sin(\theta_3) + l_2 \sin(2\theta_2 + \theta_3))) / 2
\end{aligned}$$

$$\begin{aligned}
C_{12} = & -(\dot{\theta}_1(m_3 \sin(2\theta_2) l_2^2 + 2m_3 \sin(2\theta_2 + \theta_3) l_2 l_{c3} + 2l_1 m_3 \sin(\theta_2) l_2 + \\
& m_2 \sin(2\theta_2) l_{c2}^2 + 2l_1 m_2 \sin(\theta_2) l_{c2} + m_3 \sin(2\theta_2 + 2\theta_3) l_{c3}^2 \\
& + 2l_1 m_3 \sin(\theta_2 + \theta_3) l_{c3})) / 2
\end{aligned}$$

$$\begin{aligned}
C_{13} = & -(l_{c3} m_3 \dot{\theta}_1 (l_{c3} \sin(2\theta_2 + 2\theta_3) + 2l_1 \sin(\theta_2 + \theta_3) \\
& + l_2 \sin(\theta_3) + l_2 \sin(2\theta_2 + \theta_3))) / 2
\end{aligned}$$

$$\begin{aligned}
C_{21} = & \dot{\theta}_1 ((m_3 \sin(2\theta_2) l_2^2) / 2 + m_3 \sin(2\theta_2 + \theta_3) l_2 l_{c3} + \\
& l_1 m_3 \sin(\theta_2) l_2 + (m_2 \sin(2\theta_2) l_{c2}^2) / 2 + l_1 m_2 \sin(\theta_2) l_{c2} + \\
& (m_3 \sin(2\theta_2 + 2\theta_3) l_{c3}^2) / 2 + l_1 m_3 \sin(\theta_2 + \theta_3) l_{c3})
\end{aligned}$$

$$C_{22} = -l_2 l_{c3} m_3 \dot{\theta}_3 \sin(\theta_3)$$

$$C_{23} = -l_2 l_{c3} m_3 \sin(\theta_3) (\dot{\theta}_2 + \dot{\theta}_3)$$

$$\begin{aligned}
C_{31} = & (l_{c3} m_3 \dot{\theta}_1 (l_{c3} \sin(2\theta_2 + 2\theta_3) + 2l_1 \sin(\theta_2 + \theta_3) + l_2 \sin(\theta_3) \\
& + l_2 \sin(2\theta_2 + \theta_3))) / 2
\end{aligned}$$

$$C_{32} = l_2 l_{c3} m_3 \dot{\theta}_2 \sin(\theta_3)$$

$$C_{33} = 0$$

Para el cálculo de la matriz de gravedad se consideró un vector de gravedades de tres componentes ( $g_x, g_y, g_z$ ). Dicho vector depende de la orientación instantánea de la base del hexápodo. En caso de que la plataforma se encuentre paralela al piso, el vector solo tiene una componente distinta de cero, la componente en z que es igual en magnitud a la gravedad a la que esté sometido el robot. En caso de que la orientación de la plataforma varíe, es necesario calcular los valores del vector de gravedades para poder obtener de manera más exacta la dinámica del robot. La matriz de gravedad está dada por:

$$\begin{aligned} G(\theta) = & (g_y \cos(\theta_1) - g_x \sin(\theta_1))(l_1 m_2 + l_1 m_3 + l_{c1} m_1 + l_2 m_3 \cos(\theta_2) + \\ & l_{c2} m_2 \cos(\theta_2) + l_{c3} m_3 \cos(\theta_2 + \theta_3)), g_z l_2 m_3 \cos(\theta_2) + \\ & g_z l_{c2} m_2 \cos(\theta_2) + g_z l_{c3} m_3 \cos(\theta_2) \cos(\theta_3) - g_x l_2 m_3 \cos(\theta_1) \sin(\theta_2) - \\ & g_x l_{c2} m_2 \cos(\theta_1) \sin(\theta_2) - g_y l_2 m_3 \sin(\theta_1) \sin(\theta_2) - g_y l_{c2} m_2 \\ & \sin(\theta_1) \sin(\theta_2) - g_z l_{c3} m_3 \sin(\theta_2) \sin(\theta_3) - g_x l_{c3} m_3 \\ & \cos(\theta_1) \cos(\theta_2) \sin(\theta_3) - g_x l_{c3} m_3 \cos(\theta_1) \cos(\theta_3) \\ & \sin(\theta_2) - g_y l_{c3} m_3 \cos(\theta_2) \sin(\theta_1) \sin(\theta_3) - g_y l_{c3} m_3 \\ & \cos(\theta_3) \sin(\theta_1) \sin(\theta_2), l_{c3} m_3 (\cos(\theta_3) (g_z \cos(\theta_2) - \\ & \sin(\theta_2) (g_x \cos(\theta_1) + g_y \sin(\theta_1))) - \sin(\theta_3) (g_z \sin(\theta_2) \\ & + \cos(\theta_2) (g_x \cos(\theta_1) + g_y \sin(\theta_1)))) \end{aligned}$$

Para poder encontrar los torques de los motores debido a las fuerzas en el extremo de las patas del robot, es necesario calcular el término  $J^T f$  de la ecuación, siendo  $J^T$  la matriz jacobiana transpuesta y siendo  $f$  el vector de fuerzas y momentos. Se considera que los momentos en el efecto final son despreciables, por lo que dicho vector estará dado por  $[f_x, f_y, f_z, 0, 0, 0]^T$ . Los valores  $f_x, f_y, f_z$  son las fuerzas que se ejercen en el extremo de una pierna del robot. Dichas fuerzas se deben principalmente al peso del robot y a la fuerza de fricción del extremo del robot con la superficie donde éste se está desplazando. Las fuerzas involucradas pueden observarse en la Figura 24. Sea el vector  $\tau_f$  la contribución de torque debido a las fuerzas externas en el extremo, entonces:

$$\tau_f = \begin{bmatrix} \tau_{f1} \\ \tau_{f2} \\ \tau_{f3} \end{bmatrix} = J^T \begin{bmatrix} f_x \\ f_y \\ f_z \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

El valor de la matriz jacobiana dependerá del sistema de referencias a partir del cual se calculen las fuerzas. Si se toma el sistema de referencia como el mostrado en la Figura 24, es decir, que las fuerzas se calculan en función del eje fijo a la base, las componentes de torque debido a las fuerzas en los extremos estarán dadas por:

$$\begin{aligned}
\tau_{f1} &= f_y \cos(\theta_1) (l_1 + l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) - \\
&\quad f_x \sin(\theta_1) (l_1 + l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\
\tau_{f2} &= f_z (\cos(\theta_2 + \theta_3) (l_3 + l_2 \cos(\theta_3)) + l_2 \sin(\theta_2 + \theta_3) \sin(\theta_3)) - \\
&\quad f_y (\sin(\theta_2 + \theta_3) \sin(\theta_1) (l_3 + l_2 \cos(\theta_3)) - l_2 \cos(\theta_2 + \theta_3) \sin(\theta_1) \sin(\theta_3)) - \\
&\quad f_x (\sin(\theta_2 + \theta_3) \cos(\theta_1) (l_3 + l_2 \cos(\theta_3)) - l_2 \cos(\theta_2 + \theta_3) \cos(\theta_1) \sin(\theta_3)) \\
\tau_{f3} &= f_z l_3 \cos(\theta_2 + \theta_3) - f_x l_3 \sin(\theta_2 + \theta_3) \cos(\theta_1) - \\
&\quad f_y l_3 \sin(\theta_2 + \theta_3) \sin(\theta_1)
\end{aligned}$$

En caso de que las fuerzas se calculen a partir de los ejes del efecto final, y sean  $F_x, F_y, F_z$  las componentes de las fuerzas en dichos ejes, entonces las componentes de torque debido a las fuerzas en los extremos estarán dadas por:

$$\begin{aligned}
\tau_{f1} &= -F_z (l_1 + l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\
\tau_{f2} &= F_y (l_3 + l_2 \cos(\theta_3)) + F_x l_2 \sin(\theta_3) \\
\tau_{f3} &= F_y l_3
\end{aligned}$$

Para finalizar el desarrollo de dinámica inversa del robot, se modelan los efectos de fricción en las articulaciones. Para ello se tienen en cuenta tanto la componente de fricción viscosa como la componente de fricción de Coulomb. La primera corresponde a un torque proporcional a la velocidad angular de la articulación y la segunda produce un torque que se opone a que el actuador comience a moverse. Sean  $b_1, b_2, b_3$  los coeficientes de fricción viscosa de cada articulación y sean  $T_{c1}, T_{c2}, T_{c3}$  las fricciones de Coulomb, entonces el torque que debe realizar cada actuador debido a la fricción en las articulaciones está dado por:

$$F = \begin{bmatrix} \tau_{fa1} \\ \tau_{fa2} \\ \tau_{fa3} \end{bmatrix} = \begin{bmatrix} b_1 \dot{\theta}_1 + T_{c1} \operatorname{sig}(\dot{\theta}_1) \\ b_2 \dot{\theta}_2 + T_{c2} \operatorname{sig}(\dot{\theta}_2) \\ b_3 \dot{\theta}_3 + T_{c3} \operatorname{sig}(\dot{\theta}_3) \end{bmatrix}$$

Obteniendo de esta forma el modelo dinámico de una pata del robot.

Para simular el movimiento del manipulador se hace uso del modelo dinámico calculado. La simulación requiere resolver la ecuación dinámica:

$$\ddot{\theta} = M(\theta)^{-1} (\tau - C(\theta, \dot{\theta}) \dot{\theta} - F(\dot{\theta}) - G(\theta) - J^T f)$$

Por lo que es posible aplicar técnicas de integración numérica para integrar la aceleración y calcular velocidades y posiciones futuras.

A continuación se utilizan las ecuaciones dinámicas inversas para estimar los torques que deben realizarse en los actuadores cuando el robot se desplaza sobre el suelo.

Para desarrollar la dinámica general del robot se pueden utilizar diversas estrategias. A nivel analítico, la dinámica lagrangiana puede obtenerse separando

el robot en diversos subsistemas para luego vincularlos a través de los multiplicadores de Lagrange. Esta es una de las mejores estrategias para la resolución de la dinámica de sistemas híbridos. Este análisis se puede encontrar en [22]. En las próximas subsecciones se hace un análisis de las fuerzas de contacto del robot para luego estimar los torques de las articulaciones.

### 7.1. Análisis de la fuerza de contacto

Para obtener los torques que se deben generar debido a las fuerzas de reacción en los extremos de las patas del robot, es necesario estimar dichas fuerzas en función del peso de la estructura, se hace un análisis similar al realizado en [23].

Para el caso de la operación normal se supone que el robot se desplaza usando *tripod gait*, es decir, que en cada momento hay tres patas en fase de impulsión y tres en fase de apoyo. Es importante recalcar que para la selección de actuadores debe considerarse el caso más desfavorable. Este caso se produce cuando el robot se encuentra soportado por solo tres patas totalmente extendidas. Dicha configuración atraerá consigo los máximos torques en los actuadores.

La ecuación de balance de fuerzas y momentos cuando tres patas se encuentran en fase de apoyo está dada por:

$$AR = W$$

Siendo

$$R = [f_{z1}, f_{z3}, f_{z5}]^T$$

$$W = [0, 0, W]^T$$

$$A = \begin{bmatrix} x_{fz1} & x_{fz3} & x_{fz5} \\ y_{fz1} & y_{fz3} & y_{fz5} \\ 1 & 1 & 1 \end{bmatrix}$$

Donde  $f_{zi}$  son las fuerzas de reacción con el suelo de la pierna i, W es el peso del robot y  $x_{fzi}$  y  $y_{fzi}$  son las componentes de posición del pie i con respecto al sistema de referencia fijo a la base del robot.

Es posible por lo tanto obtener las fuerzas de reacción con el suelo de la forma:

$$R = A^{-1}W$$

Para evitar el resbalamiento del robot se debe satisfacer la siguiente consideración.

$$\sqrt{f_{xi}^2 + f_{yi}^2} \leq \mu f_{zi}$$

donde  $\mu$  es el coeficiente de fricción estática del suelo.

## 7.2. Estimación de torques de los actuadores

Para estimar los torques en los actuadores se utiliza Simscape Multibody. Esta herramienta desarrollada por MathWorks provee un entorno de simulación para sistemas mecánicos 3D. Permite modelar sistemas complejos usando bloques que representen cuerpos, articulaciones, restricciones, elementos de fuerza y sensores.

Para realizar las estimaciones de torque en las articulaciones, se crea un modelo simplificado de una sola pata en Simscape. Este modelo puede observarse en la Figura 25.

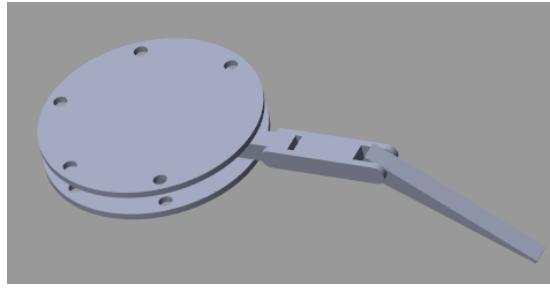
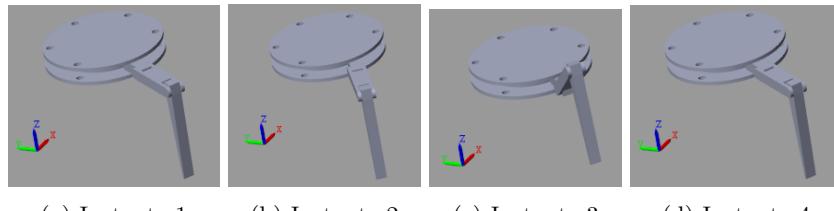


Figura 25: Modelo de la pata en Simscape.

A partir de las ecuaciones de la dinámica del robot y de la estimación de las fuerzas de contacto obtenidas en las secciones anteriores se procede a realizar las simulaciones correspondientes para obtener los torques necesarios en los actuadores. En la simulación se utilizan las mismas masas estimadas del robot real, así como las mismas dimensiones de las patas para asegurar que los resultados obtenidos se asemejen lo más posible a los resultados de la aplicación real.

Se realizan simulaciones que corresponden a los movimientos habituales del robot. Una de ellas puede observarse en la Figura 26. Dicho movimiento corresponde al movimiento que debería realizar una pata del robot cuando se produce una rotación del mismo y correspondería al caso de funcionamiento normal más desfavorable para la primera articulación.



(a) Instante 1      (b) Instante 2      (c) Instante 3      (d) Instante 4

Figura 26: Trayectoria habitual de una pierna del hexápedo.

Se definieron trayectorias conformadas por cinco puntos para simular las fases de impulsión y de apoyo de una pata del hexápedo. Los puntos que con-

forman las trayectorias fueron unidas mediante trayectorias formadas por polinomios de orden cinco para evitar picos de aceleración que no podrían obtenerse en el robot real y de esta forma tener un modelo más realista.

Para realizar las estimaciones, se supone que el peso del robot se distribuye uniformemente sobre tres patas que sostienen al robot, por lo que la fuerza que actúa sobre cada pata es igual a un tercio del peso total del robot. Se considera un coeficiente de fricción estático de 0.6 (coeficiente de fricción del caucho sobre madera), ya que las patas están recubiertas por aislantes de caucho. Se realizan las simulaciones para la obtención de los torques en los actuadores, esto puede verse en la Figura 27. En dicha figura se pueden observar los torques de cada articulación para un movimiento normal de la pata del robot.

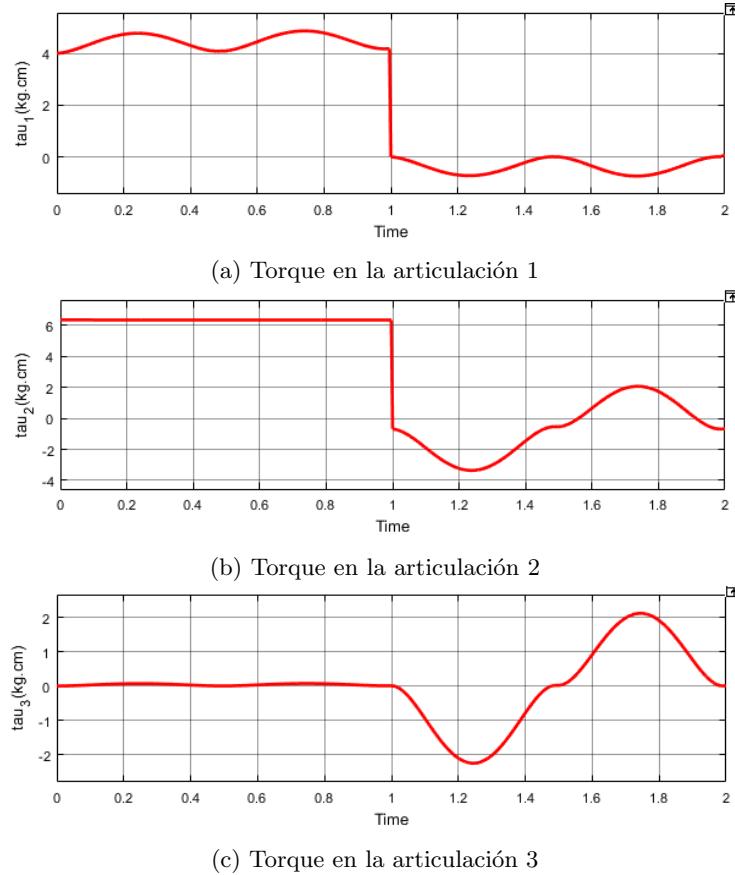


Figura 27: Trayectoria habitual de una pierna del hexápedo.

Desde el tiempo  $t = 0\text{ s}$  hasta  $t = 1\text{ s}$  la pierna se encuentra en fase de apoyo, por lo que el extremo de la pierna se encuentra en contacto con el suelo y sometida a las fuerzas debidas a la fricción y al mismo peso del robot. Desde el tiempo  $t = 1\text{ s}$  hasta  $t = 2\text{ s}$  la pierna se encuentra en el aire y el torque se

debe solamente a las inercias y masas de los eslabones.

Entre  $t = 0\text{ s}$  y  $t = 1\text{ s}$  puede observarse que la articulación 2 es la que presenta el torque más elevado, esto se debe al propio peso del robot. La articulación 1 presenta un alto torque debido a los efectos de fricción, mientras que la articulación 3 presenta un torque casi nulo debido a que el brazo de palanca es cercano a cero. Entre  $t = 1\text{ s}$  y  $t = 2\text{ s}$  los torques son, en general, menores debido a que no se produce contacto de las piernas con suelo, por lo que las articulaciones no están sometidas a fuerzas de fricción y a fuerzas debidas al peso del robot. Se puede observar un mayor torque en la segunda articulación, sin embargo es pequeño en comparación al obtenido en el primer periodo de tiempo.

Otra condición desfavorable, esta vez para la tercera articulación, puede observarse en la Figura 28.

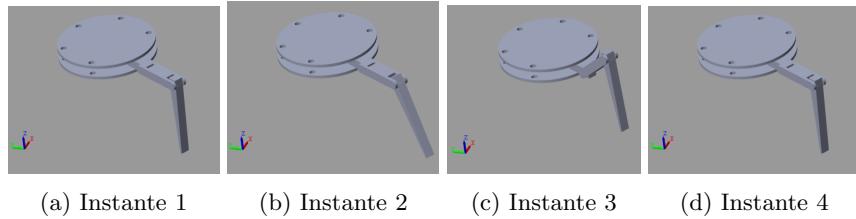


Figura 28: Trayectoria habitual de una pierna del hexápodo.

Este caso se produce durante la traslación del robot. La articulación 3 es la que produce la fuerza necesaria en el extremo de la pata para poder movilizar el robot. Es decir, las fuerzas de fricción tendrán efecto únicamente en esta articulación. En la Figura 29 pueden verse los torques obtenidos en la simulación para cada una de las articulaciones de la pata.

Se puede apreciar que la primera articulación el torque es nulo, dado que no se produce ningún tipo de movimiento en dicho eje y el actuador no está sometido a esfuerzos estáticos. La segunda articulación está sometida a una torque elevado debido a los efectos del peso de robot, el torque es similar a la simulación realizada anteriormente. En cambio, la articulación 3 es la encargada de producir las fuerzas necesarias en extremo del último eslabón. Dichas fuerzas son debidas a la fricción con el suelo. Es por ello que los torques son superiores y del orden de los  $6\text{ kg.cm}$ .

Luego de realizar simulaciones similares para distintos movimientos del hexápodo se obtiene que la segunda articulación es, en general, la más forzada para la mayoría de movimientos normales. Esto se debe al mayor brazo de palanca que está ejercido sobre esta articulación. Dicho brazo de palanca puede aumentar bastante en caso de que se extiendan las patas del robot. Sería por ello conveniente utilizar actuadores con mayor torque en la segunda articulación.

Las simulaciones fueron realizadas teniendo en cuenta condiciones normales de funcionamiento del robot. Sin embargo, existen distintas condiciones en donde los torques pueden ser mucho más elevados a los obtenidos anteriormente.

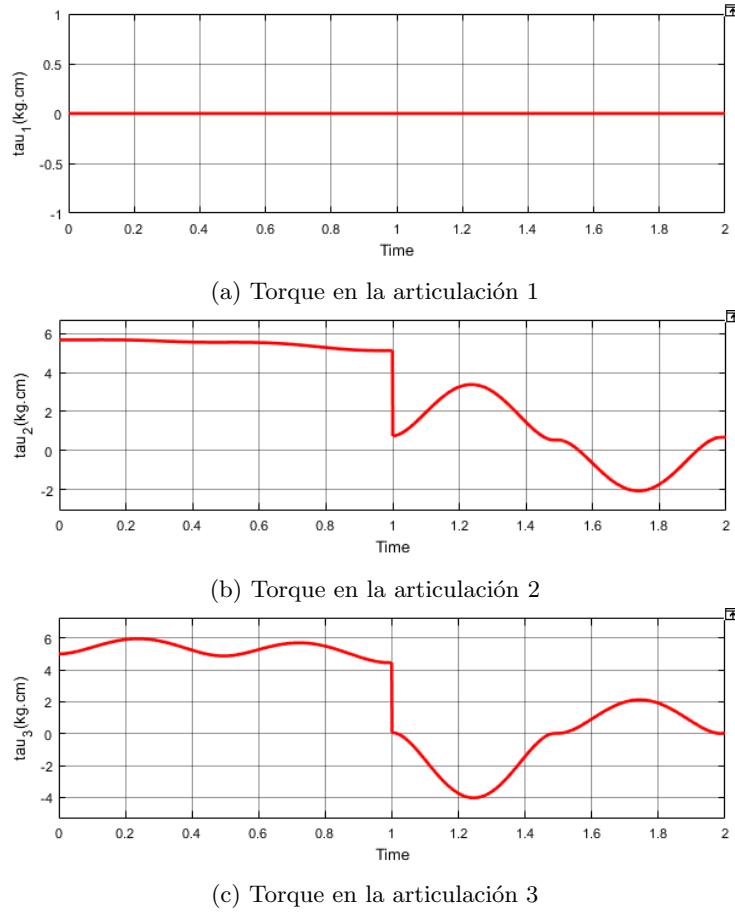


Figura 29: Trayectoria habitual de una pierna del hexápedo.

Algunas de estas condiciones pueden ser por ejemplo, la translación y rotación del robot con brazos extendidos, inclinación del suelo, mayor carga del robot, entre muchos otros. Es por ello necesario sobredimensionar los motores del hexápodo para tener en cuenta, al menos en cierta medida, la operación en alguna de estas condiciones.

## 8. Selección del hardware

Para poder desarrollar la aplicación es necesario seleccionar el hardware apropiado. Este debe permitir las comunicaciones inalámbricas con el operador y el correcto movimiento del robot. Se debe seleccionar la placa controladora y los motores de forma tal que el robot pueda moverse de forma fluida sobre la superficie sin que se excedan los torques máximos de las articulaciones.

## 8.1. Placa controladora

Con el fin de recibir los comandos de parte del operador y accionar los 18 motores según las trayectorias deseadas se utiliza una Raspberry Pi 4. Ésta puede observarse en la Figura 30.

Esta placa posee un procesador ARMv8 Cortex-A72 de cuatro núcleos y de 1.5GHz. Posee una memoria SDRAM de 4 GB, interfaces de comunicación WiFi (IEEE 802.11ac) y Bluetooth 5.0. Además la Raspberry Pi 4 dispone de 40 pines GPIO para poder dar señales de consigna a los motores.

Varios sistemas operativos pueden ser instalados, tales como Raspbian, ArchLinux, Windows 10, entre otros. Esto facilita el desarrollo del código necesario para realizar las tareas requeridas, dado que se puede optar por utilizar diversos lenguajes de programación.



Figura 30: Raspberry Pi 4.

Otras ventajas son su bajo costo y su tamaño compacto. Esto permite que la placa pueda fácilmente ser instalada en el robot sin que esto conlleve una variación importante en el diseño ni una variación significativa en el peso del mismo.

## 8.2. Selección de actuadores

Con el objetivo de mover las articulaciones del robot se utilizan servomotores de radio-control. El torque que deben realizar cada uno de estos actuadores va a depender de la posición y orientación tanto de la plataforma como de las patas del robot en relación con el suelo. Sin embargo, haciendo un análisis estático de fuerzas y realizando las simulaciones de movimiento podemos apreciar que en la mayoría de las posiciones del robot la segunda articulación es, en general, la más forzada. Si bien existen configuraciones donde, por ejemplo, la tercera articulación debería realizar un torque mayor a la segunda, estas configuraciones son más inusuales.

En un comienzo se realizaron pruebas utilizando servomotores analógicos similares a los que se muestran en la Figura 31. Éstos son más económicos y presentan un menor consumo energético. Sin embargo poseen ciertas desventajas que perjudican el funcionamiento del robot. Su lenta respuesta, su amplio ancho de banda muerta y su baja resolución producen movimientos poco precisos. Es por ello que se optó por utilizar servomotores digitales, cuya única desventaja reside en un mayor consumo energético, debido al incremento en la frecuencia dentro de un mismo ciclo de trabajo.

Se decidió utilizar servomotores digitales TD-8120MG de mayor torque para la segunda articulación y servomotores digitales MG996R de menor torque para la primera y la tercera. Esta decisión se realizó debido a razones técnicas y económicas. Es necesario remarcar que sería conveniente utilizar, si fuera económicamente posible, motores de mayor torque para todas las articulaciones. De esta forma sería posible obtener más libertad de movimiento del robot aun para condiciones más desfavorables.

Cada uno de estos motores presenta un consumo de corriente considerable en los casos más desfavorables (del orden de los 2,5 A) por lo que es necesario tener una fuente de alimentación apropiada para poder alimentarlos.

#### 8.2.1. Servomotores MG996R

El MG996R es un servomotor digital de engranajes metálicos que permite utilizar movimientos de 0 a 180 grados. Este motor presenta un torque de 9,4 kg.cm a 4,8 V y de 11 kg.cm a 6 V. Este tiene una velocidad de operación 0.17 s/60°. La masa es de 55 gramos, se utilizan 12 de estos motores de los cuales 6 son utilizados en la articulación 1 y los restantes 6 son utilizados en la articulación 3. Este puede visualizarse en la Figura 31.



Figura 31: MG996R.

### 8.2.2. Servomotores TD-8120MG

Para la segunda articulación de cada pata se utilizan servomotores digitales TD-8120MG. Estos motores también poseen engranajes metálicos y presentan torques de 18,5 kg.cm a 4,8 V y de 21,8 kg.cm a 6,6 V. Permiten rotaciones entre 0 y 180 grados. En cuanto a su velocidad, éstos pueden llegar 0,18 s/60° a 4,8 V y 0,14 s/60° a una tensión de 6,6 V. El motor en cuestión puede observarse en la Figura 32.



Figura 32: TD-8120MG.

### 8.2.3. Placa controladora PCA9685

Para poder controlar los motores se recurre a una placa PCA9685, dado que la Raspberry Pi no cuenta con la cantidad de pines necesarios para controlar la cantidad de actuadores requeridos. La placa permite controlar hasta 16 servomotores a través de I2C. Esto permite reducir la cantidad de pines necesarios para mover los motores y simplificar la programación requerida. Dado que se tienen 18 motores a controlar, se requiere utilizar dos de estas placas conectadas entre sí, las cuales controlan 9 motores cada una. Esta se muestra en la Figura 33.



Figura 33: Placa controladora PCA9685.

#### 8.2.4. Baterías 18650

Para la alimentación de los motores se utilizan pilas Li-On recargables Sony Konion US18650VTC5A de 3,7 V, 2200 mAh y 25 A de corriente máxima. Éstas pueden observarse en la Figura 34. Se utilizan cuatro de estas baterías dispuestas en serie para alimentar a los motores, dándole la posibilidad al robot de desplazarse sin necesidad de estar conectado a una fuente externa. Para regular la tensión y mantenerla en el rango de voltaje de los motores se utilizaron reguladores de tensión LM2596 que permiten regular tensiones de entrada de entre 4 V - 35 V a valores de entre 1.23 V - 30 V.



Figura 34: Baterías 18650.

### 8.3. Selección de sensores

#### 8.3.1. Potenciómetros

Los servomotores utilizados en cada una de las articulaciones cuentan con un potenciómetro el cual es utilizado como sensor de posición, permitiendo conocer la posición exacta del eje de giro del servomotor. Esto puede verse en la Figura 35.



Figura 35: Composición del servomotor utilizado.

Con el sensor se puede corregir la posición del eje para que en todo momento se encuentre en la posición deseada. Esto es realizado por el circuito de control

que se muestra en la figura. Dicho circuito forma parte del actuador y no se ha realizado cambios en el lazo de control, es decir, se ha limitado a utilizar los servomotores a lazo abierto.

### 8.3.2. Sensores no implementados

Existen distintos sensores que podrían ser utilizados para mejorar el funcionamiento del hexápodo:

- Giróscopos: para controlar la orientación de la base mediante un lazo de control cerrado.
- Cámara: la cual brindaría información acerca del entorno del robot.
- GPS: para verificar la posición absoluta del hexápodo y corregir los errores de trayectoria debido a resbalamientos o defectos mecánicos

La selección de los sensores dependerá de la aplicación en cuestión y si bien se ha analizado la utilización de diversos sensores, su implementación excede las capacidades de este proyecto y será realizada en trabajo futuro.

## 9. Diseño mecánico

El diseño del hexápodo tuvo como premisa fundamental la versatilidad motriz, por lo que se basó en una distribución angular de  $60^{\circ}$  entre todas sus patas. Esta particularidad le otorga al robot una mayor facilidad para poder avanzar en distintas direcciones sin la necesidad de rotar sobre su eje, así como también simplificó el análisis de estabilidad debido a la simetría que posee. El diseño final del robot puede observarse en la Figura 36.



Figura 36: Renderizado del hexápodo en KeyShot 8.

Utilizando software CAD Solid Edge se diseñó toda la estructura del robot, la cual determina las longitudes fundamentales  $l_0$ ,  $l_1$ ,  $l_2$  y  $l_3$  que rigen la cinemática del robot.

A grandes rasgos el robot consta de la base y las 6 extremidades que le brindan la capacidad de desplazamiento y sustentación.

### 9.1. Base

Como se observa en la Figura 37 y la Figura 38 la base se forma a partir de dos placas paralelas de forma estrella-hexagonal.

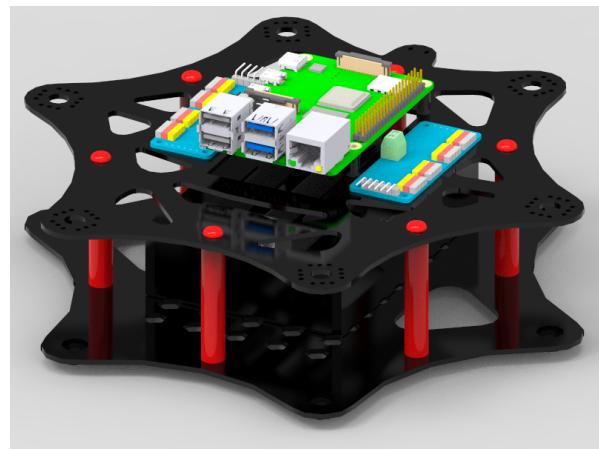


Figura 37: Vista superior de la base en KeyShot 8.

Las placas se encuentran vinculadas por 6 soportes uniformemente distribuidos, los cuales le otorgan mayor rigidez una vez insertados los motores.

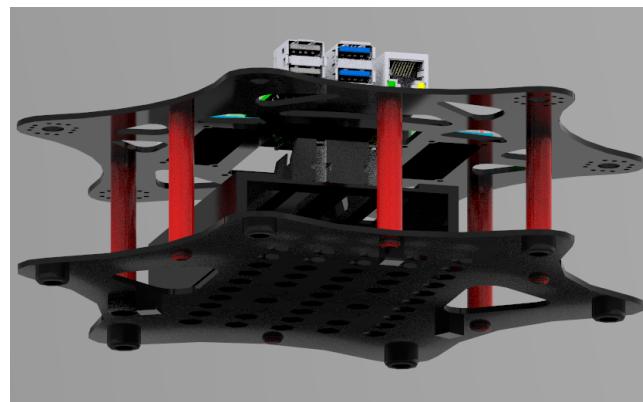


Figura 38: Vista inferior de la base en KeyShot 8.

En las figuras anteriores se puede observar la placa Raspberry Pi 4, y los dos módulos controladores de servos PCA9685. Cada uno de estos módulos está a cargo de controlar 9 de los 18 motores del hexápedo.

En el compartimento interno, se ubica el banco de baterías en el cual se encuentran separadas las baterías que alimentan los motores, de la batería que alimenta la placa controladora.

Para embellecer la parte central del robot, se diseñó la carcasa que se muestra en la Figura 39, la cual cubre la parte superior de la base.

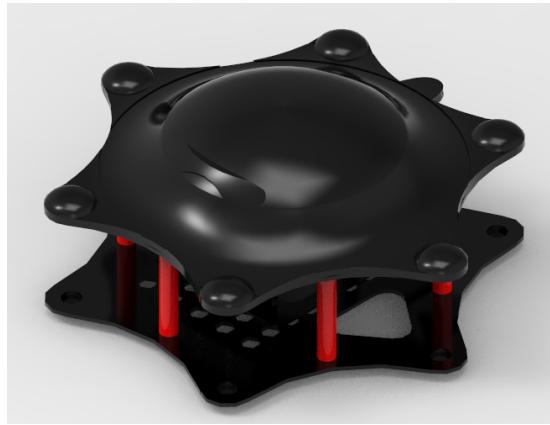


Figura 39: Base con carcasa en KeyShot 8.

Observando las cotas ubicadas en las 3 direcciones de la Figura 40, se puede observar que todos los motores de la primera articulación de cada pata, equidistan del centro del robot.

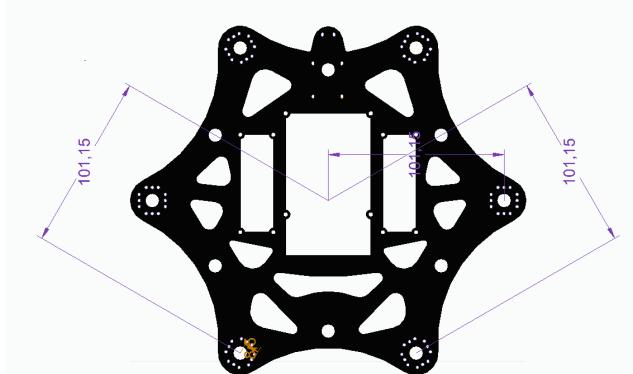


Figura 40: Sección superior en la que  $l_0$  se encuentra acotada.

## 9.2. Extremidades

En este robot, si bien la base tiene 6GDL, si lo tomamos como un robot serie-paralelo, cuando el hexápedo comienza a comportarse como robot móvil sus 6 patas son las que les brindan el movimiento. Cada una de las patas consta de 3 articulaciones y 3 eslabones, como se observa en la Figura 41.



Figura 41: Pata en KeyShot 8

Una perspectiva desde abajo de la pata puede verse en la Figura 42.

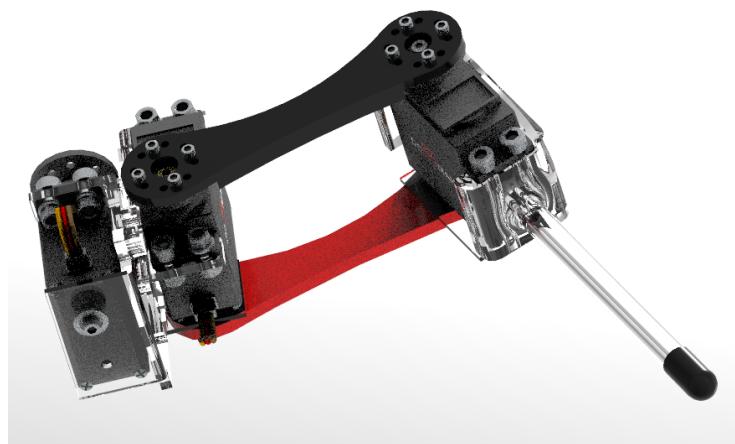


Figura 42: Pata en KeyShot 8, perspectiva desde abajo.

### 9.2.1. Eslabón 1: Coxa

El primer eslabón de las extremidades del hexápedo se encuentra ligado a la articulación 1 y la articulación 2. Ambos amarrados por tornillos métrica 4. La articulación 1 posee dos puntos de vinculación con la base, uno motriz (junta superior), Figura 43, y el otro que funciona de pivote (rodamiento inferior), Figura 44.

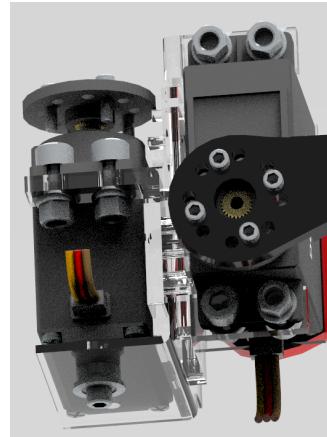


Figura 43: Coxa en KeyShot 8.

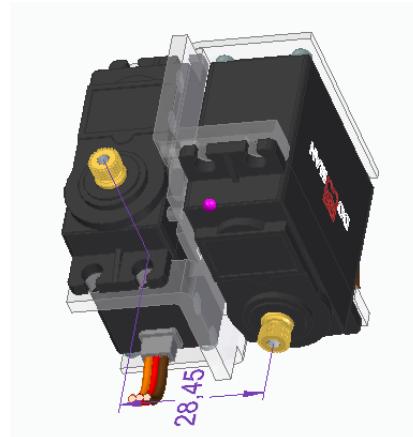


Figura 44: Coxa en Solid Edge 2020.

Como se observa en la Figura 44, la longitud  $l_1$  se ve acotada entre la articulación 1 y 2.

### 9.2.2. Eslabón 2: Fémur

El eslabón número 2 se encuentra ligado a la articulación 2 y 3, idealmente estas articulaciones deben ser paralelas. Inicialmente este eslabón estaba conformado por una sola pieza que vinculaba los motores a los amarres. Sin embargo, una vez construido el robot, se pudo observar ciertos alabeos en los ejes. Dichos alabeos producían esfuerzos de torsión en el eslabón.

Para contrarrestar este par torsional, en primera instancia se maximizó el grosor de la pieza, y en segunda instancia se le agregó una compensación en la parte posterior de los eslabones 1 y 3 las cuales poseen un rodamiento en cada una. Esto le brinda mayor estabilidad al eslabón 2, evitando deformaciones en el amarre. En la Figura 45 se puede ver, en rojo, la compensación en paralelo ubicada por detrás.

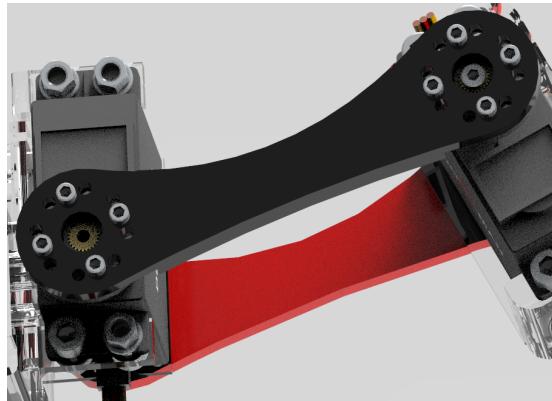


Figura 45: Fémur con compensación paralela en rojo.

La longitud  $l_2$  se ve acotada entre la articulación 2 y 3, como se observa en la Figura 46. En dicha imagen puede observarse las dimensiones utilizadas en el fémur.

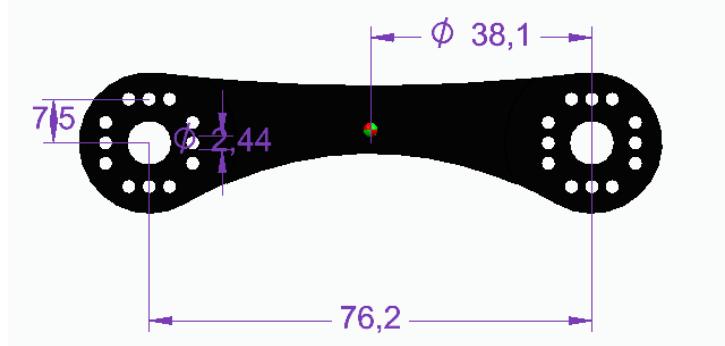


Figura 46: Fémur en Solid Edge 2020.

### 9.2.3. Eslabón 3: Tibia

El tercer eslabón tiene la particularidad de estar vinculado con la tercera articulación y con el suelo. Dependiendo de la rugosidad del suelo, es la fuerza de contacto que se genera entre el efecto final (el extremo del tercer eslabón) y el suelo. Para evitar deslizamiento y aumentar el agarre en el diseño se consideró que el extremo de este eslabón sería de un material diferente, por lo que se le agregó una especie de cobertura en el punto de contacto.

La longitud  $l_2$  se ve acotada entre la articulación 2 y 3. Las tibias pueden verse en la Figura 47 y Figura 48.

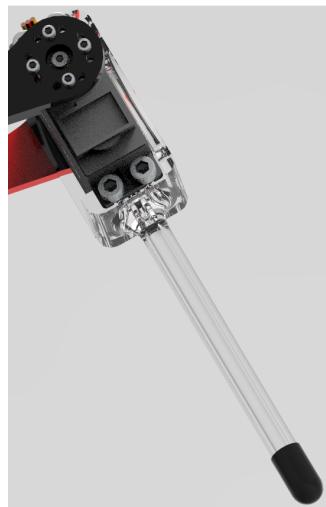


Figura 47: Eslabón 3 en KeyShot 8.

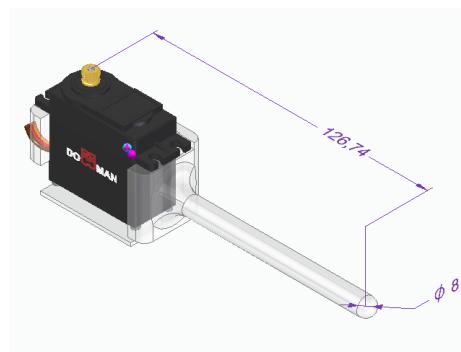


Figura 48: Eslabón 3 en Solid Edge.

## 10. Arquitectura de control

La arquitectura de control del robot puede observarse en la Figura 49. Se utiliza un sistema de control de lazo abierto en el nivel superior, para ejecutar consignas de movimiento del robot completo, y un sistema de lazo cerrado en cada servomotor independiente. El operador envía consignas al robot, este ingresa al planificador de trayectorias de las patas que calcula la cantidad de pasos a realizar y la distancia de estos. Posteriormente las consignas de este planificador ingresan al planificador de trayectoria del robot que, a partir de la cinemática inversa de estas, genera la consigna de posición para motores.

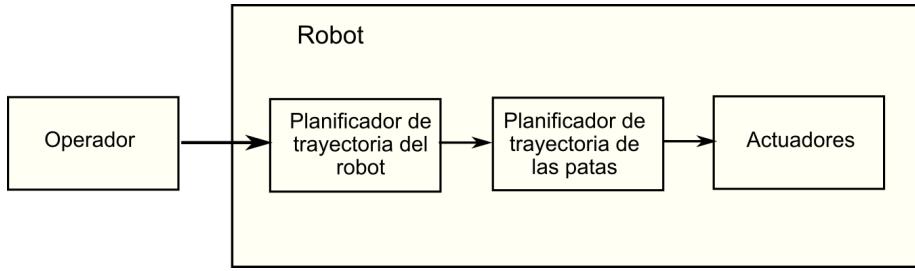


Figura 49: Arquitectura de control.

Cada uno de los servomotores posee un sistema de control de lazo cerrado basado en la lectura de un potenciómetro que varía su salida en función de la posición angular del motor. Esto les permite satisfacer las consignas de control eliminando perturbaciones.

En caso de que se tuviese algún tipo de sensor que permitiera obtener la posición absoluta del robot en el espacio, sería posible realizar un control de posición absoluta cuando el robot se desplaza.

Se podrían producir diferencias entre el desplazamiento de consigna y el desplazamiento real del robot debido a resbalamientos, desperfectos mecánicos, imperfección en la calibración de los ejes, errores de precisión de las variables en el programa y otros tipos de problemas. Por lo que un controlador de posición absoluta podría ser la solución para este tipo de problema. Un controlador de dichas características ya fue diseñado en [24], sin embargo la utilización de dicho controlador excede las capacidades de este proyecto y no será implementado.

A continuación se explican los fundamentos básicos de los bloques que conforman la arquitectura de control propuesta. La implementación del sistema de control se realiza en las secciones de simulación y programación, en donde se explica con mayor detalle el funcionamiento de cada uno de los bloques.

### 10.1. Planificador de trayectorias del robot

El planificador de trayectorias del robot tiene como objetivo definir los desplazamientos que deben realizarse por parte del robot. La cantidad de pasos que deben efectuarse, la dirección y su longitud deben ser determinados por dicho

planificador. Estos valores son obtenidos en función de los comandos ingresados por el operador.

El planificador tiene como entradas las posiciones en X y en Y a las cuales debe desplazarse el robot. En función de dichas entradas y en función de la posición angular actual del robot, se define el ángulo en el cual el hexápodo debe moverse. En función de la distancia de paso, se calcula la cantidad de pasos que debe realizar el robot para que éste alcance la posición deseada. Es decir que la salida del planificador está conformada por la cantidad de pasos a realizar y su dirección en *rad*.

La información obtenida es llevada al planificador de trayectorias de las patas el cual genera el vector de variables articulares que debe satisfacer el hexápodo.

## 10.2. Planificador de trayectorias de las patas

El planificador de trayectorias de las patas tiene como objetivo generar los pasos del hexápodo, es decir, definir los valores de posición en *x*, *y* y *z* que tomarán cada una de las piernas del robot en función de las distancias de paso requeridas y los tiempos necesarios provistos por el planificador del robot. Una vez calculado los vectores de variables cartesianas y sus vectores de tiempo correspondiente, estos son almacenados y llevados al bloque de cinemática inversa para poder dar consignas a los actuadores. Si suponemos que las piernas impares comienzan con la fase de impulsión y las pares con la fase de apoyo, las salidas del planificador serán, para las piernas impares,

$$\begin{aligned} l_{ix} &= \left( x_0, \quad x_0 + \frac{dx_i}{4}, \quad x_0 + \frac{dx_i}{2}, \quad x_0 + \frac{dx_i}{4}, \quad x_0 \right), \\ l_{iy} &= \left( y_0, \quad y_0 + \frac{dy_i}{4}, \quad y_0 + \frac{dy_i}{2}, \quad y_0 + \frac{dy_i}{4}, \quad y_0 \right), \\ l_{iz} &= \left( z_0, \quad z_0 + h, \quad z_0, \quad z_0, \quad z_0 \right). \end{aligned}$$

Y para las piernas pares,

$$\begin{aligned} l_{ix} &= \left( x_0, \quad x_0 - \frac{dx_i}{4}, \quad x_0 - \frac{dx_i}{2}, \quad x_0 - \frac{dx_i}{4}, \quad x_0 \right), \\ l_{iy} &= \left( y_0, \quad y_0 - \frac{dy_i}{4}, \quad y_0 - \frac{dy_i}{2}, \quad y_0 - \frac{dy_i}{4}, \quad y_0 \right), \\ l_{iz} &= \left( z_0, \quad z_0, \quad z_0, \quad z_0 + h, \quad z_0 \right). \end{aligned}$$

Donde  $l_i$  son las coordenadas cartesianas que debe tomar el extremo de la *i*-ésima pierna en función del tiempo,  $x_0$ ,  $y_0$  y  $z_0$  son las coordenadas iniciales de dicho extremo con respecto al sistema de referencia de la pata,  $h$  es la altura a la que sube la pierna durante la fase de impulsión y  $dx_i$  y  $dy_i$  son los valores de desplazamiento de las piernas calculados en la sección de traslación. Dichos vectores están asociados a un vector de tiempo que especifican el momento donde deben realizarse dichas acciones. Dicho vector de tiempo tiene la forma,

$$t_i = (t_0, \quad t_1, \quad t_2, \quad t_3, \quad t_4)$$

En el tiempo  $t_0$  la pierna *i* se encuentra en su posición inicial  $(x_0, y_0, z_0)$ , en contacto con el suelo, esto puede verse en la Figura 50.

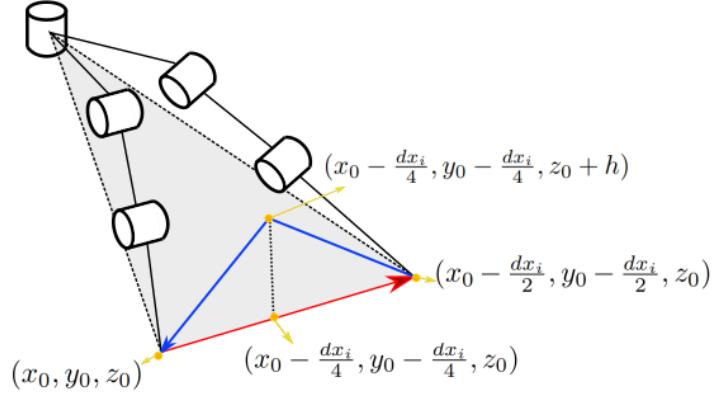


Figura 50: Trayectoria de una de las patas del robot.

Si consideramos que la pierna es par, ésta entra en la fase de apoyo. En el tiempo  $t_1$  la pierna se ha desplazado en  $x$  y en  $y$  pero se mantiene con la misma  $z$ , ahora se encontrará en la coordenada  $(x_0 - \frac{dx_i}{4}, y_0 - \frac{dx_i}{4}, z_0)$ . La fase de apoyo continúa hasta el instante  $t_2$ , donde el extremo de la pierna llega a la posición  $(x_0 - \frac{dx_i}{2}, y_0 - \frac{dx_i}{2}, z_0)$ . En ese momento la pierna entra en fase de impulsión y empieza moverse en sentido contrario en  $x$  y en  $y$ , la pierna ya no se encontrará en contacto con el suelo sino que se elevará. En el instante  $t_3$  el extremo de la pata se encontrará en las coordenadas  $(x_0 - \frac{dx_i}{4}, y_0 - \frac{dx_i}{4}, z_0 + h)$ , ubicándose en la posición de máxima altura. En  $t_4$ , la pierna volverá a su posición inicial  $(x_0, y_0, z_0)$  cumpliendo, de esta forma, un ciclo de movimiento. Una posible mejora que podría implementarse sería utilizar perfiles de trayectoria más redondeados que disminuyan las aceleraciones necesarias para poder llevar a cabo el movimiento. Esto permitiría una mayor suavidad en el desplazamiento de las piernas. Dicho mejora será realizada en trabajos futuros.

En caso de que se tuviesen sensores que permitan determinar el contacto de cada una de las piernas con el suelo, podría optarse por utilizar un *blind gait* [11]. Este tipo de paso permite que el hexápodo pueda moverse de forma más adecuada en terrenos irregulares. En este tipo de terrenos cuando la pierna se mueve hacia abajo puede contactar el piso en cualquier momento. El tiempo en que la pierna está en fase de impulsión no está determinado dado que la pierna se mueve hacia abajo con una velocidad constante mientras que la distancia al piso es desconocida. El algoritmo consiste en seleccionar la pierna que debe moverse, desplazar dicha pierna hacia adelante y hacia arriba simultáneamente, y luego mover hacia abajo hasta hacer contacto con el suelo, almacenando los valores obtenidos para ser utilizados en el próximo paso, para luego repetir el ciclo. Esto puede realizarse modificando las consignas de la sección anterior ubicando los puntos en posiciones de  $z$  arbitrariamente lejanas de la base y detener la pierna una vez se produce el contacto con el suelo, almacenando los valores obtenidos.

## 11. Simulación del robot

Con el objetivo de corroborar el comportamiento del robot previo a la puesta en marcha de éste, se utiliza una simulación. Ésta se realizó utilizando el motor de videojuegos Unity que provee herramientas de gran utilidad para la simulación de robots. La validación de los desarrollos matemáticos mostrados previamente en el informe fueron realizados en [24] usando Simscape, una librería de Matlab/Simulink que provee un entorno de simulación para sistemas mecánicos 3D. Si bien los resultados fueron acertados y se validó el desarrollo matemático propuesto, los tiempos de simulación no fueron adecuados para realizar una simulación en tiempo real del robot. Además, no fue posible simular terrenos irregulares, por lo que no se pudo evaluar el comportamiento del robot en estos tipos de terrenos.

Por lo tanto, para eliminar las restricciones previamente detalladas se optó por utilizar Unity. Dicho programa posee las herramientas que facilitan la generación de terrenos y la simulación de cuerpos rígidos. Además en Unity 2020.1a15 se agregaron funciones que facilitan la simulación de articulaciones de robots. Se utilizó la versión Unity 2020.1.0b5, la última beta en el momento en que se escribió este informe.

Para que la simulación sea lo más realista posible y el usuario pueda visualizar con mejor detalle la posición las patas y las dimensiones reales de los actuadores y de los eslabones, se utilizaron las mismas piezas que en el robot real.

Como se mencionó anteriormente las piezas fueron realizadas utilizando el software de diseño 3D Solid Edge. Los archivos en Solid Edge tienen formato .PAR para el caso de las partes y .ASM para el caso de los conjuntos. Ninguno de los formatos anteriores es soportado por Unity. Por ello es necesario cambiar el formato a .OBJ. Se guardan los archivos en Solid Edge en el formato .VRML para luego ser transformados a .OBJ utilizando el software MeshLab<sup>5</sup>. Éste es un programa Open Source para el editado y procesado de mallas en 3D. Una vez se tiene el formato requerido, las partes pueden ser fácilmente importadas en Unity. Una vez ingresadas las partes en Unity fue necesario realizar el ensamblado para obtener el robot. Éste puede observarse en la Figura 51.

Se utilizó los componentes ArticulationBody con el fin de realizar las conexiones entre los distintos eslabones del robot. Fue necesario ubicar los ejes de rotación de cada articulación de forma manual. Se le asignaron las masas e inercias a cada uno de los eslabones y a la plataforma del robot. Se asignó las fricciones estimadas, así como la rigidez de los eslabones con el fin de simular de la forma más exacta posible el comportamiento del robot real. Cada uno de los elementos del robot posee un *collider*, estos definen la forma de un objeto para los propósitos de colisiones físicas. Esto permite que la interacción entre el robot y el suelo, permitiendo que el robot se desplace sobre este. Esto puede observarse en la Figura 52

Una vez definida la estructura del robot. Se procedió a verificar los movi-

---

<sup>5</sup><https://www.meshlab.net/>

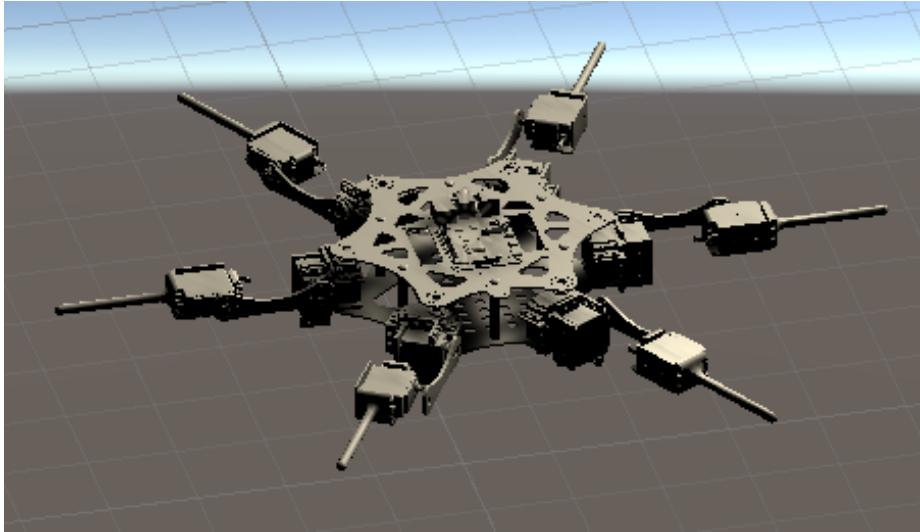


Figura 51: Robot en Unity, sin la compensación en la segunda articulación.

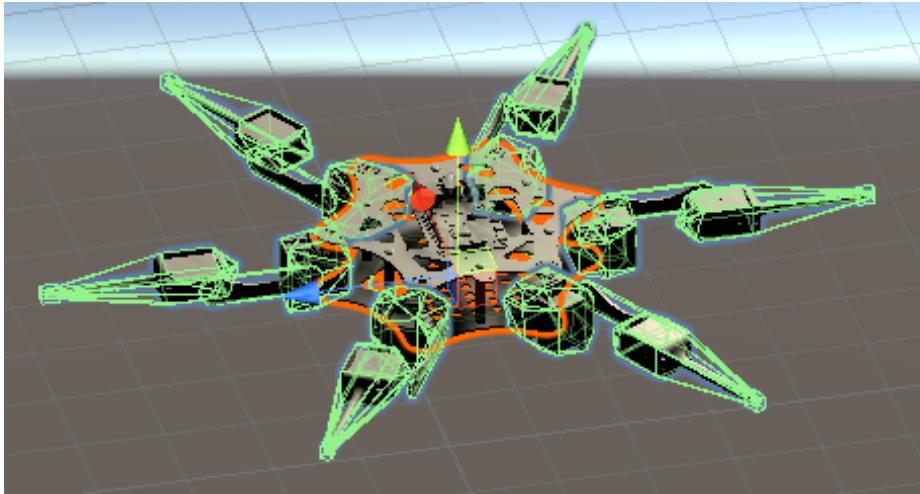


Figura 52: *Colliders* del robot.

mientos de éste. Para ello se creó una interfaz que permite al usuario visualizar las posiciones angulares de cada una de las articulaciones del robot así como también las posiciones y orientaciones de la plataforma, también posibilita establecer dichas variables. Además permite al usuario definir los modos de operación del robot, provee joysticks que permiten el movimiento del robot y de la cámara. La interfaz provista al operador puede observarse en la Figura 53

Como se mencionó anteriormente, la interfaz provee distintos elementos para



Figura 53: Interfaz gráfica en Unity.

visualizar el estado del robot y para darle comandos a los actuadores. A la izquierda de la Figura 53 puede visualizarse la posición y orientación instantánea de la plataforma (Platform), así como la posición angular instantánea de cada una de las 18 articulaciones del robot (Joints). Además se puede ver en todo momento el coeficiente de estabilidad estática del robot (Stability).

Otra característica consiste en que el operador puede dar comandos de *roll*, *pitch* y *yaw*, así como posiciones de X, Y e Z de la plataforma (centro de la figura). También puede ingresar el valor de las variables articulares de cada una de las articulaciones.

El usuario puede modificar algunos parámetros relevantes para el funcionamiento del robot, entre ellos se encuentran:

- Vel: Su velocidad.
- DS: Paso del robot.
- SH: Altura de paso.

Como otra alterenativa el operador tiene la opción de ingresar consignas de:

- Posicion: Coordenadas X e Y utilizando (Move to).
- Rotación: Sentido horario (C), Antihorario (A)

Tiene un joystick para que el robot camine en la dirección ingresada por el operador y otro joystick para movilizar la cámara de la simulación alrededor del robot para mejorar su visualización.

A la derecha puede observarse una barra que permite mostrar u ocultar los elementos de la interfaz. Además existe la opción *Uneven* que permite que la simulación utilice un *blind gait*.

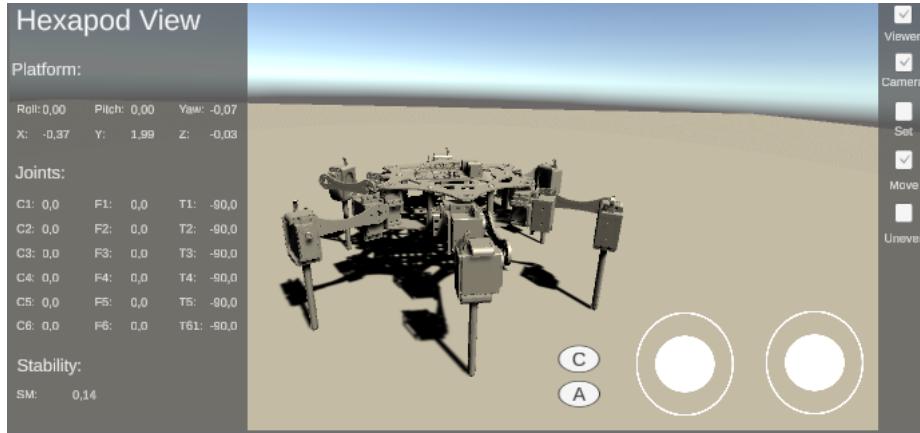


Figura 54: Robot en su posición inicial junto con la interfaz gráfica.

El robot junto con la interfaz puede verse en la Figura 54. El hexápodo se encuentra en la posición inicial. Se coloca un piso liso bajo el robot.

La arquitectura de control es la misma que la explicada en la sección anterior, existe un planificador de trayectorias del robot cuyas salidas ingresan en el bloque del planificador de trayectorias de las patas que a su vez dan consignas a los actuadores. Las salidas del planificador de trayectorias de las patas son un par de vectores de las posiciones angulares y de los tiempos en los cuales deben satisfacerse dichas posiciones. En cada instante de tiempo se interpola linealmente los valores de posición angular, definidas por el vector de posiciones angulares previamente calculado, en función del tiempo, definidos por el vector de tiempos, para dar consignas a los actuadores del hexápodo. Esto se realiza mediante el método  *FixedUpdate*, un método de Unity que se ejecuta en períodos tiempo fijo (cada 0,02 s). La escritura en los actuadores se realiza mediante el método  *Update* que se ejecuta en cada  *frame*.

Una vez realizada la interfaz en función de las necesidades del operador y ya definida la arquitectura de control, se procede a la programación de los movimientos del robot. La programación de las rutinas fue realizada en C# dado que es el lenguaje de programación más popular para la programación de aplicaciones en Unity. Se realizaron las rutinas para calcular la cinemática inversa, tanto de las patas y como de la plataforma en función de las ecuaciones obtenidas previamente. Se programaron las rutinas de movimientos y se les asignaron a los controles previamente mencionados.

El funcionamiento de las rutinas de rotación y de traslación de la plataforma pueden observarse en la Figura 55 y Figura 56 respectivamente.

Además se validó el comportamiento de la cinemática inversa de la plataforma. En la Figura 57 puede observarse el funcionamiento del hexápodo cuando a este se le da como consigna un ángulo de 20° en  *pitch*.

Se valida de esta forma los modelos matemáticos calculados y los algoritmos desarrollados para el movimiento del hexápodo. Para la programación del ro-

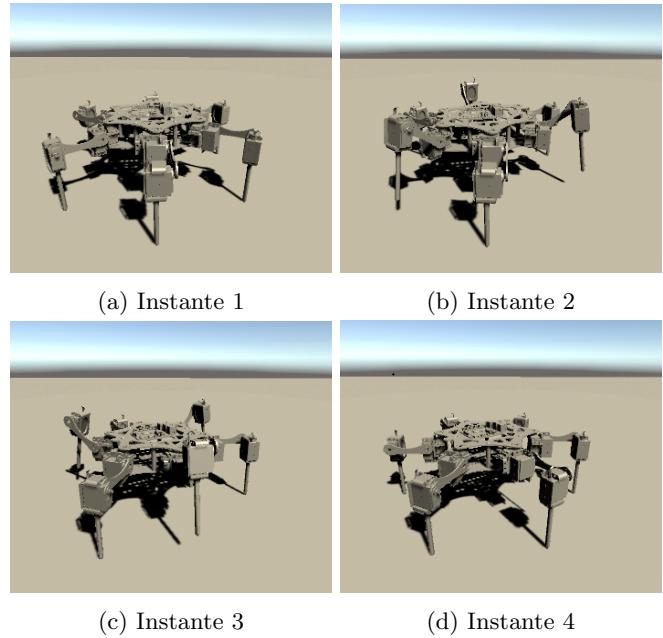


Figura 55: Rotación del hexápodo.

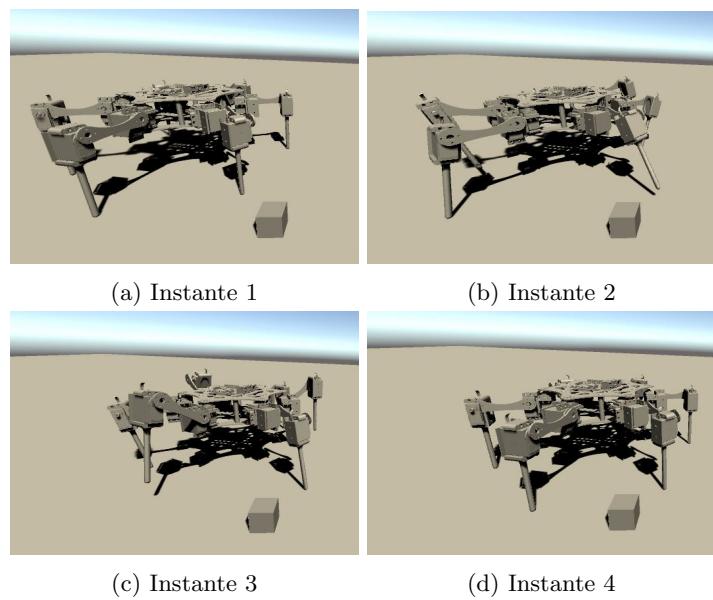


Figura 56: Traslación del hexápodo.

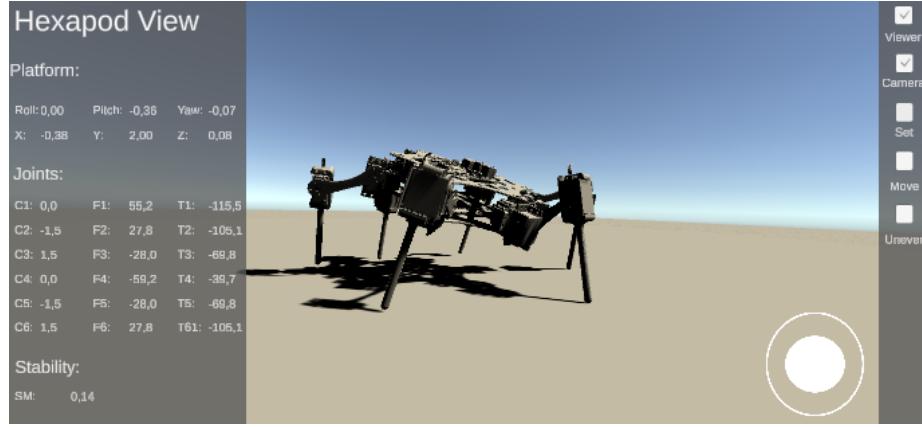


Figura 57: Cinemática inversa de la plataforma.

bot real se utilizarán estos mismos desarrollos modificados para funcionar en un robot real. Si bien el lenguaje de programación será distinto y el programa correrá en un sistema embebido, el funcionamiento del programa y la arquitectura de control serán, en esencia, los mismos. Esto se explicará con mayor detalle en la siguiente sección.

Una vez corroborado el funcionamiento del robot en un terreno liso, se procede a observar el comportamiento del robot en terrenos irregulares. Unity posee herramientas que facilitan la generación de este tipo de terrenos, siendo esta una de las razones por las que se optó por utilizar este *software*. Se diseña un terreno irregular utilizando en *Terrain toolbox* y se procede nuevamente a hacer caminar al robot sobre dicho terreno, como puede verse en la Figura 58.

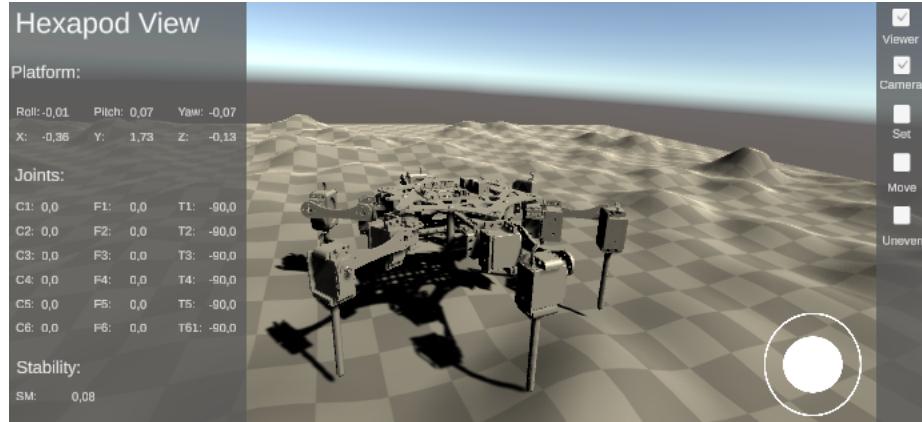


Figura 58: Robot en superficie irregular.

El hexápodo pudo desplazarse adecuadamente. Sin embargo, en ocasiones

en las que debe trasladarse o rotar sobre imperfecciones muy marcadas en el terreno, los movimientos que el robot debe realizar no son apropiados, perjudicando la estabilidad de paso y produciendo torques elevados en los actuadores. Dichos torques se deben al hecho de que, al no tener información sobre el terreno, las rutinas son las mismas cualquiera sea la superficie. Por lo tanto, en caso de que se presente una elevación en el terreno en alguna de las patas, la pata en consideración levantará la plataforma del robot produciendo una inclinación de la misma. Este comportamiento no deseado, a su vez, aumentará los torques necesarios en los actuadores del robot cuando este camine.

Por eso se optó por utilizar un *blind gate* que utiliza la detección de colisiones en las patas para obtener información del terreno donde se está circulando. Para ello se usan *colliders* en las puntas de las piernas para detectar las colisiones y actuar en función de éstas deteniendo las patas cuando éstas entran en contacto con el suelo. Se obtuvieron mejores resultados utilizando este tipo de pasos, mejorando la estabilidad y produciendo movimientos más suaves del centro de masa del robot. Aunque este tipo de paso puede ser implementado en un robot real utilizando sensores de contacto en las patas del robot, esto traía mayores complejidades en cuanto al diseño mecánico y a la programación, por lo que dicha implementación excede las capacidades de este proyecto y será implementada en trabajos futuros.

Además la aplicación en Unity permite al operador comunicarse con el robot mediante una conexión WiFi, la aplicación actúa como servidor y gestiona las conexiones, mientras que el hexápedo actúa como cliente. Una vez ingresado un comando dentro de la interfaz, la información es codificada en una cadena de caracteres de 20 bytes y enviada al robot para ser interpretada. El robot realizará los mismos movimientos que en el caso de la simulación, con algunos retardos debidos a las comunicaciones y a los tiempos de procesamiento del robot real. La interfaz permite al operador el control sencillo del robot a distancia, permitiéndole un gran control de los movimientos y parámetros de funcionamiento del robot.

## 12. Programación del robot

Una vez que se simula el comportamiento del robot y se verifica que los movimientos y desempeño de este son satisfactorios, se comienza a programar el robot real. La arquitectura del programa es similar a la arquitectura de la simulación. El código implementado corre en la Raspberry Pi 4 y tiene como objetivo recibir comandos por parte del operador y calcular las consignas de posición angular para el movimiento de las 18 articulaciones, posteriormente se envía al operador los datos relevantes de su operación. Como lenguaje de programación se utiliza Python 3.7. El programa corre en el sistema operativo GNU/Linux Raspberry Pi OS.

Se utilizan dos procesos, uno a cargo de la comunicación y el cálculo cinemático de los movimientos de las articulaciones, y otro destinado a la escritura de los comandos en el bus I2C, el cual se conecta a las PCA9685. El primero escribe

un arreglo el cual contiene los comandos a realizar, que luego es leído por el siguiente.

En un principio se intentó ejecutar el programa en un solo proceso, sin embargo los resultados no fueron satisfactorios. Se produjeron movimientos demasiado toscos de las articulaciones. Con el fin aumentar la fluidez de los desplazamientos, se optó por distribuir la ejecución del programa.

El proceso encargado de la escritura en el bus I2C es el más sencillo. Se utiliza la librería *adafruit\_servokit*<sup>6</sup> para mover los motores a través de las PCA9685. Se inicializan las comunicaciones I2C en las direcciones 0x40 y 0x41 para cada placa. Posteriormente se define las frecuencias máximas y mínimas de los motores para que puedan moverse entre 0 y 180° y se definen los offsets de las articulaciones (explicado con mayor detalle en la sección de Ensamblado). Una vez definidas estas variables, se ingresa en un *loop* que tiene como objetivo escribir los comandos de cada una de las articulaciones en el bus. Este está permanentemente leyendo y escribiendo dichos comandos y solo finaliza cuando se termina la ejecución del programa. Previo a la escritura en el bus se verifica que los comandos a escribir se encuentren en el rango de valores máximos y mínimos de cada articulación, en caso de no ser así no se escribe en el bus.

El proceso encargado del cálculo de los comandos corre 20 hilos distintos, un hilo principal que tiene como finalidad gestionar las comunicaciones, un hilo para el planificador de trayectorias del robot y 18 hilos encargados del cálculo de movimientos de cada una de las articulaciones. Habrá un hilo por cada articulación, los cuales comparten un objeto global en donde leen y escriben.

El hilo principal funciona de dos formas diferentes. El operador puede elegir que el robot reciba comandos por parte del servidor en Unity mediante una comunicación WiFi, actuando como cliente, recibiendo cadenas de caracteres para posteriormente interpretarlas y producir el movimiento requerido. Otra opción consiste en recibir comandos a partir de una interfaz gráfica que corre en la Raspberry Pi. Esta última fue programada utilizando la librería Tkinter y tiene como finalidad operaciones de prueba y debugging, si bien la interfaz puede ser fácilmente usada utilizando un servidor VNC. Dicha interfaz puede observarse en la Figura 59.

De la misma forma que la interfaz en Unity, ésta provee al operador la posibilidad de modificar la posición y orientación de la plataforma y establecer los valores de las 18 variables articulares. Además permite al operador dar consignas para que el robot camine en una cierta dirección, permite la rotación horaria (C) y antihoraria (CC) cierto ángulo de paso e ingresar comandos de posición relativa del robot para que se desplace en X y en Y una cierta distancia. A su vez, provee la capacidad de establecer parámetros tales como la velocidad (Vel), la distancia de paso (S\_D), la altura de paso (S\_H) y el paso de rotación (R\_S). Todos los valores relacionados con ángulos se encuentran en grados [°] mientras que los valores relacionados con desplazamientos lineales se encuentran en milímetros [mm].

El hilo relacionado con el planificador de trayectorias del robot tiene como

---

<sup>6</sup>[https://github.com/adafruit/Adafruit\\_CircuitPython\\_ServoKit](https://github.com/adafruit/Adafruit_CircuitPython_ServoKit)

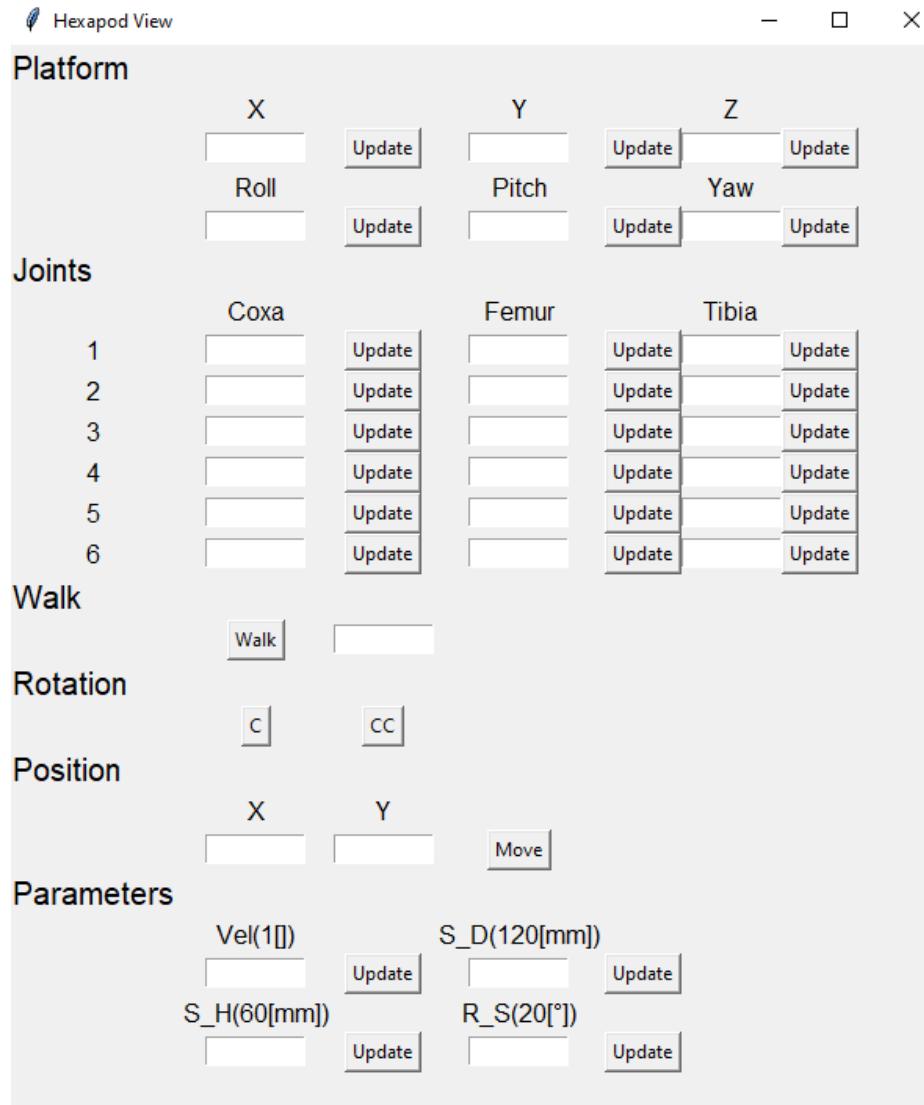


Figura 59: Interfaz gráfica programada usando Tkinter.

objetivo gestionar los movimientos cuando se ingresan consignas de posición a las cuales el robot debe desplazarse. Una vez que se ingresan consignas de posición, se calcula la dirección de desplazamiento y la cantidad de pasos a realizar. El hilo tiene como objetivo verificar si quedan pasos a realizar para satisfacer las consignas. En caso de que queden pasos a realizar, el hilo dará los comandos necesarios para que el robot continúe su caminata. Se podría utilizar este mismo hilo con el objetivo de recibir comandos de rotación absoluta del robot por parte

del operador y definir la cantidad de pasos de rotación a realizar. Sin embargo esta funcionalidad no ha sido implementada y será realizada en trabajos futuros.

Los hilos de cada articulación tienen como objetivo el cálculo de las posiciones deseadas de éstas en el tiempo. Se utilizan hilos con el fin de asegurar que el programa funcione de forma concurrente. Cada uno de ellos funciona como un *timer* ejecutando una acción cada un periodo de tiempo definido por el operador. En cada instante se realiza una interpolación lineal entre una posición angular inicial y una final. Estos valores son obtenidos a partir del generador de trayectorias de las piernas.

Al ser ingresados los comandos por parte del operador, el funcionamiento del programa dependerá del tipo de comando ingresado. Si el operador detalla la posición en X e Y de consigna del robot, entonces se pasa por el generador de trayectoria del robot que calcula la cantidad de pasos a realizar y posteriormente estos pasos son llevados al generador de trayectorias de las patas que definen los movimientos angulares que deben realizar las articulaciones en el tiempo. En caso que se le introduzcan consignas de posición angular de las articulaciones, consignas de rotación, caminata, posición u orientación de la plataforma, el programa utiliza únicamente el generador de trayectorias de las patas para dar consignas a las articulaciones. Si al ingresar comandos de traslación del robot la plataforma se encuentra con alguna inclinación, el hexápodo vuelve al paralelismo con el suelo e inicia la rutina para trasladarse.

El generador de trayectorias de las patas calcula las posiciones cartesianas que deben seguir cada una de las piernas del robot para que éste pueda realizar los movimientos requeridos, luego calcula las posiciones angulares de cada articulación a partir de la cinemática inversa y por último establece un arreglo de posiciones angulares de manera tal de que éstas puedan ser interpoladas linealmente por los *timers* asociados a cada articulación. Una vez que se obtienen los valores, éstos son escritos en el arreglo compartido entre procesos para luego dar consigna a los motores.

La estructura del programa puede observarse en la Figura 60.

El *main.py* inicializa los hilos relacionados con el planificador de trayectoria del robot, los hilos asociados a las articulaciones, el proceso para la escritura en los motores y ejecuta la interfaz gráfica o las comunicaciones WiFi según sea el caso. En *GUI.py* se almacenan todas las funciones relacionadas con la interfaz gráfica, mientras que en *wifi.py* se almacenan las funciones relacionadas con las comunicaciones WiFi.

El *servo\_process.py* gestiona la escritura de comandos en el bus I2C de los motores. En *joint\_timer.py* se hacen los cálculos relacionados con el movimiento de las articulaciones, utilizando las funciones de *robot\_movement.py* y *robot\_math.py* que definen las funciones relacionadas con el planificador de trayectoria de las patas y las funciones relacionadas con la cinemática inversa del robot respectivamente.

El robot puede alcanzar cualquier posición en el espacio utilizando cualquiera de las interfaces gráficas planteadas. Para ello primero debe establecerse la posición en X e Y del robot. Una vez dadas las consignas, el robot se traslada a dicha posición. Si se desea que el robot rote en dicho punto, puede utilizarse los

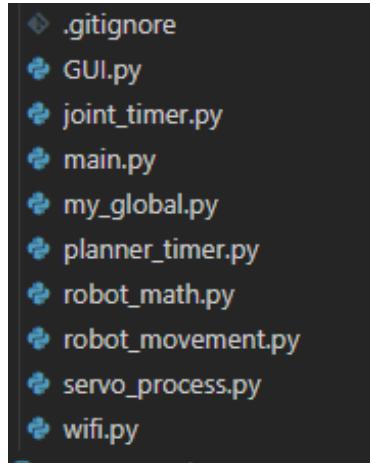


Figura 60: Ficheros del programa en Python.

botones de rotación que permiten al robot movilizarse un cierto ángulo. Si se requieren ángulos precisos, el ángulo de paso puede ser establecido en la GUI. La rotación debe realizarse de forma manual presionando los botones tantas veces como sea necesario para obtener el valor deseado. Una vez que el robot se encuentra en la posición y ángulo deseado, puede establecerse la posición y orientación de la plataforma a través de la interfaz. Además pueden establecerse las posiciones de cualquiera de las 18 variables articulares. Permitiendo de esta forma que el robot pueda alcanzar cualquier posición que se desee.

Una vez verificado el correcto funcionamiento del programa se procedió a continuar con las comunicaciones del robot.

## 13. Comunicaciones

La teleoperación de robots o telerobótica se refiere al control de robots a distancia. La telerobótica implica una barrera entre el usuario y el robot [25]. Esta barrera puede ser solucionada mediante control de forma remota. Este tipo de sistemas generalmente está dividido en dos sitios, un sitio local donde se encuentra el operador humano y todos los elementos necesarios para realizar el mando y las comunicaciones, y un sitio remoto donde se encuentra el robot, los sensores y los elementos de control necesarios para realizar los movimientos deseados. Las razones por las cuales en ocasiones es preferible optar por este tipo de sistemas pueden ser de distinta naturaleza. Éstas pueden ser financieras, humanas o técnicas y muchas veces las aplicaciones de teleoperación pueden ser la única solución apropiada a los problemas propuestos. La presencia de un operador humano hace que la teleoperación de robots sea utilizada en entornos desconocidos y no estructurados.

Existen distintas opciones que permiten la comunicación inalámbrica del ro-

bot, cada una con sus ventajas y limitaciones. Dentro de ellas se encuentra el Bluetooth (IEEE 802.15.4). Si bien la latencia es adecuada para realizar las operaciones necesarias, este tipo de comunicación está limitada en cuanto al alcance de la señal [26], por lo que se descartó esta opción. Otra posibilidad consiste en utilizar radiofrecuencia (ZigBee, Lora, LoraWan, etc.). Este tipo de tecnologías generalmente permiten alcances más largos y menores consumos, pero su latencia se ve aumentada y los costos del hardware son elevados. Es por ello que se optó por utilizar Wifi (IEEE 802.11), usando sockets para comunicar ambas aplicaciones. Si bien el consumo energético es mayor, las latencias son adecuadas para permitir la transferencia de información y las distancias de comunicación son mayores. Además la Raspberry Pi 4 y el ordenador en el cual se ejecuta la aplicación en Unity ya poseen las interfaces necesarias para su utilización, por lo que no se recurre a costos adicionales en hardware.

El programa del robot ha sido realizado con la idea de modularizar lo más posible las comunicaciones del robot de manera de poder intercambiar su tipo, sin mayores problemas, en caso de ser necesario. Por lo que, si bien por las razones previamente mencionadas se recurrió a utilizar Wi-Fi, el tipo de comunicación puede ser fácilmente modificado para que se adapte a la tarea del robot, de manera de que esta sea la más apropiada para la aplicación a realizar.

La aplicación en Unity actúa como servidor gestionando las conexiones. El robot actúa como cliente y se conecta al servidor para comenzar las comunicaciones. El programa que gestiona las conexiones en el servidor corre en un hilo aparte para permitir la ejecución de la aplicación aun cuando ningún cliente se encuentra conectado. La transferencia de datos se realiza a partir de cadenas de 20 caracteres codificados en el servidor y posteriormente interpretados por el cliente mediante un intérprete de comandos. Se utilizan caracteres para determinar el inicio y el fin de la cadena, y definir el tipo de comando o variable a modificar. Las cadenas son de la forma:

:  $Xvvv, vvv ======$

Se comienza con : determinando el inicio de la trama. El segundo carácter  $X$  determina el tipo de comando (R: Rotación, C: Traslación cartesiana, J: posición angular de las articulaciones, P: posición y orientación de la plataforma, W: caminata en el ángulo determinado por el joystick, V: Velocidad, H: Altura de paso, D: Distancia de paso, S: Paso de rotación). Posteriormente la trama contiene los valores a enviar  $vvv$  separados por comas, estos valores son números enteros. La cantidad de caracteres depende del tipo de comando. Por ejemplo, en caso de que se quiera dar comando de posición del robot, solo es necesario enviar la posición en X e Y separados por comas. Mientras que si se quiere enviar valores de posición y orientación de la plataforma es necesario enviar valores de posición ( $x, y, z$ ) y orientación (roll, pitch, yaw) separados, también, por comas. La cadena se rellena con signos = hasta cumplir los 20 caracteres. El carácter = se considera el fin de trama y toda la información posterior es ignorada hasta leer nuevamente el símbolo de inicio de trama.

Las cadenas son interpretadas por el intérprete de comandos para dar consignas al hexápodo. Este se ejecuta en el hilo principal del robot y es llamado

cada vez que se recibe información por parte del servidor.

## 14. Fabricación, ensamblaje y puesta en marcha

### 14.1. Fabricación

Para llevar a cabo la construcción de la gran mayoría de las partes mecánicas que conforman el robot, se recurre a la impresión 3D utilizando PLA (Ácido poliláctico).

El proceso de fabricación de las piezas parte del diseño mecánico el cual se realiza en el software CAD Solid Edge. Estas piezas diseñadas son exportadas al *slicer* REALvision<sup>7</sup>, en el cual se configuran las preferencias de cada pieza antes de imprimir. Siendo las más relevantes:

- Altura de capa: expresa el espesor en milímetros que posee cada capa de impresión a medida. Se utilizó 0.2 mm para todas las piezas a excepción de la carcasa la cual fue impresa en 0.15 mm.
- Geometría de filamento interno: Dependiendo la geometría utilizada, las propiedades mecánicas como la inercia, la tensión de rotura, resistencia a la flexión puede variar. Los más comunes son lineal, cruzado y hexagonal. Pero existen otros más complejos que adquieren formas tridimensionales, como 3D Gyroid, Double Gyroid, 3D Isomax, entre otros. Particularmente en este proyecto la gran mayoría de las piezas son delgadas, por lo que solo se recurrió a utilizar filamento interno cruzado en la coxa.
- Contornos: El número de contornos, el cual le da mayor rigidez a la pieza a medida que aumenta su cantidad. Se utilizaron 3 contornos en todas las piezas.
- Capas inferior y superior: Es el número de capas sólidas de la base y de la parte superior de la pieza. En el software utilizado, se mide en milímetros de espesor. En todas las piezas se utilizó tanto 1 milímetro en la base como en la superficie superior.
- Temperatura de la cabeza del extrusor: Depende del material con el que se imprime, normalmente si se utiliza PLA se trabaja en un rango de 190-220 °C.
- Temperatura de la cama: Temperatura que va a adquirir la cama sobre la cual se va a imprimir la pieza.

Esto se observa en la Figura 61.

Luego se convierte la pieza en Código G, el cual le otorga las coordenadas y configuraciones necesarias para que la impresora 3D realice el proceso.

---

<sup>7</sup><https://www.createitreal.com/3d-printing-slicing-software-realvision/>

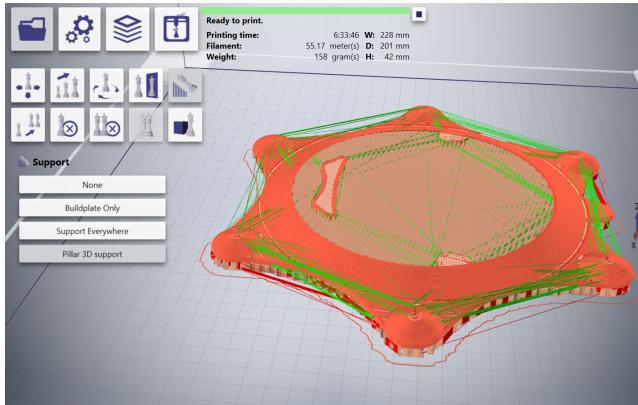


Figura 61: Rebanadas de la carcasa en REALvision.

En el proceso de fabricación se utilizan diversas impresoras debido a que algunas piezas del robot poseen mayor volumen, por lo tanto requieren de impresoras con un volumen de trabajo acorde. Estas son las impresoras utilizadas:

- Weistek IdeaWerk Speed X: para los eslabones de las patas y soportes de la plataforma.
- Creality CR-10: las tres placas de la base han sido impresas en esta impresora.
- Raise3D E2: para la carcasa.

## 14.2. Ensamblaje

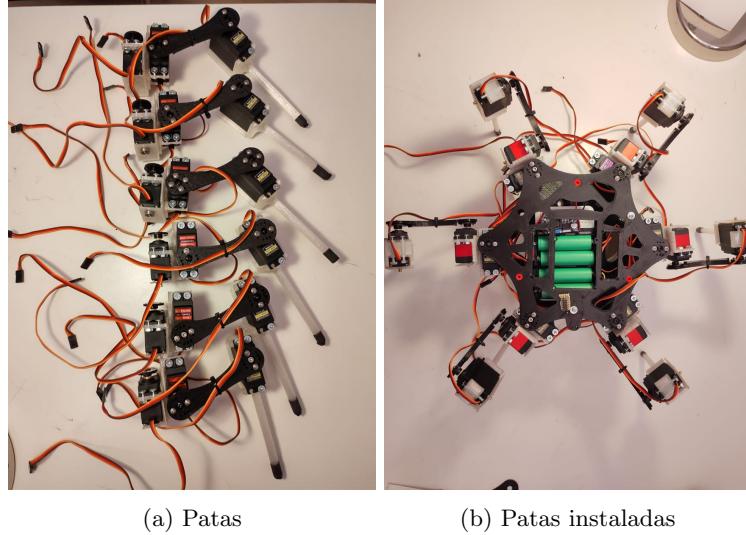
Una vez definido el diseño mecánico, impresas las piezas, adquiridos los actuadores y ya programadas las rutinas de movimiento del robot, se procede al ensamblado.

### 14.2.1. Verificación de motores

Este paso consiste en el ensayo de los 18 motores en el cual se corroboran los anchos de pulso necesarios para que el motor pueda moverse hacia sus posiciones extremas (0 - 180°).

### 14.2.2. Vinculación de base, eslabones y motores

Luego se fijan los motores en los eslabones de manera tal que se permitan los movimientos deseados del robot, teniendo en cuenta que los motores pueden moverse en un rango angular restringido. Las patas del robot pueden observarse en la Figura 62.



(a) Patas

(b) Patas instaladas

Figura 62: Vinculación de patas.

#### 14.2.3. Definición de los offsets

Posteriormente se deben establecer los offsets de cada motor. Se comandan los motores desde la Raspberry Pi 4 para establecer los ceros de cada uno de ellos. De modo que los ceros de la articulación coincidan con los ceros establecidos en la etapa de diseño.

Tanto en la definición del los offset como en la verificación de los motores se utilizó una fuente de alimentación de 5 V con un límite de 22 A.

### 14.3. Puesta en marcha

En esta etapa se realizan pruebas, en las cuales se aumenta la exigencia de los motores de forma gradual.

Esto implica las siguientes variables:

- Contacto con el suelo: cuando los extremos de las patas entran en contacto con el suelo u otro objeto, por fuerzas de acción y reacción, generan un aumento en el torque requerido en los motores.
- Fuente de alimentación
  - Fuente externa de tensión constante 5 V hasta 22 A.
  - Baterías Li-Ion 18650, entre 2 y 4 de estas.
  - Banco de energía portátil 5 V, 1-2,4 A

- Carga: La inercia del robot varía considerablemente si el robot carga con el peso de las baterías, provocando un aumento de la demanda corriente en los motores dado que se requiere mayor torque para mantener la estructura.

#### **14.3.1. Pruebas en vacío**

- Sin contacto con el suelo.
- Fuente externa de 5 V.
- Carga de las placas que se considera despreciable en comparación con las baterías y la estructura del hexápedo.

Se imprimió una base soporte con el fin de suspender el robot y así poder probar el movimiento de las patas. Esto puede observarse en la Figura 63.

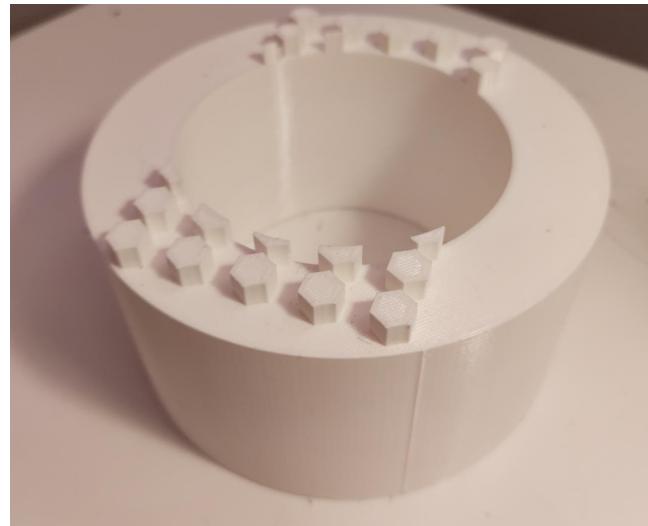


Figura 63: Base soporte para pruebas en vacío.

Se testea pata por pata en todo el espacio de trabajo, se evita el contacto con el suelo para así cerciorarse de que cada pata tiene un comportamiento adecuado.

Se mide la corriente consumida, durante el movimiento de cada una de las patas la cual no superaba los 200 mA.

#### **14.3.2. Pruebas en el suelo**

- Con contacto con el suelo.
- Fuente externa de 5 V y luego 2 a 4 baterías 18650.

- Carga de las placas y la estructura del hexápodo.

Habiendo probado exitosamente las rutinas con el robot en el aire, se procede a colocar el robot en el suelo, primero apoyando tres patas desfasadas 120 grados entre sí de forma estática, tomando valores entre 550 mA y 650 mA.

Luego se procede a colocar las otras 3 patas aumentando el consumo de corriente entre 800 mA y 900 mA. A continuación se realizan movimientos en los ejes de roll, pitch y yaw, en los cuales se obtienen valores de corriente entre 850 mA y 1,5 A.

En las pruebas de rotación horario y anti-horario los valores de corriente son similares a los de roll, pitch y yaw.

Finalmente se realiza la prueba de caminata en la cual los valores de corriente oscilaron entre 1,2 A y 2,5 A.

Se prueban las rutinas programadas y se observa el comportamiento de las articulaciones ante los esfuerzos mecánicos debidos al mismo peso del robot. Se verifica la rigidez de las articulaciones y se observa deformaciones despreciables en los eslabones del robot. El robot ensamblado puede visualizarse en la Figura 64. Es necesario remarcar que en caso de que se coloque una mayor cantidad de carga sobre el robot, se deberán esperar mayores corrientes.

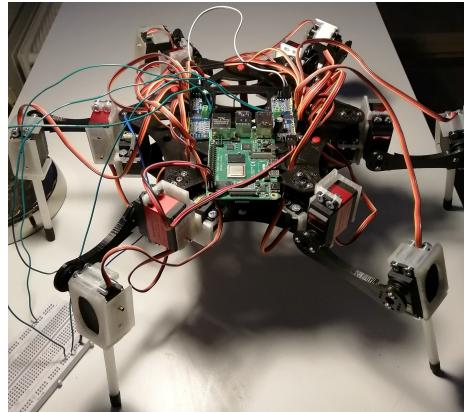


Figura 64: Modelo que fue sometido a pruebas en el suelo.

## 14.4. Iteraciones

La estructura del hexápodo diseñada inicialmente fue ligeramente modificada en lo que respecta a la base y a las patas. Debido a diversos factores que serán mencionados a continuación.

### 14.4.1. Base

Esta modificación surge de la complejidad a la hora de acceder a la parte central del cuerpo del robot. En la primera iteración, el acceso a la base estaba

restringido por los soportes y los motores de la primera articulación de cada pata. Inicialmente para las pruebas en las que el robot era energizado por una fuente externa, no se generó inconveniente alguno. Luego se incorporaron las baterías, las cuales requieren de recarga frecuente. Por lo que se forzaba la extracción de una pata y un soporte cada vez que se requería retirar alguna de ellas. La configuración de la base antigua puede observarse en la Figura 65.

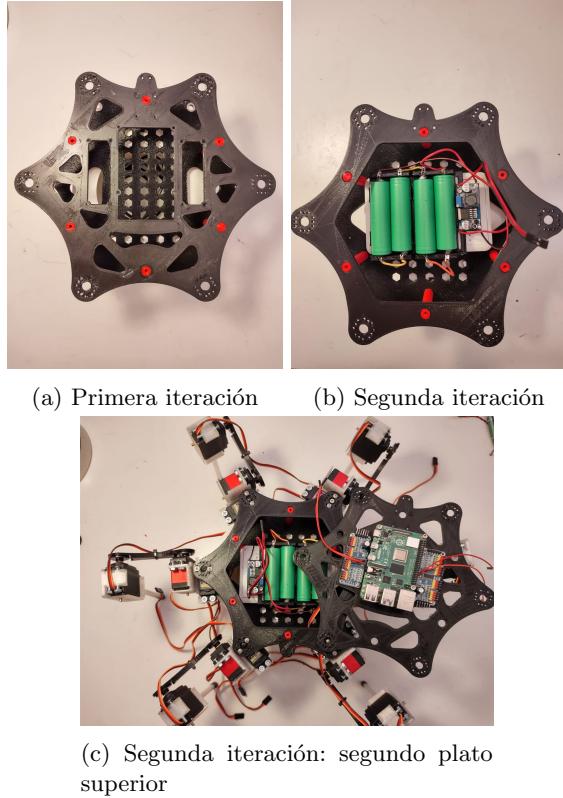


Figura 65: Iteraciones de la base.

Esto fue motivación suficiente para rediseñar la base, pudiendo así acceder al centro de ésta desde la parte superior sin necesidad alguna de retirar motores o soportes.

La solución consistió en independizar el plato superior de la base que estaba vinculada tanto a los motores como a las placas Raspberry Pi y PCA9685. Por lo que se imprimió un nuevo plato hueco, el cual está vinculado a las patas y permite el acceso al cuerpo. El resultado final puede observarse en Figura 66.

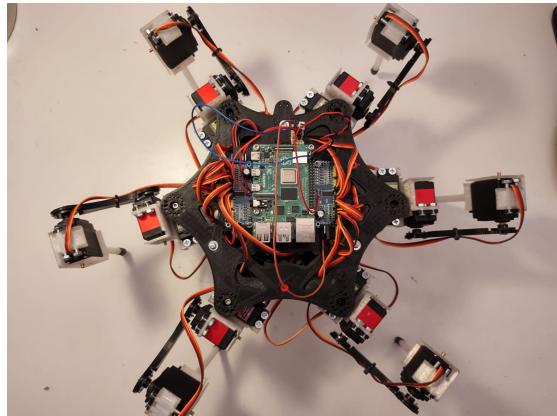


Figura 66: Cuerpo cerrado por el segundo plato superior.

#### 14.4.2. Patas

En lo que respecta a las patas, el primer cambio que se realizó fue sobre el fémur. Este eslabón se ve sometido a torsión, la solución inmediata, fue duplicar la sección de la pieza que se veía afectada, otorgando la rigidez necesaria para contrarrestar este efecto no deseado.

Como segunda instancia se adhirió un recubrimiento en las puntas las cuales le proporcionaban un mayor agarre en superficies de bajo coeficiente de rugosidad.

Finalmente se le adhirió una compensación paralela a este eslabón como se puede observar en la Figura 67.



(a) Iteración 1

(b) Iteración 2

(c) Iteración 3

Figura 67: Iteraciones de las patas.

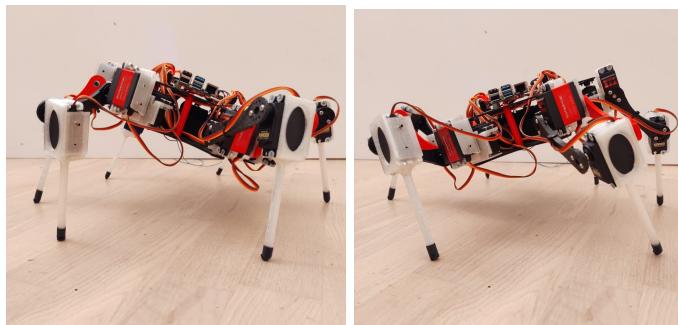
## 15. Resultados

Una vez realizado el ensamblaje del robot, se procede a evaluar el desempeño éste. Esto se realiza visualizando su comportamiento en cada una de sus rutinas de funcionamiento. La operación del robot fue realizada de forma remota mediante la interfaz en Unity, y se ensayó la capacidad del hexápedo de trasladarse, rotar y variar la orientación y la posición de su base. Además, se verificó la capacidad del operador de variar los parámetros de funcionamiento. Se probó el comportamiento a distintas velocidades y con distintos pasos. Se comparó el desempeño del robot en contraste con la simulación realizada, que representa un caso de funcionamiento ideal. El robot se comportó de forma apropiada, pudiendo desplazarse y rotar sin problemas en el suelo. El desempeño del robot se vio influenciado por la fricción del suelo, pudiendo producirse algunos resbalamientos cuando el robot se mueve a mayores velocidades. Dicho problema podría solucionarse utilizando en las patas del robot materiales con mayor coeficiente de fricción o reduciendo la velocidad del robot.

### 15.1. Orientación y posición estáticas

#### 15.1.1. Rotación: Roll, Pitch & Yaw

A través de técnicas de comparación de imágenes se evaluó la orientación angular de la base. El cumplimiento de los ángulos de roll, pitch y yaw pueden observarse en la Figura 68, Figura 69 y Figura 70 respectivamente.

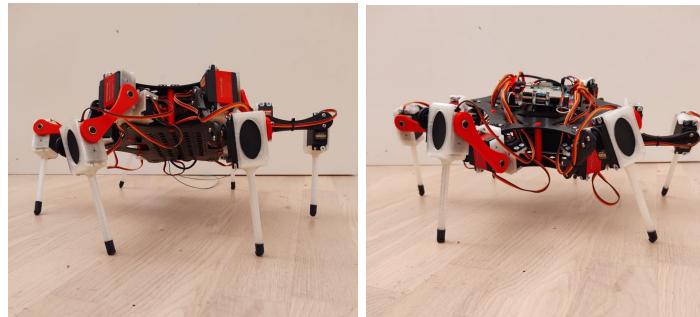


(a) Ángulo de roll  $20^\circ$

(b) Ángulo de roll  $-20^\circ$

Figura 68: Ángulo de roll.

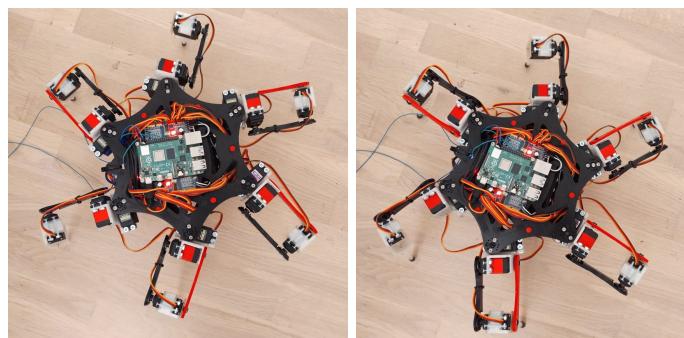
El hexápedo orientó la base de forma correcta en cada uno de los ángulos de consigna. Posteriormente se probó con combinaciones de ángulos de roll, pitch y yaw. Esto puede observarse en la Figura 71, se dan consignas de  $10^\circ$  en cada uno de los ángulos.



(a) Ángulo de pitch  $20^\circ$

(b) Ángulo de pitch  $-20^\circ$

Figura 69: Ángulo de pitch.



(a) Ángulo de yaw  $25^\circ$

(b) Ángulo de yaw  $-25^\circ$

Figura 70: Ángulo de yaw.

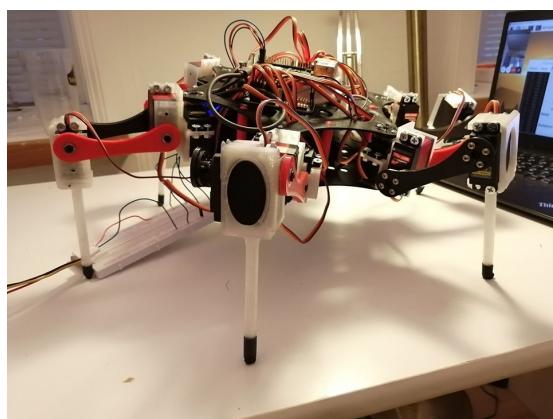
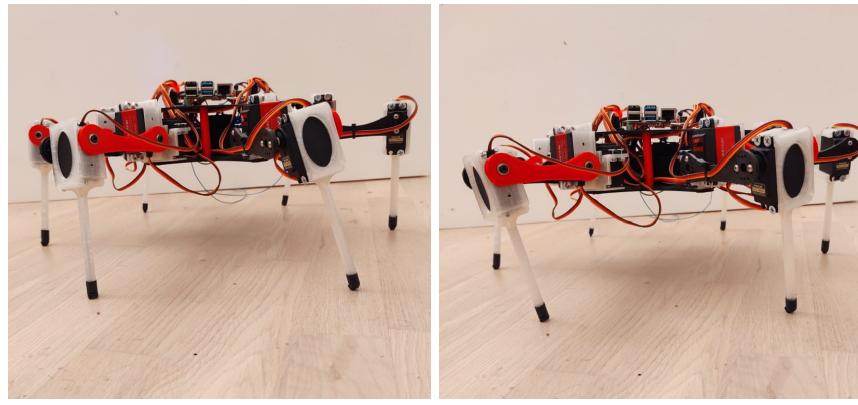


Figura 71: Combinación roll, pitch y yaw ( $10^\circ$  en cada eje).

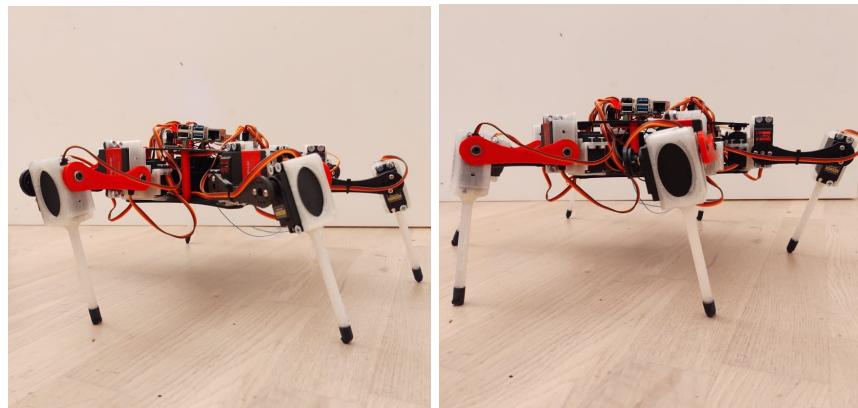
### 15.1.2. Posición: X, Y & Z

El desplazamiento de la base en los 3 ejes de translación X, Y e Z fue óptimo. Como se pueden observar en las Figuras 72, 73 y 74.



(a) Desplazamiento a 20mm (b) Desplazamiento a -20 mm

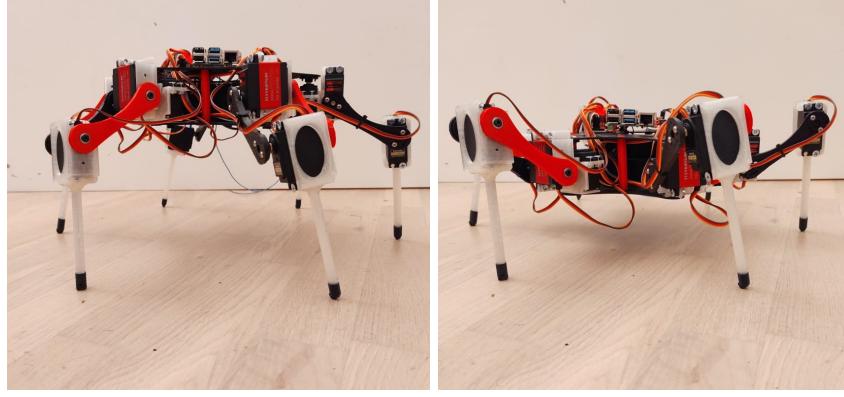
Figura 72: Desplazamiento de base en X.



(a) Desplazamiento a 20 mm (b) Desplazamiento a -20 mm

Figura 73: Desplazamiento de base en Y.

Tanto en X como en Y, se establecieron +/-20 mm como los límites de desplazamiento de la plataforma y en Z +/-30 mm. Es posible extender la plataforma en esos ejes con mayor amplitud, pero por cuestiones de seguridad se tomaron esos valores como límites.



(a) Desplazamiento a 30 mm

(b) Desplazamiento a -30 mm

Figura 74: Desplazamiento de base en Z.

## 15.2. Desplazamientos del robot

Finalizado el análisis de las posiciones estáticas, se procede a evaluar los desplazamientos básicos del robot, traslación en superficie plana y rotación en superficie plana. A continuación se muestran los resultados obtenidos para ambos movimientos.

### 15.2.1. Traslación

En la prueba de traslación se realizaron desplazamientos con distintas velocidades y pasos en los cuales se pudo observar el correcto desplazamiento del robot aun en superficies de baja fricción. Como se mencionó anteriormente, hubo problemas de resbalamientos cuando el robot se desplazó a altas velocidades por el suelo (3 veces la velocidad por defecto, siendo la velocidad por defecto 4 segundos por paso). Sin embargo, a menos velocidad el robot se movió sin casi resbalamientos de las patas. El problema podría solucionarse utilizando materiales de mayor fricción en extremo del efecto final.

El ciclo de desplazamiento del robot puede verse en la Figura 75. Pueden observarse cuatro instantes del ciclo de desplazamiento del robot.

Todos los movimientos propuestos desde un principio fueron realizados de forma exitosa. De esta forma se validan los trabajos realizados desde el punto de vista mecánico, desde el punto de vista de la programación y de la electrónica.

### 15.2.2. Rotación

Una vez verificada la traslación del hexápedo, se procede a evaluar el comportamiento de la rotación. Para ello se configura al robot para que gire un ángulo de  $30^\circ$  y se hace rotar en sentido antihorario. El resultado puede observarse en la Figura 76. En el instante 1, el robot se ubica en la posición inicial,

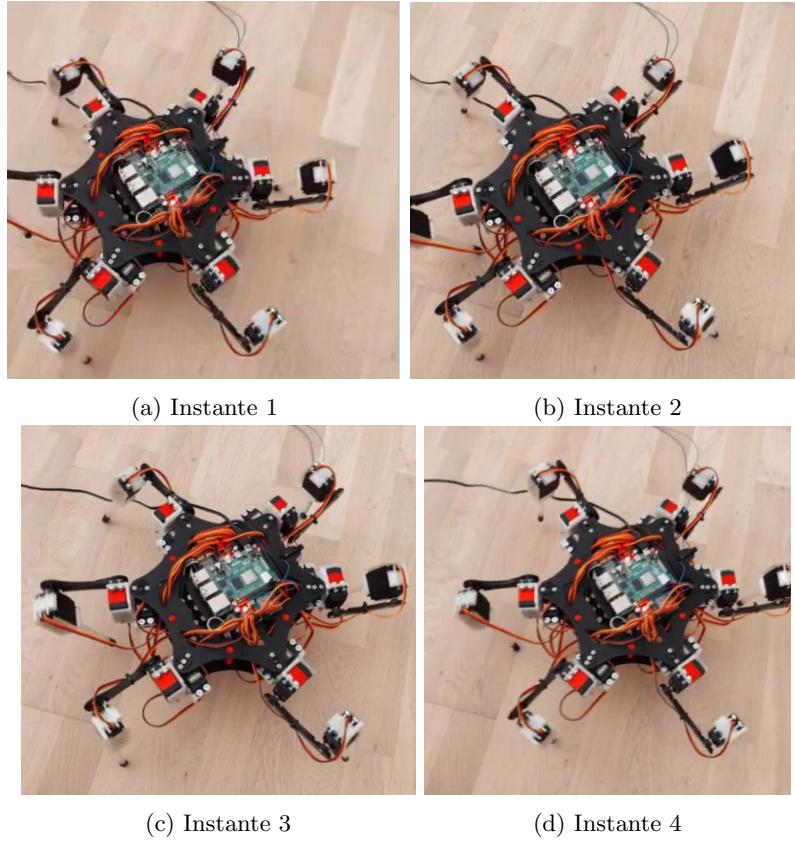


Figura 75: Desplazamiento del hexápodo.

se comienza con la rutina de rotación. En el instante 2, tres de las patas se encuentran en fase de impulsión mientras que las otras tres se encuentran en fase de apoyo. Las fases se invierten en el instante 3. El robot sigue rotando hasta el instante 4, en donde cesa su rotación ya habiendo girado los  $30^\circ$  de consigna.

Se obtuvieron problemas similares a los obtenidos durante la traslación del robot. A altas velocidades se produjeron resbalamientos considerables. Sin embargo, se obtuvieron buenos resultados a velocidades más bajas, validando los algoritmos desarrollados para la rotación del robot.

Una vez corroborado el correcto funcionamiento del robot en sus distintos movimientos básicos, se procede a agregar las baterías necesarias para su alimentación y la carcasa protectora que cubre la electrónica del hexápodo. El diseño impreso final del robot puede observarse en la Figura 77.

De esta forma se finaliza el desarrollo del robot hexápodo planteado.

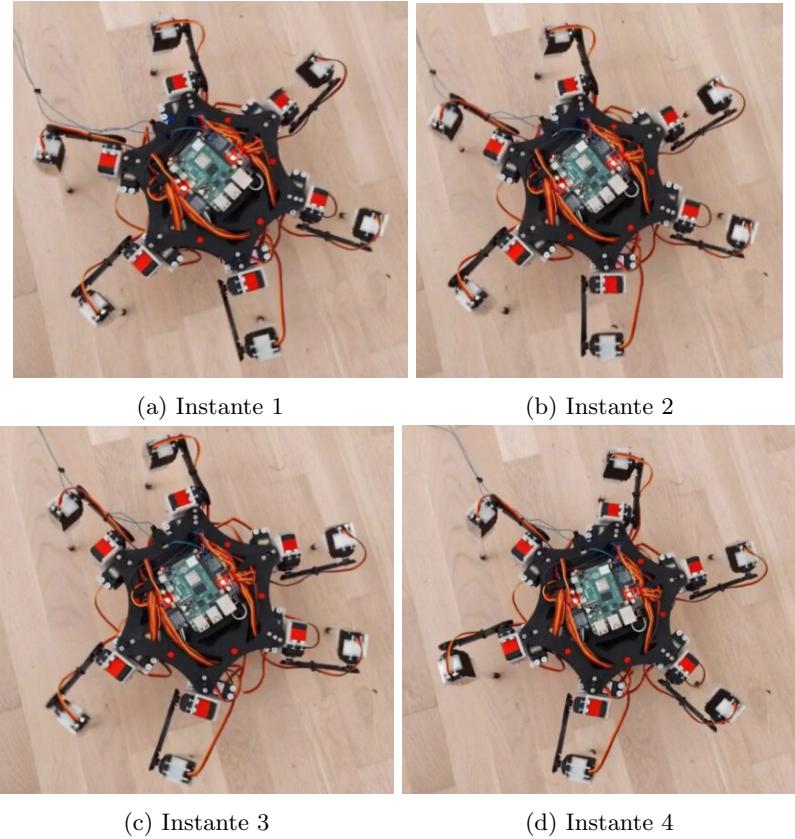


Figura 76: Rotación del hexápodo.



Figura 77: Diseño final del robot.

## 16. Recursos implicados

Los recursos implicados están relacionados con el tiempo utilizado para la desarrollo del proyecto y con el precio de los componentes que conforman el robot.

En cuanto al tiempo utilizado, el cronograma de actividades puede verse en la Figura 78. En dicha figura puede observarse el estimado de tiempo en semanas que fue empleado en cada actividad. La totalidad del proyecto se desarrolló en 16 semanas aproximadamente.

Actividades	Tiempo en semanas															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Informe																
Investigación sobre los fundamentos del robot																
Desarrollo de los modelos cinemático y dinámico																
Diseño mecánico																
Selección del hardware																
Arquitectura de control																
Prueba de los actuadores																
Creación de la interfaz de simulación																
Programación del robot																
Desarrollo de las comunicaciones																
Construcción del robot																
Testeo, puesta en marcha y conclusiones																
Presentación																

Figura 78: Cronograma de actividades.

En cuanto a los materiales usados, estos pueden observarse en la Figura 79. Allí se encuentran listados los componentes utilizados para la construcción del robot, su cantidad, así como su precio en euros y pesos argentinos según las cotizaciones del día 21 de septiembre de 2020.

Es necesario remarcar que el precio de las impresiones de las piezas son estimaciones realizadas por la empresa especializada en modelado e impresiones 3D *Create It Real*<sup>8</sup>.

## 17. Conclusión

A partir de los resultados obtenidos durante el desarrollo del proyecto, se ha podido visualizar un buen comportamiento del robot hexápedo propuesto. Se

<sup>8</sup><https://www.createitreal.com/>

Item#	Descripción	Cantidad	Precio(Euros)	Precio(AR\$)
1	RaspBerry Pi 4	1	75	6669.37
2	PCA9685	2	17.29	1537.51
3	MG996r(ServoMotor)	14	50.39	4480.93
4	TD-8120MG(ServoMotor)	6	30.52	2713.99
5	Baterias	4	30.92	2749.56
6	Bulonería(x40Tornillos)	5	20.13	1790.06
7	Rodamientos	10	12.08	1074.21
8	Coxa	6	73.16	6505.75
9	Femur + compensación	6	63.54	5650.29
10	Tibia	6	52.22	4643.66
11	Base	1	150	13338.74
12	Carcaza	1	195.67	17399.95
13	Envío		17.63	1567.75
14	Impuestos		29.1	2587.72
Total			817.65	72709.49

Figura 79: Precio de los recursos utilizados.

ha podido, además, observar las limitaciones y los distintos puntos de mejora que podrían ser implementados en trabajos futuros. Algunas de las conclusiones obtenidas se muestran a continuación.

Durante la ejecución del proyecto se ha realizado exitosamente el análisis de un robot hexápedo de 18 articulaciones. Se ha estudiado la estructura y principio de funcionamiento de este tipo de robot y se han implementado los modelos matemáticos necesarios para poder realizar los movimientos deseados. Se hizo el desarrollo de la cinemática directa e inversa de las patas del robot. Se implementó la cinemática inversa de la plataforma desarrollada en [5], adaptándola a los sistemas de referencia utilizados. Se desarrollaron las ecuaciones matemáticas para generar los movimientos de rotación y traslación del robot. Se implementó el cálculo de la estabilidad y se hizo el desarrollo de la dinámica de las patas del hexápedo. Los modelos matemáticos fueron validados a través de una simulación. Ésta permitió visualizar el comportamiento del robot en un entorno virtual, permitiendo evaluar la arquitectura de control propuesta y los algoritmos para el movimiento previo a la construcción del robot real. Se investigaron los sensores y actuadores necesarios para el funcionamiento del robot y se seleccionaron aquellos que técnica y económicamente fueron más apropiados para la aplicación en cuestión. Se investigó y se definió el tipo de comunicación a realizar. Se realizó un diseño mecánico del robot mediante un software de modelado 3D y se imprimieron las partes. Una vez construido el robot se evaluó el desempeño de éste.

A partir de los resultados obtenidos pudo observarse un buen comportamiento del hexápedo. Los modelos matemáticos fueron validados exitosamente y permitieron movimientos precisos de las articulaciones. En cuanto a la progra-

mación del robot, se desarrollaron algoritmos flexibles y fáciles de modificar que permitirían generar movimientos complejos con poco trabajo adicional. Se desarrolló una simulación para permitir al operador verificar los algoritmos, previo a su aplicación del robot real, dicha simulación funcionó eficientemente y provee una idea del desempeño idealizado del robot. El usuario tiene una interfaz gráfica donde puede cambiar muchos parámetros relevantes para el funcionamiento del robot, así como definir fácilmente los movimientos del hexápodo. Se le da al operador la posibilidad de operar el robot de forma remota mediante una comunicación Wifi. En caso de que se deseé utilizar otro medio para realizar las comunicaciones, el programa es fácilmente modificable. Desde el punto de vista mecánico, el diseño permitió al robot desplazarse sobre distintos terrenos sin presentar deformaciones importantes en los eslabones. Los actuadores fueron adecuados y permitieron una gran flexibilidad en los desplazamientos del robot aun en posiciones desfavorables de los actuadores.

## 18. Trabajo futuro

Si bien el robot se comportó adecuadamente y los resultados fueron positivos, hay muchas posibles mejoras que pueden implementarse para aumentar el desempeño del robot. Estas mejoras van desde el desarrollo del modelo matemático, así como desde el punto de vista del hardware y del software.

Algo que podría mejorar el desempeño del robot sería cambiar el lenguaje de programación por otro lenguaje que provea una mayor rapidez a la hora de realizar los cálculos y manipular las *GPIO* de la Raspberry Pi. Las rutinas del hexápodo fueron realizadas utilizando Python 3.7 y produjeron buenos resultados, sin embargo, se observan retardos en los movimientos de las articulaciones. Estos producen pequeñas diferencias entre los tiempos de simulación y de movimiento real del robot. Dichas diferencias podrían tal vez minimizarse utilizando un lenguaje de programación compilado en vez de interpretado, como puede ser C o C++. Estos lenguajes de más bajo nivel presentan una mayor velocidad a la hora de manipular las entradas y salidas de la placa y permitirían obtener rutinas más eficientes a costa de un mayor esfuerzo de programación. Sin embargo, dado que no se han medido los retardos debidos a las comunicaciones y a la mecánica del robot, no es posible asegurar que un cambio en el lenguaje de programación pueda producir una mejora apreciable en el funcionamiento del robot.

Otro problema consiste en la fluidez de los movimientos de las articulaciones, pudo observarse que a velocidades reducidas las articulaciones realizan movimientos toscos. A más velocidad se obtienen movimientos más fluidos, este problema se debe a dos razones principales, los tiempos de escritura en los motores y la precisión de éstos. El problema podría solucionarse utilizando actuadores de mayor resolución. Otra solución consiste en cambiar el hardware para la escritura en los motores por uno más eficiente o realizando una escritura distribuida. Además, para mejorar la suavidad de los movimientos podrían utilizarse trayectorias más redondeadas de las piernas, reduciendo las aceleraciones

angulares máximas que deben imprimirse en cada una de las articulaciones.

En cuanto a los movimientos del robot en terrenos irregulares, se podría mejorar el desempeño del robot implementando un *blind gate*. Para ello se requeriría modificar el diseño de las patas para poder ubicar sensores de contacto en las puntas de éstas. Además se debería modificar el programa para tener en cuenta dicha información en el comportamiento del robot. En simulaciones se demuestra que esta implementación permitiría una mejora apreciable del funcionamiento del robot en este tipo de terrenos.

El robot podría tener imprecisiones en cuanto al desplazamiento debido a imperfecciones mecánicas, resbalamientos y descalibración de los actuadores. Por lo que en desplazamientos relativamente grandes, la diferencia entre la posición deseada y la real puede ser apreciable. Es por ello que se podría utilizar un controlador como el ya desarrollado en [24] para verificar la diferencia entre el valor deseado y el valor real y definir la cantidad, dirección y longitud de paso en función de dicha diferencia. Para ello se requeriría utilizar un sensor que provea la posición absoluta del robot.

Otro punto de mejora consiste en utilizar sensores que permitan obtener más información del entorno donde circula el robot y de su estado. Podrían utilizarse giróscopos para determinar la orientación de la base. Esto daría al operador más información acerca del estado del robot y de la superficie por donde circula. Además permitiría realizar un control a lazo cerrado de la orientación de la plataforma. Podría utilizarse una cámara para obtener información sobre el entorno, permitiendo al operador sobrepasar obstáculos presentes en su espacio de trabajo. La Raspberry Pi 4 facilita la utilización de la cámara a lo que se añadiría algoritmos de reconocimiento de imágenes para la detección de posibles obstáculos.

Por último, en cuanto a los movimientos generales del robot, se podrían poner en práctica movimientos más complejos que no han sido implementados. Algunos de ellos podrían ser rotación y traslación simultánea del robot, traslación y rotación con una orientación específica de la plataforma, la utilización de otros tipos de pasos, entre muchas otras. Esto le daría mayor libertad al operador para poder realizar el control del robot.

Estos y algunos otros representan los trabajos a realizar en el futuro para mejorar el desempeño del robot. Si bien existen muchos puntos de mejora, creamos que se alcanzaron exitosamente los objetivos propuestos. Esperamos que el trabajo realizado pueda ser utilizado en futuras investigaciones y proyectos.

## Referencias

- [1] F. Tedeschi and G. Carbone, “Design issues for hexapod walking robots,” *Open Access robotics*, 2014.
- [2] Z. Wang, X. Ling, and A. Rovetta, “Analysis of typical locomotion of a symmetric hexapod robot,” *Cambridge University Press*, 2009.

- [3] C. S. Gurel, “Hexapod modelling, path planning, and control,” *University of Manchester*, 2009.
- [4] M. Ghayour and A. Zareei, “Direct kinematic analysis of a hexapod spider-like mobile robot,” *Trans Tech Publications*, 2011.
- [5] M. M. A. Hajiabadi, “Analytical workspace, kinematics, and foot force based stability of hexapod walking robots,” *Worcester Polytechnic Institute*, 2013.
- [6] M. Sorin and M. NiÑulescu, “Hexapod robot leg dynamic simulation and experimental control using matlab,” *FAC Symposium on Information Control Problems in Manufacturing*, 2012.
- [7] D. Thilderkvist and S. Svensson, “Motion control of hexapod robot using model-based design,” *Department of Automatic Control, Lund University*, 2015.
- [8] C. Mahfou, K. Djouani, M. Bouaziz, and S. Rechak, “Dynamic modelling and control in operational space of hexapod robot,” *5th WSEAS Int. Conf. Signal Processing*, 2006.
- [9] X. Y. Sandoval-Castro, M. Garcia-Murillo, L. A. Perez-Resendiz, and E. Castillo-Castañeda, “Kinematics of hex-piderix - a six-legged robot - using screw theory,” *International Journal of Advanced Robotic Systems*, 2012.
- [10] F. A. Raheem and H. Z. Khaleel, “Static stability analysis of hexagonal hexapod robot for the periodic gaits,” *IJCCCE Vol.14, No.3*, 2014.
- [11] J. Mrva, “Design of motion primitives for a hexapod walking robot operating in a rough environment,” *Czech Technical University in Prague*, 2014.
- [12] J. V. Sanjurjo and F. S. Sesto, “Robot hexápodo de 18 gdl,” *Universidad Nacional de Cuyo*, 2017.
- [13] F. Tedeschi and G. Carbone, “Design issues for hexapod walking robots,” *Open access robotics*, 2014.
- [14] J. P. Merlet, “Parallel robots,” *Second Edition*, 2006.
- [15] P. Corke, “Robotics, vision and control,” *Primera edición*, 2011.
- [16] J. J. Craig, “Robótica,” *Tercera edición*, 2006.
- [17] B. He, S. Xu, Y. Zhou, and Z. Wang, “Mobility properties analyses of a wall climbing hexapod robot,” *Journal of Mechanical Science and Technology*, 2018.
- [18] G. Carbone and F. Gomez-Barvo, “Motion and operation planning of robotics systems,” *Primera Edición*, 2015.

- [19] R. Woering, “Simulating the first steps of a walking hexapod robot,” *Tech-nische Universiteit Eindhoven*, 2011.
- [20] J. B. Mrinal Kalakrishnan, P. Pastora, M. Mistry, and S. Schaal, “Fast, robust quadruped locomotion over challenging terrain,” *2010 IEEE International Conference on Robotics and Automation*, 2010.
- [21] A. Barrientos, “Fundamentos de robótica,” *Segunda edición*, 1997.
- [22] K. Holmes, Full and Guckenheimer, “The dynamics of legged locomotion: Models, analyses, and challenges,” *SIAM Review (SIAM REV)*, 2006.
- [23] V. Kumar and K. J. Waldron, “Force distribution in walking vehicles,” *ASME*, 1990.
- [24] G. Romero Román and R. Gonzalez, “Modelado y control cinemático de hexápodo de 18 grados de libertad,” *XVIII Reunión de trabajo en Procesamiento de la Información y Control*, 2019.
- [25] G. Niemeyer, C. Preusche, and G. Hirzinger, “Handbook of robotics,” *Springer*, 2008.
- [26] A. A. Khan and A. Kabir, “Comparison among short range wireless networks: Bluetooth, zigbee wi-fi,” *Daffodil international University journal of science and technology*, 2016.