

PROPUESTA DE UNA PLATAFORMA AGRÍCOLA TERRESTRE BASADA EN UN  
ROBOT TIPO SERPIENTE QUE PUEDA DESPLAZARSE EN TERRENOS  
IRREGULARES



Universidad  
del Cauca®

Trabajo de Grado Ciclo I Doctorado en Ciencias de la Electrónica

**Jairo José Marín Arciniegas**

Director

*PhD. Oscar Andrés Vivas Albán*

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DOCTORADO EN CIENCIAS DE LA ELECTRÓNICA  
POPAYÁN, 2023

Jairo José Marín Arciniegas

PROPUESTA DE UNA PLATAFORMA AGRÍCOLA  
TERRESTRE BASADA EN UN ROBOT TIPO SERPIENTE  
QUE PUEDA DESPLAZARSE EN TERRENOS  
IRREGULARES

Trabajo de grado presentado a la Facultad de Ingeniería  
Electrónica y Telecomunicaciones de la Universidad del Cauca  
para la culminación del Ciclo I del Doctorado en Ciencias de la  
Electrónica .

Popayan,  
2023

A mis padres, Luz Miryam y José, por estar siempre a mi lado y  
ser mi fuente de inspiración.

A mis hermanos, por sus impulsos, motivaciones y ayudarme siempre  
a perseguir mis sueños.

A mi director, PhD Andrés Vivas, por sus sabios consejos, ayuda y  
orientación a lo largo de este trabajo.

A mis compañeros y amigos, especialmente a Cristian, John y  
Santiago, por compartir su conocimiento, experiencia y  
su infinito apoyo en todo momento.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	2
1.2. Objetivos . . . . .	3
1.2.1. Objetivo general . . . . .	3
1.2.2. Objetivos específicos . . . . .	3
1.3. Estructura del documento . . . . .	4
<b>2. Revisión de la literatura</b>	<b>5</b>
2.1. Estado del arte . . . . .	5
2.2. Marco teórico . . . . .	9
2.2.1. Clasificación de los robots . . . . .	9
2.2.2. Biomimética y robots bioinspirados . . . . .	10
2.2.3. Robots serpientes . . . . .	11
2.2.4. Locomoción de robots serpientes . . . . .	11
<b>3. Metodología de la investigación</b>	<b>15</b>
3.1. Modelado matemático de un robot serpiente . . . . .	15
3.2. Diseño del robot . . . . .	17
3.2.1. Actuador . . . . .	17
3.2.2. Fuente de alimentación . . . . .	19
3.2.3. Sensor . . . . .	20
3.2.4. Módulo de comunicación . . . . .	20
3.2.5. Material para el chasis . . . . .	20
3.3. Diseño CAD . . . . .	21
3.4. Simulador . . . . .	21
3.5. Modos de locomoción . . . . .	22
3.5.1. Modo de locomoción rectilínea . . . . .	23
3.5.2. Modo de locomoción ondulación lateral . . . . .	23
3.5.3. Modo de locomoción <i>sidewinding</i> . . . . .	25
3.5.4. Modo de locomoción Rolling lateral . . . . .	26
<b>4. Presentación de resultados</b>	<b>27</b>
4.1. Parámetros del experimento . . . . .	27
4.1.1. Configuración de las pruebas para el terreno regular . . . . .	28

4.1.2.	Configuración de las pruebas para el terreno irregular . . . . .	28
4.2.	Resultados locomoción rectilínea . . . . .	29
4.2.1.	Terreno regular . . . . .	29
4.2.2.	Terreno irregular . . . . .	33
4.3.	Resultados locomoción <i>sidewinding</i> . . . . .	34
4.3.1.	Terreno regular . . . . .	34
4.3.2.	Terreno irregular . . . . .	37
4.4.	Resultados locomoción ondulación lateral . . . . .	39
4.4.1.	Terreno regular . . . . .	39
4.4.2.	Terreno irregular . . . . .	42
4.5.	Resultados locomoción Rolling lateral . . . . .	44
4.5.1.	Terreno regular . . . . .	44
<b>5.</b>	<b>Conclusiones y trabajos futuros</b>	<b>47</b>
	<b>Apéndices</b>	<b>1</b>
<b>A.</b>	<b>Selección de componentes</b>	<b>3</b>
A.1.	Análisis de requerimientos . . . . .	3
A.2.	Selección de componentes y solución electrónica . . . . .	4
A.2.1.	Dispositivos considerados en cada uno de los planos . . . . .	4
A.2.2.	Selección de la solución electrónica . . . . .	6
A.2.3.	Reducción del espacio morfológico . . . . .	10
A.2.4.	Solución electrónica definitiva . . . . .	12
<b>B.</b>	<b>Selección del software de simulación</b>	<b>13</b>
B.0.1.	Software de simulación robótica . . . . .	13
B.0.2.	Comparación de los simuladores de robótica . . . . .	18
B.0.3.	Simulador Webots . . . . .	20

# Lista de Figuras

2.1. Clasificación de los robots [1] . . . . .	10
2.2. Robot serpiente de la Universidad Carnegie Mellon [2]. . . . .	11
2.3. Ondulación lateral - vista lateral y superior [3]. . . . .	12
2.4. Locomoción concertina o acordeón [4]. . . . .	12
2.5. Locomoción rectilínea [5]. . . . .	12
2.6. Locomoción sidewinding [6]. . . . .	13
3.1. Curva serpentinoide de Hirose [7]. . . . .	16
3.2. Combinación de curvas lateral y dorsal [8]. . . . .	16
3.3. Diagrama general del robot en SimMechanics. . . . .	17
3.4. Diagrama de bloques de los cuerpos del robot. . . . .	18
3.5. Diagrama de bloques del primer cuerpo. . . . .	18
3.6. Simulación de locomoción en las primeras versiones del robot en Simulink. . . . .	18
3.7. Torques de las articulaciones 2 y 6. . . . .	19
3.8. Módulo del robot serpiente. . . . .	21
3.9. Prototipo del robot serpiente. . . . .	21
3.10. Simulación del robot en el software Webots. . . . .	22
3.11. Modelo de robot serpiente de 12 eslabones ajustado a la locomoción rectilínea. . . . .	23
3.12. Modelo de robot serpiente de 12 eslabones ajustado a la locomoción de ondulación lateral. . . . .	24
3.13. Modelo de robot serpiente de 12 eslabones ajustado a la locomoción <i>side- winding</i> . . . . .	25
3.14. Modelo de robot serpiente de 12 eslabones ajustado a la locomoción Rolling lateral. . . . .	26
4.1. Configuración para experimento terreno regular. . . . .	28
4.2. Entornos de los experimentos en terreno irregular. . . . .	29
4.3. Forma de onda de los tipos de marcha rectilínea. . . . .	30
4.4. Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha rectilínea. . . . .	30
4.5. Ángulo Yaw (rumbo) para el patrón de marcha rectilíneo en terreno regular. . . . .	31
4.6. Ángulo Yaw (rumbo) para el patrón de marcha rectilíneo en terreno irregular. . . . .	33
4.7. Forma de onda de los tipos de marcha <i>sidewinding</i> . . . . .	35

4.8. Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha <i>sidewinding</i> . . . . .	35
4.9. Ángulo Yaw (rumbo) para el patrón de marcha <i>sidewinding</i> en terreno regular. . . . .	36
4.10. Ángulo Yaw (rumbo) para el patrón de marcha <i>sidewinding</i> en terreno irregular. . . . .	38
4.11. Forma de onda de los tipos de marcha ondulación lateral. . . . .	40
4.12. Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha ondulación lateral. . . . .	40
4.13. Ángulo Yaw (rumbo) para el patrón de marcha ondulación lateral en terreno regular. . . . .	41
4.14. Ángulo Yaw (rumbo) para el patrón de marcha ondulación lateral en terreno irregular. . . . .	43
4.15. Forma de onda de los tipos de marcha rolling lateral. . . . .	44
4.16. Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha rolling lateral. . . . .	45
4.17. Ángulo Yaw (rumbo) para el patrón de marcha rolling lateral en terreno regular. . . . .	45
A.1. Diagrama de bloques del mecanismo general del robot. . . . .	4
A.2. Componentes opcionales. . . . .	7
A.3. Diagrama de bloques con la solución de electrónica definitiva. . . . .	12
B.1. Resultados promedio de 20 ejecuciones de los elementos medidos para comparar los simuladores en un enfoque cuantitativo. . . . .	19
B.2. Interfaz del software de simulación Webots. . . . .	20
B.3. Jerarquía principal del nodo World. . . . .	21



# Lista de Tablas

3.1. Parámetros para los modos de locomoción rectilínea. . . . .	24
3.2. Parámetros para los modos de locomoción de ondulación lateral. . . . .	24
3.3. Parámetros para los modos de locomoción <i>sidewinding</i> . . . . .	25
3.4. Parámetros para los modos de locomoción Rolling lateral. . . . .	26
4.1. Comparación de resultados de la locomoción rectilínea en terreno regular. .	32
4.2. Comparación de resultados de la locomoción rectilínea en terreno irregular.	34
4.3. Comparación de resultados de la locomoción <i>sidewinding</i> en terreno regular.	37
4.4. Comparación de resultados de la locomoción <i>sidewinding</i> en terreno irregular.	39
4.5. Comparación de resultados de la locomoción ondulación lateral en terreno regular. . . . .	42
4.6. Comparación de resultados de la locomoción ondulación lateral en terreno irregular. . . . .	44
4.7. Comparación de resultados de la locomoción rolling lateral en terreno regular.	46
A.1. Especificaciones de los dispositivos de alimentación. . . . .	8
A.2. Especificaciones de los actuadores. . . . .	9
A.3. Especificaciones de los sensores. . . . .	9
A.4. Especificaciones de los dispositivos de comunicación para los actuadores. . .	9
A.5. Especificaciones de los dispositivos de comunicación para los sensores. . . .	10
B.1. Características principales de los simuladores. . . . .	18



# Capítulo 1

## Introducción

El sector agrícola suministra recursos para satisfacer las principales necesidades humanas. Sin embargo, la no optimización de las prácticas y tareas realizadas en este entorno pueden afectar negativamente el medio ambiente y en consecuencia, conducir a desastres ecológicos. A la anterior dificultad que debe enfrentar este sector se suma el aumento necesario en la producción agrícola para satisfacer la demanda alimentaria, debido al crecimiento de la población. Con el fin de abordar estas cuestiones, es necesario innovar en los procesos y la tecnología que se utiliza en la industria agrícola, en busca de mejorar la productividad y competitividad. Este tipo de innovaciones se deben implementar en un entorno que considere el campo, los cultivos y sus alrededores, con el objetivo de facilitar información al agricultor para tomar mejores decisiones y poder actuar maximizando el rendimiento, de dónde proviene el concepto de agricultura de precisión.

El término de agricultura de precisión se define como un tipo de agricultura fundamentado en el manejo de técnicas de monitoreo, recolección e intervención de datos del cultivo a través del despliegue de tecnologías de detección y automatización [9], con la finalidad de apoyar en la toma de decisiones y así actuar únicamente donde es imprescindible la aplicación y uso de los recursos. Un enfoque de la agricultura de precisión para conseguir una gestión óptima de los recursos es la introducción de robots en el entorno agrícola.

La aplicación de los robots agrícolas ha experimentado un aumento en la investigación, a causa de que el uso de estas plataformas es inevitable por las múltiples ventajas que se obtienen debido a que existen tareas agrícolas susceptibles de automatización, por ejemplo, de labores altamente repetitivas, arduas y peligrosas. Sin embargo, la introducción de robots en la agricultura tiene más complicaciones que en la industria. Principalmente se debe a que los robots deben desplazarse en entornos dinámicos, por ello se requieren robots robustos y capaces de adaptarse. En la agricultura se hace uso de plataformas robóticas tanto terrestres como aéreas, Los robots terrestres deben viajar en suelos irregulares, poco homogéneos y fangosos, mientras que los vehículos aéreos deben operar durante largos períodos de tiempo, en diferentes condiciones climáticas.

Los robots agrícolas actuales utilizan diseños basados en tecnología usada en otros

sectores o se complementan con plataformas existentes. Pero la mayoría de estas soluciones no están completamente optimizadas para el desarrollo de todas las tareas. Por lo anteriormente citado, en este proyecto se busca incluir un robot tipo serpiente en entornos agrícolas, ya que estas plataformas son innovadoras y tienen un gran alcance en la realización de varias aplicaciones en el mundo real. Su alto número de grados de libertad les dan el potencial de adaptarse a terrenos complejos, además, de desplazarse en espacios estrechos y restringidos. Se espera que el uso de estas plataformas equipadas con sensores apropiados pueda explorar el cultivo y obtener información beneficiosa para los productores.

## 1.1. Planteamiento del problema

Las plataformas agrícolas son anunciadas como parte integral de la agricultura de precisión. La aplicación de esta tecnología tiene como principio mejorar la productividad y competitividad en los campos con el fin de satisfacer la demanda alimentaria generada por el permanente crecimiento de la población mundial [10], [11]. Adicionalmente, los avances tecnológicos han llevado a que los robots agrícolas ofrezcan la posibilidad de utilizar de forma eficaz e inteligente los recursos naturales como el agua y tierra cultivable, ya que estos se encuentran muy limitados [12], disminuir el consumo de fertilizantes y pesticidas, reduciendo los impactos ambientales en el ecosistema [13], [14].

Durante el siglo XX se obtuvo un aumento considerable en la productividad agrícola gracias al desarrollo de maquinaria de gran tamaño. Actualmente en gran parte del sector agrícola aún se utiliza esta maquinaria, causando daños de fertilidad y compactación permanente en el suelo [15]. Una solución práctica es reemplazar estos mecanismos pesados, por lo cual se espera que en un futuro cercano los pequeños robots móviles automatizados, desempeñen un papel importante en la agricultura [16], [17]. Idealmente estos pequeños robots deben ejecutar con un alto grado de precisión las actividades agrícolas programadas y adaptar su navegación al terreno del cultivo. Aunque para realizar una navegación autónoma hay diversas soluciones, estas están bien establecidas para los campos pequeños e instalaciones interiores con terreno estructurado, tales como invernaderos y viveros [18], pero este problema sigue sin resolverse para los cultivos con entornos abiertos, donde se observan terrenos densos e irregulares, además de tener que enfrentarse a los cambios en las condiciones climatológicas que se puedan presentar [19]. Por ello se requiere la implementación de robots robustos capaces de adaptarse y desplazarse por los cultivos haciendo uso de sistemas de muestreo y detección, con el fin de ayudar a reducir los costos de producción [9]. A pesar de que en la robótica agrícola surjan muchos prototipos y estén muy avanzados, seguirán siendo poco prácticos e inviables si no se cuenta con una infraestructura de soporte que deberá incluir todos los servicios y tecnologías que necesitan los robots agrícolas mientras estén en funcionamiento. Esto incluye una comunicación inalámbrica confiable, un marco para compartir y reutilizar software, y un entorno eficaz para la interacción entre robots, trabajadores agrícolas y agricultores [20].

Últimamente se han generado múltiples investigaciones sobre vehículos agrícolas terrestres, debido a que los productores necesitan reunir información confiable para tomar las mejores decisiones sobre sus cultivos. En algunas ocasiones, obtener la información más confiable significa realizar una detección proximal o exploración del campo a nivel de suelo [21], ya que el análisis de esta información ofrece un gran rendimiento en la productividad y ayuda a mitigar el daño ambiental que se genera en este sector [22]. Sin embargo, faltan llenar algunos vacíos comunes para la adaptación de estas plataformas a entornos dinámicos no estructurados [23]. Estas dificultades llevan a los investigadores a proponer nuevos sistemas como lo son los mecanismos bio-inspirados. En este caso en particular un robot zoomórfico tipo serpiente, el cual pertenece a una clase prometedora de mecanismos para aplicaciones del mundo real, como la búsqueda y rescate urbano, inspección industrial y la agricultura [24]. Los robots serpientes son mecanismos hiper-redundantes, inspirados en las serpientes biológicas, que cuentan con gran cantidad de enlaces accionados en serie. Además, sus múltiples grados de libertad les brindan la capacidad de adaptarse y navegar en entornos complejos, pudiendo ser manipulados en espacios reducidos [7], [25]. Sin embargo, los robots serpiente a diferencia de otras clases de robots móviles, están siendo estudiados por un número relativamente pequeño de grupos de investigación [26]. Esto se debe a que estas plataformas tienen ciertas limitaciones como la mala eficiencia energética y la dificultad del control en la locomoción en entornos desafiantes, pero hay ciertas características que permiten concluir que la innovación en este tipo de plataformas puede superar estas limitaciones, potenciando las capacidades de locomoción y de esta forma llevar a este mecanismo a un futuro exitoso. De acuerdo con lo expuesto previamente, este proyecto busca proponer un robot modular tipo serpiente que incorpore un algoritmo de control para un modo de marcha adaptativo, con el propósito de que el robot pueda navegar en campos agrícolas con entornos dinámicos. De lo anterior surge la siguiente pregunta de investigación: *¿Cuáles serán las características de una plataforma agrícola terrestre basada en un robot modular tipo serpiente que pueda desplazarse en entornos irregulares, con el propósito de explorar un ambiente a nivel de suelo?*

## 1.2. Objetivos

### 1.2.1. Objetivo general

Proponer una plataforma agrícola terrestre basada en un robot tipo serpiente que pueda desplazarse en terrenos irregulares con el fin de explorar un ambiente a nivel de suelo.

### 1.2.2. Objetivos específicos

1. Determinar los dispositivos electrónicos que faciliten la implementación de una locomoción en terreno irregular para el desarrollo de un robot serpiente.
2. Seleccionar un ambiente virtual para la simulación de un robot tipo serpiente, que permita observar la precisión de las trayectorias de navegación del algoritmo de marcha propuesto.

3. Validar un primer prototipo de un robot serpiente a partir de los resultados obtenidos en diferentes pruebas de campo y en presencia de terreno irregular.

### **1.3. Estructura del documento**

Una visión general del tema en estudio se presenta en el capítulo 2, donde se realiza la revisión literaria compuesta por estado del arte, y el marco conceptual. El capítulo 3 muestra la metodología llevada a cabo en el proyecto. Posteriormente, en el capítulo 4, se presentan los resultados de las simulaciones y pruebas del robot real obtenidos con la implementación de los diferentes algoritmos de locomoción. Finalmente, en el capítulo 5 se presentan las conclusiones y recomendaciones para futuros trabajos.

# Capítulo 2

## Revisión de la literatura

### 2.1. Estado del arte

El sector agrícola, de igual manera que el sector industrial, se actualiza constantemente gracias al surgimiento de nuevas tecnologías. Actualmente, los robots se están introduciendo en este entorno, haciendo uso de plataformas tanto aéreas como terrestres, las cuales han demostrado que pueden desarrollar de forma precisa actividades como la siembra, cosecha, monitoreo y medición de propiedades de las plantas en diversos cultivos [9].

Los vehículos aéreos no tripulados, usados en el contexto de la agricultura, cuentan con la ventaja de que no necesitan adaptarse a los entornos irregulares y en estos se pueden adaptar múltiples sensores con el fin de controlar el déficit de riego, detectar enfermedades, medir el estrés hídrico y niveles de fluorescencia de la clorofila, entre otros [27], [28]. Pero el uso de esta tecnología es complejo y está alcanzando un estado estacionario, debido a que la mayoría de las aplicaciones utilizan visión artificial, y el uso de esta disciplina requiere que la plataforma sobrevuele el cultivo a ciertas horas del día para evitar las sombras de vegetación en el suelo. Generalmente estas sombras generan errores, haciendo desafiante el posterior análisis y procesamiento de datos y mosaico de las imágenes [29].

Los vehículos terrestres equipados con los sensores apropiados pueden explorar campos a nivel del suelo (detección proximal) y obtener información ventajosa para los productores, porque a medida que estos vehículos se mueven cerca del cultivo, los datos adquiridos aumentan en precisión, y las resoluciones de una o más muestras por metro son factibles, estando solamente limitadas por las especificaciones de los sensores implementados [21]. Un ejemplo de aplicación de estas plataformas se encuentra en el robot terrestre Rosphere desarrollado en la Universidad Politécnica de Madrid [30], el cual tiene como propósito realizar monitoreo de humedad bajo la premisa de agricultura de precisión.

Pero el diseño y construcción de robots agrícolas terrestres para que operen fuera de un ambiente de laboratorio requieren de esfuerzos desafiantes, ya que estas plataformas deben moverse en entornos dinámicos y semi estructurados. Estas dificultades, sumado al estado actual de la técnica de automatización, ha llevado a un creciente interés en el diseño

y desarrollo de nuevos robots agrícolas [31]. Entre las soluciones propuestas, vale la pena mencionar a los robots que son bio-inspirados, como lo son los robots con extremidades capaces de subir montañas para exploración forestal [32], y los robots hexápodos basados en la locomoción de las arañas, capaces de plantar y cosechar maíz de manera autónoma [33]. Sin embargo, la mayoría de ellos están todavía en desarrollo, o sólo existen como prototipos de investigación, pero cada vez más están surgiendo variedad de propuestas de vehículos terrestres innovadores, los cuales probablemente en el futuro sean planteados como soluciones robóticas que se incorporen en el mercado de las plataformas agrícolas. En las dos últimas décadas se han generado múltiples innovaciones versátiles en plataformas tanto aéreas como terrestres en el campo de la bioingeniería, gracias al estudio de los animales y cómo estos se adaptan a su entorno circundante. Una de estas novedades son las plataformas robóticas tipo serpiente, mecanismos diseñados para moverse como serpientes biológicas [34]. La ventaja de tales robots es su cuerpo largo y flexible, que les da el potencial de satisfacer la creciente necesidad de moverse y operar en entornos desconocidos y desafiantes, como pueden ser los entornos agrícolas [24], [35]. Estos robots son dispositivos hiper-redundantes [36], consisten en módulos conectados en serie que pueden girar en uno o más planos, generando muchos grados de libertad, haciendo que sean difíciles de controlar, pero por otra parte les proporcionan una capacidad de desplazamiento en una amplia gama de entornos irregulares que supera la movilidad de los robots con patas, ruedas y orugas [37].

La investigación sobre robots serpientes se ha llevado a cabo durante varias décadas. El primer estudio analítico de los métodos de locomoción de serpientes lo realizó Gray en 1946 [38], en el cual planteó una descripción matemática de todas las fuerzas que actúan cuando una serpiente biológica se mueve, donde una de las principales conclusiones fue que el movimiento de desplazamiento sobre una superficie plana requiere la existencia de fuerzas externas que actúen en la dirección normal hacia el cuerpo de la serpiente. En 1972 [39], Hirose descubrió que una aproximación cercana a la forma de una serpiente biológica durante la ondulación lateral viene dada por una curva plana cuya curvatura varía sinusoidalmente. De esa investigación salió la famosa ecuación de la curva serpenoide (la forma más común de locomoción de las serpientes), además se desarrolló el primer robot serpiente equipado con ruedas pasivas a lo largo de su cuerpo, las cuales le facilitan al mecanismo lograr una propulsión en superficies planas. Esta ha sido una característica muy usada durante los últimos 20 años en el desarrollo de este tipo de plataformas [40], [41], sin embargo, las ruedas pasivas no permiten un buen control en entornos no estructurados. Por lo tanto, tales mecanismos son adecuados solo para experimentos de control de navegación sobre terrenos planos [42]. Por lo anterior, en la literatura existe una tendencia creciente en optimizar la locomoción del robot serpiente, permitiendo que estas plataformas se muevan en entornos irregulares y desafiantes [43].

En el año 2007, Borenstein [44], realizó la cuarta versión de OmniTread (OT-4), el cual se compone de 6 articulaciones accionadas por fuelles neumáticos, lo cual facilita el control simultáneo de la posición y rigidez de cada junta mediante la ejecución de la locomoción serpentina mientras atraviesa un terreno irregular. Una característica de él es



el uso de correas de propulsión tipo oruga, haciendo que el robot no detenga su desplazamiento debido a los vuelcos que se pueden presentar en terrenos accidentados, lo cual se demuestra con pruebas en entornos con presencia de escombros y rocas. Rasakatla y otros [45] presentaron en 2010 un robot serpiente que hace una transición automática de la marcha dependiendo del tipo de terreno, utilizando una cámara para diferenciar entre una alfombra de césped artificial y el pavimento. Para el primer entorno se requiere una marcha rectilínea o tipo oruga, ya que los primeros módulos del robot al entrar en contacto con el césped requieren un mayor coeficiente de fricción que permita tirar del resto del cuerpo y así obtener una locomoción regular. Siguiendo esta línea de adaptación al entorno se desarrolla un prototipo robótico llamado HITSZ-Serpiente I [46], el cual cuenta con 10 articulaciones modulares y una programación de los modos de locomoción típicos de estas plataformas, incluyendo el modo de marcha de peristaltismo, rodante y curva serpenoide, los cuales ayudarán al robot a ajustar la postura y adaptarse al ambiente cambiante al cruzar obstáculos como rocas, raíces y hierba.

Las investigaciones anteriores hacen una gran contribución al progreso de los robots serpientes. Pero todavía existen algunos problemas típicos de estos mecanismos como lo es: la baja velocidad, gran tamaño y control complejo [47]. Estos problemas obviamente obstaculizan el progreso de la investigación y dificultan que estos robots jueguen un papel más importante en el mundo real. Debido a los problemas mencionados anteriormente, en el año 2012, el laboratorio de biorobótica de la Universidad de Carnegie Mellon, llevó a cabo el desarrollo de un robot serpiente siguiendo un nuevo enfoque de construcción para que este sea pequeño, ligero y con fuertes mecanismos de actuación, que le permita navegar en diversos entornos, como terreno irregular, pendiente, canales, trepar postes y árboles [48]. Esto se logra gracias a sus 16 actuadores incorporados en una arquitectura modular, los cuales se alternan en posición vertical y horizontal, con el fin de que las articulaciones impares controlen la rotación lateral y las pares la rotación dorsal. Además cuentan con una cámara y un láser que permiten que el robot realice tareas de búsqueda y rescate, detección de minas e inspección industrial [49]. Fabico y colaboradores [50], plantearon un pequeño robot biomórfico e hiper-redundante que pueda ser desplegado en áreas con escombros de desastres con el fin de llevar a cabo actividades de búsqueda y rescate. El robot cuenta con 10 segmentos y está equipado con sensores de proximidad para detección de obstáculos, IMU para detección de orientación y una cámara inalámbrica para que el controlador vea el entorno en el que opera el robot. Los experimentos sobre terrenos rocosos, con túneles, pavimento y hierba, mostraron que el algoritmo de movimiento proporciona una locomoción similar a las serpientes biológicas, permitiendo obtener una velocidad máxima de 70 cm/s y una altura de ascenso que posibilita sobrepasar obstáculos de hasta 22 cm.

La locomoción del robot serpiente se basa en la hipótesis de que la marcha inteligente y eficiente en entornos desconocidos y desordenados requiere que el robot pueda sentir su entorno [51]. Una eficiente detección de la fuerza de contacto le permite al robot detectar cuándo el cuerpo está en contacto con un punto de empuje y también controlar la fuerza ejercida sobre ese punto. Dado que la suma de las fuerzas de contacto a lo largo del cuerpo de la serpiente son las que lo impulsan hacia adelante, la capacidad de medir estas

fuerzas es importante para controlar la propulsión. Por la importancia de que el robot pueda sentir su entorno, el centro de robótica avanzada Robotnor en Noruega, presentó al robot serpiente Mamba en el año 2014 [52], el cual tiene una capacidad de detección del entorno muy novedosa en comparación con plataformas anteriores, ya que se requería que la plataforma fuera a prueba de agua y polvo, facilitando aplicaciones del robot en entornos al aire libre y bajo el agua. Sin embargo, esta combinación de requerimientos es difícil ya que los sistemas de detección se producen a expensas de la electrónica que está expuesta al medio ambiente, pero se logró aislar el actuador dentro de cada módulo de unión con sensores de fuerza basados en galgas extensiométricas diseñadas a medida.

Las plataformas robóticas tipo serpiente que implementan modos de marcha comunes, a la hora de ejecutar actividades con dichos modos, proporcionan un campo visual con información muy limitada. En consecuencia, los robots serpiente deberían levantar la cabeza para obtener un campo de visión de  $360^\circ$  y así aumentar su espacio de trabajo. Desde el punto de vista biológico, es irregular ver este movimiento y es difícil describir su forma por completo por medio de un modelo matemático. Pero Zhang y otros [53], con base a las múltiples observaciones realizadas a estos animales, determinó que el movimiento para levantar su cabeza es similar a la de una curva espiral, por este motivo proponen un modelo para levantar la cabeza de estos mecanismos, el cual sea adaptable a diferentes plataformas con diversas longitudes, números de módulos y números de espiral: Pero para obtener un modelo exitoso se deben resolver varios problemas. Primero, la curva espiral propuesta es una función por partes y tiene puntos suaves, lo que conduce a cambios bruscos en las trayectorias de movimiento; segundo, el modelo propuesto considera solo la planificación cinemática, es decir, no se considera el problema dinámico, teniendo en cuenta que las restricciones dinámicas son un factor importante para la implementación exitosa del modelo en un robot real. El método de curva espiral propuesto por Zhang y colaboradores se implementó únicamente para robots con 3 grados de libertad por módulo, y en las pruebas se observó que este método limita el rango de movimiento de elevación de la cabeza. Para resolver los problemas generados por la curva espiral, Zhou y otros [54], presentaron una nueva estrategia de control para el levantamiento de la cabeza en los robots modulares tipo serpiente basada en cinemática segmentada, que hace uso de un algoritmo de optimización para mejorar la estabilidad de su movimiento. Además, se propone un método de planificación de trayectoria del cuerpo en general en el espacio cartesiano, para evitar colisiones entre módulos.

En el año 2020, se presenta una combinación de dos plataformas robóticas para tareas de búsqueda y rescate, un robot móvil con sistema de tracción tipo oruga y un robot serpiente [55]. El robot serpiente puede manipular objetos cuando está encima del robot móvil, pero puede separarse para examinar el entorno cuando en este se presentan espacios estrechos donde la plataforma móvil no puede navegar más. Actualmente, el robot serpiente una vez separado no puede reincorporarse a la plataforma móvil, por lo que los autores buscarán una solución en un trabajo futuro mediante la implementación de un nuevo modo de marcha.

En Colombia, el laboratorio de investigación KM-RoBoTa en articulación con la Pontificia Universidad Javeriana han llevado a cabo varios trabajos de investigación y desarrollo sobre el robot modular tipo serpiente dentro del proyecto Lola-OPTM: Este proyecto cuenta con una plataforma robótica de hardware abierto, para la cual se desarrollaron tanto software como controladores de manipulación, herramientas de simulación y control [56], [57], [58]. Además, se realizaron análisis comparativos de diferentes modos de locomoción para el desplazamiento sobre tuberías horizontales, donde se demostró que el modo de marcha de laminado lateral fue el más adecuado y eficiente, con un resultado en la velocidad máxima de 46.5 cm/s [59].

Finalmente, en el sector agrícola se ejecutan diversas actividades, las cuales son susceptibles a errores humanos, ya que se realizan en condiciones de mal tiempo, densa vegetación y presencia de vida salvaje, además son tediosas, repetitivas y peligrosas. Estas dificultades al momento de realizar las actividades agrícolas hacen que este sector requiera de una innovación tecnológica. Actualmente, debido a la implementación de la agricultura de precisión, se han desarrollado múltiples plataformas robóticas con el fin de aumentar la productividad del campo y enfrentar las dificultades presentes en la agricultura. De acuerdo a lo descrito anteriormente y las condiciones en las que deben ejecutarse labores dentro de estos cultivos, de la presente revisión se logra evidenciar que el robot serpiente puede estar equipado de sensores y herramientas que le permitan extraer información del ambiente mientras atraviesa una amplia gama de entornos complejos. Estas propiedades hacen que el robot serpiente se presente como una gran alternativa de innovación al momento de la realización de actividades agrícolas autónomas, ya que cuenta con características únicas de adaptación a entornos desconocidos, dinámicos y no estructurados.

## 2.2. Marco teórico

En esta sección se documentan los conceptos necesarios para entender el contexto de este trabajo. En primer lugar, se propone una clasificación de los robots. Posteriormente, el capítulo se centra en los principales desafíos de investigación de los robots hiper-redundantes y modulares bioinspirados en las serpientes.

### 2.2.1. Clasificación de los robots

Existen diferentes definiciones para lo que se conoce como “robot”, una de ellas es la descrita por Karel Čapek en la obra *R.U.R (Rossum’s Universal Robots)* [60], donde se define a un robot como una entidad programable y automática capaz de realizar tareas previstas de forma autónoma, sin ninguna intervención humana, especialmente las agotadoras, repetitivas o peligrosas. Además, un robot también se puede identificar como un sistema virtual o mecánico, pero en este documento se utilizará en el último sentido.

Los robots pueden clasificarse siguiendo cinco características diferentes: según la funcionalidad del robot, el número de grados de libertad, el número de módulos, los diferentes

tipos de actuación y su ubicación, y los materiales de los que están hechos.

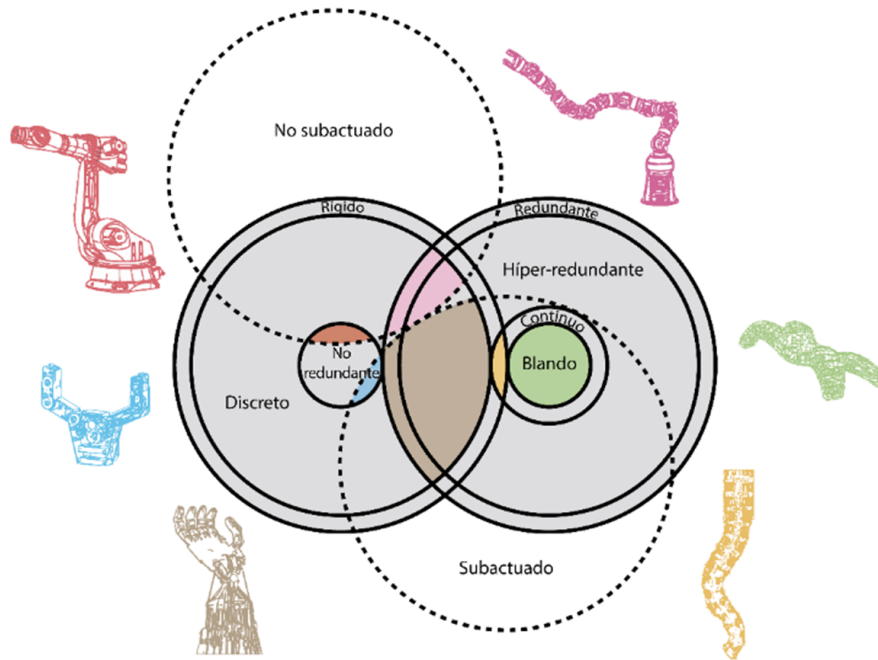


Figura 2.1: Clasificación de los robots [1]

Los robots generalmente están diseñados para diferentes propósitos. En función de esta clasificación es posible diferenciar robots manipuladores, móviles o híbridos hiper-redundantes. Los primeros tienen una base fija mientras que su efector final es capaz de moverse para manipular objetos. Sin embargo, los robots hiper-redundantes móviles tienen capacidades de locomoción. En otras palabras, habilidades para mover todo el cuerpo de un lugar a otro en diversos terrenos.

Un robot hiper-redundante tiene mayores capacidades cinemáticas y son más tolerantes a fallas que uno redundante. Por lo tanto, pueden funcionar correctamente incluso después de la falla en algunas de sus articulaciones.

El diseño mecánico de los robots hiper-redundantes es inherentemente complejo. Por lo general, consiste en el diseño de un módulo que se repite secuencialmente a lo largo de su estructura. Esta estrategia permite la creación de sistemas hiper-redundantes modulares, que son reconfigurables y más fáciles de reparar en caso de falla.

### 2.2.2. Biomimética y robots bioinspirados

El término biomimética deriva del griego antiguo: bios “vida” y mimesis “imitación”. Así, pueden entenderse como la imitación de los modelos, sistemas y elementos de la naturaleza con el fin de resolver problemas humanos complejos.

Los organismos vivos han evolucionado hasta convertirse en estructuras muy bien adaptadas a través de la selección natural. En este contexto, ahora es posible encontrar diseños sumamente inteligentes que pueden ser aprovechados, imitando o modificando sus estructuras o comportamientos. La biomimética no pretende suplantar a la naturaleza con mejores diseños, sino tomar aquellos aspectos beneficiosos con interés potencial para un fin específico.

En relación con esto último, en el ámbito de la robótica hiper-redundante, estos dispositivos a menudo se inspiran en modelos biomecánicos como serpientes, tentáculos de calamar, pulpos o trompas de elefante, entre otros.

### 2.2.3. Robots serpientes

Un robot serpiente es una clase de robots de inspiración biológica que se construyen para emular las características estructurales de las serpientes reales. Además, los robots serpiente pertenecen a la familia de mecanismos robóticos hiper-redundantes, donde su estructura física consta de la unión de módulos idénticos conectados en serie que les dan la capacidad de movimiento en uno o más planos, lo que los hace muy adecuados para la locomoción en entornos complejos y no estructurados.



Figura 2.2: Robot serpiente de la Universidad Carnegie Mellon [2].

### 2.2.4. Locomoción de robots serpientes

La versatilidad de la locomoción de las serpientes les permite prosperar en terrenos muy diversos. Escalan acantilados rocosos empinados, atraviesan pasajes estrechos, trepan árboles, navegan a través de densos arbustos y vegetación con muy pocos tipos de locomoción. A continuación, se detallan algunos tipos de locomoción inspirados por el comportamiento biológico de una serpiente real.

- **Ondulación lateral:** también conocida como locomoción serpentina, es el modo de andar más utilizado por las serpientes. Es un movimiento continuo de todo el

cuerpo en un patrón ondulatorio similar al de una curva sinusoidal [61]. El vientre de la serpiente está en contacto constante con la superficie sobre la que viaja, como se muestra en la Figura 2.3. Las ondas se propagan lateralmente desde la cabeza hasta la cola empujando contra la superficie del suelo y generando un movimiento hacia adelante. Durante la locomoción, cada punto del cuerpo sigue el mismo camino establecido por la cabeza y el cuello.

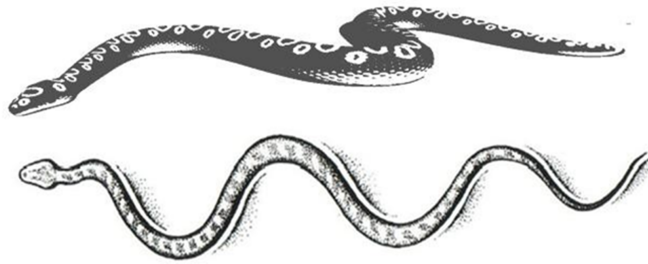


Figura 2.3: Ondulación lateral - vista lateral y superior [3].

- **Concertina o acordeón:** este tipo de patrón de locomoción se usa a menudo para espacios estrechos donde el posible rango de movimiento es pequeño. El movimiento se lleva a cabo al estirar y doblar su cuerpo para avanzar. La parte doblada se mantiene en una posición fija mientras que el resto del cuerpo se empuja o tira. Luego, las dos partes intercambian roles. El movimiento hacia adelante se obtiene cuando la fuerza necesaria para empujar hacia atrás la parte fija del cuerpo de la serpiente es mayor que las fuerzas de fricción en la parte móvil del cuerpo [62], ver la Figura 2.4.

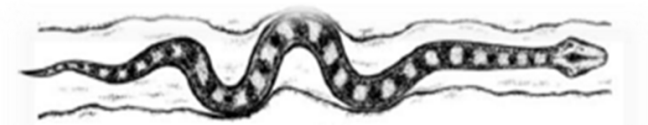


Figura 2.4: Locomoción concertina o acordeón [4].

- **Locomoción rectilínea:** En este tipo de movimiento, la serpiente agarra su cuerpo en puntos específicos que parecen moverse continuamente hacia la cola, como se muestra en la Figura 2.5. Las ondas sinusoidales se dirigen a lo largo del robot serpiente, impulsándolo hacia adelante o hacia atrás [63].



Figura 2.5: Locomoción rectilínea [5].

- **Sidewinding:** es probablemente el paso más asombroso de observar y es utilizado principalmente por serpientes en el desierto [64]. La serpiente levanta y curva su cuerpo en una posición hacia arriba y hacia adelante y luego se acuesta de nuevo en un movimiento de balanceo en la dirección del movimiento como se muestra en la Figura 2.6. El movimiento es periódico y cada segmento del cuerpo realiza acciones de los segmentos anteriores. Al igual que con otros modos de andar, la ubicación del ángulo de la cabeza determina la dirección de la serpiente.

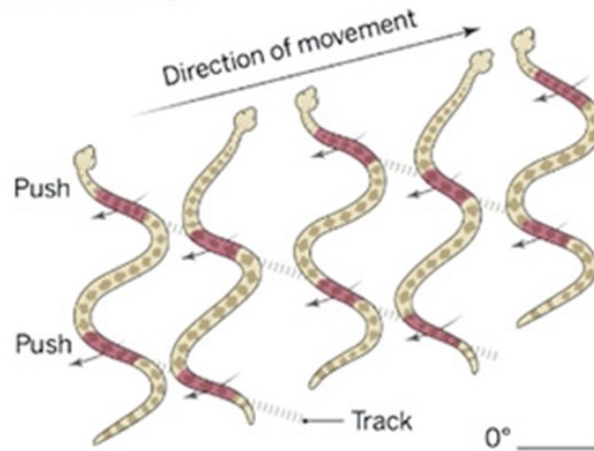


Figura 2.6: Locomoción sidewinding [6].

- En la literatura también se han implementado a robots serpientes otros tipos de locomoción no bioinspirados, tales como: Rolling lateral [65], marcha de fusión [66], locomoción asistida por obstáculos [67] y Hélice [65].





# Capítulo 3

## Metodología de la investigación

En este trabajo de investigación se desarrolló un robot serpiente, el cual incluye 12 módulos idénticos, además de un módulo para la cabeza y cola. Cada articulación tiene un grado de libertad y se conectan al módulo anterior de tal forma que el eje de giro sea perpendicular, permitiendo al robot serpiente tener dos ejes de movimiento, uno en el plano dorsal y otro en el lateral. Tal configuración permite que el robot realice movimientos en las tres dimensiones.

### 3.1. Modelado matemático de un robot serpiente

Los mecanismos altamente articulados son por definición sistemas sub determinados en términos de su control cinemático. Por lo tanto, brindan múltiples soluciones que satisfacen el mismo problema. Esta es una gran ventaja frente a los robots tradicionales no redundantes, pero también es un gran desafío en términos de encontrar la solución más adecuada. La optimización para gestionar el modelo matemático para la locomoción de robots con forma de serpiente dependerá completamente del contexto del robot, su diseño mecánico, las tareas y el entorno de operación.

Un inconveniente de las plataformas hiper-redundantes tipo serpiente al tener en cuenta un modelo matemático clásico basado en la cinemática inversa, es que se requiere una mayor potencia de cálculo comparado con los enfoques bioinspirados, los cuales necesitan menos recursos computacionales para su implementación.

Teniendo en cuenta lo antes mencionado, el robot serpiente diseñado debe tener la habilidad de ejecutar diferentes modos de marcha para adaptarse a entornos irregulares. En este trabajo, para lograr imitar el movimiento de una serpiente, se implementaron modos de locomoción basados en la curva serpentinoide de Hirose y sus variaciones tridimensionales [65]. Estas curvas se pueden describir como ondas sinusoidales que se propagan a través del cuerpo, una para el movimiento horizontal y otra para el posible movimiento vertical, como se demuestra en la ecuación 3.1

$$\alpha(n, t) = \begin{cases} A_h * \sin(\omega_h * t + n * \delta_h), & n = \text{impar} \\ A_v * \sin(\omega_v * t + n * \delta_v + \phi), & n = \text{par} \end{cases} \quad (3.1)$$

Donde  $n$  es el número del motor,  $A_h$ ,  $A_v$  son las amplitudes de la onda,  $\delta_h$ ,  $\delta_v$  es la frecuencia espacial,  $\omega_h$ ,  $\omega_v$  representan la frecuencia temporal, y  $\phi$  es la diferencia de fase entre las ondas sinusoidales en el plano horizontal y vertical. Estos parámetros se ilustran cualitativamente en la siguiente figura.

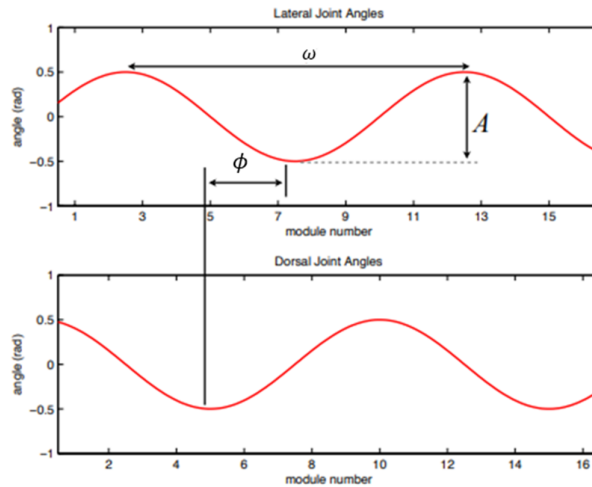


Figura 3.1: Curva serpentina de Hirose [7].

Los parámetros en la ecuación 3.1 se pueden modificar para crear diferentes modos de locomoción. Para modelar un patrón de movimiento bidimensional,  $A_h$  se establecen en cero para que solo una onda horizontal se propague a través del cuerpo en lugar de dos. Para que sea posible un movimiento tridimensional, se deben aplicar simultáneamente al robot las dos curvas como se presenta en la siguiente figura.

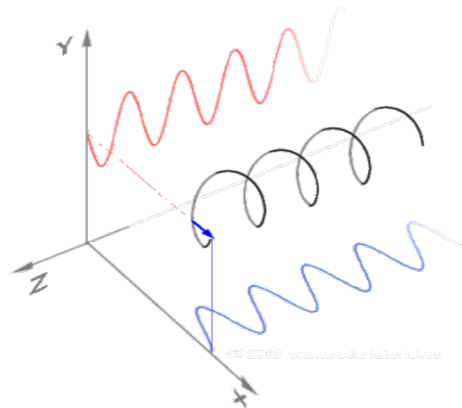


Figura 3.2: Combinación de curvas lateral y dorsal [8].

## 3.2. Diseño del robot

El objetivo en la construcción del robot serpiente fue diseñar un mecanismo modular, dado que el uso de una arquitectura modular facilita la sustitución o la adición de módulos para disminuir o aumentar los grados de libertad. El diseño actual consta de 12 módulos similares para cada uno de los enlaces, más un módulo de cabeza y uno de cola.

Se utilizó la metodología de diseño basada en una matriz morfológica para seleccionar el hardware más adecuado. Este proceso implica examinar cada componente por separado y elegir la mejor solución para cada uno. El desarrollo de este estudio morfológico se encuentra en el apéndice A y a continuación se describen los componentes seleccionados.

### 3.2.1. Actuador

Antes de comenzar la construcción del robot, es conveniente analizar en un entorno virtual las ecuaciones matemáticas de movimiento. Para este análisis se realiza una simulación del robot utilizando el software Matlab – Simulink con la combinación de la herramienta SimScape, la cual permite una simulación multicuerpo eficiente y rápida, formulando y resolviendo las ecuaciones de movimiento de cualquier sistema mecánico.

Las simulaciones mostrarán el torque requerido para crear los patrones de movimiento y permitirán comprobar si los parámetros escogidos para las ecuaciones de movimiento producen el resultado deseado. En consecuencia el modelo se puede depurar en el entorno de simulación, sin correr el riesgo de provocar movimientos inesperados y sobrecargar los motores.

En la Figura 3.3 se puede observar el diagrama en bloques que representa el robot, el piso y las ecuaciones de locomoción.

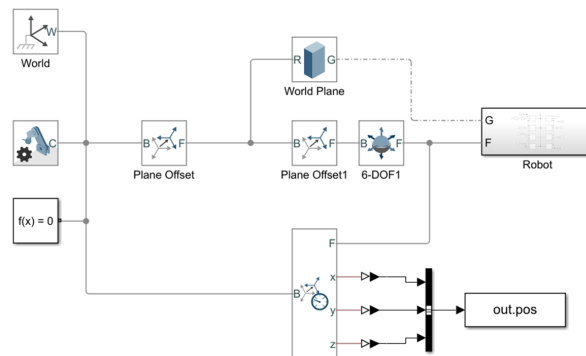


Figura 3.3: Diagrama general del robot en SimMechanics.

El diagrama anterior se descompone en varios bloques. Para determinar el marco de referencia global del modelo se usa el bloque *World frame* y se implementa una unión de 6 grados de libertad (6-DOF-1), la cual representa la estructura virtual de orientación y

posicionamiento para robots con base flotante. Esta unión permite establecer la posición inicial del robot con respecto a la referencia global del sistema. Dentro del bloque Robot de la Figura 3.3, se encuentran representados los doce cuerpos, la cola y la cabeza del robot, los cuales están conectados entre sí (ver Figura 3.4). Adicionalmente, se pueden observar algunos bloques llamados *Transform*, donde se especifican las rotaciones y traslaciones de los ejes de cada uno de los enlaces.

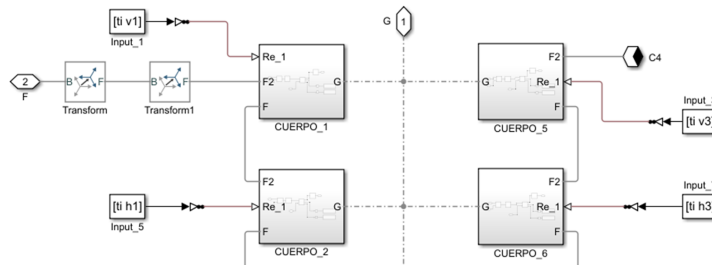


Figura 3.4: Diagrama de bloques de los cuerpos del robot.

Los bloques de los cuerpos en su interior contienen cada uno una articulación rotacional y una salida de señal de posición y torque del enlace (Figura 3.5). Además, el bloque del cuerpo\_1 contiene la cabeza del robot.

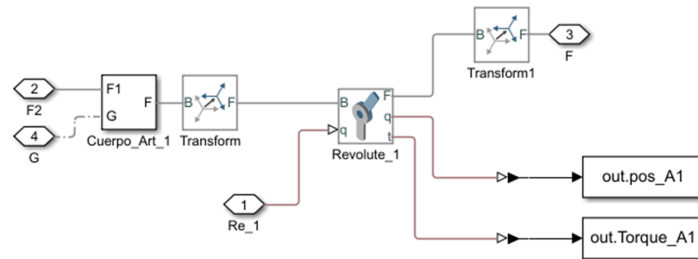


Figura 3.5: Diagrama de bloques del primer cuerpo.

Siguiendo la misma configuración del diagrama de bloques que se describió anteriormente, se simularon dos versiones iniciales del robot tipo serpiente. En cada uno de ellas se programaron diferentes modos de locomoción adaptados a este tipo de mecanismo. Estas ecuaciones preliminares fueron tomadas de trabajos previos y estudiadas mediante la revisión del estado del arte. Con esta primera aproximación del funcionamiento del robot, se demostró que diferentes modos de locomoción se podían crear con la ecuación 3.1.

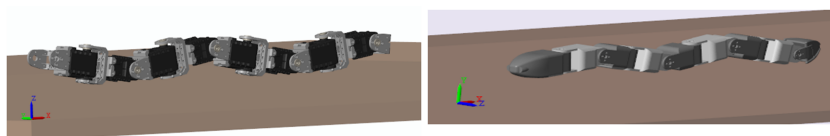


Figura 3.6: Simulación de locomoción en las primeras versiones del robot en Simulink.

Además de cada una de las simulaciones que se realizaron, se obtuvo un valor máximo del par al realizar las mediciones en todas las articulaciones, valor que se tuvo en cuenta al momento de elegir el actuador para el robot serpiente. De la Figura 3.7 se puede observar que el par máximo es aproximadamente 0.48 Nm para la sexta articulación, la cual es la articulación que más par exige durante el modo de marcha *sidewinding*.

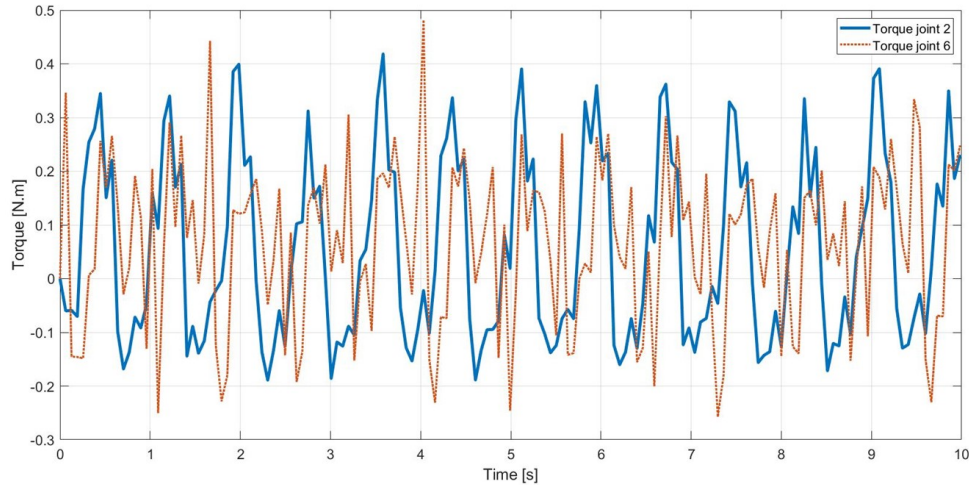


Figura 3.7: Torques de las articulaciones 2 y 6.

### Selección del Actuador

Se pueden usar muchos tipos diferentes de accionamiento para los robots serpiente, como neumáticos, motores eléctricos, servos o ruedas motrices [26]. Para este proyecto en específico se consideraron servomotores.

Teniendo en cuenta los resultados de las simulaciones mostradas en la Figura 3.7 respecto a los pares máximos requeridos, se eligió el servomotor Dynamixel AX-12, el cual cuenta con un microprocesador incorporado para proporcionar una comunicación en serie TTL o RS-485 y una velocidad de conexión en cadena de 1-3 Mbps. También cuenta con velocidad de par ajustable y control de respuesta con retroalimentación de posición, carga, voltaje, velocidad y temperatura, lo que permite la implementación de un sistema de control con relativa facilidad. Esta referencia de servomotores funciona a 12V, pesan 55 g y proporcionan un torque de 1,47 Nm, dando un buen margen por encima de los 0.48Nm requeridos en la simulación.

#### 3.2.2. Fuente de alimentación

El robot serpiente se alimenta de una fuente de 12 V a través de un cable de 1.5 m, el cual va hasta el módulo de cola. En un comienzo del proyecto se consideró el uso de baterías, pero estas darían una autonomía de solo 30 – 45 minutos aproximadamente. Con

este tipo de alimentación se pudieron realizar diversas pruebas y usarse mediante periodos prolongados, gracias a su fuente externa de voltaje constante.

### 3.2.3. Sensor

El robot serpiente está diseñado para operar en diversos entornos donde no hay garantía de que el GPS pueda funcionar y, como tal, no es una opción viable. Debido a lo anterior, se agregó una IMU BNO055 de la marca Adafruit, la cual contiene un giroscopio de 3 ejes, un acelerómetro de 3 ejes y un magnetómetro de 3 ejes. El sistema IMU está atado al módulo de la cabeza a través de un marco especialmente diseñado para garantizar su conexión rígida y precisión de instalación. Con el fin de permitir una estimación de estado más precisa, la instalación se realiza en la cabeza del robot, teniendo en cuenta que este es un módulo pasivo que no cuenta con un servomotor y no genera un campo magnético que pueda afectar la lectura de la IMU.

### 3.2.4. Módulo de comunicación

El uso de la comunicación inalámbrica agrega complejidad al sistema, ya que debe mantener una conexión confiable con la computadora principal, independientemente del entorno. Como los servomotores elegidos se pueden comunicar a través de un bus serial TTL mediante el mismo cable usado para su alimentación, se decidió implementar dicho modelo de comunicación, siendo este más fácil de realizar y teniendo la ventaja de una conexión confiable en todo momento de la ejecución de los algoritmos de locomoción.

Para llevar a cabo la comunicación de la computadora con los servomotores, fue necesario el uso de un convertidor de USB a TTL. El elegido para esta tarea fue el U2D2 de la marca Robotis, siendo este dispositivo capaz de controlar y operar los servomotores Dynamixel.

Los datos del sensor se procesan de forma remota en una computadora, por lo que también fue necesario el uso de un microcontrolador Arduino nano.

### 3.2.5. Material para el chasis

La opción más eficiente para fabricar el chasis fue mediante impresión 3D, siendo el material elegido termoplástico PLA, dado que presenta un menor tiempo de producción y permite una mayor personalización en el diseño de los componentes del robot. Además, como complemento para generar mayor fricción entre los módulos y el ambiente, se decidió usar una capa adicional en la parte inferior de cada uno de ellos. Se optó por usar pedazos de cinta antideslizante para la obtención de una mayor fricción entre el cuerpo del robot y la superficie de cada uno de los entornos de prueba.

### 3.3. Diseño CAD

Los módulos del robot serpiente se diseñaron en el software SolidWorks para posteriormente ser impresos en 3D. Cada módulo tiene como objetivo funcionar como una articulación rotacional con un grado de libertad, como se puede observar en la Figura 3.8. El módulo consta de una carcasa, un servomotor Dynamixel y un canal interno que permite que los cables de comunicación y alimentación pasen formando un bus serial.

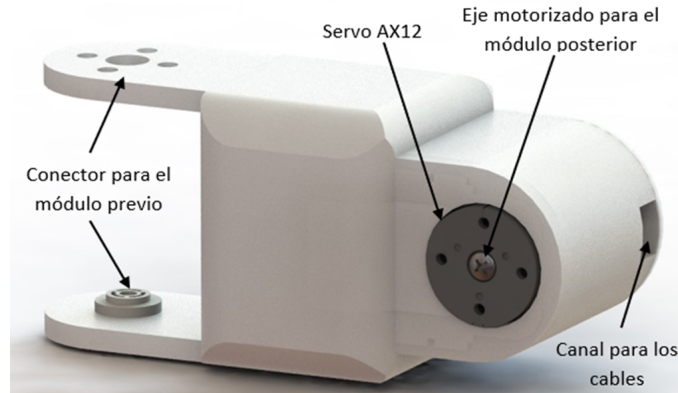


Figura 3.8: Módulo del robot serpiente.

Cada módulo del robot se conecta al módulo anterior de tal forma que el eje de giro sea perpendicular, permitiendo un giro  $\pm 90^\circ$  respecto a su predecesor, lo cual puede observarse en la Figura 3.9. Esta configuración nos permite tener dos ejes de movimiento en cada par de módulos lo que se traduce en que el robot pueda usar modos de marcha en tres dimensiones. El diseño actual consta de un módulo de cabeza, uno de cola y 12 módulos simétricos, que se pueden modificar según la aplicación. En términos dimensionales el robot mide 113 cm de largo, con un diámetro aproximado de 5cm, y un peso total de 2 Kg.



Figura 3.9: Prototipo del robot serpiente.

### 3.4. Simulador

Primero se llevó a cabo un estudio para comparar diferentes softwares de simulación, cuyos resultados se encuentran en el apéndice B. El objetivo de este estudio fue seleccionar el software más adecuado para simular los movimientos del robot serpiente en un

entorno controlado que se asemejara a un ambiente real. Después de la evaluación, se eligió el simulador de robótica Webots, desarrollado por Cyberbotics Ltd. Este software ofrece múltiples herramientas para el prototipado rápido, modelado, programación y simulación de robots móviles. En él, es posible definir parámetros físicos como la matriz de inercia, densidad, textura, masa y fricción para cada objeto añadido al programa. Además, se puede equipar al robot con diversos sensores y actuadores.

La forma del robot se importó directamente de los modelos CAD diseñados en el software SolidWorks (Figura 3.10). Se incluyeron en el robot simulado sus propiedades físicas, la velocidad máxima, y el par de los motores. El modelo dispone también de sensores tipo IMU y GPS para obtener la posición global exacta del robot. Webots permitió evaluar rápidamente su desempeño, y en particular su comportamiento sobre terreno totalmente llano.

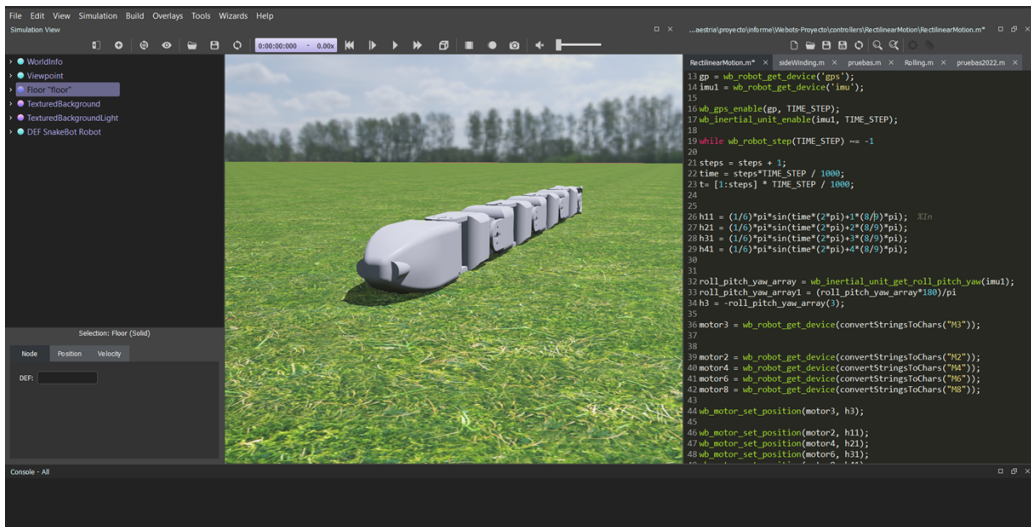


Figura 3.10: Simulación del robot en el software Webots.

### 3.5. Modos de locomoción

En este proyecto se tendrán en cuenta los tres de los cuatro modos de locomoción naturales y uno no bioinspirado como lo es el rolling lateral. Los tres movimientos bioinspirados serán: *sidewinding*, ondulación lateral y rectilíneo, modos que permiten al robot adaptarse a diferentes entornos. La marcha concertina o acordeón, la cual es el cuarto modo de locomoción bioinspirada, no se consideró en este trabajo debido a que ella genera un movimiento intermitente a lo largo de la estructura, lo que no permite que el momento se conserve, por lo tanto, el movimiento es lento y no es eficiente en términos de consumo de energía [68].

Además, para cada uno de los modos de marcha bioinspirados se tendrán en cuenta dos conjuntos de parámetros con el fin de obtener diferentes condiciones en la locomoción, como



lo son la amplitud, formas de las curvas y puntos de apoyo, las cuales permiten analizar cómo puede afectar la estabilidad del robot variando la velocidad final de mecanismo sin ayuda de algoritmos de control y sensores adicionales que adquieran datos del terreno.

### 3.5.1. Modo de locomoción rectilínea

La marcha rectilínea se define como una onda plana que viaja rostralmente (que remata en punta), y es inspirada en las marchas observadas en las orugas. La propagación de una onda de cuerpo sinusoidal vertical desde la cola hasta la cabeza genera una progresión hacia adelante de la serpiente robótica. En consecuencia, la componente de onda de cuerpo horizontal o motores impares representados en la ecuación 3.1,  $\alpha_h(n, t)$  está restringida a una línea recta a lo largo de una abscisa paralela al plano de tierra. El componente de onda de cuerpo vertical,  $\alpha_v(n, t)$ , se define como una senoide que varía espacial y temporalmente.

La amplitud de onda vertical ( $A_v$ ), la frecuencia espacial ( $\delta_v$ ) y la frecuencia temporal ( $\omega_v$ ) son parámetros ajustables que definen la marcha rectilínea. En la Figura 3.11, se ilustra la onda del cuerpo en movimiento. El componente de onda de cuerpo horizontal,  $\alpha_h$ , es una línea recta parametrizada alineada con la dirección de viaje prevista. El componente de onda de cuerpo vertical,  $\alpha_v$ , es la senoide viajera parametrizada a lo largo de la onda de cuerpo horizontal.

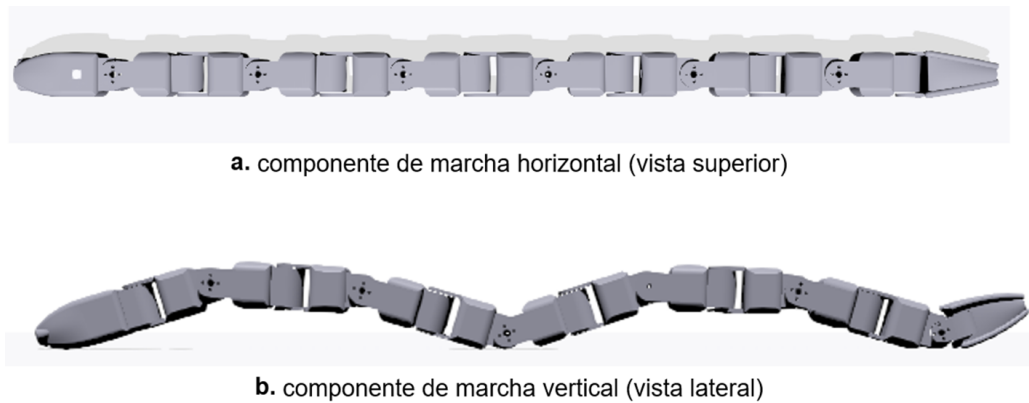


Figura 3.11: Modelo de robot serpiente de 12 eslabones ajustado a la locomoción rectilínea.

Los valores de los parámetros usados para dos velocidades del modo de locomoción rectilínea, para el robot serpiente, son presentados en la tabla 3.1.

### 3.5.2. Modo de locomoción ondulación lateral

En este modo de marcha se incorporó el levantamiento de los senos nasales de la serpiente. Para emplear productivamente la ondulación lateral, la magnitud de las fuerzas netas dirigidas rostralmente debe exceder las fuerzas contraproducentes producidas en los segmentos de la curvatura máxima del cuerpo. El levantamiento de senos nasales, tal como lo utilizan las serpientes, implica elevar segmentos de la curvatura máxima del cuerpo

Modo de Marcha Rectilínea	Parámetros			
	Amplitud	F. Temporal	F. Espacial	Dif. de fase
<b>Velocidad 1</b>	$A_h = 0$	$\omega_h = 0$	$\delta_h = 0$	$\phi = 0$
	$A_v = \frac{1}{6}\pi$	$\omega_v = 2\pi$	$\delta_v = \frac{8}{9}\pi$	
<b>Velocidad 2</b>	$A_h = 0$	$\omega_h = 0$	$\delta_h = 0$	$\phi = 0$
	$A_v = \frac{1.3}{6}\pi$	$\omega_v = \frac{9}{5}\pi$	$\delta_v = \frac{6}{9}\pi$	

Tabla 3.1: Parámetros para los modos de locomoción rectilínea.

fuera de la superficie de locomoción para minimizar las fuerzas contraproducentes.

Los componentes de onda corporal horizontal y vertical que definen la ondulación lateral con elevación del seno se ilustran en la Figura 3.12.

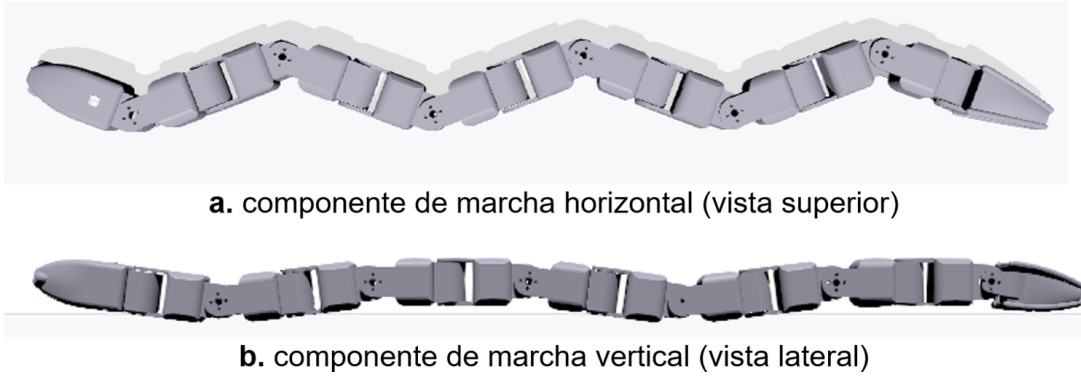


Figura 3.12: Modelo de robot serpiente de 12 eslabones ajustado a la locomoción de ondulación lateral.

Los valores de los parámetros que se usarán para la ejecución de dos velocidades del modo de locomoción de ondulación lateral, para el robot serpiente, son presentados en la tabla 3.2.

Modo de Marcha Ond. Lateral	Parámetros			
	Amplitud	F. Temporal	F. Espacial	Dif. de fase
<b>Velocidad 1</b>	$A_h = \frac{1}{3}\pi$	$\omega_h = \frac{1}{3}\pi$	$\delta_h = \pi$	$\phi = 0$
	$A_v = \frac{1}{9}\pi$	$\omega_v = 3\pi$	$\delta_v = \frac{2}{3}\pi$	
<b>Velocidad 2</b>	$A_h = \frac{2}{9}\pi$	$\omega_h = \frac{2}{3}\pi$	$\delta_h = \pi$	$\phi = 0$
	$A_v = \frac{2}{15}\pi$	$\omega_v = \frac{3}{2}\pi$	$\delta_v = \frac{2}{3}\pi$	

Tabla 3.2: Parámetros para los modos de locomoción de ondulación lateral.

### 3.5.3. Modo de locomoción *sidewinding*

Se define el componente de onda del cuerpo horizontal de la marcha lateral como una senoide que varía en el tiempo y en el espacio. Y la componente de onda del cuerpo vertical se parametriza con respecto a lo largo de la onda del cuerpo horizontal del robot.

La frecuencia de la onda de cuerpo vertical,  $\omega_v$ , se iguala a la frecuencia de la onda del cuerpo horizontal,  $\omega_h$ . asegurando que la frecuencia espacial de la onda del cuerpo vertical sinusoidal encaje dentro de la frecuencia espacial de la onda del cuerpo horizontal.

En la figura 3.13 se muestran tanto los componentes de la onda del cuerpo como el ajuste resultante del modelo de serpiente robótica, así como también el desplazamiento de fase de  $\phi = -\frac{\pi}{4}$ , en la onda de cuerpo vertical. Además, se alinean los componentes de onda horizontal y vertical de tal manera que los segmentos de la onda del cuerpo horizontal obtengan dos pendientes 'hacia arriba'.



Figura 3.13: Modelo de robot serpiente de 12 eslabones ajustado a la locomoción *sidewinding*.

Los valores de los parámetros que se usarán para la ejecución de dos velocidades del modo de locomoción *sidewinding*, para el robot serpiente, son presentados en la tabla 3.3.

Modo de Marcha Sidewinding	Parámetros			
	Amplitud	F. Temporal	F. Espacial	Dif. de fase
Velocidad 1	$A_h = \frac{1}{18}\pi$	$\omega_h = \pi$	$\delta_h = \frac{2}{3}\pi$	$\phi = -\frac{1}{4}\pi$
	$A_v = \frac{5}{36}\pi$	$\omega_v = \pi$	$\delta_v = \frac{2}{3}\pi$	
Velocidad 2	$A_h = \frac{1}{6}\pi$	$\omega_h = \pi$	$\delta_h = \frac{2}{3}\pi$	$\phi = -\frac{1}{4}\pi$
	$A_v = \frac{5}{36}\pi$	$\omega_v = \pi$	$\delta_v = \frac{2}{3}\pi$	

Tabla 3.3: Parámetros para los modos de locomoción *sidewinding*.

### 3.5.4. Modo de locomoción Rolling lateral

El Rolling lateral es un modo de andar que no se encuentra en las serpientes biológicas, pero es un modo simple y eficaz para los robots con forma de serpiente, demostrado por primera vez por Kevin Dowling [69]. La marcha de Rolling lateral requiere que el robot se flexione periódicamente en un arco y se enderece como se puede observar en la figura 3.14, lo que da como resultado un movimiento lateral. Esta locomoción es también un movimiento tridimensional, pero sin la ondulación que se encuentra en el modo *sidewinding*. La curva del cuerpo de esta forma de andar se describe como la composición de la rotación para el componente de onda del cuerpo horizontal y vertical, siendo la diferencia de fase entre las dos curvas de  $\phi = \frac{\pi}{2}$ .

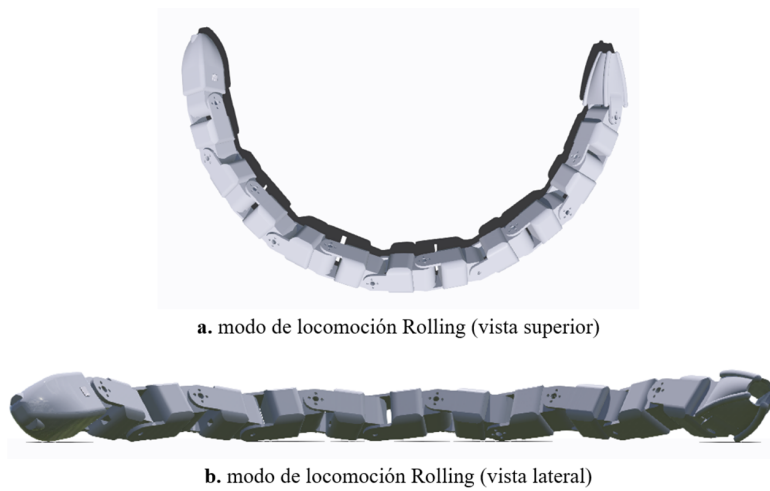


Figura 3.14: Modelo de robot serpiente de 12 eslabones ajustado a la locomoción Rolling lateral.

EL modo de locomoción Rolling se puede modificar para trepar postes o rodar por una tubería formando una hélice. Pero a pesar de que este modo sea eficiente y veloz, en este trabajo debido a las limitaciones mecánicas y de diseño solo se probará la manera más simple de este movimiento el cual es el Rolling lateral. Ya que al ejecutar este modo de locomoción el cable de alimentación y el de comunicación conectados en la cola se enrollan y tensionan dificultando la puesta en marcha del movimiento. Teniendo en cuenta lo anterior, se probará solo una combinación de parámetros y se llevará a cabo en una superficie llana y controlada. Los valores de los parámetros que se usarán para la ejecución del modo de locomoción Rolling lateral, para el robot serpiente, son presentados en la tabla 3.4.

Modo de Marcha Rolling lateral	Parámetros			
	Amplitud	F. Temporal	F. Espacial	Dif. de fase
Velocidad 1	$A_h = \frac{1}{6}\pi$	$\omega_h = \pi$	$\delta_h = 0$	$\phi = \frac{1}{2}\pi$
	$A_v = \frac{1}{6}\pi$	$\omega_v = \pi$	$\delta_v = 0$	

Tabla 3.4: Parámetros para los modos de locomoción Rolling lateral.

# Capítulo 4

## Presentación de resultados

Esta sección se establece el diseño y protocolo de experimentación y se describen los entornos donde se ejecutarán los modos de marcha del robot sobre un terreno regular y tres tipos de terrenos irregulares. Por último, se condensa la información adquirida de los mismos, para su posterior análisis.

### 4.1. Parámetros del experimento

Probar robots serpiente con un banco de pruebas real es especialmente importante debido a la complejidad del propio robot y la aleatoriedad de los terrenos previstos. Debido a lo anterior, se realizaron experimentos en un terreno regular en los que se ejecutaron los cuatro modos de locomoción: rectilíneo, *sidewinding*, ondulación lateral y Rolling lateral. De esta manera, para cada uno de los modos de marcha bio-inspirados se ejecutarán dos configuraciones donde se varían los parámetros de la ecuación de la curva serpentinoide con el fin de obtener dos diferentes tipos de velocidades y solo un tipo de configuración se llevará a cabo para el modo de marcha rolling lateral.

Para el análisis en terreno irregular solo se utilizaron los tres movimientos bioinspirados. Además, para evaluar y comparar el funcionamiento del robot, estos movimientos se repitieron en tres tipos de superficies distintas.

Una vez realizados todos los experimentos, se cotejan estadísticamente los datos para hallar los promedios respectivos al número de experimentos realizados. En las siguientes secciones se presentarán una serie de cuadros y se muestran también graficas con los resultados más relevantes de los experimentos ejecutados en los diferentes entornos. Los cuadros mostrarán los valores de la distancia recorrida, tiempo, velocidad y ángulos de desviación.

#### 4.1.1. Configuración de las pruebas para el terreno regular

La configuración para el experimento del terreno regular, tanto en simulación como con el robot real, se da en una superficie totalmente llana como se muestra en la Figura 4.1. El robot serpiente cuenta con una IMU BNO055 en la cabeza, la cual le ayudará a verificar la desviación o ángulo de rumbo durante la ejecución de los modos de marcha. El robot serpiente se ubica en el mismo punto en cada inicio de experimento, siendo la condición inicial para cada uno de los servomotores que su ángulo sea igual a cero.

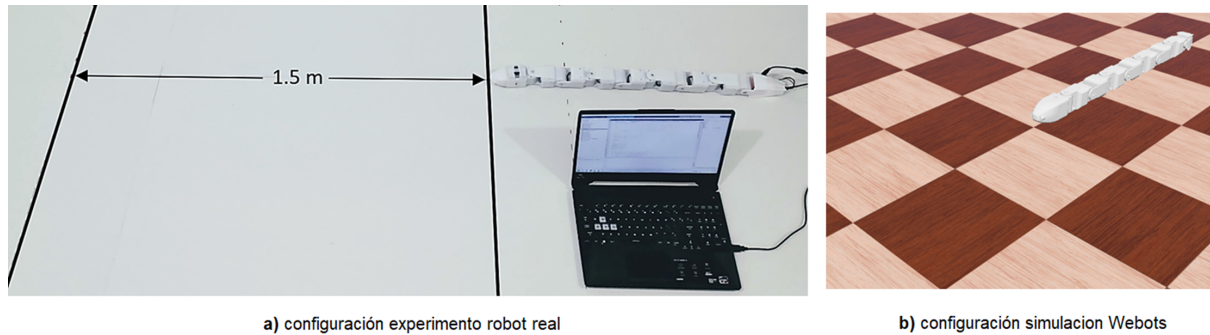


Figura 4.1: Configuración para experimento terreno regular.

Se realizaron diez pruebas por cada combinación de marcha y configuración de parámetros **obteniendo un total de 140 ejecuciones realizadas (70 en simulación y 70 con el robot real)**. La distancia desde el punto inicial hasta el punto final fue de 1,5 m para los modos de locomoción rectilíneo, ondulación lateral y Rolling lateral. Para el modo de marcha *sidewinding* las distancias del experimento se modificaron teniendo en cuenta que en este tipo de locomoción el robot se mueve en diagonal en cada ciclo de marcha y al modificar los parámetros de la ecuación 3.1 este se desplaza con una mayor velocidad hacia un lado, lo que generó limitaciones en la dimensión en algunos de los terrenos seleccionados. Por lo anterior, se estandarizó que para la configuración de velocidad 1 la distancia que recorrería el robot hacia adelante sería de 1,2 m y de 0,5 m para la configuración de velocidad 2.

Como se mencionó previamente, las métricas para evaluar el desempeño del robot requieren solo de la medida del tiempo y ángulos de desviación. Además, la velocidad se obtiene mediante el cálculo de la distancia conocida en cada uno de los experimentos y el tiempo que le tomo al robot en terminar cada prueba.

#### 4.1.2. Configuración de las pruebas para el terreno irregular

La configuración de distancias para las pruebas en terreno irregular fue idéntica a las mencionadas en las pruebas de terreno regular. También se realizaron 10 experimentos por cada combinación de parámetros en los modos de marcha, que en este caso solo serán las bioinspiradas: Rectilínea, *sidewinding* y ondulación lateral. Estos experimentos se repitieron en tres tipos de superficies diferentes: hormigón, tierra y césped, cada una de las superficies se ilustran en la figura 4.2. **Obteniendo un total de 180 ejecuciones realizadas.**

- La superficie de hormigón se compone de una textura relativamente uniforme, con pequeños granos y algunas grietas.
- La superficie en tierra consiste en una mezcla de grava, rocas pequeñas, ramas y una pequeña inclinación en todo el recorrido.
- En la superficie césped se puede observar un pasto cortado a diferentes alturas y se tiene la presencia de una inclinación en todo el entorno.

Todos los ambientes donde se realizaron las pruebas de locomoción eran superficies previamente existentes.

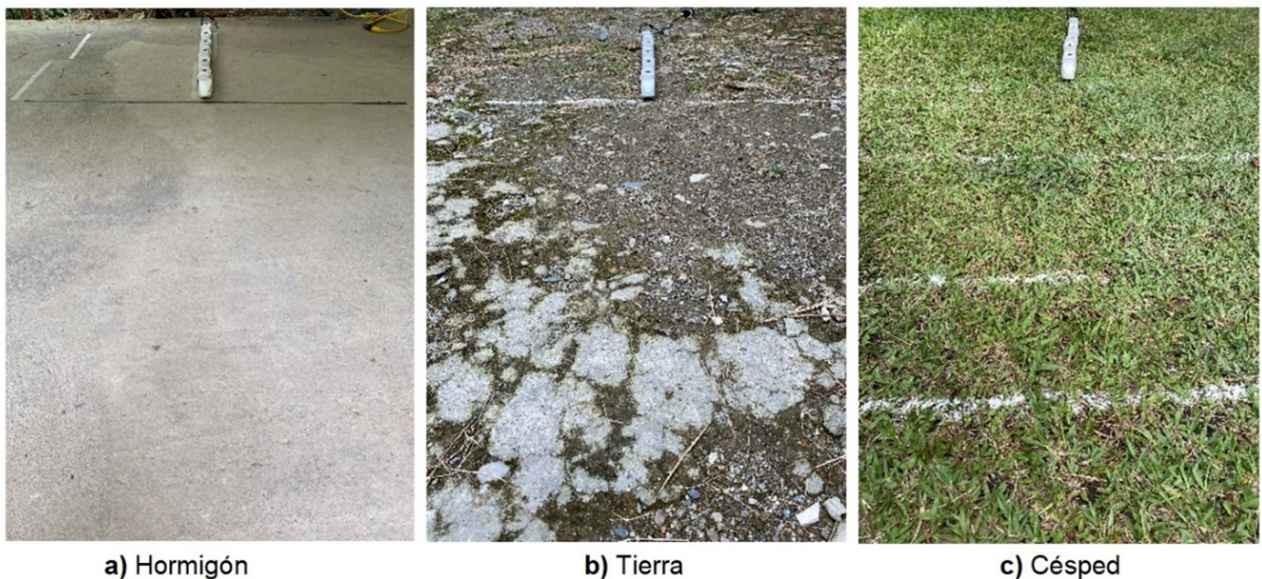


Figura 4.2: Entornos de los experimentos en terreno irregular.

## 4.2. Resultados locomoción rectilínea

### 4.2.1. Terreno regular

El movimiento rectilíneo es común entre animales como las orugas y boas. En la Figura 4.3 se muestran las curvas que forma el cuerpo del robot al ejecutarse las dos configuraciones previstas en este tipo de marcha, donde se puede observar que la amplitud de la curva del componente vertical que se forman con los parámetros de la segunda configuración es mayor. Este cambio de los parámetros en la ecuación 3.1, conlleva a tener un punto menos de apoyo en cada ciclo de marcha, pasando de 4 a 3 puntos de contacto.

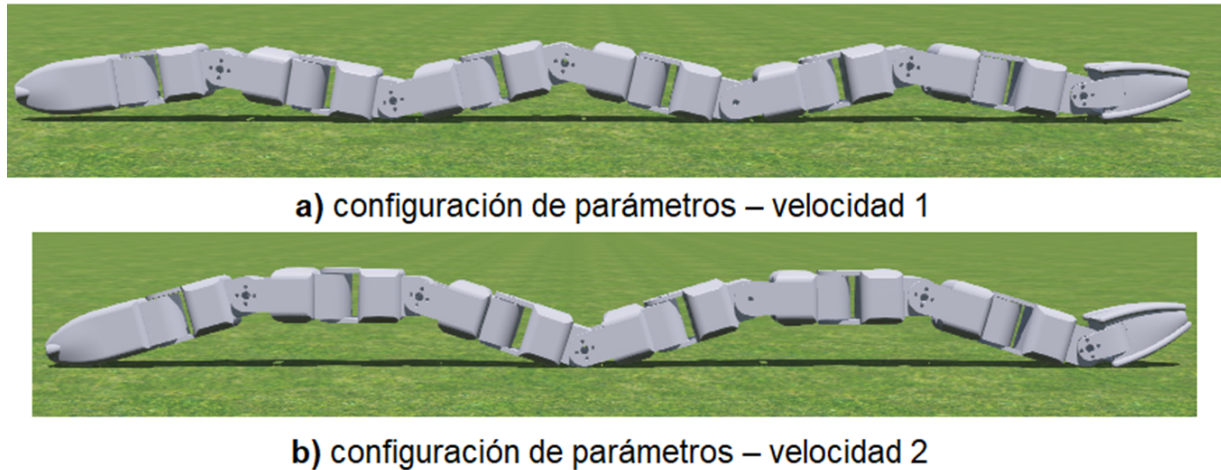


Figura 4.3: Forma de onda de los tipos de marcha rectilínea.

Aunque este tipo de locomoción bidimensional no es muy lenta, su movimiento es poco eficiente al solo usar el componente vertical de los motores, pero dicho modo es común cuando un robot serpiente debe pasar a través de tuberías estrechas de forma circular o de tipo lineal, en las que es difícil realizar otras formas de marcha.

En la Figura 4.4 se presentan los resultados de seguimiento de la posición de la cabeza del robot, en simulación, para las dos configuraciones de velocidad.

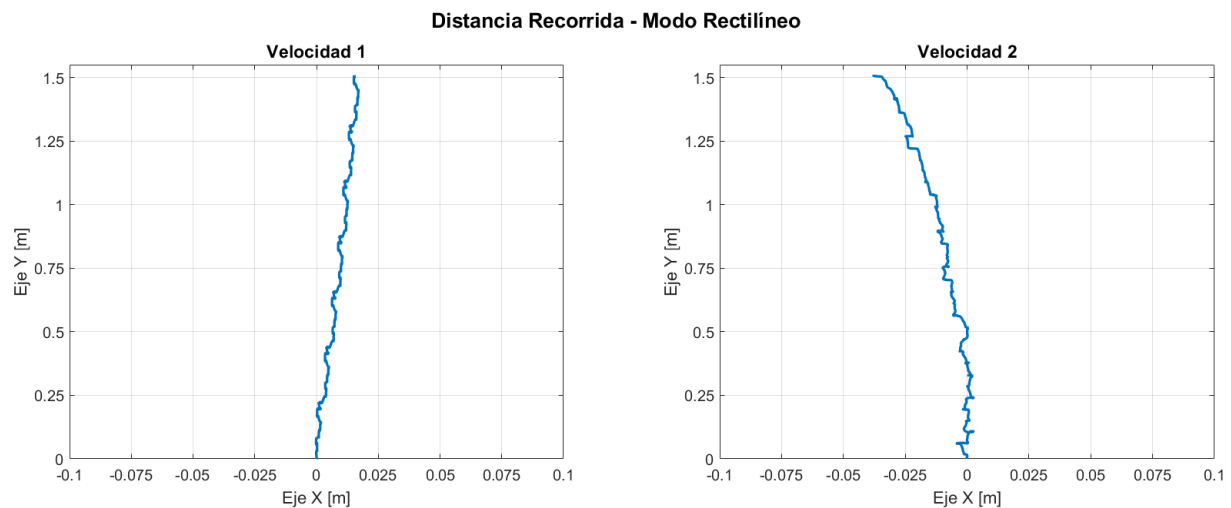


Figura 4.4: Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha rectilínea.

De la figura anterior se puede observar que el robot no tiene un comportamiento totalmente rectilíneo, presentándose un ligero desvío hacia la derecha de 1.6 cm sobre el eje X para la velocidad 1 y un desvío hacia la izquierda de 3.8 cm para la velocidad 2, las cuales



son distancias insignificantes en recorridos cortos.

Posteriormente se llevaron a cabo los 10 experimentos para cada una de las dos configuraciones, con el robot real. Dado que este robot no cuenta con un sensor o un sistema de captura de movimiento, los datos del desplazamiento se midieron de manera manual y posteriormente se comprobó el sentido y se realizó una aproximación del desvío con el ángulo entregado por la IMU. Con el robot real se observa también una pequeña desviación en cada una de las pruebas realizadas, donde para las pruebas con la velocidad 1 este desvío fue menor a  $\pm 3$  cm y para la velocidad 2 los desvíos siempre fueron negativos, es decir el robot terminaba con una inclinación hacia la izquierda en un rango de 5 – 10 cm. Las anteriores afirmaciones se pueden verificar con la información presentada en la Figura 4.5, donde se muestra el ángulo Yaw o ángulo de rumbo, medido por la IMU ubicada en la cabeza del robot respecto al eje vertical perpendicular a éste. La figura presenta datos de la simulación y de 5 de las 10 pruebas realizadas con el robot real.

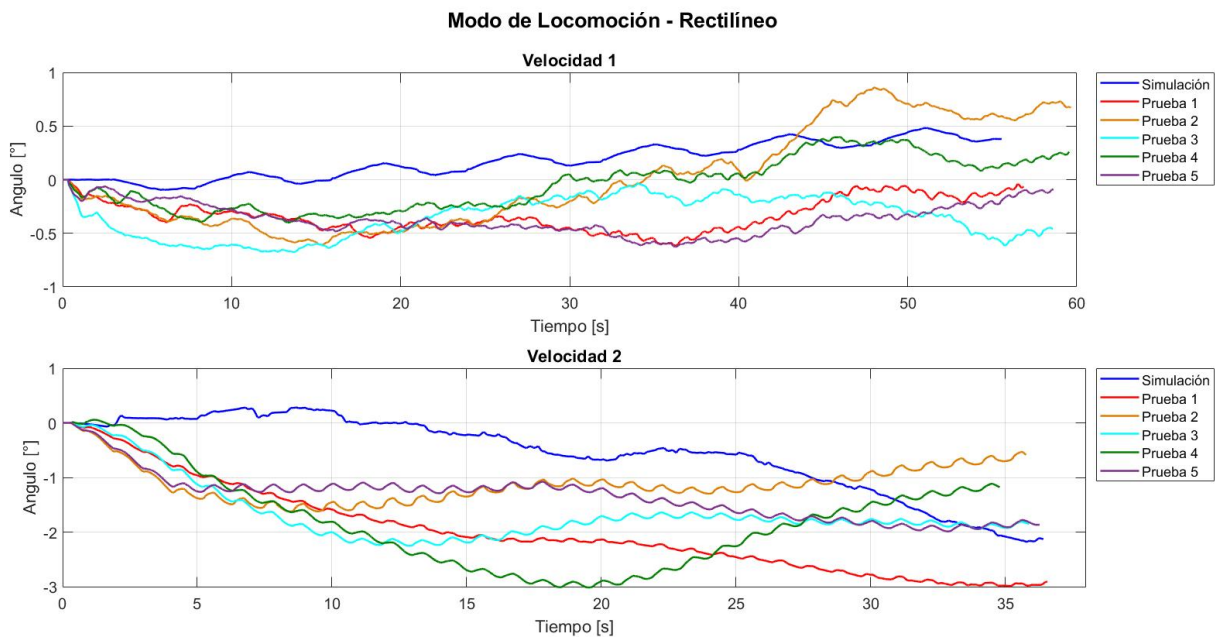


Figura 4.5: Ángulo Yaw (rumbo) para el patrón de marcha rectilíneo en terreno regular.

Se puede observar que el ángulo de giro en simulación se mantiene relativamente constante con un valor cercano a cero durante los primeros 20 segundos y después comienza a inclinarse a la derecha con un ángulo promedio de  $0.38^\circ$  para la primera configuración y de  $-2.13^\circ$  para la segunda configuración. En el caso del robot real los ángulos presentan una mayor oscilación, pasando a ángulos de desviación con medias de  $-1.38^\circ \pm 1.15$  y  $-2.57^\circ \pm 1.74$  para la primera y segunda configuración de velocidad. Los resultados en términos de la desviación cuentan con valores pequeños y cercanos al valor de simulación, los cuales no afectan al recorrido del robot.

Esta desviación puede ser consecuencia del diseño y construcción de cada uno de los bordes redondeados de los cuerpos del robot, así como del esfuerzo que hace el robot al arrastrar la fuente de alimentación externa. Además, durante este modo de marcha los motores de las articulaciones impares no deben moverse, conservando en todo momento una consigna igual a cero, lo cual actualmente no se está controlando de manera precisa en el algoritmo de locomoción.

En la tabla 4.1 se presenta la comparación de los promedios del tiempo, velocidad y ángulos de rumbo de la ejecución de todas las pruebas realizadas en una superficie llana, para el modo de locomoción rectilínea. Además se toma como referencia los resultados de simulación para calcular el error del tiempo y velocidad. Donde se observa que la velocidad en las ejecuciones para la primera configuración de parámetros fue 2.96% más lenta respecto a la simulación y solo un 0.36% más rápido para la segunda configuración.

Modo de Marcha rectilínea Terreno Regular		Simulación		Robot real		
		Promedio	Desviación	Promedio	Desviación	Error (%)
Velocidad 1 $d_y = 150cm$	Tiempo (s)	55.55	0.020	57.33	2.283	3.20
	Velocidad (cm/s)	2.70	0.001	2.62	0.102	-2.96
	Ángulo Yaw - rumbo (°)	0.38	0.001	-1.38	1.154	—
Velocidad 2 $d_y = 150cm$	Tiempo (s)	36.42	0.000	36.29	0.611	-0.33
	Velocidad (cm/s)	4.12	0.000	4.13	0.069	0.36
	Ángulo Yaw - rumbo (°)	-2.13	0.002	-2.57	1.74	—

Tabla 4.1: Comparación de resultados de la locomoción rectilínea en terreno regular.

Aunque existen varias fuentes de error que influyen en los datos adquiridos por el robot real, vemos que los gráficos que comparan la simulación y los resultados experimentales proporcionan una descripción satisfactoria del sistema antes de validar el robot propuesto. Además, el software de simulación y los parámetros físicos incluidos en él forman un modelo bastante preciso del robot serpiente.

### 4.2.2. Terreno irregular

A continuación, en la figura 4.6 se presentan los resultados aleatorios de 5 experimentos de los 10 realizados, para cada combinación de configuración de velocidad y terreno.

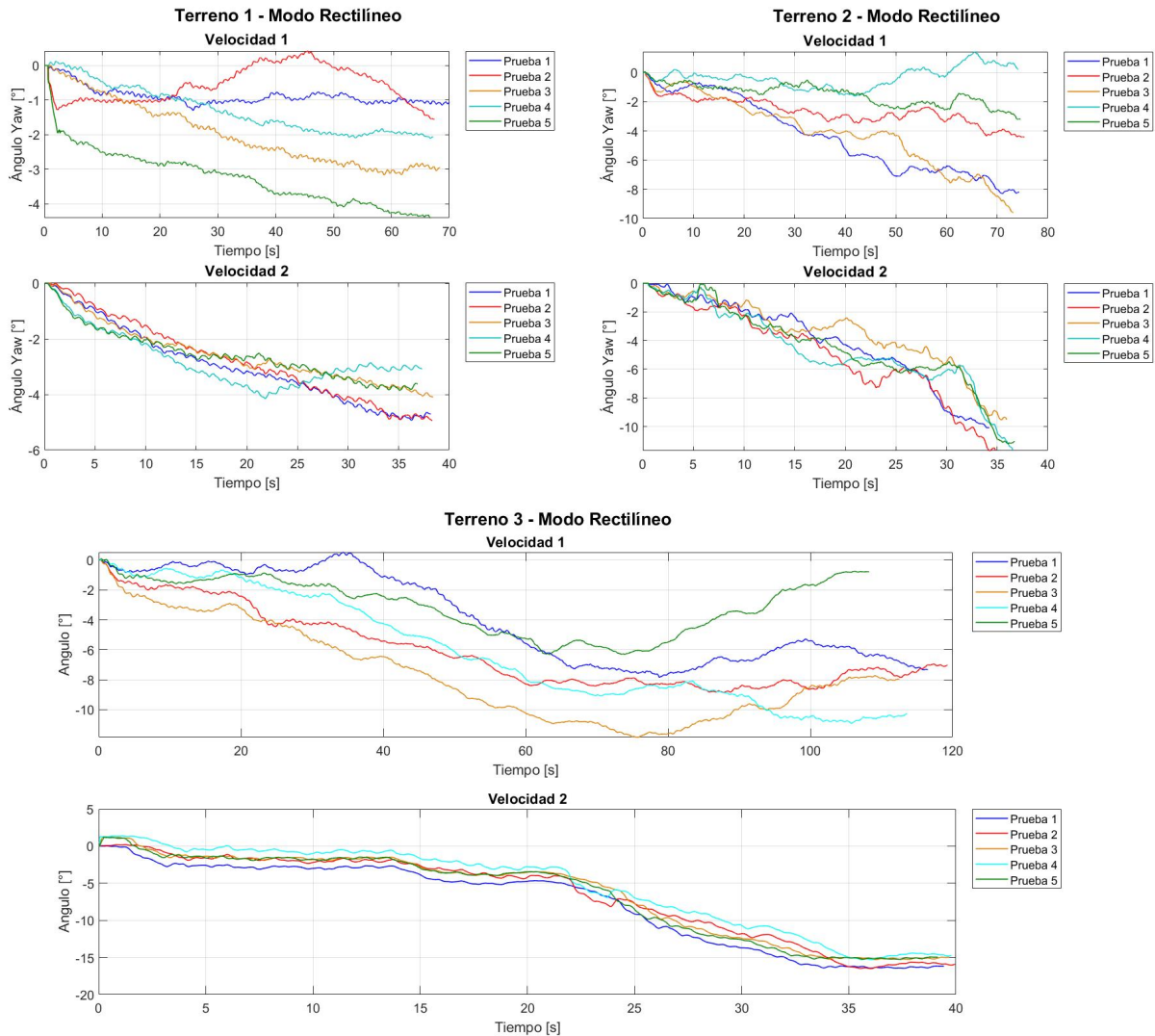


Figura 4.6: Ángulo Yaw (rumbo) para el patrón de marcha rectilíneo en terreno irregular.

De los anteriores resultados se puede concluir que para la configuración de velocidad 2, al tener una amplitud mayor en cada ciclo de marcha, los pequeños obstáculos como rocas, ramas o la inclinación, presentes en los terrenos 2 y 3, los puede superar con facilidad obteniendo una velocidad promedio mayor que la primera configuración de parámetros, por ejemplo en el terreno 2 la velocidad pasó de  $2.02 \pm 0.06$  cm/s a  $4.29 \pm 0.24$  cm/s. Aunque se presenta una mayor velocidad este movimiento genera una mayor inestabilidad al contacto de los obstáculos más sobresalientes, obteniéndose como resultado unos ángulos de inclinación mayores con respecto a los resultados de la configuración con la velocidad

1 o los obtenidos en un terreno plano. Los ángulos de rumbo van aumentando con la dificultad del terreno, donde los menores desvíos se presentan en el entorno con hormigón y los mayores desvíos en el césped. Además, los ángulos que se observan una vez finalizado cada experimento es que con la segunda configuración sus resultados están más agrupados a la media, caso contrario a lo observado con los parámetros de la velocidad 1 que tienen una mayor desviación estándar.

En la tabla 4.2 se condensan los resultados promedios de tiempo, velocidad y ángulo de rumbo después de ejecutados todos los experimentos con la marcha rectilínea en los entornos irregulares propuestos.

Modo de Marcha Rectilínea Terreno Irregular		Terreno 1		Terreno 2		Terreno 3	
		Promedio	Desviación	Promedio	Desviación	Promedio	Desviación
Velocidad 1 $d_y = 150cm$	Tiempo (s)	67.26	2.030	74.43	2.36	116.82	5.342
	Velocidad (cm/s)	2.23	0.069	2.02	0.061	1.29	0.058
	Ángulo Yaw - rumbo (°)	-2.72	1.331	-5.66	3.185	-8.57	3.823
Velocidad 2 $d_y = 150cm$	Tiempo (s)	38.35	0.660	35.08	1.806	40.65	1.296
	Velocidad (cm/s)	3.91	0.068	4.29	0.241	3.69	0.116
	Ángulo Yaw - rumbo (°)	-4.43	0.630	-10.20	1.219	-14.78	0.752

Tabla 4.2: Comparación de resultados de la locomoción rectilínea en terreno irregular.

### 4.3. Resultados locomoción *sidewinding*

#### 4.3.1. Terreno regular

En esta sección se presentan los resultados obtenidos mediante la ejecución en simulación y en un entorno controlado sobre una superficie plana del patrón de marcha *sidewinding*. Este modo se incluyó debido a que se usa comúnmente en proyectos relacionados con robots serpientes. Por lo cual se quiere demostrar que utilizando la ecuación 3.1 y los parámetros descritos en la Tabla 3.3, se obtiene el movimiento deseado hacia adelante. Para lograr esto se coloca un valor distinto de 0 en la diferencia de fase  $\phi$ . Por ejemplo, para un valor de  $-\frac{\pi}{4}$ , el robot deberá avanzar hacia la izquierda con respecto a la dirección a la que apunta la cabeza del robot.

En la Figura 4.7 se muestran las ondas verticales y horizontales que forman el cuerpo del robot al ejecutar las dos configuraciones de velocidad para el tipo de locomoción *sidewinding*.

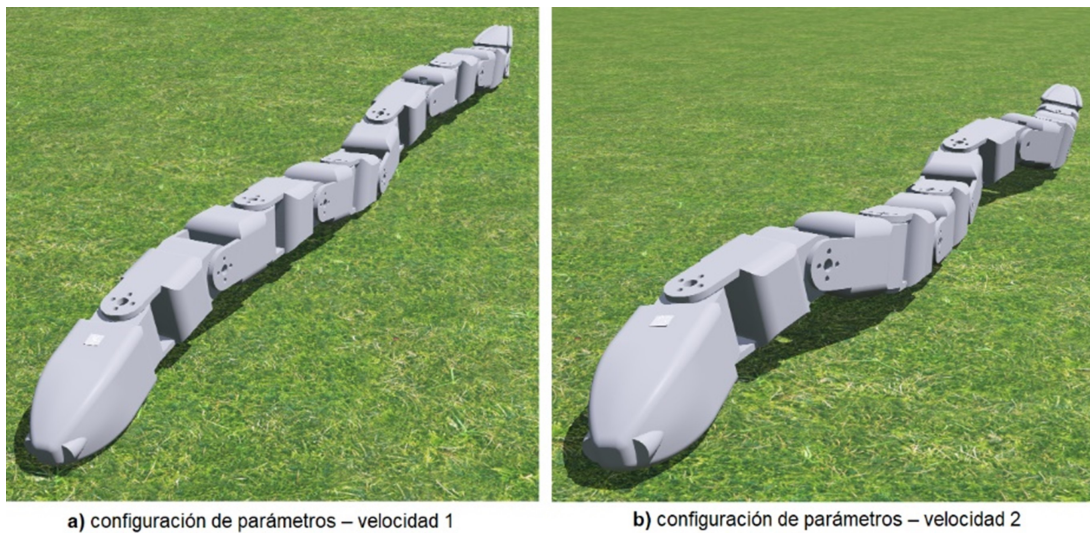


Figura 4.7: Forma de onda de los tipos de marcha *sidewinding*.

Para la simulación en Webots del tipo de locomoción *sidewinding*, el robot debía alcanzar una distancia sobre el eje Y de 1.20 m para la primera configuración de velocidad y 0.5 m para la segunda. El experimento se repitió 10 veces, en los cuales la distancia sobre el eje X que alcanzaba al completar cada prueba, eran aproximadamente las mismas, con un resultado de 118cm y 152cm para cada una de las configuraciones(ver Figura 4.8).

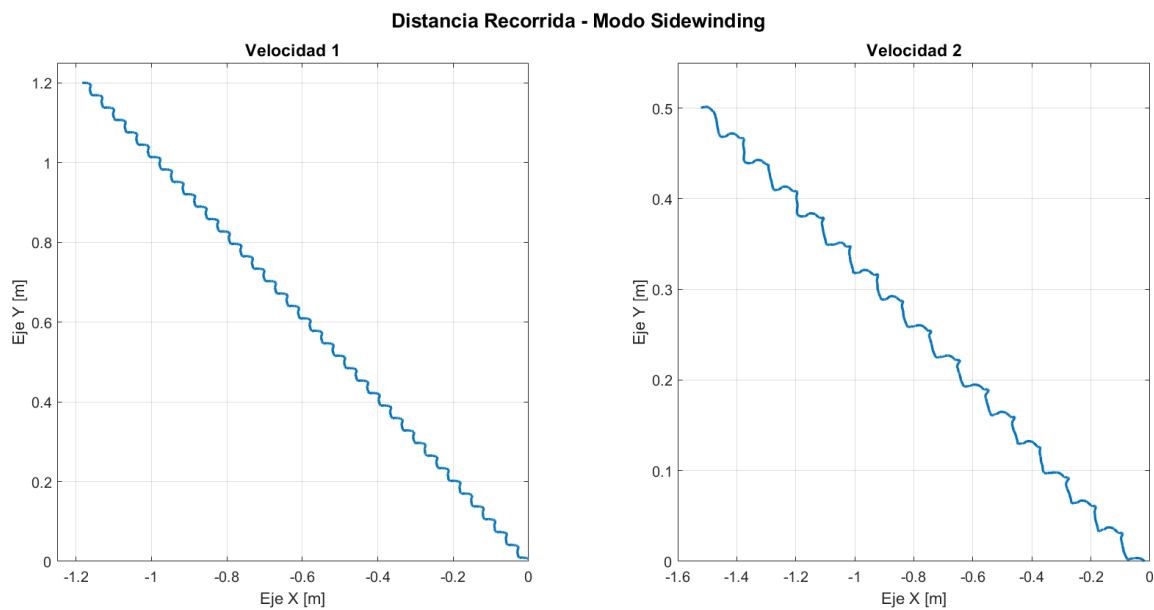


Figura 4.8: Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha *sidewinding*.

Se observa que el recorrido realizado por el robot es el esperado, realizando un desplazamiento hacia la izquierda de la cabeza del robot mientras avanza hasta alcanzar la distancia prevista para cada experimento sobre el eje Y. Además, se observa en la gráfica de la derecha que el robot avanza más rápido sobre el eje X debido a que la amplitud del componente de onda horizontal es mayor, esta comparación de los ciclos de marcha se puede evidenciar en la figura 4.7.

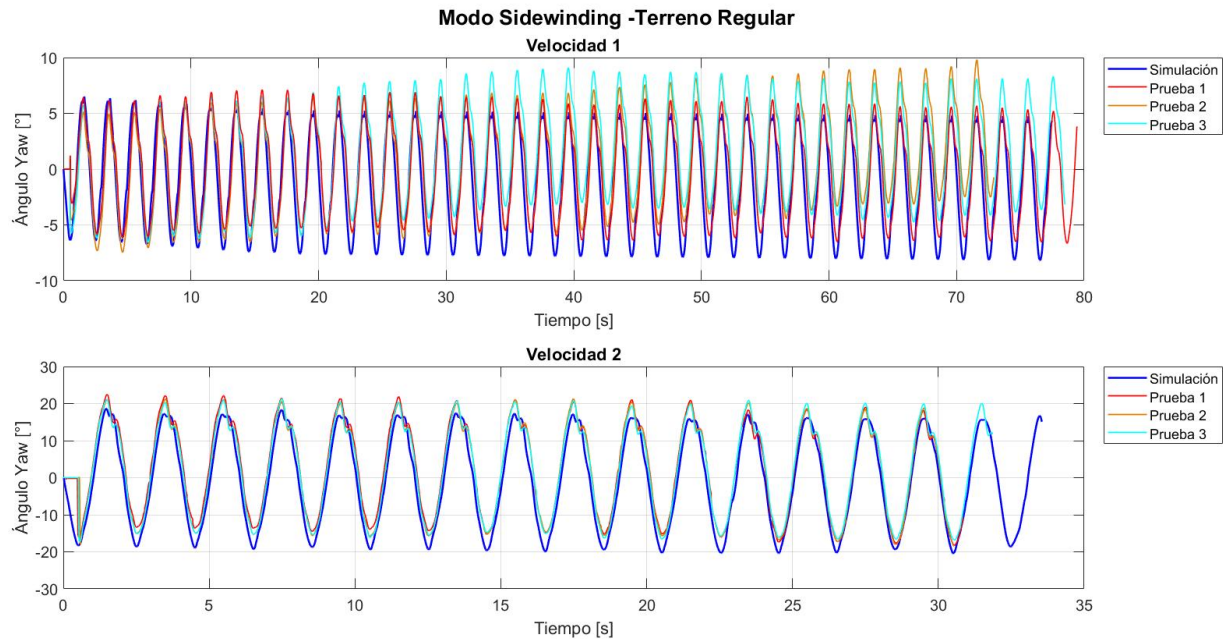


Figura 4.9: Ángulo Yaw (rumbo) para el patrón de marcha *sidewinding* en terreno regular.

En la Figura 4.9 se muestran las gráficas que comparan los valores simulados con los valores experimentales de la evolución temporal del ángulo de rumbo que registra la IMU ubicada en la cabeza del robot, en 3 de las 10 pruebas realizadas para cada configuración. En las diversas pruebas sobre la marcha *sidewinding* con la configuración de velocidad 2, se observa que los datos registrados cuentan con una gran similitud durante casi todo el recorrido respecto a la simulación, donde el ángulo de rumbo oscila entre  $-20^\circ$  a  $20^\circ$ , ubicando siempre su eje central cercano al cero con un resultado promedio de desviación de  $1.25^\circ \pm 0.67$  evidenciando que el robot intenta estar siempre perpendicular al eje longitudinal de sí mismo. Caso contrario a lo que ocurre en la primera configuración, donde la oscilación ideal durante todo el recorrido debe estar entre  $-5^\circ$  a  $5^\circ$ , pero en cada experimento se evidencio que la oscilación del ángulo de rumbo varía en promedio  $3.13^\circ$  con una desviación típica de  $1.87$ . Por ejemplo, en los resultados de la prueba 3, la oscilación final estaba entre  $11^\circ$  a  $1^\circ$  y su eje central ubicado a  $5.5^\circ$ , indicando que el robot no estaba totalmente centrado y se había desviado respecto a su eje vertical.

Una de las posibles causas de error en el desplazamiento se debe a que el robot debe halar la fuente externa, y de este modo contrarrestar el peso del cable de alimentación. Y en la primera configuración, al tener ciclos de marcha tan cortos, no se tiene la suficiente fuerza para halar el cable, haciendo que esto incline levemente el robot y el error se acumule durante todo el recorrido. Este fue uno de los múltiples factores que afectaron su movimiento respecto a los resultados obtenidos en simulación.

En la tabla 4.3 se presentan la comparación de los promedios del tiempo, velocidad y ángulo de rumbo resultados al realizar todas las pruebas para el modo de locomoción *sidewinding*. De este modo tenemos distancia recorrida en los dos ejes, notándose que los resultados son similares entre las simulaciones y los obtenidos con el robot real. Además se obtienen el error que existen entre la simulación y el robot real donde se observa que para la primera configuración de parámetros, el robot real solo es un -1,62% más lento para recorrer el objetivo de los 120cm y 4.66% más rápido en la segunda configuración para recorrer una distancia de 50cm.

Modo de Marcha <i>Sidewinding</i> Terreno Regular		Simulación		Robot real		
		Promedio	Desviación	Promedio	Desviación	Error (%)
<b>Velocidad 1</b> $d_y = 120cm$ <b>Simulación</b> $d_x = 118cm$ <b>R. Real</b> $d_x = 116cm$	<b>Tiempo</b> (s)	77.49	0.010	78.16	1.993	0.86
	<b>Velocidad</b> (cm/s)	2.17	0.000	2.14	0.057	-1.62
	<b>Ángulo Yaw</b> - rumbo (°)	-3.44	0.000	3.13	1.865	—
<b>Velocidad 2</b> $d_y = 50cm$ <b>Simulación</b> $d_x = 152cm$ <b>R. Real</b> $d_x = 140cm$	<b>Tiempo</b> (s)	33.55	0.005	29.82	1.091	-11.12
	<b>Velocidad</b> (cm/s)	4.77	0.001	4.99	0.180	4.66
	<b>Ángulo Yaw</b> - rumbo (°)	-1.97	0.000	1.25	0.670	—

Tabla 4.3: Comparación de resultados de la locomoción *sidewinding* en terreno regular.

### 4.3.2. Terreno irregular

En la figura 4.10 se presentan de manera aleatoria 3 de los 10 resultados de los ángulos de rumbo obtenidos mediante la IMU instalada en la cabeza del robot serpiente para una locomoción *sidewinding*, con las dos configuraciones de velocidades. Los resultados se dividen en los tres entornos con terreno irregular descritos anteriormente.

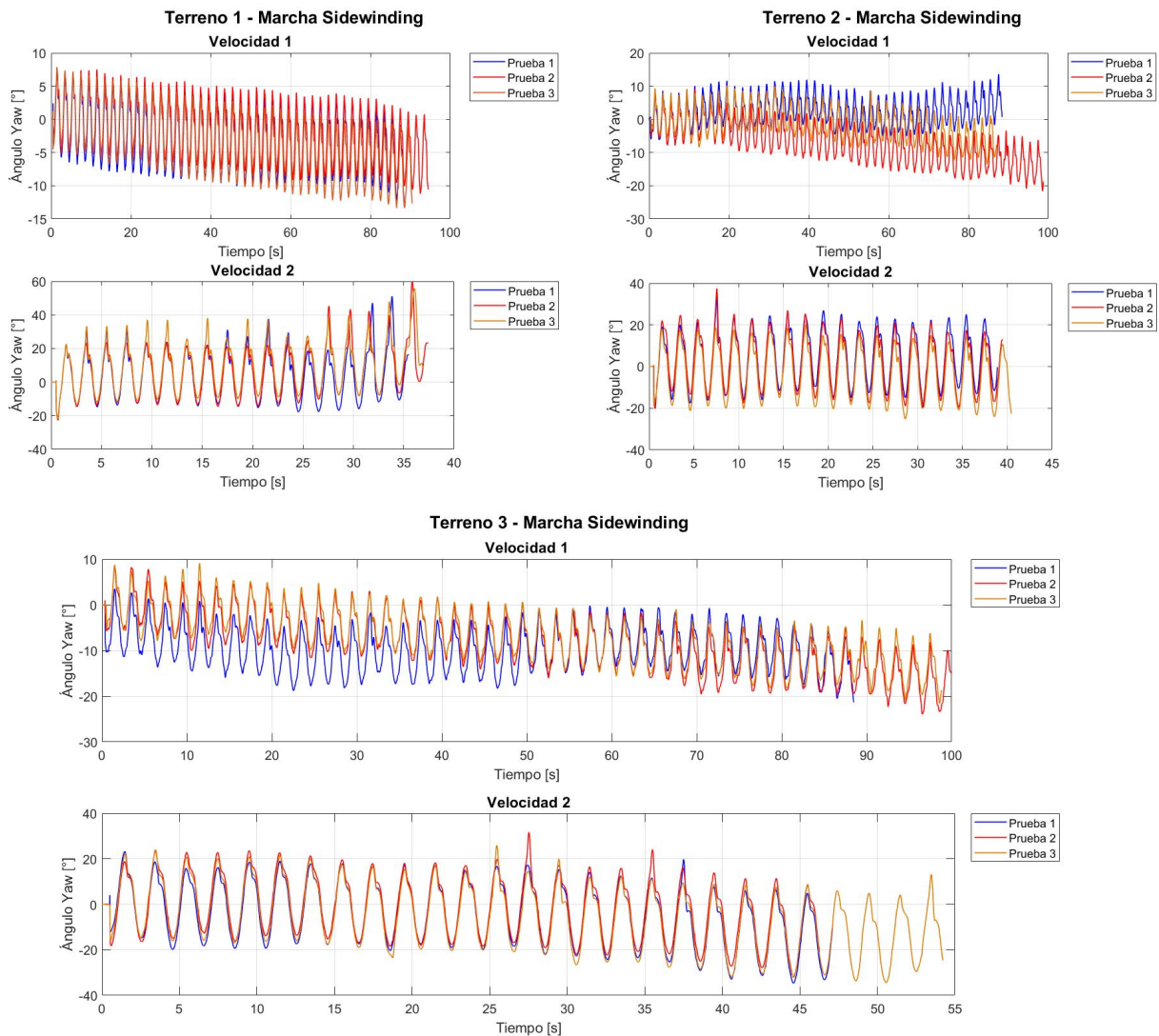


Figura 4.10: Ángulo Yaw (rumbo) para el patrón de marcha *sidewinding* en terreno irregular.

Primero se analizan los resultados de la configuración de velocidad 1 en los terrenos 1 y 3, donde se evidencia un comportamiento similar, observándose que el ángulo de rumbo del robot no oscila entre el rango ideal de  $-5^\circ$  a  $5^\circ$ , sino que se desliza continuamente sobre su eje longitudinal, produciendo un ángulo de desvío medio para el terreno 1 de  $-8.83^\circ \pm 2.02$  y para el terreno 3 de  $-11.13^\circ \pm 1.91$ , lo que indica que el robot con cada ciclo de marcha se va desviando respecto a su eje vertical. Además, cabe aclarar que en el terreno 3, el césped presenta una pequeña pero continua inclinación durante todo el recorrido, dificultando aún más el arrastre de la fuente de alimentación. En el terreno 2, se presentan diversos obstáculos distribuidos de manera no uniforme, y al presentarse uno de estos durante la marcha, la cabeza del robot se desliza aleatoriamente, generando cambios continuos y muy diferentes al ángulo de rumbo en cada ejecución de la locomoción obteniendo un ángulo de rumbo promedio de  $-6.94^\circ$  pero con una desviación estándar de  $7.69^\circ$ , indicándonos lo



dispersos que están los datos con respecto a la media.

En la configuración para la velocidad 2, se puede observar que para la locomoción en el terreno 2, su ángulo de orientación oscila constantemente entre los  $-20^\circ$  a  $20^\circ$ , indicándonos que el robot no tiende a desviarse demasiado con respecto a su vertical como lo observado en los otros dos entornos, siendo la media de desvío de  $-1.26^\circ \pm 2.88$ . En términos de tiempo y velocidad, para la primera configuración se logra evidenciar que los resultados son muy similares en todos los terrenos, obteniéndose una velocidad promedio aproximada de 1.82 cm/s. Esto no ocurre con la segunda configuración, donde la velocidad media fue cercana a los 4.16 cm/s para los dos primeros terrenos y para el tercer terreno donde se presenta una inclinación de  $10^\circ$  durante todo el recorrido, el robot es más propenso a avanzar más hacia el lateral que hacia adelante, reduciendo su velocidad promedio a 3.28 cm/s. Un resumen de los resultados de los promedios del tiempo, velocidad y ángulo de rumbo obtenidos durante la ejecución del modo de marcha sidewinding se presenta en la tabla 4.4.

Modo de Marcha <i>Sidewinding</i> Terreno Irregular		Terreno 1		Terreno 2		Terreno 3	
		Promedio	Desviación	Promedio	Desviación	Promedio	Desviación
<b>Velocidad 1</b> $d_y = 120cm$ T1 - $d_x = 116cm$ T2 - $d_x = 124cm$ T3 - $d_x = 128cm$	Tiempo (s)	91.83	3.750	93.63	4.109	100.74	2.048
	Velocidad (cm/s)	1.82	0.079	1.86	0.159	1.74	0.036
	Ángulo Yaw - rumbo ( $^\circ$ )	-8.83	2.019	-6.94	7.868	-11.13	1.914
<b>Velocidad 2</b> $d_y = 50cm$ T1 - $d_x = 153cm$ T2 - $d_x = 162cm$ T3 - $d_x = 168cm$	Tiempo (s)	36.60	1.680	40.94	3.051	54.39	2.665
	Velocidad (cm/s)	4.41	0.215	4.16	0.281	3.28	0.458
	Ángulo Yaw - rumbo ( $^\circ$ )	11.92	3.360	-1.26	2.884	-12.074	2.371

Tabla 4.4: Comparación de resultados de la locomoción *sidewinding* en terreno irregular.

## 4.4. Resultados locomoción ondulación lateral

### 4.4.1. Terreno regular

El movimiento de locomoción de ondulación lateral es un modo de marcha tridimensional. En la Figura 4.11 se puede observar las ondas verticales y horizontales que se forman en el cuerpo del robot para las dos configuraciones de velocidad. Para su primera configuración, los puntos de apoyo se intercalan en cada ciclo pasando de 4 a 3 puntos, generando una mayor estabilidad que en la segunda configuración, donde se pasa de 3 a 2 puntos de contacto.

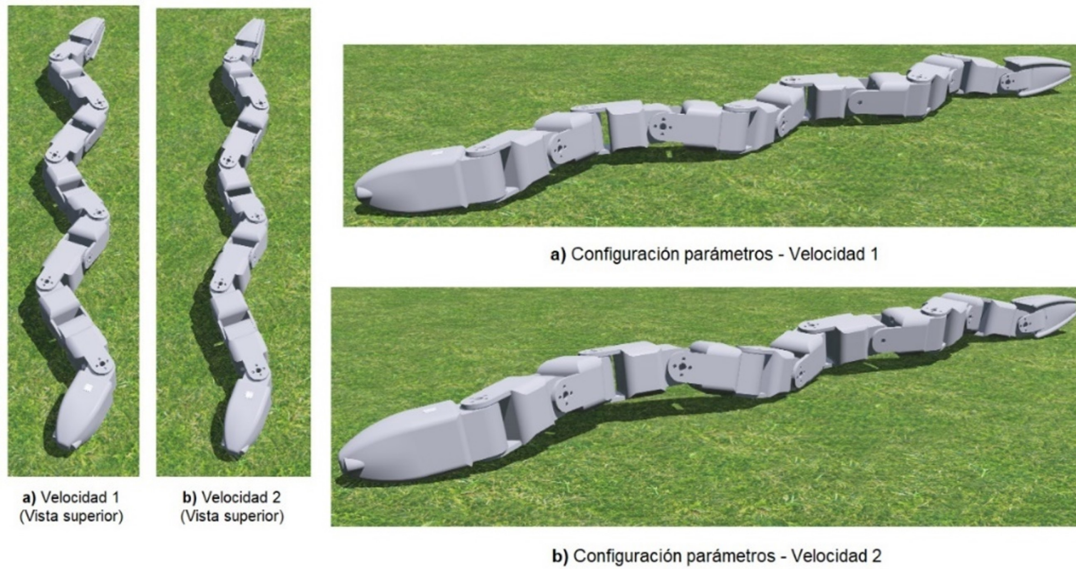


Figura 4.11: Forma de onda de los tipos de marcha ondulación lateral.

En la Figura 4.12 se presenta la posición o trayectoria de robot al ejecutar las dos configuraciones de velocidad en el simulador Webots.

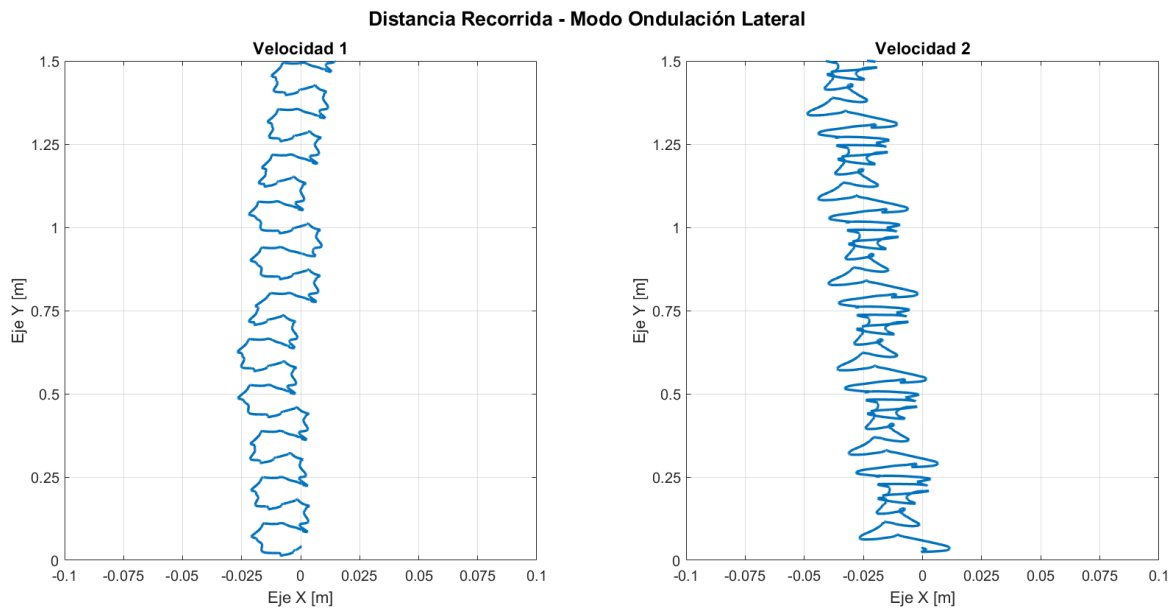


Figura 4.12: Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha ondulación lateral.

En la anterior figura se observa como el robot se balancea mientras se contrae y estira para generar un avance en dirección del eje Y. Con la configuración de la velocidad 1 se obtiene un movimiento más ordenado y sincronizado, el cual permite al robot mantener

su cabeza centrada cercana a la posición cero del eje X con una velocidad media de 2.27 cm/s. Un comportamiento muy diferente se observa con la configuración 2, donde se muestra que el movimiento no es similar en cada ciclo y se ve afectado por tener menos puntos de apoyo durante su ejecución, lo que hace que el robot tienda a desviarse hacia la izquierda y obteniendo una velocidad media de 2.08 cm/s.

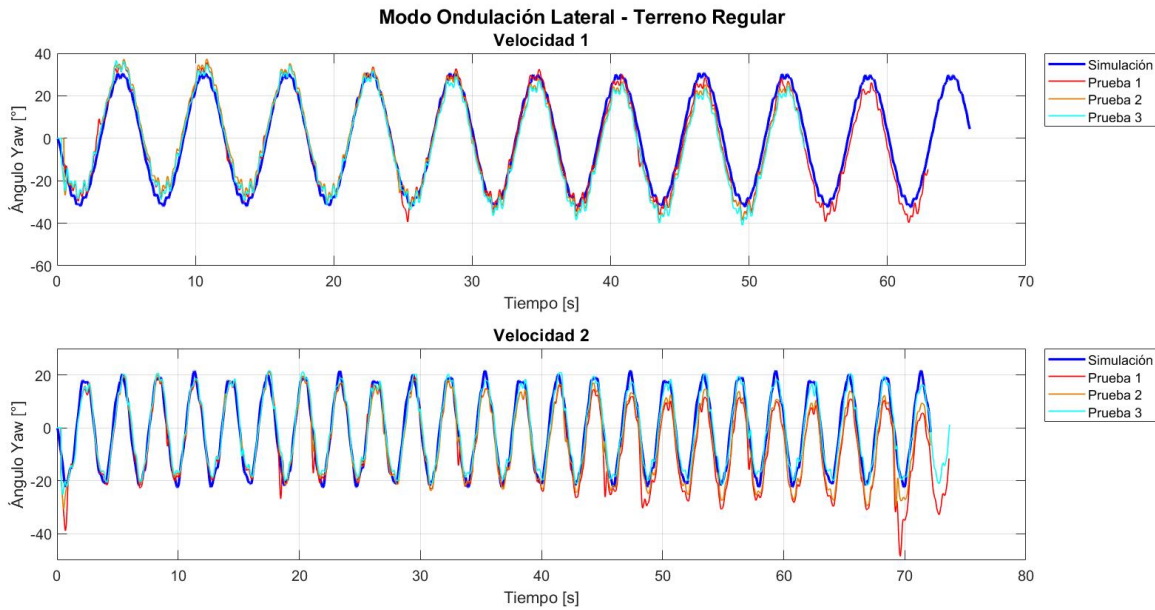


Figura 4.13: Ángulo Yaw (rumbo) para el patrón de marcha ondulación lateral en terreno regular.

En la figura 4.13 se presenta el ángulo de rumbo o Yaw que se obtuvieron en simulación y tres pruebas al ejecutar las dos configuraciones de velocidad del algoritmo del modo de marcha ondulación lateral. Como se mencionó anteriormente, con la configuración 1 el movimiento es más sincronizado y presenta pocas perturbaciones que afecten una armonía a las ondas que se forman, las cuales oscilan entre  $-30^\circ$  a  $30^\circ$ , manteniendo su eje central cercano a cero con solo una desviación promedio de  $-1.72^\circ \pm 0.96$ , indicándonos que el robot no tiende a desviarse hacia sus laterales como en los modos de marcha descritos en las secciones anteriores. En cambio, en la segunda configuración se puede observar que con los parámetros seleccionados, el robot genera unos balanceos no deseados durante cada ciclo de marcha, los cuales a partir del segundo 40 aproximadamente hacen que el robot presente una leve desviación respecto a su eje vertical, siendo en promedio un ángulo de  $-2.13^\circ \pm 1.88$ .

En la siguiente tabla se indican los promedios del tiempo, velocidad y ángulo de rumbo que resultaron de la ejecución de las 10 pruebas realizadas en simulación y el robot real con las dos configuraciones de parámetros. En ellos se puede evidenciar las diferencias que hubo con respecto a las ejecuciones en los dos ambientes donde en la plataforma de simulación no hubo mayor diferencia en términos de velocidad pasando de 2.27 cm/s en la primera configuración a 2.08 cm/s en la segunda. Pero en las pruebas llevadas a cabo con el robot real con la primera configuración de parámetros se obtuvo un 18.68 % más de velocidad comparado con los resultados de simulación y para la segunda configuración la velocidad se redujo en un 2.22 %.

Modo de Marcha Ondulación Lateral Terreno Regular		Simulación		Robot real		
		Promedio	Desviación	Promedio	Desviación	Error (%)
<b>Velocidad 1</b> $d_y = 150cm$	Tiempo (s)	65.95	0.002	55.70	2.915	-15.55
	Velocidad (cm/s)	2.27	0.000	2.70	0.131	18.68
	Ángulo Yaw - rumbo (°)	-1.414	0.003	-1.72	0.960	—
<b>Velocidad 2</b> $d_y = 150cm$	Tiempo (s)	72.14	0.001	73.79	1.099	2.30
	Velocidad (cm/s)	2.08	0.000	2.03	0.030	-2.22
	Ángulo Yaw - rumbo (°)	-0.38	0.000	-2.13	1.88	—

Tabla 4.5: Comparación de resultados de la locomoción ondulación lateral en terreno regular.

#### 4.4.2. Terreno irregular

En la figura 4.14 se muestra la variación del ángulo de rumbo registrado durante tres de las diez pruebas realizadas en cada combinación de terreno y configuración de parámetros para la marcha de ondulación lateral. El resultado indica las oscilaciones generadas por la locomoción con la primera configuración de velocidad presentan una menor desviación en términos de ángulo de rumbo donde obtenemos para el terreno 1 un promedio de ángulo Yaw de  $-3.46 \pm 0.617$ ,  $-4.79 \pm 2.54$  para el terreno 2 y  $-1.41 \pm 2.02$  para el terreno 3. Caso contrario a los resultados con la segunda configuración donde se presenta una media superior de ángulos de inclinación con una mayor desviación estándar. Además, se puede identificar que para el terreno 1 – velocidad 2, es más notorio que el ángulo de rumbo cambia de giro algunos grados hacia el eje negativo indicándonos una inclinación del robot respecto a su eje vertical. Además, debido a la presencia frecuente de pequeños obstáculos

como ramas y piedras ubicados aleatoriamente en el terreno 2 y una inclinación durante todo el recorrido del terreno 3, se observó un efecto más estocástico en las oscilaciones del ángulo Yaw sobre estos terrenos, en relación con los mismos experimentos sobre el terreno 1.

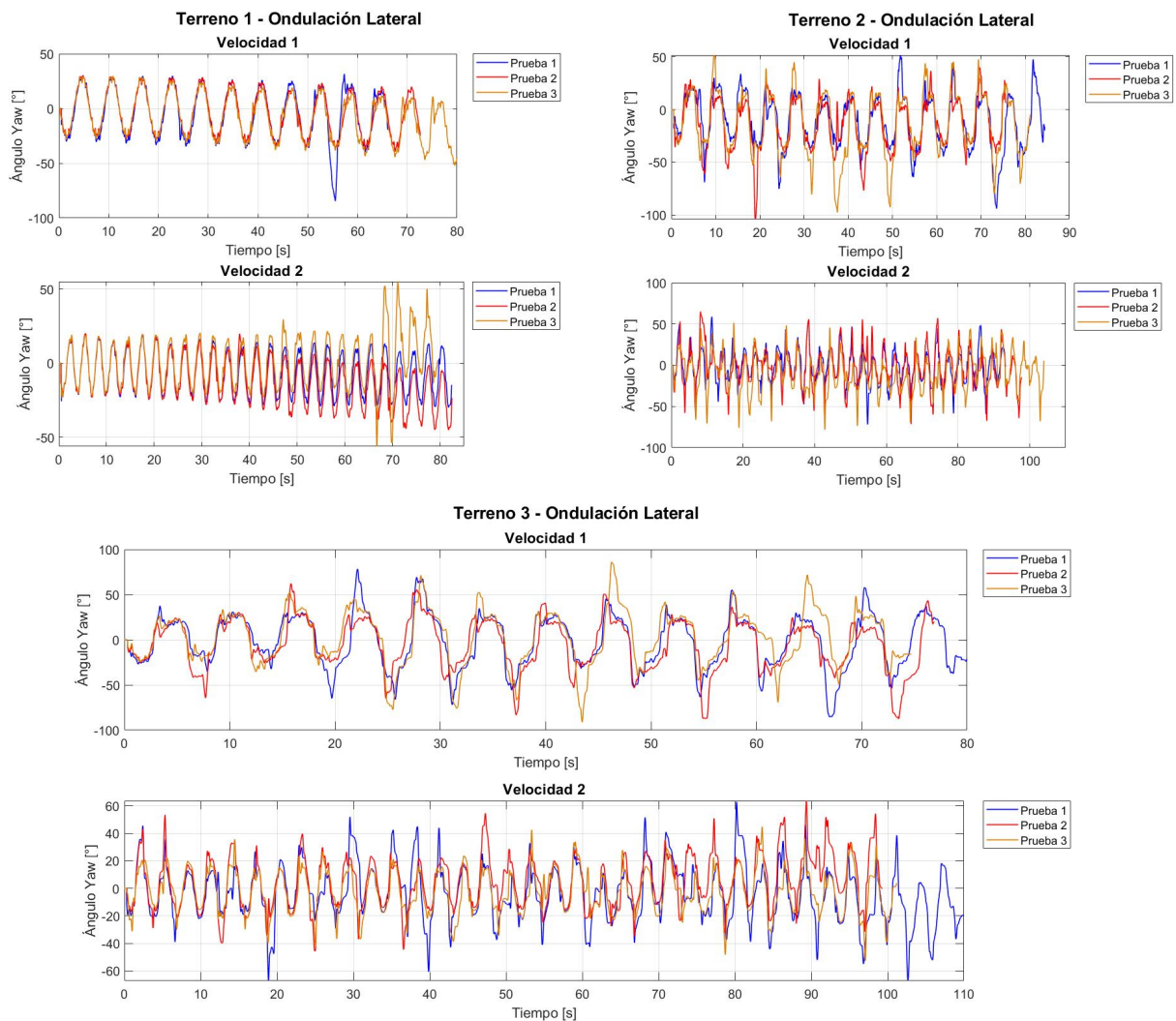


Figura 4.14: Ángulo Yaw (rumbo) para el patrón de marcha ondulación lateral en terreno irregular.

En la tabla 4.6 se presentan la comparación de las medias y desviaciones estándar de los tiempos, velocidades y ángulos de rumbo producto de realizar todas las pruebas en los tres terrenos con las dos configuraciones de parámetros para el modo de locomoción ondulación lateral.

Modo de Marcha Ond. Lateral Terreno Irregular		Terreno 1		Terreno 2		Terreno 3	
		Promedio	Desviación	Promedio	Desviación	Promedio	Desviación
Velocidad 1 $d_y = 150cm$	Tiempo (s)	73.75	3.750	83.31	2.354	77.79	2.734
	Velocidad (cm/s)	2.05	0.174	1.81	0.124	1.93	0.068
	Ángulo Yaw - rumbo (°)	-3.46	0.617	-4.79	2.543	-1.41	2.022
Velocidad 2 $d_y = 150cm$	Tiempo (s)	82.23	3.770	104.52	5.563	105.41	4.285
	Velocidad (cm/s)	1.83	0.080	1.46	0.184	1.43	0.102
	Ángulo Yaw - rumbo (°)	0.113	6.638	-9.46	5.726	-3.43	6.616

Tabla 4.6: Comparación de resultados de la locomoción ondulación lateral en terreno irregular.

## 4.5. Resultados locomoción Rolling lateral

### 4.5.1. Terreno regular

En la figura 4.15 se muestra la forma que adopta el robot en este modo de locomoción. A continuación, se podrá observar que esta marcha es la más rápida debido a sus simples movimientos de contraerse y expandirse en forma de arco. Este tipo de locomoción solo se ejecutó en un terreno regular y controlado, como se explicó en la sección 3.54.

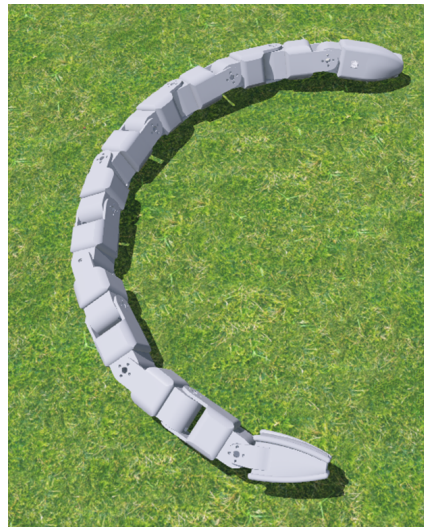


Figura 4.15: Forma de onda de los tipos de marcha rolling lateral.

La posición y dirección que toma el robot a la hora de ejecutar el modo de locomoción Rolling lateral se muestran en la figura 4.16. Se muestra como avanza el robot en cada ciclo de marcha, al contraerse y expandirse en esta forma que adopta el robot. Se debe tener en cuenta que el centro de la masa del robot serpiente es el punto donde se produce

la rotación y la traslación durante este modo de locomoción, dando como resultado que la dirección que toma el robot al rodar se desviará hacia el lado donde se encuentra el centro de masa, por lo que la trayectoria de movimiento ya no será recta.

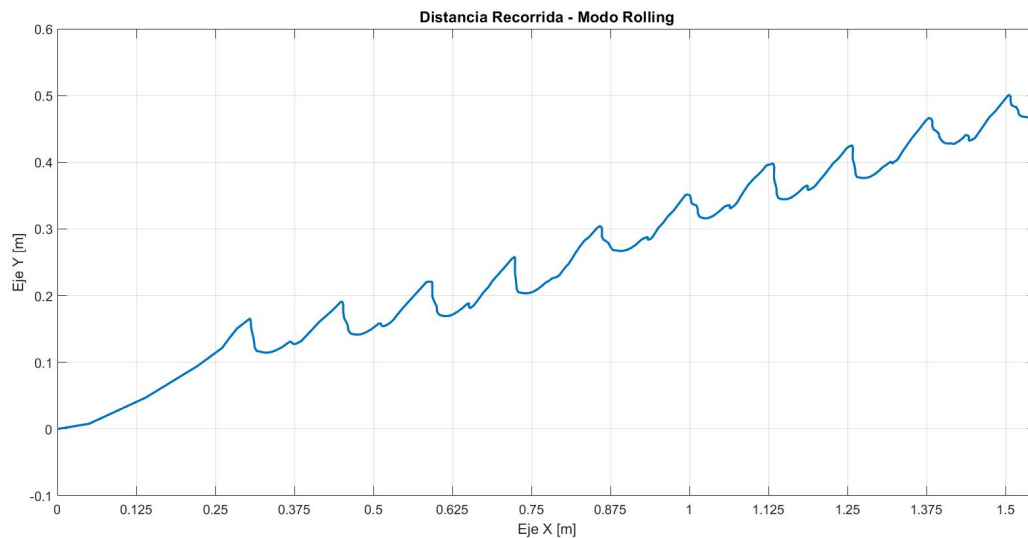


Figura 4.16: Recorrido de la posición de la cabeza del robot en la simulación del patrón de marcha rolling lateral.

A continuación, en la figura 4.17 se presenta la comparación de los ángulos roll y pitch que se generan al ejecutar este modo de locomoción, con los mismos ángulos entregados por la IMU en las pruebas con el robot real.

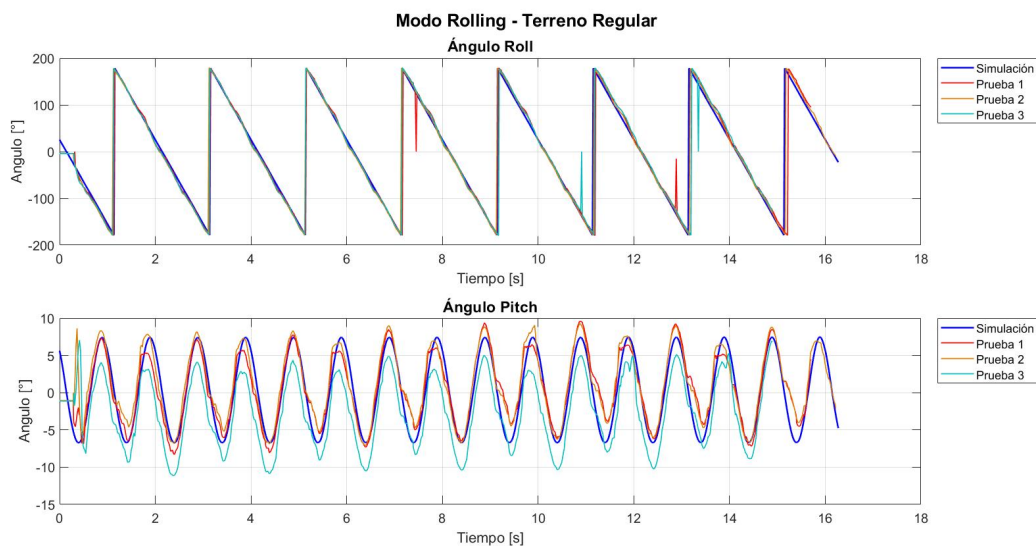


Figura 4.17: Ángulo Yaw (rumbo) para el patrón de marcha rolling lateral en terreno regular.

Se puede evidenciar que el comportamiento del robot real es muy similar a lo obtenido en simulación tanto en el ángulo de rotación Roll, el tiempo de ejecución para lograr una distancia recorrida de 150 cm con un promedio de 15.67 s, siendo este menor al de simulación en un 3.68 %. Además, se observa que hay pequeñas variaciones en el ángulo Pitch, presentando un ángulo con media de  $1.43^\circ$  y una desviación estándar de  $1.43^\circ$  a lo largo de todo el recorrido. Estas oscilaciones son consecuencia del efecto dinámico de los cables que están unidos a la cola del robot, los cuales se tensan con el movimiento. Esta tensión afecta la masa central de la serpiente al desplazarse más hacia la cola.

En la tabla 4.7 se presenta el promedio y desviaciones del tiempo, velocidad y ángulo Roll y Pitch de la ejecución de todas las pruebas realizadas en una superficie llana, para el modo de locomoción Rolling lateral.

Modo de Marcha Rolling Lateral Terreno Regular		Simulación		Robot real		
		Promedio	Desviación	Promedio	Desviación	Error (%)
Velocidad 1 $d_y = 150\text{cm}$	Tiempo (s)	16.27	0.001	15.67	0.67	-3.68
	Velocidad (cm/s)	9.22	0.000	9.59	0.41	4.00
	Ángulo Roll ( $^\circ$ )	0.07	0.001	0.11	0.83	—
	Ángulo Pitch ( $^\circ$ )	0.40	0.002	0.81	1.433	—

Tabla 4.7: Comparación de resultados de la locomoción rolling lateral en terreno regular.



# Capítulo 5

## Conclusiones y trabajos futuros

El presente documento mostró el diseño de un robot tipo serpiente, su modelado en un software CAD y su posterior simulación. A partir de la revisión de la literatura, se verificó el estado de la robótica de serpientes, centrándose en el diseño de diferentes tipos de robots serpientes que se han construido a lo largo de las últimas décadas, junto con los mecanismos y funciones implementadas actualmente. Con base en lo anterior se diseñó y simuló un robot serpiente, el cual fue luego construido utilizando una impresora 3D. Este robot cuenta con doce servomotores Dynamixel AX-12, los cuales facilitan una conexión en cadena y una comunicación TTL en serie. El robot cuenta también con un sensor de medida inercial (IMU) para obtener datos de los ángulos de rotación. En total cuenta con diez motores para el cuerpo y dos más que hacen de cabeza y cola del mismo.

La simulación de un primer prototipo se realizó en el ambiente Matlab – Simulink, haciendo uso de la herramienta SimScape, la cual presenta la ventaja de poder simular sistemas con múltiples grados de libertad, sin tener que recurrir al modelo matemático de los mismos. Con esta primera aproximación se probó el modelo del robot con diferentes variables para la ecuación de locomoción, generándose diversos modos de marcha, donde se obtuvieron los pares necesarios para los motores. Luego se utilizó el software de simulación de robótica móvil Webots, que permite la síntesis y prueba sobre un terreno llano de diversos patrones de movimiento en dos y tres dimensiones, basados en el modelo matemático propuesto para el robot serpiente. Finalmente se realizaron experimentos para demostrar la versatilidad en la locomoción sobre el robot real, sobre un terreno llano y sobre 3 tipos diferentes de superficies irregulares, replicando los mismos patrones de movimiento probados en simulación. Los modos de locomoción probados fueron: *sidewinding*, rectilíneo, ondulación lateral y Rolling lateral, cada uno de ellos con dos velocidades diferentes.

Los resultados presentados demuestran que el rendimiento asociado a [los modos de locomoción bio-inspirados se benefician del diseño propuesto](#), con una estructura modular y un sistema en serie de varios grados de libertad. Además, los resultados muestran dos comportamientos interesantes: primero, las pruebas con el robot real [presentan una gran similitud con las pruebas realizadas en el software Webots, donde se obtuvo un promedio de error para el tiempo de -3.48 % y 2.99 % para la velocidad en relación a las medias obte-](#)

nidas en simulación para los 4 modos de marcha ejecutados en este trabajo. Además, nos permite evidenciar que la velocidad del robot depende de parámetros como la amplitud y la longitud de las ondas empleadas, así como la estabilidad del movimiento. Segundo, el comportamiento del robot real es satisfactorio en recorridos cortos, con valores pequeños en la media y desviación estándar de los ángulos de rumbo en cada uno de los experimentos realizados teniendo en cuenta las dos configuraciones de parámetros y los cuatro modos de locomoción. Donde la marcha de ondulación lateral con la configuración de velocidad 1 fue la que presento los menores promedios de ángulo de desviación, indicándonos que el robot finaliza su recorrido de manera casi completamente vertical respecto a su eje longitudinal. Por último, los errores de desviación obtenidos en este trabajo son atribuibles principalmente al hecho de tener que arrastrar el robot la fuente que lo alimenta energéticamente, aparte de la influencia de los pequeños bordes redondeados en cada uno de sus módulos.

Como trabajos futuros se propone implementar otros modos de locomoción, además de probar su comportamiento en terrenos irregulares de mayor complejidad, con el fin de aplicar los correctivos tanto en la arquitectura del mismo como la selección de materiales de construcción, para así permitir un mayor agarre en la marcha. También se podría considerar la impermeabilización del robot, con el fin de permitir operaciones en campo abierto, y en su control, para que pueda realizar tareas con mayor versatilidad. Esto se podría lograr añadiendo sensores que puedan adquirir datos del terreno para superar superficies irregulares. También se podría incorporar la tarjeta de control y varias baterías dentro del robot, para así tener un mecanismo completamente autónomo, el cual pueda ser utilizado en labores agrícolas.

# Bibliografía

- [1] A. Barrio, S. Terrile, A. Barrientos, and J. del Cerro, “Robots Hiper-Redundantes: Clasificación, Estado del Arte y Problemática,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 4, pp. 351–362, Sep. 2018. [Online]. Available: <https://polipapers.upv.es/index.php/RIAI/article/view/9207>
- [2] X. Yang, L. Zheng, D. Lü, J. Wang, S. Wang, H. Su, Z. Wang, and L. Ren, “The snake-inspired robots: a review,” *Assembly Automation*, vol. 42, no. 4, pp. 567–583, Jan. 2022, publisher: Emerald Publishing Limited. [Online]. Available: <https://doi.org/10.1108/AA-03-2022-0058>
- [3] N. T. Tran, “Rapid GAIT adoption for snake-like locomotion (RGASL),” PhD Thesis, Toronto Metropolitan University, May 2021. [Online]. Available: [https://rshare.library.torontomu.ca/articles/thesis/Rapid\\_GAIT\\_adoption\\_for\\_snake-like\\_locomotion\\_RGASL\\_/14656650/1](https://rshare.library.torontomu.ca/articles/thesis/Rapid_GAIT_adoption_for_snake-like_locomotion_RGASL_/14656650/1)
- [4] Y. F. Lee, “Snake-like robot (mechanical part),” PhD Thesis, Universiti Tunku Abdul Rahman, Apr. 2011. [Online]. Available: <http://eprints.utar.edu.my/151/>
- [5] E. Prada, Miková, R. Surovec, and M. Kenderová, “Complex kinematic model of snake-like robot with holonomic constraints,” Hradec Králové, Česká republika, Dec. 2012, pp. 11–14.
- [6] D. Tenenbaum, “Secrets of the sidewinder | The Why Files,” 2014. [Online]. Available: <https://whyfiles.org/2014/secrets-of-the-sidewinder/index.html>
- [7] D. Rollinson, “Control and Design of Snake Robots,” PhD Thesis, Carnegie Mellon University, Pittsburgh, USA, Jun. 2014.
- [8] W. Hickey, “7 Animated GIFs That Will Make You Instantly Understand Trigonometry,” 2013. [Online]. Available: <https://www.businessinsider.com/7-gifs-trigonometry-sine-cosine-2013-5>
- [9] T. Duckett, S. Pearson, S. Blackmore, and B. Grieve, “Agricultural Robotics: The Future of Robotic Agriculture,” *arXiv:1806.06762 [cs]*, Aug. 2018, arXiv: 1806.06762. [Online]. Available: <http://arxiv.org/abs/1806.06762>

- [10] A. Bautista and S. Wane, “ATLAS Robot: A Teaching Tool for Autonomous Agricultural Mobile Robotics,” in *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*, Oct. 2018, pp. 264–269.
- [11] J. Rosell, F. Auat, E. Gregorio, D. Andújar, and L. Puigdomènech, “Chapter Three - Advances in Structured Light Sensors Applications in Precision Agriculture and Livestock Farming,” in *Advances in Agronomy*. Academic Press, Jan. 2015, vol. 133, pp. 71–112. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0065211315001078>
- [12] F. Narvaez, G. Reina, M. Torres, G. Kantor, and F. Cheein, “A Survey of Ranging and Imaging Techniques for Precision Agriculture Phenotyping,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2428–2439, 2017.
- [13] A. Oliveira, Gomes, M, and Leite, A, “Adaptação de um Automodelo para Aplicações de Robótica Móvel na Agricultura,” in *XXII Congresso Brasileiro de Automática*, João Pessoa, Brazil, 2018. [Online]. Available: <http://www.swge.inf.br/proceedings/paper/?P=CBA2018-0742>
- [14] M. Fawakherji, A. Youssef, D. Bloisi, A. Pretto, and D. Nardi, “Crop and Weeds Classification for Precision Agriculture Using Context-Independent Pixel-Wise Segmentation,” in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, Feb. 2019, pp. 146–152.
- [15] I. Hameed, “A Coverage Planner for Multi-Robot Systems in Agriculture,” in *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Kandima, Maldivas, Aug. 2018, pp. 698–704, iSSN: null.
- [16] J. Billingsley, A. Visala, and M. Dunn, “Robotics in Agriculture and Forestry,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1065–1077. [Online]. Available: [https://doi.org/10.1007/978-3-540-30301-5\\_47](https://doi.org/10.1007/978-3-540-30301-5_47)
- [17] K. Cavender-Bares and J. B. Lofgren, “Robotic platform and method for performing multiple functions in agricultural systems,” US Patent US9 265 187B2, Feb., 2016. [Online]. Available: <https://patents.google.com/patent/US9265187B2/en>
- [18] H. Mousazadeh, “A technical review on navigation systems of agricultural autonomous off-road vehicles,” *Journal of Terramechanics*, vol. 50, no. 3, pp. 211–232, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022489813000220>
- [19] M. Xaud, A. Leite, and P. From, “Thermal Image Based Navigation System for Skid-Steering Mobile Robots in Sugarcane Crops,” in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019, pp. 1808–1814.

- [20] S. Hajjaj and K. Sahari, “Review of agriculture robotics: Practicality and feasibility,” in *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, Tokyo, Japan, Dec. 2016, pp. 194–198.
- [21] V. Saiz, F. Rovira, and C. Millot, “Performance Improvement of a Vineyard Robot through its Mechanical Design,” ser. ASABE Paper No. 1701120. St. Joseph, USA: ASABE, 2017, p. 1. [Online]. Available: <http://elibrary.asabe.org/abstract.asp?aid=48039&t=5>
- [22] P. Astolfi, A. Gabrielli, L. Bascetta, and M. Matteucci, “Vineyard Autonomous Navigation in the Echord++ GRAPE Experiment,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 704 – 709, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896318315271>
- [23] A. Bechar and C. Vigneault, “Agricultural robots for field operations. Part 2: Operations and systems,” *Biosystems Engineering*, vol. 153, pp. 110–128, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1537511015301926>
- [24] C. Nidhi and S. G. Shruti, “A Review Study on Future Applicability of Snake Robots in India,” *IOSR Journal of Computer Engineering*, vol. 17, no. 5, pp. 3–6, Sep. 2015. [Online]. Available: <https://doi.org/10.9790/0661-17510306>
- [25] X. Qi, H. Shi, T. Pinto, and X. Tan, “A Novel Pneumatic Soft Snake Robot Using Traveling-Wave Locomotion in Constrained Environments,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1610–1617, 2020.
- [26] R. Bogue, “Snake robots: A review of research, products and applications,” *Industrial Robot: An International Journal*, vol. 41, no. 3, pp. 253–258, May 2014. [Online]. Available: <https://doi.org/10.1108/IR-02-2014-0309>
- [27] P. Zarco, V. González, and J. Berni, “Fluorescence, temperature and narrow-band indices acquired from a UAV platform for water stress detection using a micro-hyperspectral imager and a thermal camera,” *Remote Sensing of Environment*, vol. 117, pp. 322–337, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0034425711003555>
- [28] F. Garcia, S. Sankaran, J. M. Maja, W. S. Lee, J. Rasmussen, and R. Ehsani, “Comparison of two aerial imaging platforms for identification of Huanglongbing-infected citrus trees,” *Computers and Electronics in Agriculture*, vol. 91, pp. 106–115, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169912002748>
- [29] V. Saiz and F. Rovira, “From Smart Farming towards Agriculture 5.0: A Review on Crop Data Management,” *Agronomy*, vol. 10, no. 2, p. 207, 2020. [Online]. Available: <https://www.mdpi.com/2073-4395/10/2/207>

- [30] J. Hernández, J. Barrientos, J. del Cerro, A. Barrientos, and D. Sanz, “Moisture measurement in crops using spherical robots,” *Industrial Robot: An International Journal*, vol. 40, no. 1, pp. 59–66, 2013. [Online]. Available: <https://doi.org/10.1108/01439911311294255>
- [31] F. Rovira, I. Chatterjee, and V. Sáiz, “The role of GNSS in the navigation strategies of cost-effective agricultural robots,” *Computers and Electronics in Agriculture*, vol. 112, pp. 172–183, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169914003275>
- [32] F. Rovira Más, Q. Zhang, and A. C. Hansen, “Off-road Vehicle Dynamics,” in *Mechatronics and Intelligent Systems for Off-road Vehicles*, F. Rovira Más, Q. Zhang, and A. C. Hansen, Eds. London: Springer, 2011, pp. 21–42. [Online]. Available: [https://doi.org/10.1007/978-1-84996-468-5\\_2](https://doi.org/10.1007/978-1-84996-468-5_2)
- [33] D. Dorhout, “Prospero: The Robot Farmer.” [Online]. Available: [http://www.dorhoutrd.com/prospero\\_robot\\_farmer](http://www.dorhoutrd.com/prospero_robot_farmer)
- [34] F. Reyes and S. Ma, “Snake robots in contact with the environment: Influence of the friction on the applied wrench,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 2017-September, 2017, pp. 5790–5795.
- [35] S. Manzoor, U. Khan, and I. Ullah, “Serpentine and Rectilinear Motion Generation in Snake Robot Using Central Pattern Generator with Gait Transition,” *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, no. 3, pp. 1093–1103, Dec. 2019. [Online]. Available: <https://link.springer.com/article/10.1007/s40998-019-00301-8>
- [36] G. Chirikjian and J. Burdick, “The kinematics of hyper-redundant robot locomotion,” *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 781–793, Dec. 1995. [Online]. Available: <https://doi.org/10.1109/70.478426>
- [37] P. Liljebäck, S. Fjerdingen, K. Y. Pettersen, and Stavdahl, “A snake robot joint mechanism with a contact force measurement system,” in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3815–3820.
- [38] J. Gray, “The Mechanism of Locomotion in Snakes,” *Journal of Experimental Biology*, vol. 23, no. 2, pp. 101–120, Dec. 1946. [Online]. Available: <https://doi.org/10.1242/jeb.23.2.101>
- [39] T. Owen, “Biologically Inspired Robots: Snake-Like Locomotors and Manipulators,” *Robotica*, vol. 12, no. 3, pp. 282–282, May 1994.
- [40] M. Mori and S. Hirose, “Three-dimensional serpentine motion and lateral rolling by active cord mechanism ACM-R3,” in *IEEE International Conference on Intelligent Robots and Systems*, vol. 1, 2002, pp. 829–834.

- [41] H. Yamada, S. Chigisaki, M. Mori, K. Takita, K. Ogami, and S. Hirose, “Development of amphibious snake-like robot ACM-R5,” in *36th International symposium on robotics*, Tokyo, Japan, 2005, p. 133.
- [42] P. Liljebäck, K. Pettersen, Stavdahl, and J. Gravidahl, “A review on modelling, implementation, and control of snake robots,” *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 29–40, Jan. 2012. [Online]. Available: <https://doi.org/10.1016/j.robot.2011.08.010>
- [43] W. Yang, “Biomorphic Hyper-Redundant Snake Robot: Locomotion Simulation, 3D Printed Prototype and Inertial-Measurement-Unit-Based Motion Tracking,” Master’s thesis, University of Nevada, Reno, USA, 2016. [Online]. Available: <http://hdl.handle.net/11714/2349>
- [44] J. Borenstein, M. Hansen, and A. Borrell, “The OmniTread OT-4 serpentine robot—design and performance,” *Journal of Field Robotics*, vol. 24, no. 7, pp. 601–621, 2007. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20196>
- [45] S. Rasakatla and K. Krishna, “Snake P3: A semi-autonomous Snake robot,” in *2010 IEEE International Conference on Robotics and Biomimetics*, Tianjin, China, Dec. 2010, pp. 1502–1506.
- [46] H. Wang, Wang, H, Xu, X, and Mu, Z, “Development and experiment of a snake-like robot composed of modularized isomorphic joints,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, Oct. 2016, pp. 003 160–003 165.
- [47] X. Su, J. Gao, Z. Zhao, Q. Wu, and C. Huang, “Structural analysis and design of round belt drive snake-like robot,” in *IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012)*, Shenzhen, China, Dec. 2012, pp. 1–5.
- [48] X. Xiao and R. Murphy, “A review on snake robot testbeds in granular and restricted maneuverability spaces,” *Robotics and Autonomous Systems*, vol. 110, pp. 160–172, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889018300861>
- [49] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, and D. Rollinson, “Design and architecture of the unified modular snake robot,” in *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, USA, May 2012, pp. 4347–4354.
- [50] K. Fabico, J. Hernandez, S. Plata, K. Puzon, J. Sipin, and A. Bandala, “Development of a biomorphic and hyper-redundant caecilian based robots,” in *TENCON 2017 - 2017 IEEE Region 10 Conference*, Penang, Malaysia, Nov. 2017, pp. 2369–2373.
- [51] P. Liljebäck, K. Pettersen, O. Stavdahl, and J. Gravidahl, *Snake Robots - Modelling, Mechatronics, and Control*, ser. Advances in Industrial Control. London:

- Springer-Verlag, 2013. [Online]. Available: <https://www.springer.com/gp/book/9781447129950>
- [52] P. Liljebäck, Stavdahl, K. Pettersen, and J. Gravdahl, “Mamba - A waterproof snake robot with tactile sensing,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, USA, Sep. 2014, pp. 294–301.
- [53] X. Zhang, J. Liu, Z. Ju, and C. Yang, “Head-Raising of Snake Robots Based on a Predefined Spiral Curve Method,” *Applied Sciences*, vol. 8, no. 11, 2018. [Online]. Available: <https://www.mdpi.com/2076-3417/8/11/2011>
- [54] Y. Zhou, Y. Zhang, F. Ni, and H. Liu, “A Head Control Strategy of the Snake Robot Based on Segmented Kinematics,” *Applied Sciences*, vol. 9, no. 23, p. 5104, 2019. [Online]. Available: <https://www.mdpi.com/2076-3417/9/23/5104>
- [55] T. Kamegawa, T. Akiyama, S. Sakai, K. Fujii, K. Une, and E. Ou, “Development of a separable search-and-rescue robot composed of a mobile robot and a snake robot,” *Advanced Robotics*, vol. 34, no. 2, pp. 132–139, 2020. [Online]. Available: <https://doi.org/10.1080/01691864.2019.1691941>
- [56] K. Melo, J. Leon, J. Monsalve, V. Fernandez, and D. Gonzalez, “Simulation and control integrated framework for modular snake robots locomotion research,” in *2012 IEEE/SICE International Symposium on System Integration (SII)*, Fukuoka, Japan, Dec. 2012, pp. 523–528, iSSN: null.
- [57] K. Melo, J. Leon, A. di Zeo, V. Rueda, D. Roa, M. Parraga, D. Gonzalez, and L. Paez, “The Modular Snake Robot Open Project: Turning animal functions into engineering tools,” in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Linköping, Sweden, Oct. 2013, pp. 1–6, iSSN: 2374-3247.
- [58] J. Monsalve, J. Leon, and K. Melo, “Modular snake robot oriented open simulation software,” in *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, Hong Kong, China, Jun. 2014, pp. 546–550, iSSN: null.
- [59] K. Melo and L. Paez, “Modular snake robot gaits on horizontal pipes,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012, pp. 3099–3104, iSSN: 2153-0858.
- [60] A. Barrio, “Design, Modelling, Control and Teleoperation of Hyper-Redundant Robots,” Ph.D. dissertation, Universidad Politécnica de Madrid, Madrid, 2020, publisher: Industriales. [Online]. Available: <https://oa.upm.es/65161/>
- [61] M. Sato, M. Fukaya, and T. Iwasaki, “Serpentine locomotion with robotic snakes,” *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 64–81, Feb. 2002, conference Name: IEEE Control Systems Magazine. [Online]. Available: <https://doi.org/10.1109/37.980248>



- [62] I. Virgala, M. Dovica, M. Kelemen, E. Prada, and Z. Bobovský, “Snake Robot Movement in the Pipe Using Concertina Locomotion,” *Applied Mechanics and Materials*, vol. 611, pp. 121–129, Aug. 2014. [Online]. Available: <https://www.scientific.net/AMM.611.121>
- [63] K. Lipkin, I. Brown, A. Peck, H. Choset, J. Rembisz, P. Gianfortoni, and A. Naaktgeboren, “Differentiable and piecewise differentiable gaits for snake robots,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2007, pp. 1864–1869.
- [64] I. Tanev, T. Ray, and A. Buller, “Evolution, robustness, and adaptation of sidewinding locomotion of simulated snake-like robot,” in *Genetic and Evolutionary Computation – GECCO 2004*, vol. 3102. Springer, 2004, pp. 627–639. [Online]. Available: [https://link.springer.com/chapter/10.1007/978-3-540-24854-5\\_65](https://link.springer.com/chapter/10.1007/978-3-540-24854-5_65)
- [65] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, “Parameterized and Scripted Gaits for Modular Snake Robots,” *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, Jan. 2009. [Online]. Available: <https://doi.org/10.1163/156855309X452566>
- [66] K. Wang, W. Gao, and S. Ma, “Snake-Like Robot with Fusion Gait for High Environmental Adaptability: Design, Modeling, and Experiment,” *Applied Sciences*, vol. 7, no. 11, p. 1133, Oct. 2017. [Online]. Available: <https://doi.org/10.3390/app7111133>
- [67] F. Sanfilippo, J. Azpiazu, G. Marafioti, A. A. Transeth, Stavadahl, and P. Liljebäck, “Perception-Driven Obstacle-Aided Locomotion for Snake Robots: The State of the Art, Challenges and Possibilities,” *Applied Sciences 2017*, vol. 7, no. 4, p. 336, Mar. 2017. [Online]. Available: <https://www.mdpi.com/2076-3417/7/4/336/htmlhttps://www.mdpi.com/2076-3417/7/4/336>
- [68] R. Buckingham, “Snake arm robots for flexible delivery,” *Insight*, vol. 44, pp. 150–151, 2002.
- [69] K. Dowling, “Limbless locomotion: learning to crawl,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 4, 1999, pp. 3001–3006 vol.4.



# APÉNDICES



# Apéndice A

## Selección de componentes

En esta sección se presenta el proceso de selección de los diferentes componentes hardware que son necesarios para el diseño y construcción de nuestro primer prototipo robótico basado en una serpiente.

### A.1. Análisis de requerimientos

Después de investigar sobre los avances que la comunidad científica ha tenido sobre los robots serpiente, se lograron identificar los requerimientos para el diseño y funcionalidad de esta plataforma.

1. Los actuadores angulares deben tener la capacidad de realizar un movimiento periódico y preciso.
2. La fuente de propulsión debe contrarrestar el peso del robot y las demandas de torque exigidas.
3. La fricción entre el robot y el suelo debe ser lo suficientemente anisotrópica para que los modos de locomoción a implementar sean efectivos.
4. La fuente de energía tiene que entregar suficiente energía para impulsar los actuadores.
5. El dispositivo de procesamiento debe poder manejar el algoritmo de control y comunicarse con el hardware
6. El diseño debe ser flexible en tres dimensiones y poder moverse en diferentes ángulos, como una serpiente.
7. La estructura y el material del robot deben ser lo suficientemente fuertes y estables como para soportar el estrés que se producirá durante el movimiento del robot.
8. La plataforma robótica debe asemejarse a una serpiente real.

## A.2. Selección de componentes y solución electrónica

La solución electrónica, se fundamenta con base en cuatro planos, en la figura A.1, se puede apreciar de manera general como es la articulación del sistema y la forma en como cada uno de los planos se relaciona con los otros.

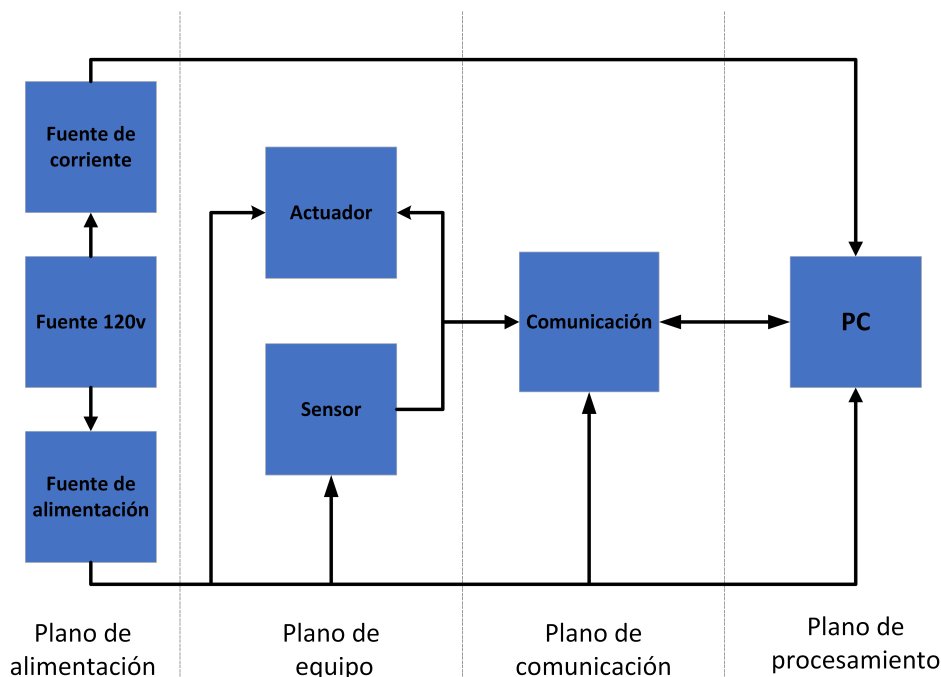


Figura A.1: Diagrama de bloques del mecanismo general del robot.

- **Plano de alimentación:** se encarga de suministrar la energía eléctrica necesaria para el correcto funcionamiento de los dispositivos electrónicos que serán parte de la solución electrónica definitiva.
- **Plano de equipo:** es el plano donde se capturan las señales provenientes de los sensores.
- **Plano de comunicación:** cuyo rol es tener un puente de comunicación entre los equipos utilizados en el desarrollo del robot con los dispositivos de procesamiento.
- **Plano de procesamiento:** se realizan los cálculos pertinentes y se ejecuta el algoritmo correspondiente a los procedimientos de funcionamiento del robot.

### A.2.1. Dispositivos considerados en cada uno de los planos

#### Plano de alimentación

la primera opción es tener en el último módulo del robot un mecanismo de anclaje para un cable de alimentación externo, conectado a una fuente de alimentación DC. La segunda

opción es tener una batería ubicada en cada módulo del robot proporcionando energía a los actuadores y una batería en el módulo de cabeza para el sensor y el dispositivo de comunicación.

### **Dispositivos de Alimentación**

- Unidad de alimentación externa.
- Baterías Turnigy 2.2.
- Baterías de litio HP-G570-0850S3
- Baterías Turnigy 5.0.

### **Plano de equipo**

Los tipos de actuadores más adecuados para esta aplicación son los motorreductores, por su baja velocidad y gran fuerza, o los servomotores, por su precisión. La elección del tipo de motor, y de su cantidad, es muy importante ya que el diseño y las dimensiones del módulo dependen en una buena parte de estos elementos.

### **Actuadores**

- HSR-5990TG.
- Dynamixel AX-18A
- Dynamixel AX-12+
- HS-5955TG
- Hitec 7950TH
- Futaba S3305

Con el fin de obtener información sobre el ángulo de desviación del robot durante la ejecución de los diversos modos de locomoción, en este trabajo se instalará un sensor de medición inercial - IMU en el modulo de la cabeza de la plataforma robótica.

### **Sensores**

- MPU9250
- BNO055
- NGIMU
- BNO080

## Plano de comunicación

Para la comunicación se evaluará si los módulos serán independientes o no. Por lo tanto, se presentan soluciones como dispositivos que permitan una comunicación inalámbrica o cableado, entre los motores, sensores y el dispositivo de procesamiento en este caso un portátil.

### Dispositivos de comunicación para los actuadores

- Modulo wireless Nrf24l01
- ConnectBlue OWS451
- U2D2
- USB2AX
- CP2102 USB to ttl
- Xbee pro 900

### Dispositivos de comunicación para el sensor

- Arduino Nano
- Arduino uno
- Raspberry Pi 3 B
- ODROID XU4

## Plano de procesamiento

Para procesar los datos del sensor y de las trayectorias del robot serpiente enviadas a los actuadores, el dispositivo elegido fue un portátil gaming Asus TUF A15 con 16Gb de RAM y una tarjeta gráfica Nvidia GTX 1650ti.

### A.2.2. Selección de la solución electrónica

Para elegir la solución electrónica definitiva ha sido necesario acudir a la técnica de Análisis Morfológico, ya que permite analizar y determinar mediante el uso de múltiples combinaciones la solución más apropiada de acuerdo al propósito del proyecto.

Esta herramienta fue desarrollada e implementada por Fritz Zwicky, un reconocido astrofísico y científico aeroespacial. El análisis morfológico es un método para estructurar e investigar la totalidad de relaciones contenidas en un problema multidimensional, complejo y esencialmente no cuantificable.



### Construcción del espacio morfológico

El espacio morfológico es el campo de los posibles y está compuesto por las todas las combinaciones que se obtuvieron al realizar la descomposición del sistema y de las hipótesis formuladas en cada uno de los subsistemas. Para construir el espacio morfológico se deben seleccionar cuidadosamente los subsistemas que componen el sistema bajo inspección, además realizar una hipótesis sobre cómo se comportan los subsistemas y cómo van a evolucionar.

En la Figura A.2, se pueden apreciar los dispositivos que han sido contemplados en cada uno de los planos anteriormente mencionados y que se consideran como posibilidades a ser los constituyentes de la solución electrónica definitiva. También se pueden observar la gran cantidad de escenarios que resultan de la combinación de cada una de estas posibilidades, esto es, el espacio morfológico.

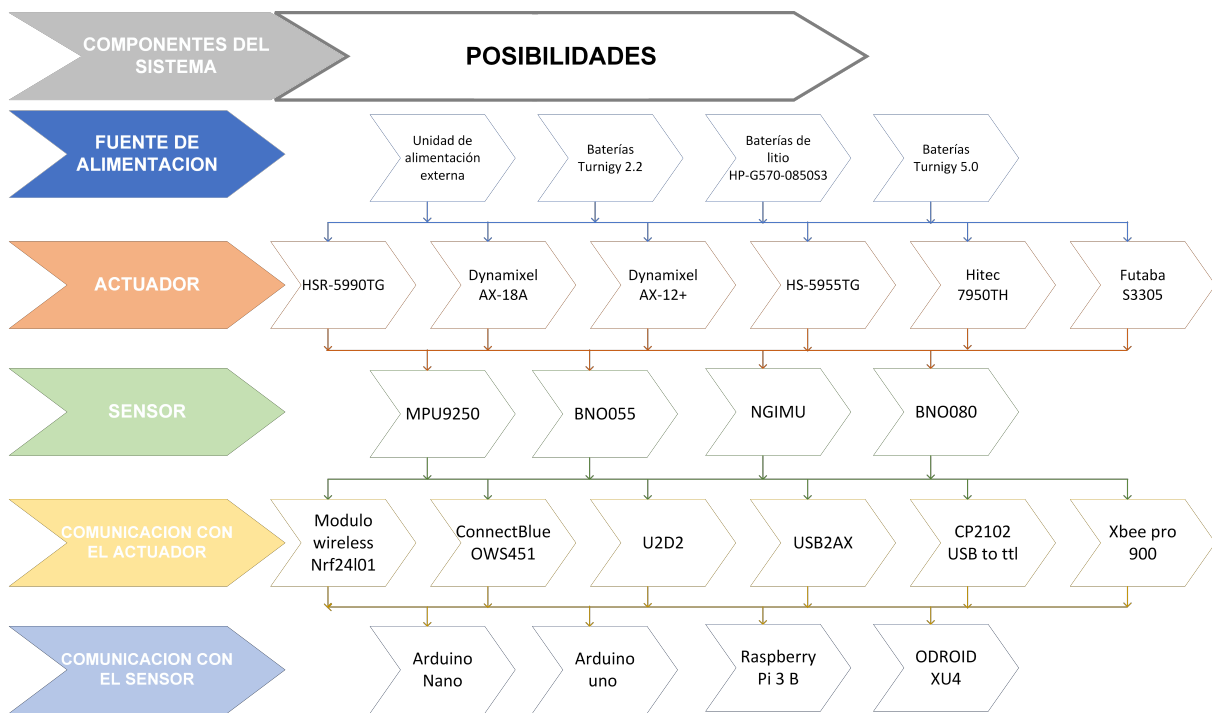


Figura A.2: Componentes opcionales.

Con base en la Figura A.2 y de acuerdo al número de dispositivos que hacen parte de cada plano, es posible determinar de manera cuantitativa el espacio morfológico, procediendo de la siguiente forma:

$$E.M = D.Alimentación * Actuadores * Sensores * D.Comunicación A * D.Comunicación S \quad (A.1)$$

**E.M** = Espacio Morfológico.

**Dispositivos de alimentación** = 4

**Actuador** = 6

**Sensor** = 4

**Dispositivos de comunicación actuador** = 6

**Dispositivos de comunicación sensor** = 4

$$E.M = 4 * 6 * 4 * 6 * 4 = 2304 \quad (A.2)$$

Estas 2304 posibilidades reflejan 2304 escenarios, que deben ser analizados con el propósito de optar por el escenario más conveniente.

### Descripción de los componentes electrónicos

Para determinar la elección más viable y funcional, es necesario considerar cada una de las características de los dispositivos, pues es aquí en donde es posible realizar un análisis profundo que permitirá establecer criterios que posibiliten descartar diferentes dispositivos.

Por esta razón, se presentan las siguientes tablas informativas con el fin de conocer las características eléctricas, electrónicas, mecánicas y económicas de cada uno de los dispositivos, para que de este modo sea posible determinar qué dispositivos harán parte de la solución electrónica definitiva mediante criterios de exclusión.

### Fuente de alimentación

	turnigy 2.2	HP-G570-0850S3	turnigy 5.0	Unidad de alimentación externa
<b>Voltaje</b>	11.1 V	11.1 V	11.1 V	12 V
<b>Capacidad</b>	2200 mah	850 mah	5000 mah	8 A
<b>Velocidad de descarga</b>	50 Cmax	70 Cmax	30 Cmax	—
<b>Peso</b>	188 g	73 g	360 g	210 g
<b>Dimensiones</b>	105x33x24 mm	76x26x19 mm	143x51x23 mm	186x84x50 mm.
<b>Precio (dólares)</b>	25.69	13.95	28.13	27.80

Tabla A.1: Especificaciones de los dispositivos de alimentación.

## Actuadores

	HSR-5990TG	Dynamixel AX-18A	Dynamixel AX-12+	HS-5955TG	HS-7950TH	Futaba S3305
<b>Voltaje</b>	6 ~7.4 V	9 ~12 V	9 ~12 V	4.8 ~6 V	6 ~7.4 V	4.8 ~6 V
<b>Torque</b>	2.94 N.m 30 Kg.cm (7.4v ~5.2 A)	1.8 N.m 18.35 Kg.cm (12v ~2.2 A)	1.52 N.m 15.3 Kg.cm (12v ~1.5 A)	2.35 N.m 24 Kg.cm (6v ~4.2A)	3.43 N.m 35 Kg.cm (7.4v ~9A)	0.87 N.m 8.9 Kg.cm (6v ~A)
<b>Peso</b>	30 Kg.cm	55.9 g	54.6 g	61.5 g	61.5 g	47 g
<b>Dimensión</b>	(7.4v ~5.2 A)	32x50x40	32x50x40	40x20x37	40x20x38	40x20x38
<b>Velocidad</b>	114 rpm	97 rpm	59rpm	—	—	—
<b>Rango (grados)</b>	180°	0 - 300°	0 - 300°	180 °	180 °	90 °
<b>Temperatura de trabajo</b>	-20 ~60 °C	-5 ~70 °C	-5 ~70 °C	-20 ~60 °C	-20 ~60 °C	-20 ~60 °C
<b>Material</b>	Metal	Plástico	Plástico	Plástico / aluminio	Metal	Plástico
<b>Retroalimentación</b>	Posición.	posición, temperatura, carga, voltaje de entrada, etc.	posición, temperatura, carga, voltaje de entrada, etc.	—	—	—
<b>Precio (dólares)</b>	120	95	45	115	130	40

Tabla A.2: Especificaciones de los actuadores.

## Sensores

	MPU9250	BNO055	NGIMU	BNO080
<b>Tipo de salida</b>	I2C	2xSPI; 2xI2C; AUX I/F; OIS I/F; 2xinterrupciones	Wi-Fi, Puerto serie, USB, almacenamiento tarjeta microSD	2C; UART; SPI; UART-RVC
<b>Dimensiones</b>	20x16x3 mm	20x27x4 mm	56x39x18 mm	26x23x4.6 mm
<b>Frecuencia</b>	157 Hz	100 Hz	400 Hz	100 Hz
<b>Fusión de sensores</b>	No	Si	Si	Si
<b>Precio (dólares)</b>	18.30	24.95	348.64	32.50

Tabla A.3: Especificaciones de los sensores.

## Dispositivos de comunicación para los actuadores

	U2D2	Modulo Nrf24l01	ConnectBlue OWS451	USB2AX	CP2102	Xbee pro 900
<b>Descripción</b>	es un convertidor de comunicación USB de tamaño pequeño que permite controlar y operar los dynamixel con PC.	El NRF24L01 es un chip de comunicación inalámbrica fabricado por Nordic Semiconductor que podemos conectar a un procesador	El adaptador OWS451 de connectBlue es ideal cuando se reemplaza un cable serie con una conexión LAN inalámbrica.	El USB2AX es una interfaz para controlar servos Dynamixel desde un ordenador. Se conecta a un puerto USB y tiene un conector Dynamixel de 3 pines para conectarse a los servos.	El conversor CP2102 facilita la comunicación entre una PC y un microcontrolador utilizando el protocolo USB.	El módulo 900 de RF XBee-PRO es ideal para aplicaciones de redes de punto a multipunto de baja latencia.
<b>Protocolos de comunicación</b>	TTL RS-485 UART	SPI	RS232 RS422/485	TTL	TTL UART	DigiMesh
<b>Dimensiones</b>	48x18x14.6 mm	16.6x45.5 mm	23x36x3 mm	16x36 mm	21x16 mm	32.92x22 mm
<b>Tasa de baudios</b>	6 Mbps	250 Kbps ~ 2 Mbps	500 kbps	1 Mbps	12 Mbps	200 Kbps
<b>Precio (dólares)</b>	49.90	4.5	55.90	39.95	8.45	59

Tabla A.4: Especificaciones de los dispositivos de comunicación para los actuadores.

## Dispositivos de comunicación para el sensor

	Arduino Nano	Arduino uno	Raspberry Pi 3 B	ODROID XU4
Core	ATmega328	ATmega328P	Broadcom BCM2837, Cortex-A53 (ARMv8)	Exynos5422
Tamaño de registro	8 bits	8 bits	64 bits	32 bits
Reloj principal	16 MHz	16 MHz	1.2 GHz	2 GHz
Dimensiones	18x45 mm	68.6x53.4 mm	85x56x17 mm	82x58x22 mm
RAM	2 kB	2 kB	1 Gb	2 Gb
Flash	32 kB	32 kB	tarjeta SD	MicroSD
Voltaje de entrada	7 ~12 V	7 ~12 V	5 V	5 V
Puertos USB	—	—	4 x USB 2.0	2 x USB 3.0 x USB 2.0
Software	Arduino	Arduino	Raspbian y Ubuntu	Ubuntu
Precio (dólares)	13.98	23	36.99	59

Tabla A.5: Especificaciones de los dispositivos de comunicación para los sensores.

## A.2.3. Reducción del espacio morfológico

En esta etapa y gracias al manejo de criterios de exclusión, es posible reducir el espacio morfológico a un número de posibilidades manejables. Este análisis consiste en descartar todas aquellas opciones que son inconsistentes o que no son compatibles.

Los criterios de exclusión permiten seleccionar los dispositivos que hacen parte de un mismo plano y que garantizan la viabilidad y la calidad de la solución electrónica. Los criterios que han sido definidos son los siguientes:

1. **Disponibilidad:** hace referencia a la facilidad de adquirir el producto en el mercado.
2. **Escalabilidad:** hace referencia a la capacidad que tiene el dispositivo de permitir la incorporación de nuevas funcionalidades en el transcurso del tiempo.
3. **Incompatibilidad:** hace referencia a la falta de compatibilidad de los dispositivos para los propósitos establecidos.
4. **Soporte y material de apoyo:** hace referencia a la cantidad y calidad de información que permitan conocer la configuración y estructura de los dispositivos.
5. **Difícil implementación:** hace referencia a la dificultad de implementar el dispositivo.
6. **Sobredimensionamiento:** se refiere a que un dispositivo pese a que presente grandes cualidades, resulta ser exagerado para su implementación.
7. **Alto Costo:** se espera que el desarrollo de la plataforma sea de bajo costo, por lo tanto, este es un criterio determinante para tener en cuenta en la selección del dispositivo

8. **Requiere de dispositivos o circuitos adicionales:** se requiere la adquisición de dispositivos adicionales para la implementación de un dispositivo.
9. **Herramientas de desarrollo:** hace referencia a la disponibilidad y compatibilidad de herramientas que pueden ser utilizadas según el gusto del desarrollador.

Con los criterios de exclusión previamente establecidos, se procede a realizar el análisis respectivo de cada uno de los dispositivos, con el propósito de descartar uno o varios de ellos.

### **Fuente de alimentación:**

Una batería puede alimentar el robot durante 30-45 minutos aproximadamente. Debido a esto, su costo y la adquisición de componentes adicionales para conectarlas al robot (criterio 7 y 8), se selecciona para el primer prototipo una alimentación mediante cables externos. Como resultado, el robot puede usarse durante períodos prolongados.

### **Actuadores:**

El séptimo criterio (alto costo) o el octavo criterio (requerimiento de dispositivos adicionales) para obtener una retroalimentación de la información necesaria para programar el robot son criterios de exclusión para los siguientes servomotores: HSR-5990TG, Dynamixel AX-18A, HS-5955TG, HS-7950TH y futaba S3305. Teniendo en cuenta lo anterior el actuador escogido fue el Dynamixel AX-12A.

### **Sensores:**

Para la selección del sensor o IMU, se descartó en primer lugar el sensor NGIMU teniendo en cuenta el séptimo criterio (alto costo). Además, dado que los tres sensores presentan características similares, para la selección final de este componente nos basamos en el primer criterio (disponibilidad), debido a que ya se cuenta con el sensor BNO055 de la marca Adafruit y se ha trabajado con él en proyectos anteriores.

### **Dispositivo de comunicación para el actuador:**

El tercer criterio de exclusión (incompatibilidad) nos hacen descartar los módulos Nrf24l01, connectBlue OW451 y Xbee pro-900, debido a que para nuestro primer prototipo robótico no se ha planteado que la plataforma cuente con un sistema inalámbrico para la comunicación con los actuadores. Además, mediante el quinto criterio (difícil implementación), descartamos a los módulos USB2AX y CP2102. Obteniendo como resultado el uso del modulo U2D2, el cual tiene una gran compatibilidad con los actuadores anteriormente seleccionados.

### Dispositivo de comunicación para el sensor:

Con base a los criterios número seis (sobredimensionamiento) y número siete (alto costo) se descartan los dispositivos: Arduino uno, Raspberry pi 3 B y Odroid XU4. Teniendo como resultado y dispositivo a usar en el proyecto para la comunicación entre el sensor y nuestro dispositivo de procesamiento a una tarjeta Arduino nano.

#### A.2.4. Solución electrónica definitiva

Con base en los criterios de exclusión ha sido posible determinar los elementos que no harán parte del sistema, de este modo, la propuesta a la selección de dispositivos vista desde cada uno de los planos correspondientes al sistema se puede ver en la siguiente figura A.3.

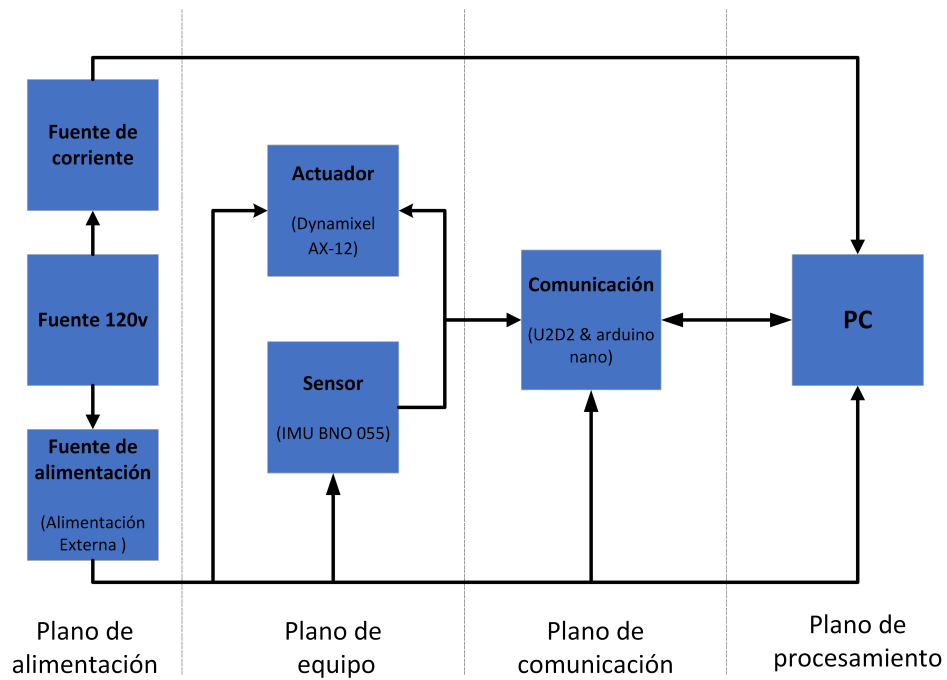


Figura A.3: Diagrama de bloques con la solución de electrónica definitiva.

# Apéndice B

## Selección del software de simulación

Antes de implementar aplicaciones o algoritmos en máquinas reales, probarlos en simuladores es un paso necesario, porque tal práctica puede ayudar a minimizar los riesgos y gastos potenciales involucrados en las pruebas de plataformas robóticas en el mundo real y mejorar la seguridad y la productividad.

### B.0.1. Software de simulación robótica

Los simuladores son programas que permiten al usuario probar y crear comportamientos para robots y otros componentes, sin tener que interactuar físicamente con ellos.

La mayoría de los simuladores modernos incluyen una variedad de características que permiten una simulación realista y útil, como la capacidad de mostrar modelos 3D de robots y entornos, ya sea con herramientas de modelado incluidas o la capacidad de importar modelos externos, motores de física, módulos de cálculo y secuencias de comandos con comportamientos en los lenguajes de codificación comúnmente utilizados.

Los beneficios de usar un simulador son inmensos, en la forma en que reduce los costos, ahorrando tiempo y dinero, permitiendo que se prueben varias alternativas sin costos ni riesgos y sin tiempo de inactividad. También permite al usuario optimizar la cantidad de vehículos necesarios, las áreas más problemáticas o qué estrategias de gestión usar.

Hay muchas herramientas que se pueden usar en la simulación. Desde herramientas de simulación gratuitas y hasta software con la tarifa de licencia. A continuación se puede encontrar una lista completa con todo el software de simulación que se utiliza actualmente en robótica.

- **ODE (Open Dynamics Engine):** es un motor físico de código abierto de alto rendimiento para simular la dinámica rígida del cuerpo. Cuenta con todas las características, estable e independiente de la plataforma con una API C/C++ fácil de usar. Tiene avanzados tipos de articulaciones y detección de colisión integrada con

fricción. ODE es útil para simular vehículos, objetos en entornos de realidad virtual y criaturas virtuales. Actualmente se utiliza en muchos juegos de computadora, herramientas de autoría 3D y herramientas de simulación.

- **Bullet:** es un motor de física de código abierto que simula detección de colisiones, y la dinámica de cuerpos suaves y rígidos. Se ha utilizado en videojuegos, así como para efectos visuales en películas. La última versión (v.2.82) también admite el algoritmo del cuerpo articulado de Featherstone y un problema de complementariedad lineal mixto. Esto lo hace más adecuado para aplicaciones de robótica, ya que la dinámica se resuelve en coordenadas conjuntas y la resolución de los contactos es más estable.
- **DART (Dynamic Animation and Robotics Toolkit):** es un motor físico, multiplataforma y de código abierto creada por Graphics Lab y Humanoid Robotics Lab en Georgia Institute of Technology con contribuciones continuas del Laboratorio de Robótica de la Universidad de Washington y Open Source Robotics Foundation. La biblioteca proporciona estructuras de datos y algoritmos para aplicaciones cinemáticas y dinámicas en robótica y animación por computadora. DART se distingue por su precisión y estabilidad debido a su uso de coordenadas generalizadas para representar sistemas articulados de cuerpo rígido y el algoritmo del cuerpo articulado de Featherstone para calcular la dinámica del movimiento. Brinda acceso completo a cantidades cinemáticas internas y dinámicas, como la matriz de masas, Coriolis y fuerzas centrífugas, matrices de transformación y sus derivadas. También proporciona un cálculo eficiente de las matrices jacobianas para puntos corporales arbitrarios y marcos coordinados. Tiene aplicaciones en robótica y animación por computadora porque cuenta con un simulador dinámico multicuerpo y varias herramientas cinemáticas para el control y la planificación del movimiento.
- **Gazebo:** es un simulador que permite de manera precisa y eficiente trabajar con poblaciones de robots en entornos interiores y exteriores complejos. Es de carácter libre, y se ha convertido en un proyecto vibrante con una comunidad muy activa, la cual ha permitido disponer a los usuarios de una gran cantidad de modelos y complementos de simulación. Además, cuenta con un motor robusto de física, gráficos de alta calidad e interfaces programáticas y gráficas convenientes. Admite múltiples motores de física (ODE, Bullet, DART, Simbody) y utiliza OGRE para proporcionar una representación realista de entornos que incluyen iluminación, sombras y texturas de alta calidad. Gracias a su estructura modular y basada en complementos, se puede ampliar fácilmente con nuevas funciones. Fue diseñado idealmente para trabajar con múltiples robots en entornos al aire libre. Es soportado por la Open-Source Robotics Foundation (OSRF) y cuenta con un gran soporte para conectar con ROS. Es la herramienta de software oficial para DARPA Robotics Challenge.
- **ARGoS:** es un simulador de robot multi-física. Puede simular enjambres de robots heterogéneos en gran escala de manera eficiente. Se puede personalizar fácilmente agregando nuevos complementos. Tiene un enfoque profundamente modular, esto implica que todos los aspectos relevantes de una simulación pueden ser anulados,



para una flexibilidad máxima. Los módulos se implementan como complementos cargados en tiempo de ejecución. Con respecto a los simuladores existentes, ARGoS permite agregar funcionalidad en forma de nuevos sensores, actuadores, componentes de robot, visualizaciones e incluso nuevos motores de física y nuevos “medios” de comunicación. Un medio, es un complemento que implementa algoritmos adecuados para simular medios de comunicación del robot (por ejemplo, alcance y soporte, WiFi, stigmergy a través de RFID, etc.). ARGoS se ejecuta para realizar simulaciones detalladas. Los controladores de robot usan una API específica del robot llamada interfaz de control. La misma API se implementa en el robot, lo que permite una transición fluida del código de la simulación a la realidad. Respecto a los motores físicos, el usuario puede elegir para usar en una simulación. Además, el espacio físico se puede dividir en varias regiones, cada una controlada por un motor de física específico. En otras palabras, ARGoS permite simulaciones multimotor. Este simulador se ejecuta en Linux, Mac OSX y recientemente en Windows. Se publica bajo los términos de la licencia de MIT.

- **CoppeliaSim:** anteriormente conocido como V-REP, es un simulador de robots con entorno de desarrollo integrado, el cual se basa en una arquitectura de control distribuido, es decir, cada objeto y/o modelo puede controlarse individualmente a través de un script incrustado, un complemento, un nodo ROS o BlueZero, un cliente API remoto o un sistema de solución personalizado. Esto hace que V-REP sea muy versátil e ideal para aplicaciones multi-robot. Los controladores se pueden escribir en C/C++, Python, Java, Lua, Matlab u Octave. Se utiliza para el desarrollo de algoritmos rápidos, simulaciones de automatización de fábricas, creación rápida de prototipos y verificación, educación relacionada con la robótica, monitoreo remoto, doble verificación de seguridad, etc. Es un software producido por Coppelia Robotics. Al igual que Gazebo, es compatible con múltiples motores de física (ODE, Bullet, Vortex).
- **Webots:** es un software de simulación de robótica móvil que le proporciona un entorno de prototipado rápido para modelar, programar y simular robots móviles. Bibliotecas de robots proporcionadas permiten transferir sus programas de control a varios robots móviles reales disponibles comercialmente. Permite definir y modificar una configuración completa de robótica móvil, incluso varios robots diferentes que comparten el mismo entorno. Para cada objeto, puede definir varias propiedades, como la forma, el color, la textura, la masa, la fricción, etc. Puede equipar a cada robot con una gran cantidad de sensores y actuadores disponibles. Puede programar estos robots utilizando su entorno de desarrollo favorito, simularlos y, opcionalmente, transferir los programas resultantes a sus robots reales. Webots™ ha sido desarrollado en colaboración con el Instituto Federal Suizo de Tecnología en Lausana, probado exhaustivamente, bien documentado y mantenido continuamente durante más de 7 años. Ahora es el principal producto comercial disponible de Cyberbotics Ltd.
- **OpenRave:** proporciona un entorno para probar, desarrollar y desplegar algoritmos

de planificación de movimiento en aplicaciones de robótica en el mundo real. El foco principal está en la simulación y análisis de información cinemática y geométrica relacionada con la planificación del movimiento. Su naturaleza independiente permite que se integre fácilmente en los sistemas de robótica existentes. Proporciona muchas herramientas de línea de comandos para trabajar con robots y planificadores, y el núcleo en tiempo de ejecución es lo suficientemente pequeño como para ser utilizado dentro de controladores y marcos más grandes. Una aplicación objetivo importante es la automatización de la robótica industrial.

- **Robotran:** es un software que genera modelos simbólicos de sistemas de múltiples cuerpos (MBS), que se pueden analizar y simular en Matlab y Simulink. Es desarrollado por el grupo de investigación Multicuerpo del Centro de Investigación en Mecatrónica (CEREM) que forma parte del instituto del Instituto de Ingeniería Mecánica, de Materiales e Ingeniería Civil (iMMC) de la Universidad Católica de Lovaina (UCL).
- **Vortex Studio:** es una plataforma de simulación en tiempo real unificada que le permite respaldar la investigación de sistemas robóticos y mecatrónicos, y enseñar conceptos de simulación y prototipos virtuales basados en modelos a estudiantes que utilizan herramientas de nivel profesional. Le permite transformar modelos CAD estáticos en mecanismos virtuales interactivos, lo que le permite probar sistemas inteligentes y diseño de equipos robóticos en entornos 3D envolventes.
- **OpenSim:** es un conjunto de herramientas para el modelado musculoesquelético y la simulación dinámica del movimiento, desarrollado en la Universidad de Stanford y respaldado por los NIH de EE. UU. Y por DARPA. Está disponible de forma gratuita, de código abierto y extensible a través de complementos de usuario. El motor de física de este proyecto es SimBody una API C++ de código abierto que implementa los algoritmos de Featherstone para la mecánica corporal rígida, con soporte de diferentes modelos de contacto.
- **Adams** es definido propiamente como el software de dinámica de multicuerpos (MBD) más famoso y ampliamente utilizado en el mundo. Mejora la eficiencia de la ingeniería y reduce los costos de desarrollo de productos al permitir la validación temprana del diseño a nivel del sistema. Los ingenieros pueden evaluar y gestionar las complejas interacciones entre disciplinas, incluidos el movimiento, las estructuras, la actuación y los controles para optimizar mejor los diseños de productos en cuanto a rendimiento, seguridad y comodidad. Junto con amplias capacidades de análisis, Adams está optimizado para problemas a gran escala, aprovechando los entornos informáticos de alto rendimiento. Utilizando la tecnología de solución de dinámica multicuerpo, ejecuta dinámicas no lineales en una fracción del tiempo requerido por las soluciones FEA. Las cargas y fuerzas calculadas por las simulaciones de Adams mejoran la precisión de FEA al proporcionar una mejor evaluación de cómo varían a lo largo de una gama completa de entornos operativos y de movimiento.

- **MORSE:** es un simulador genérico para la robótica académica. Se centra en la simulación 3D realista de entornos pequeños a grandes, en interiores o al aire libre. Puede controlarse completamente desde la línea de comandos. Las escenas de simulación se generan a partir de scripts de Python. Viene con un conjunto de sensores estándar (cámaras, escáner láser, GPS, odometría, etc), actuadores (controladores de velocidad, controladores de puntos de ruta de alto nivel, controladores conjuntos genéricos) y bases robóticas (ATRV, Pioneer3DX, vehículos de ruedas, PR2, entre otros). La representación MORSE se basa en el Blender Game Engine. El Game Engine basado en OpenGL admite shaders, proporcionando opciones avanzadas de iluminación, admite texturas múltiples y utiliza la biblioteca Bullet de última generación para la simulación física.
- **MuJoCo:** es un motor dinámico desarrollado principalmente por E. Todorov y ahora propiedad de Roboti LLC. Es uno de los motores de física más recientes, concebido para simular robots y sistemas biomecánicos. Es compatible con cálculos paralelos, proporciona dinámica inversa con contactos y restricciones de igualdad, implementa varias dinámicas de contacto. Es adecuado para la optimización del control y se puede usar en tiempo real dentro de un lazo de control.
- **XDE:** es un motor de choque de realidad virtual que simula las operaciones de montaje, desmontaje y ensamblaje para el diseño de unidades mecánicas / trenes de potencia. A partir de los datos CAD que se importan automáticamente. Los códigos se pueden integrar en un entorno virtual e interactivo para simular el gesto del operador. Es desarrollado por CEA LIST. Utilizado para aplicaciones de realidad virtual en contextos industriales.
- **OpenHRP3:** (Open Architecture Human-centered Robotics Platform versión 3) es una plataforma de software integrada para simulaciones de robots y desarrollos de software. Esta versión se distribuye como un software de código abierto. Permite a los usuarios inspeccionar un modelo de robot original y un programa de control mediante simulación dinámica. Proporciona varios componentes de software y bibliotecas de cálculo que se pueden utilizar para desarrollos de software relacionados con la robótica. El motor de cálculo de dinámica está casi desarrollado por "Nakamura Lab, Depto. De Mechano Informatics, Universidad de Tokio" la interfaz gráfica está a cargo de "General Robotix, Inc". Las otras partes se desarrollan como un trabajo cooperativo de "Humanoid Research Group" "Task-Intelligence Research Group.<sup>en</sup> "Intelligent Systems Research Institute", "Instituto Nacional de Ciencia y Tecnología Industrial Avanzada (AIST)"
- **Microsoft Robotics Developer Studio:** (MRDS) es un software de simulación 3D gratuito diseñado por Microsoft con soporte para una amplia gama de sensores y otras plataformas robóticas. La plataforma es compatible con todas las versiones del sistema operativo Windows y podría usarse con C Visual Studio 2010 para crear nuevos servicios para RDS. El MRDS ofrece soporte para simular robots de servicio, incluida la tecnología Kinect. En colaboración con Parallax, Microsoft diseña una

plataforma robótica llamada Eddie utilizada para la creatividad, la innovación y la experimentación.

## B.0.2. Comparación de los simuladores de robótica

En la sección anterior se mencionaron numerosos software de simulación de robótica disponibles en la actualidad, cada uno con sus propias características y funcionalidades únicas. Sin embargo, para esta comparación en particular se han elegido tres de los software más populares y utilizados en la industria: Webots, V-REP y Gazebo. Esto se debe a que estos tres software son considerados líderes en el campo de la simulación de robótica, y se utilizan ampliamente en la industria y en la academia para la investigación, el desarrollo y la enseñanza de la robótica. Además, cada uno de estos software ofrece una amplia gama de características y herramientas que los hacen adecuados para diversas aplicaciones en el campo de la robótica móvil.

		Simulador	
Gazebo	Características	Tipos de robots	Ruedas; patas; humanoides; Brazos; drones y Otros
		Sensores y actuadores	Varios Modelos de Sensores y Motores
		Compatibilidad	Player; TCP/IP; ROS y otros.
		Motor de física	ODE; Bullet; Simbody; DART; OpenGL; GLUT y OGRE.
		Lenguajes de programación	XML; C++.
		Otras características	Los usuarios pueden agregar nuevos modelos de robots, sensores y actuadores; visualización 3D; fuente abierta; plugins; modelos de robots; simulación de nubes; herramientas de línea de comandos y de uso gratuito.
Webots	Características	Tipos de robots	Ruedas; brazos; bípedos; multi-patas; modular; automóviles; drones; submarino y aeroespacial.
		Sensores y actuadores	Varios Modelos de Sensores y Motores.
		Compatibilidad	ROS; TCP/IP; MATLAB y otros.
		Motor de física	Qt; ODE; OpenGL.
		Lenguajes de programación	C/C++; Python; Java; MATLAB y ROS.
		Otras características	Los usuarios pueden agregar nuevos modelos de robots, sensores y actuadores; visualización 3d; fuente abierta; biblioteca de robots, objetos y materiales; versiones gratuitas y de pago.
V-REP (CoppeliaSim)	Características	Tipos de robots	Multi-robots
		Sensores y actuadores	Multisensor y actuador
		Compatibilidad	ROS; BlueZero; LabView; TCP/IP.
		Motor de física	Bullet; ODE; Vortex y Newton.
		Lenguajes de programación	C/C++; Lua; Java; Python; LabView; MATLAB y Octave.
		Otras características	Biblioteca de robots, sensores y actuadores, objetos y materiales; visualización 3d; libre y versiones de pago.

Tabla B.1: Características principales de los simuladores.

En el año 2021 José Caio Camargo, realizó un análisis y comparación de los tres software anteriormente mencionados considerando elementos cuantitativos. Esta comparación considera aspectos como la CPU, la huella de memoria y el uso del disco que se evalúan utilizando un escenario de robot móvil en una situación doméstica.

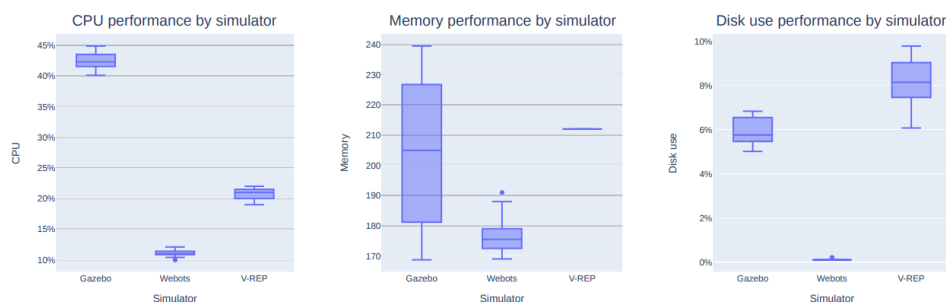


Figura B.1: Resultados promedio de 20 ejecuciones de los elementos medidos para comparar los simuladores en un enfoque cuantitativo.

La Figura B.1 resume los resultados de las 20 ejecuciones mediante la representación de diagramas de caja estadísticos para cada elemento comparativo y cada simulador. Se puede observar que tanto el uso de la CPU como del disco muestra una variación considerable entre los simuladores. En términos generales, Webots es el simulador que utiliza menos recursos para la ejecución de tareas, pudiendo simular eficientemente con el uso de CPU entre [10% - 12.1%] y consumo de memoria entre [169 MB - 191 MB]. Adicionalmente, la simulación se ejecuta de forma rápida, dado que no se requirieron operaciones de disco (valores entre [0.1% - 0.2%]). En comparación, V-REP requiere casi el doble de CPU que Webots, con un uso de CPU entre [19% - 22%], y la ejecución de la simulación realiza algunas operaciones en el disco con valores entre [6.07% - 9.78%]. Sin embargo, el consumo de memoria presentado para V-REP es estable en todas las ejecuciones; esto es una ventaja, incluso siendo un 20,39% en promedio superior a la memoria requerida por Webots. Finalmente, Gazebo presenta menos operaciones en disco que V-REP, con valores entre [5.01% - 6.83%]. Sin embargo, debido a los procesos de cliente y servidor requeridos para la ejecución de la simulación, Gazebo usa el doble de CPU que V-REP, con un uso de CPU entre [40.11% - 44.88%]. En relación a la memoria, si bien Gazebo presenta una cantidad promedio menor que V-REP, también muestra inestabilidad con grandes variaciones entre ejecuciones.

El simulador escogido fue Webots, teniendo en cuenta los resultados presentados en el anterior estudio, siendo la principal razón su ligereza en el uso de recursos computacionales y simplicidad a la hora de añadir nuevos prototipos robóticos, características indispensables en una herramienta educativa, en la que sobresalía al resto de alternativas. El siguiente paso realizado en el proyecto fue un estudio más exhaustivo del funcionamiento de Webots. Con esto se consiguieron dos cosas: la primera, conocer mejor y aprender a utilizar el software; la segunda, verificar que la selección era adecuada.

Durante este proceso se vio que Webots es un software complejo, que permite crear simulaciones con múltiples opciones de configuración y objetos, que a su vez disponen de una gran variedad de parámetros, utilizados para describir sus propiedades físicas, desde su apariencia (coeficiente difuso, coeficiente especular, textura), a sus movimientos (masa, inercia, fricción); también hay parámetros que describen la forma en la que interactúan

los objetos.

En la siguiente sección hay más detalles del simulador, concretamente de la interfaz y sus objetos. Para más información, sobre cualquier funcionalidad de Webots, se puede consultar el manual del simulador; se recomiendan los capítulos de conectividad, interfaz de usuario y entidades.

### B.0.3. Simulador Webots

Webots es un software de simulación profesional que permite programar y simular en un entorno 3D robot móviles, teniendo en cuenta propiedades físicas que se pueden establecer como la gravedad, la masa del objeto, textura, fricción entre otras, que buscan darle realismo a los movimientos y condiciones del sistema. Dentro de sus funciones Webots importa figuras, piezas o robots completos en formato VRML a los cuales se les puede aginar todo tipo de sensores y actuadores como lo son cámaras, sensores infrarrojos, servomotores entre otros. Webots permite programar en los lenguajes C, C++, Java, Python y Matlab, cuyos controladores se pueden trasladar a robots reales, teniendo de antemano el comportamiento deseado visto en el entorno de simulación.



Figura B.2: Interfaz del software de simulación Webots.

Webots trabaja con una jerarquía de nodos para el mundo virtual y la programación típicos de los mundos virtuales VRML, además de otros propios para ir dándole propiedades y capacidades para crear sistemas más complejos entre los que están:

- Nodo Worlinfo:** En este se establece las propiedades del ambiente 3D junto con la información de este, como ¿quién lo creó? y la fecha en que se hizo, además de definir propiedades físicas como la gravedad en los ejes de coordenadas, el paso del tiempo y la velocidad de simulación.

- **Nodo Viewpoint:** Permite orientar la cámara que interactúa con el mundo permitiendo rotar, hacer zoom y trasladar la cámara en sus respectivos subnodos.
- **Nodo Pointlight:** Dentro de sus subnodos permite cambiar la luminosidad, el color, intensidad, la posición y número de fuentes de luz, así como manipular las sombras generadas.
- **Nodo World:** Son nodos a los que se le asigna una figura o robot y un subnodo de transformación en el que se le asigna la apariencia y orientación del robot. Este nodo se utiliza como base para creación del robot ya que en este se define el tipo pieza, asignándole propiedades mediante nodos más específicos como los son: el nodo Robot, generalmente asignado a todo un ensamble importado, al que se le pueden asignar todas las propiedades de un robot, como lo son el controlador que usa, si posee batería y cómo sería el consumo, además de los subnodos de las articulaciones y especificar las colisiones entre estas. También se utilizan nodos más específicos para sensores y otros componentes, por ejemplo para usar servomotores se llama el Nodo Servo en el que se especifica la velocidad, torque, posición y de más características de los motores. El Nodo Solid hace referencia a un objeto y sus propiedades físicas como dimensiones, masa y material. Y también está el Nodo Shape para la apariencia y geometría para crear objetos.

En la siguiente Figura B.3, se parecía mejor la jerarquía del nodo World.

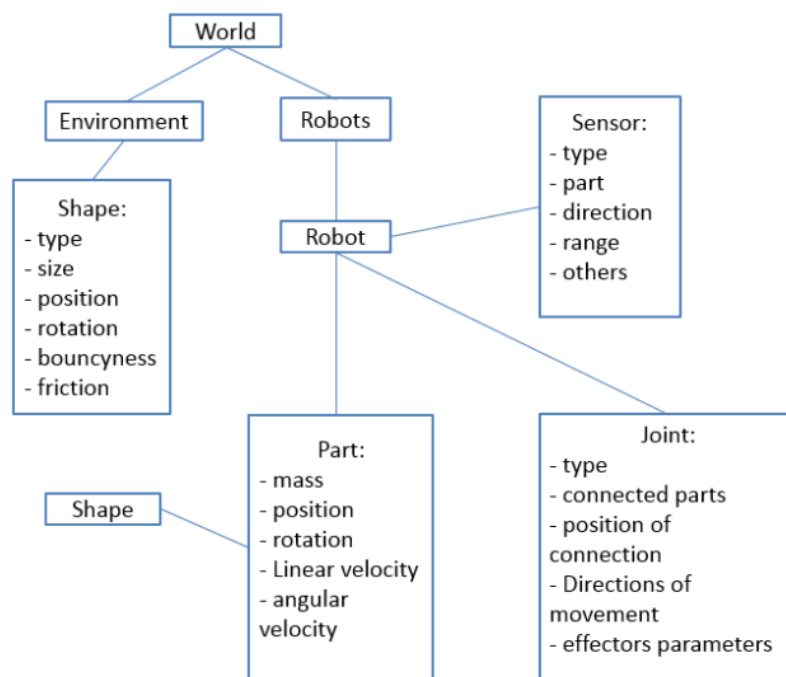


Figura B.3: Jerarquía principal del nodo World.