



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO E IMPLEMENTACIÓN DE ARQUITECTURA PARA CONTROL DE  
LOCOMOCIÓN BASADA EN GENERADORES DE PATRONES CENTRALES  
CPG PARA UN ROBOT BIOMIMÉTICO**

**Víctor Manuel Carranza Mejicanos**

Asesorado por el Ing. Jorge Augusto Balsells Orellana

Guatemala, octubre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE ARQUITECTURA PARA CONTROL DE  
LOCOMOCIÓN BASADA EN GENERADORES DE PATRONES CENTRALES  
CPG PARA UN ROBOT BIOMIMÉTICO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**VÍCTOR MANUEL CARRANZA MEJICANOS**

ASESORADO POR EL ING. JORGE AUGUSTO BALSELLS ORELLANA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN ELECTRÓNICA**

GUATEMALA, OCTUBRE DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Helmunt Federico Chicol Cabrera
EXAMINADOR	Ing. Jose Aníbal Silva De Los Ángeles
EXAMINADOR	Ing. Marvin Marino Hernández Fernández
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **DISEÑO E IMPLEMENTACIÓN DE ARQUITECTURA PARA CONTROL DE LOCOMOCIÓN BASADA EN GENERADORES DE PATRONES CENTRALES CPG PARA UN ROBOT BIOMIMÉTICO**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 16 de julio de 2020.



**Víctor Manuel Carranza Mejicanos**

Guatemala, viernes 23 de julio de 2021

Ingeniero  
Julio Cesar Solares Peñate  
Coordinador del área de electrónica  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería  
Universidad de San Carlos de Guatemala

Apreciable Ingeniero Solares:

Me permito dar aprobación al trabajo de graduación titulado "**DISEÑO E IMPLEMENTACIÓN DE ARQUITECTURA PARA CONTROL DE LOCOMOCIÓN BASADA EN GENERADORES DE PATRONES CENTRALES CPG PARA UN ROBOT BIOMIMÉTICO**" del señor Víctor Manuel Carranza Mejicanos, cuyo carnet estudiantil es "200915343" por considerar que cumple con todos los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y yo, como su asesor, nos hacemos responsables por el contenido, conclusiones y desarrollo del mismo.

Sin otro particular, me es grato saludarle.

Atentamente:

  
Ing. Jorge Augusto Balsells Orellana  
Asesor  
Colegiado 17,029

Jorge Augusto Balsells Orellana  
Ingeniero Electrónico  
Colegiado No. 17,029



Guatemala, 5 de agosto de 2021

**Señor director**  
**Armando Alonso Rivera Carrillo**  
**Escuela de Ingeniería Mecánica Eléctrica**  
**Facultad de Ingeniería, USAC**

Estimado Señor director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE ARQUITECTURA PARA CONTROL DE LOCOMOCIÓN BASADA EN GENERADORES DE PATRONES CENTRALES CPG PARA UN ROBOT BIOMIMÉTICO**, desarrollado por el estudiante **Víctor Manuel Carranza Mejicanos**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

**ID Y ENSEÑAD A TODOS**

A handwritten signature in blue ink, appearing to read 'Julio César Solares Peñate'.

**Ing. Julio César Solares Peñate**  
**Coordinador de Electrónica**



**REF. EIME 147. 2021.**

**El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; VÍCTOR MANUEL CARRANZA MEJICANOS, titulado: DISEÑO E IMPLEMENTACIÓN DE ARQUITECTURA PARA CONTROL DE LOCOMOCIÓN BASADA EN GENERADORES DE PATRONES CENTRALES CPG PARA UN ROBOT BIOMIMÉTICO, procede a la autorización del mismo.**

  
**Ing. Armando Alonso Rivera Carrillo**



**GUATEMALA, 23 DE SEPTIEMBRE 2,021.**



DTG. 478-2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE ARQUITECTURA PARA CONTROL DE LOCOMOCIÓN BASADA EN GENERADORES DE PATRONES CENTRALES CPG PARA UN ROBOT BIOMIMÉTICO**, presentado por el estudiante universitario: **Víctor Manuel Carranza Mejicanos**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Inga. Anabela Cordova Estrada

Decana



Guatemala, octubre de 2021

AACE/cc

## **ACTO QUE DEDICO A:**

- Mi madre** Luisa Mejicanos, por su incondicional apoyo y por enseñarme que el talento sin esfuerzo no llega lejos.
- Mi padre** Víctor Carranza, por su incondicional apoyo y por enseñarme a tener curiosidad por el mundo que me rodea.
- Mi abuela** Frida González, por su incondicional apoyo y por darme herramientas para ser más independiente.
- Mi abuelo** “Manolo” Carranza, por su incondicional apoyo en vida, lamentando que no pueda acompañarme este día.
- Mi tía** Claudia Carranza, por su incondicional apoyo como parte del equipo de mediadores en la familia.
- Mi hermano** Adán Carranza, por motivarme a mejorar constantemente, aún sin consciencia de ello.
- Mi hermano** Dérick Carranza, por su apoyo desinteresado para mi crecimiento personal.

**Mi primo**

Manuel Carranza, por su presencia en mi vida.

**Señora**

María Elena López, por su apoyo y amistad durante la mayor parte de mi vida.

## **AGRADECIMIENTOS A:**

<b>Universidad de San Carlos de Guatemala</b>	Por ser mi segunda casa y ayudarme a formar mi percepción del mundo.
<b>Facultad de Ingeniería</b>	Por ayudarme a entender el tipo de profesional que deseo ser, pero más aún, el que no.
<b>IEEE</b>	Por darme la oportunidad de desarrollar múltiples habilidades y ayudarme a tener impacto en mi comunidad.
<b>A mis amigos y amigas</b>	Con antigüedad y recientes, a quienes se quedaron y a quienes tomaron caminos separados.
<b>A mi asesor</b>	Jorge Balsells, por su amistad y apoyo incondicional en la realización de este trabajo.

# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN .....	XVII
1. ROBOTS MÓVILES .....	1
1.1. Definición de robot.....	1
1.2. Definición de robot móvil .....	3
1.3. Factores de diseño para robots móviles .....	4
1.4. Paradigmas de operación de robots móviles.....	5
1.4.1. Robot semiautónomo.....	5
1.4.2. Robot autónomo .....	6
1.5. Modelos tradicionales de subsistemas de locomoción de robots móviles .....	6
1.5.1. Modelo de rueda frontal directora .....	7
1.5.2. Modelo de control diferencial.....	7
2. GENERADORES DE PATRONES CENTRALES CPG.....	13
2.1. Características relevantes para ingeniería y robótica.....	15
2.2. Antecedentes en robótica .....	16
2.2.1. Pleurobot .....	16
2.2.2. Lynx.....	18
2.2.3. Cube Revolutions .....	19

3.	MODELO COMPUTACIONAL DE GENERADORES DE PATRONES CENTRALES .....	21
3.1.	Simplificación de modelos.....	21
3.2.	El modelo de Kuramoto.....	23
4.	MÉTODO PARA EL DISEÑO DEL CONTROLADOR DE LOCOMOCIÓN .....	27
4.1.	Selección de características mecánicas del robot controlado .....	27
4.2.	Selección de dispositivos para dar movimiento al robot.....	28
4.3.	Selección de la arquitectura del controlador .....	29
4.4.	Selección del modelo matemático para la implementación de CPG .....	31
4.4.1.	Parámetros de definición de patrón.....	31
4.4.2.	Parámetros de ajuste de patrón .....	33
4.5.	Selección de procesador.....	34
5.	IMPLEMENTACIÓN DE CONTROLADOR .....	35
5.1.	Hardware.....	35
5.1.1.	Estructura del robot .....	35
5.1.2.	Circuito eléctrico del controlador .....	37
5.1.2.1.	Arduino Nano .....	37
5.1.2.2.	Adafruit PCA9685 16-Channel Servo Driver.....	39
5.1.2.3.	Servomotores .....	40
5.1.2.4.	Fuente de alimentación .....	42
5.1.2.5.	Regulador de voltaje .....	43
5.2.	Programación de firmware .....	46
5.2.1.	Ambiente Integrado de Desarrollo (IDE) .....	46

5.2.2.	Librerías.....	48
5.2.2.1.	Arduino Wire.h.....	48
5.2.2.2.	Adafruit-PWM-Servo-Driver-Library .....	48
5.3.	Puesta en operación.....	49
5.3.1.	Funcionamiento de patrón de caminata.....	51
5.3.2.	Funcionamiento de patrón de trote .....	54
5.3.3.	Funcionamiento de patrón de galope .....	57
5.3.4.	Transiciones entre patrones .....	60
6.	PROPUESTAS A FUTURO.....	63
6.1.	Hardware y firmware.....	63
6.2.	Modelo matemático y computacional .....	65
	CONCLUSIONES .....	67
	RECOMENDACIONES.....	69
	BIBLIOGRAFÍA.....	71



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Rover Perseverance de la misión a Marte de la NASA en 2020 .....	4
2.	Diagramas de ventajas de robots con extremidades sobre robots con ruedas en distintas situaciones de terrenos rugosos .....	11
3.	Demostración de operación de Pleurobot .....	17
4.	Comparación de movimiento de Pleurobot y una salamandra .....	18
5.	Vista lateral de robot Lynx.....	19
6.	Vista lateral de robot Cube Revolutions .....	20
7.	Gráficas de amplitud vs tiempo de modelo con cuatro CPG para simulación de patrones de desplazamiento. (a) Caminata, (b) Trote, (c) Galope. ....	22
8.	Ilustración del método de Euler, la curva desconocida es azul y las aproximaciones poligonales se presentan en rojo, cada punto está separado horizontalmente por el valor del paso temporal.....	25
9.	Estructura interna de un servomotor .....	28
10.	Diagrama general de red de osciladores acoplados .....	30
11.	Diagramas que representan las relaciones de fase de cada extremidad de un robot cuadrúpedo para distintos patrones de desplazamiento.....	30
12.	Representación algebraica de la matriz de coeficientes de acople para un sistema de cuatro osciladores.....	32
13.	Matriz de coeficientes de acople para un sistema de cuatro osciladores con valores numéricos para las relaciones de autoacople.....	32

14.	Chip ATmega328P con encapsulado TQFP .....	34
15.	Vista simulada del diseño de “Hexapod Robot” .....	36
16.	“Hexapod Robot” impreso y adaptado con cuatro extremidades .....	36
17.	Arduino Nano .....	38
18.	Adafruit PCA9685 16-Channel Servo Driver .....	39
19.	Diagrama de conexión entre Arduino UNO, Adafruit PCA9685 16- Channel Servo Driver y un servomotor .....	40
20.	Imagen de servomotor MG996R.....	41
21.	Imagen de Eyeboot DC Universal Regulated Switching Power Supply AC to DC.....	43
22.	Imagen del regulador de voltaje utilizado.....	44
23.	Diagrama de conexión del circuito de control del sistema .....	45
24.	Placa de prueba montada.....	45
25.	Vista inicial del IDE de Arduino 1.8.10 .....	47
26.	Esquema de implementación de osciladores sobre el robot móvil .....	49
27.	Inicio de operación de CPG con matriz de patrón de caminata .....	52
28.	Comportamiento a mediano plazo de CPG con matriz de patrón de caminata .....	53
29.	Comportamiento a largo plazo de CPG con matriz de patrón de caminata .....	54
30.	Inicio de operación de CPG con matriz de patrón de trote .....	55
31.	Comportamiento a mediano plazo con matriz de patrón de trote .....	56
32.	Comportamiento a largo plazo de matriz de patrón de trote .....	57
33.	Inicio de operación de CPG con matriz de patrón de galope .....	58
34.	Comportamiento a mediano plazo de matriz de patrón de galope.....	59
35.	Comportamiento a largo plazo de matriz de patrón de galope .....	59
36.	Transición entre patrones de caminata a trote.....	60
37.	Transición entre patrones de trote a caminata.....	61

## TABLAS

I.	Subsistemas de locomoción resultantes de las combinaciones de los modelos básicos de locomoción (traducida de inglés a español).....	8
II.	Patrones de movimiento típicos y sus respectivas relaciones de fase entre osciladores para un robot cuadrúpedo.....	31
III.	Matrices de acople de diferentes patrones de desplazamiento.....	33
IV.	Especificaciones técnicas de Arduino Nano.....	38
V.	Especificaciones técnicas de servomotor MG996R .....	41
VI.	Especificaciones técnicas de fuente Eyeboot DC Universal Regulated Switching Power Supply AC to DC .....	42
VII.	Especificaciones técnicas del regulador de voltaje utilizado .....	44
VIII.	Matrices de acople con valores escalados.....	50



## LISTA DE SÍMBOLOS

Símbolo	Significado
<b>A</b>	Amperios
<b>cm</b>	Centímetros
$\frac{\partial}{\partial t}$	Derivada parcial con respecto al tiempo
°	Grados
=	Igualdad
<b>kg</b>	Kilogramos
*	Multiplicación
$\pi$	Número pi
-	Resta
+	Suma
$\Sigma$	Sumatoria
<b>V</b>	Voltios
<b>W</b>	Watts



## GLOSARIO

<b>Biomimético</b>	Que emula el comportamiento de organismos biológicos.
<b>Controlador</b>	Sistema diseñado para mantener una o múltiples variables dentro de un rango o rangos deseables.
<b>CPG</b>	Generador de Patrón Central ( <i>Central Pattern Generator</i> ), circuito biológico que utiliza señales periódicas para generar un patrón de movimiento coordinado de las extremidades de un organismo o mecanismo.
<b>Grados de libertad</b>	Cantidad de posibles movimientos relativos independientes entre piezas de un mecanismo.
<b>Microcontrolador</b>	Circuito integrado con memoria que permite almacenar y ejecutar instrucciones. Posee circuitos internos especializados para realizar tareas específicas.
<b>Perturbación</b>	Fuerza o señal externa que, al ingresar a un sistema, produce una salida o respuesta diferente a la que se encontraba en curso previamente.

**Robot**

Dispositivo electromecánico diseñado para realizar tareas de forma adaptable y eficiente.

**Robot móvil**

Robot cuyo marco de referencia, para fines de desplazamiento y posicionamiento, no es inercial.

## RESUMEN

Este informe contiene la teoría introductoria a la implementación de modelos de CPG en controladores para robots móviles con extremidades, especificaciones técnicas y de diseño para tal efecto. Se resume en los próximos párrafos su contenido estructurado por capítulos.

El primer capítulo desarrolla temas y conceptos sobre los que es necesario tener conocimiento para adentrarse al resto del trabajo. Define la robótica, robótica móvil y las características de robots móviles tradicionales para contextualizar las ventajas mostradas por el enfoque del trabajo en capítulos posteriores.

El segundo capítulo define el tema de enfoque central del trabajo, los generadores de patrones centrales CPG. Describe brevemente el contexto histórico de la investigación sobre este tema en biología y cómo esto se traslada al área de robótica móvil.

El tercer capítulo describe los modelos, matemáticos y computacionales simplificados, necesarios para la implementación de características de CPG en robótica móvil.

El cuarto capítulo especifica la convergencia de los conceptos desarrollados anteriormente y su importancia para el diseño de controladores de robots móviles basados en CPG.

El quinto capítulo explica las características técnicas de los componentes individuales utilizados para la implementación del controlador basado en CPG, los diagramas de conexión, y consideraciones necesarias para que este sea funcional.

El sexto y último capítulo presenta propuestas de mejoras futuras al trabajo presentado.

# OBJETIVOS

## General

Diseñar e implementar un sistema de lazo abierto para el control de locomoción de un dispositivo robótico de una o más extremidades, con al menos 1 grado de libertad por extremidad

## Específicos

1. Adaptar una arquitectura de control basada en modelos computacionales de generadores centrales de patrones, CPG.
2. Implementar la arquitectura de control diseñada en un modelo físico de robot móvil.
3. Evaluar cuantitativa y cualitativamente el desempeño del control según varíen los parámetros de amplitud, frecuencia y desfase de los CPG implementados.
4. Establecer parámetros mínimos del controlador para generar por lo menos tres patrones de desplazamiento diferentes.



## INTRODUCCIÓN

Por décadas, múltiples disciplinas científicas han investigado el movimiento de los seres vivos. ¿Cómo se originan las señales necesarias para generarlo?, ¿Cómo se coordinan movimientos individuales para crear un patrón de desplazamiento colectivo?, ¿Qué requieren para funcionar óptimamente?; estas son algunas de las interrogantes que han motivado múltiples investigaciones al respecto, y las conclusiones obtenidas apuntan a células que funcionan como sistemas de osciladores biológicos y producen las señales necesarias para ello.

Estos sistemas están presentes en la médula espinal tanto de vertebrados como invertebrados y se conocen como Generadores de Patrones Centrales o CPG, por sus siglas en inglés. Los CPG han sido objeto de estudio constante en biología, y en décadas recientes, ha despertado interés en otras disciplinas.

En robótica móvil, el uso de múltiples extremidades como medio de desplazamiento presenta múltiples beneficios con respecto al tradicional uso de ruedas. Los robots móviles con extremidades pueden adaptarse mejor a terrenos rugosos y recuperarse con mayor facilidad de caídas o accidentes. Generar patrones movimiento de extremidades para producir desplazamiento sigue siendo un desafío, hoy en día, aún se emplean métodos tradicionales.

Los CPG presentan características deseables para hacerle frente a los desafíos del uso de extremidades como medio de desplazamiento en robots móviles, puesto que generan patrones rítmicos de movimiento que, en conjunto, inducen desplazamiento. Es por esto que en las últimas décadas se han

propuesto múltiples modelos computacionales que los utilizan para emular comportamientos biológicos en robots móviles.

Este trabajo delinea los conceptos básicos para la implementación de CPG en robots móviles con múltiples extremidades, y las especificaciones técnicas del *hardware* necesario para una prueba de concepto en un robot móvil de cuatro extremidades. Al finalizar, el lector podrá utilizar estas especificaciones para reproducir el trabajo y escalarlo para futuras pruebas.

# 1. ROBOTS MÓVILES

## 1.1. Definición de robot

Existen múltiples definiciones de robot que abarcan una amplia variedad de características. Entre todas ellas hay marcados consensos que dan una mejor idea de los requisitos mínimos que debe reunir un dispositivo para ser considerado un robot, se enlistan a continuación.

- Capacidad sensorial: el dispositivo es sensible a perturbaciones y señales físicas (eléctricas, ópticas, térmicas, entre otros), del entorno en el que se desenvuelve.
- Capacidad de procesamiento y toma de decisiones: el dispositivo está equipado con un subsistema que toma decisiones, ya sea asistidas por intervención humana o autónomas, sobre acciones a ejecutar basadas en las señales a las que es sensible y limitadas por su capacidad de interacción con el entorno.
- Capacidad de interacción con el entorno: el dispositivo está equipado con un subsistema de actuadores mecánicos que le permiten realizar acciones en el entorno en el que se encuentra (desplazarse, manipular objetos, evadir obstáculos, entre otros).

Es posible también considerar algunas características adicionales para refinar los criterios para considerar un dispositivo como un robot. Estas se basan en los consensos mencionados anteriormente, pero agregan información que

ayuda a discriminar mejor entre máquinas con procesos automatizados y dispositivos robóticos. Se describen a continuación.

- El espacio de trabajo de un robot es externo a su sistema, mientras el de una máquina automatizada suele ser interno.
- La capacidad cognitiva (combinación de sensorial y de procesamiento y toma de decisiones) que un robot requiere suele ser mayor a la de una máquina automatizada. Esto se debe a que la interacción con el entorno suele implicar una mayor cantidad de variables físicas en comparación con las presentes en un ambiente controlado e interno de una máquina automatizada.
- El sistema de actuadores mecánicos de un robot suele ser más versátil y adaptable que el de una máquina automatizada. Esto se debe a que, generalmente, las máquinas automatizadas están diseñadas para realizar tareas repetitivas bajo condiciones estables, mientras que un robot trabaja en un ambiente menos controlado y, por tanto, necesita ser más tolerante a variaciones y adaptar sus acciones para cumplir su labor.

Un robot es, entonces, una máquina o dispositivo que cumple, por lo menos, con los requisitos mencionados y puede diferenciarse de una máquina automatizada utilizando como referencia las características adicionales posteriores.

Profundizando dentro del ámbito de los robots y la disciplina (o conjunto de disciplinas), que los trata, la robótica, se clasifica a estos dentro de dos grandes grupos: robots fijos y robots móviles.

Los robots fijos son aquellos cuyas bases y áreas de trabajo se mantienen constantes en el tiempo, mientras las de los móviles se desplazan en función de este.

Siendo que es la segunda clasificación, la de robots móviles, de principal interés para este trabajo, se expandirá su definición y temas derivados en el próximo inciso.

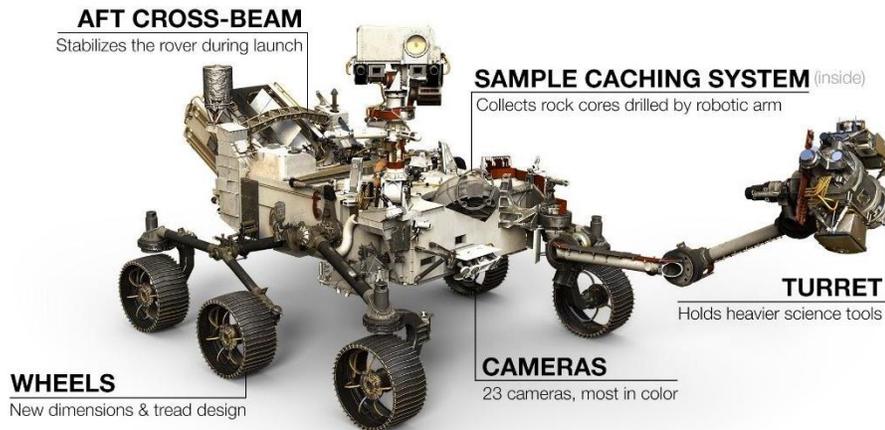
## **1.2. Definición de robot móvil**

Un robot móvil es aquel que se diseña para desplazarse sobre una superficie o dentro de un fluido específico. Puede tomar múltiples formas, según las funciones que quien lo diseña desee que ejecute.

Los robots móviles suelen ser sistemas electromecánicos robustos, con alto grado de adaptabilidad, percepción y poder computacional, características clave para desenvolverse efectivamente en espacios no controlados.

Para esto, emplean sensores y controladores que les permiten medir parámetros de su entorno, y actuadores o dispositivos que los dotan de habilidades motrices para interactuar con el mismo.

Figura 1. **Rover Perseverance de la misión a Marte de la NASA en 2020**



Fuente: NASA/JPL-Caltech. *Some differences between Mars 2020 and curiosity.*  
<https://mars.nasa.gov/resources/24716/some-differences-between-mars-2020-and-curiosity/>.  
Consulta: 26 de junio de 2021.

### 1.3. Factores de diseño para robots móviles

Actualmente, los robots móviles se diseñan y utilizan ampliamente a nivel de investigación; y cada vez son más los modelos disponibles comercialmente. Los motivos para su desarrollo varían en gran medida, ya que pueden ser de mucha ayuda en múltiples labores: exploración de áreas peligrosas o de difícil acceso (ruinas, lugares donde han ocurrido desastres, otros planetas, entre otros), monitoreo remoto de instalaciones (líneas de transmisión, enlaces submarinos de fibra óptica, entre otros); experimentos de biomecánica y más.

El terreno o medio en el que se desenvuelve el dispositivo es un factor que debe tomarse en cuenta durante la fase de desarrollo, esto dictará las bases para la selección de los componentes que se utilizarán, principalmente en subsistemas

de locomoción; sin embargo, ajustarlo solamente a un modelo específico de medio le resta adaptabilidad, porque un buen diseño debe tener holgura y flexibilidad, aún si es limitada, en sus parámetros de operación.

Se hace necesario considerar, además, las necesidades energéticas de los sistemas que componen al dispositivo. Puesto que la habilidad para desplazarse es una característica inherente a estos, estas necesidades son frecuentemente suplidas por subsistemas energéticos basados en baterías recargables; existen instancias en las que un robot móvil se alimenta de energía por medio de cables de gran longitud conectados a una fuente estática, pero, son excepciones que se dan bajo condiciones y requerimientos muy específicos.

Ningún sistema robótico móvil estaría completo sin la implementación de un subsistema de comunicación en su diseño. Esta comunicación puede ser por medio de un cable o inalámbrica, en tiempo real o fuera de línea, según los escenarios para los que se diseña.

#### **1.4. Paradigmas de operación de robots móviles**

Los robots móviles pueden desenvolverse bajo uno de los paradigmas cuyas características se describen a continuación.

##### **1.4.1. Robot semiautónomo**

Dispositivo robótico que requiere intervención humana para su operación, particularmente para el procesamiento de información y toma de decisiones. Estos robots, generalmente, se equipan con cámaras y otros sensores que envían información a un operador a distancia que hace uso de ellos para decidir y enviar señales correspondientes a acciones que debe realizar el dispositivo.

Bajo este paradigma, el poder computacional requerido se delimita, generalmente, en función de las necesidades de los subsistemas de locomoción, cámaras y sensores empleados por el dispositivo.

#### **1.4.2. Robot autónomo**

Dispositivo robótico que no requiere, o lo hace de forma insignificante, intervención humana. Este paradigma implica una demanda superior de recursos computacionales a la requerida por el descrito previamente.

Sus subsistemas suelen ser más complejos puesto que requieren de estrategias de percepción y control más avanzadas, ya que necesitan tomar datos, decisiones y acciones de forma prácticamente inmediata.

Las múltiples desventajas que se perciben inicialmente sobre este paradigma se encuentran mayormente en su fase de diseño y calibración. En la fase de implementación, son altamente efectivos y flexibles para realizar su labor.

#### **1.5. Modelos tradicionales de subsistemas de locomoción de robots móviles**

Los diseños tradicionales, tanto reales como teóricos, de subsistemas de locomoción de robots móviles se basan principalmente en dos modelos que emplean ruedas para propiciar desplazamiento.

### **1.5.1. Modelo de rueda frontal directora**

Utiliza una rueda cuyo ángulo de operación es dirigido a partir de un eje perpendicular a la superficie de contacto. Esta rueda cumple la función de dar dirección al desplazamiento del sistema, mientras un par adicional se encarga de proporcionar la fuerza motriz necesaria para el mismo. Este modelo presenta la desventaja de no permitir rotaciones de radio cero, es decir, rotaciones puras alrededor del eje imaginario que atraviesa el centro de masa del robot.

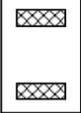
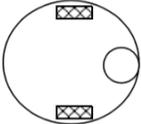
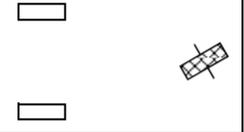
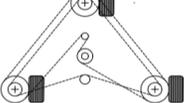
### **1.5.2. Modelo de control diferencial**

Se basa en el control independiente de cada una de las ruedas del subsistema de locomoción. Para avanzar o retroceder sin cambiar orientación, ambas ruedas giran a la misma velocidad y con la misma dirección; para avanzar o retroceder mientras se cambia la orientación, la rueda del lado hacia el que se desea que el dispositivo se gire funciona a menor velocidad que su pareja; para cambiar la orientación sin avanzar o retroceder, ambas ruedas funcionan a la misma velocidad, pero en direcciones opuestas.

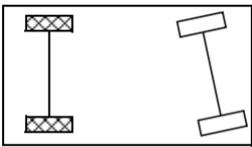
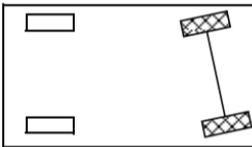
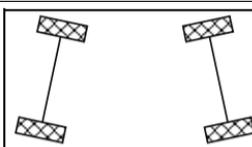
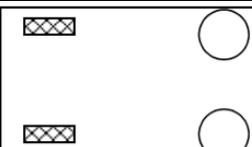
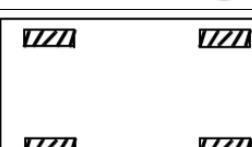
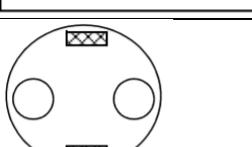
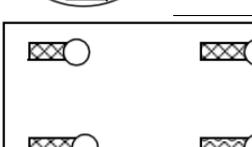
Esta última característica en particular, es la mayor ventaja que presenta este modelo con respecto al descrito previamente, la capacidad de girar sobre su propio eje.

Combinando de diferentes formas los modelos básicos descritos, se obtiene una amplia variedad de subsistemas de locomoción. Se detallan varios de ellos en la tabla I.

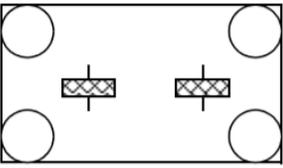
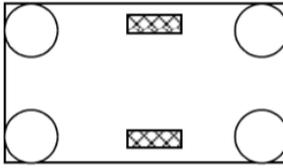
Tabla I. **Subsistemas de locomoción resultantes de las combinaciones de los modelos básicos de locomoción (traducida de inglés a español)**

Número de ruedas	Diagrama	Descripción
2		Una rueda directora al frente, una rueda de tracción trasera
		Configuración diferencial de dos ruedas con centro de masa bajo el eje
3		Configuración diferencial de dos ruedas con un tercer punto de contacto
		Dos ruedas independientes frontales o traseras con una rueda omnidireccional suelta del lado opuesto
		Dos ruedas de tracción conectadas (diferenciales) traseras con una rueda directora suelta frontal
		Dos ruedas traseras sueltas, una rueda directora de tracción frontal
		Tres ruedas suecas o esféricas motorizadas en configuración triangular
		Tres ruedas motorizadas directoras; operación no controlable

Continuación de la tabla I.

4		Dos ruedas motorizadas traseras, dos ruedas directoras frontales conectadas
		Dos ruedas motorizadas directoras frontales, dos ruedas sueltas traseras
		Cuatro ruedas directoras motorizadas
		Dos ruedas de tracción diferenciales frontales/traseras, dos ruedas omnidireccionales frontales/traseras
		Cuatro ruedas omnidireccionales
		Dos ruedas diferenciales con dos puntos de contacto adicionales
		Cuatro ruedas giratorias motorizadas directoras

Continuación de la tabla I.

6		Dos ruedas motorizadas y directoras alineadas en el centro, una rueda omnidireccional en cada esquina
		Dos ruedas diferenciales en el centro, una rueda omnidireccional en cada esquina
Íconos para cada tipo de rueda		
	Rueda omnidireccional suelta	
	Rueda sueca motorizada	
	Rueda estándar suelta	
	Rueda estándar motorizada	
	Rueda giratoria motorizada y directora	
	Rueda directora estándar	
	Ruedas conectadas	

Fuente: SIEGWART, Roland. *Introduction to autonomous mobile robots*. pp. 39-41.

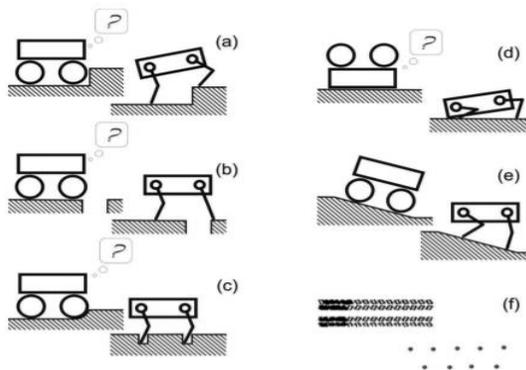
Los modelos descritos presentan ventajas durante la fase de diseño de controladores electrónicos de robots móviles, puesto que requieren pocos parámetros de control para conseguir desplazamiento. Sin embargo, a nivel mecánico, conllevan desventajas que aumentan en proporción a la complejidad del terreno o medio en el que se desenvuelven (desniveles, rugosidad, corrientes de agua, entre otros).

El área de contacto y acción del dispositivo con la superficie (en el caso de robots móviles terrestres), se concentra únicamente bajo las ruedas de estos modelos; si una o más de ellas pierden contacto con el terreno, el funcionamiento del subsistema de locomoción se ve notablemente afectado.

Esta desventaja puede verse disminuida considerablemente por el uso de arreglos de ruedas que garantizan puntos de contacto suplementarios en ausencia de uno o más de los principales, como en el caso de Curiosity, el robot de exploración de la NASA que se encuentra actualmente funcionando en el planeta Marte.

En los últimos años, se han propuesto modelos que abordan este problema a través de una distribución más amplia de puntos de contacto para el subsistema de locomoción, sustituyendo las ruedas por múltiples extremidades articuladas cuyo comportamiento se basa en el de animales con características similares, es decir, se aborda el problema desde un enfoque biomimético.

Figura 2. **Diagramas de ventajas de robots con extremidades sobre robots con ruedas en distintas situaciones de terrenos rugosos**



Fuente: SIEGWART, Roland. *Introduction to autonomous mobile robots*. p. 18.



## 2. GENERADORES DE PATRONES CENTRALES CPG

Los movimientos para desplazamiento en animales requieren múltiples patrones rítmicos distribuidos de forma que satisfagan múltiples requerimientos, como: la capacidad de generar movimiento hacia adelante, uso mínimo de energía, estabilidad o equilibrio, adaptación a terrenos complejos, entre otros.

Los Generadores de Patrones Centrales (CPG, por las siglas en inglés de *Central Pattern Generators*), son circuitos biológicos neuronales que funcionan como base de accionamiento rítmico, para los mecanismos de locomoción de organismos complejos en la naturaleza.

Son adaptaciones que permiten el control descentralizado de los procesos necesarios para producir desplazamiento, ya que forman un subsistema cuya topología no coloca al cerebro como pieza central, sino más bien como fuente de señales externas que estimulan y alteran parámetros de operación internos; esto lo hace de forma endógena, es decir, sin retroalimentación sensorial; se encuentran, entonces, embebidos en los canales de comunicación que tiene el cerebro con el resto de elementos mecánicos que hacen posible la movilidad de un organismo en su entorno: la médula espinal y el sistema nervioso periférico.

Investigadores han estudiado los CPG por décadas, principalmente desde un punto de vista biológico, lo que ha llevado a una mejor comprensión de su funcionamiento. Gracias a esto, en otra disciplina, la robótica, se ha propuesto el uso de CPG como mecanismo para generar estrategias de control eficientes para robots con extremidades basados en principios de movimiento biológicos.

Los movimientos de desplazamiento en sistemas con múltiples extremidades presentan desafíos de complejidad para estrategias de control tradicionales, siendo que implican la presencia de una gran cantidad de grados de libertad. Los CPG resuelven parcialmente este problema por medio de la generación de patrones rítmicos en lugar de posiciones y transiciones específicas.

Esto implica que son adaptables a distintas perturbaciones y escalables con mayor facilidad que otras estrategias de control tradicionales, ya que su funcionamiento no está basado en el conocimiento de la geometría y demás parámetros físicos del modelo del robot a controlar.

Otro beneficio de su uso, relacionado con esto último, es que incluso si una de las extremidades de un robot cuya estrategia de control de locomoción se basara en CPG se dañara o perdiera, no se requerirían configuraciones adicionales inmediatas, pues podría adaptarse para funcionar, de forma menos óptima, con las restantes.

Si bien los CPG presentan características que pueden ser consideradas como ventajosas para resolver problemas de ingeniería, cabe hacer mención también de sus principales desventajas, de las cuáles, algunas han sido ya parcialmente solucionadas.

La primera es la elevada cantidad de parámetros internos que deben configurarse para su generar señales rítmicas, que aumenta dependiendo de la cantidad de extremidades a controlar. Esta ha sido mitigada por medio de la simplificación de modelos de CPG, de la que se hablará más a fondo en el próximo capítulo, que reduce drásticamente la cantidad de parámetros manteniendo un comportamiento considerablemente cercano al biológico.

La última es que no existe una metodología general para la configuración óptima de los parámetros, aún en los modelos simplificados, para el diseño de estrategias de control. Este proceso se realiza, en la mayoría de casos, por prueba y error. Es precisamente esta una de las áreas de investigación de mayor interés para el uso de CPG en robótica; se utilizan, por ejemplo, estrategias para la optimización de parámetros basadas en algoritmos genéticos, esquemas de aprendizaje de máquinas (*Machine Learning*), y derivados.

No se entrará a detalles en este trabajo sobre estas metodologías puesto que se encuentran fuera del enfoque del mismo. Se hace la mención para establecer el contexto actual del uso y la investigación relacionada con CPG en robótica.

## **2.1. Características relevantes para ingeniería y robótica**

Los CPG se encuentran en una amplia variedad de organismos complejos, desde insectos y arácnidos hasta grandes mamíferos. Poseen propiedades que los hacen interesantes por sus potenciales aplicaciones en ingeniería, entre las que destacan:

- Requieren señales sencillas para modular su comportamiento
- Los mecanismos de activación e inhibición, en función de diferencias de fase, entre sus componentes u osciladores internos acoplados.
- Su capacidad para adaptarse a cambios externos de forma endógena. Un ejemplo de esto puede observarse en animales que, según las condiciones del terreno sobre el que se desplazan, caminan más lenta o rápidamente,

con pasos más largos o cortos; todo esto manteniendo secuencias rítmicas que rigen el movimiento de sus huesos y músculos de manera coordinada.

- Su requerimiento bajo o nulo de estímulos sensoriales para producir una secuencia rítmica que resulte en movimiento y desplazamiento.

Estas características han inspirado su uso en el diseño y desarrollo de entre otras cosas, sistemas robóticos móviles. Una implementación de alta fidelidad de CPG implicaría grandes costos computacionales; por este motivo, se han utilizado modelos computacionales simplificados cuyo comportamiento se asemeja notablemente al del modelo biológico.

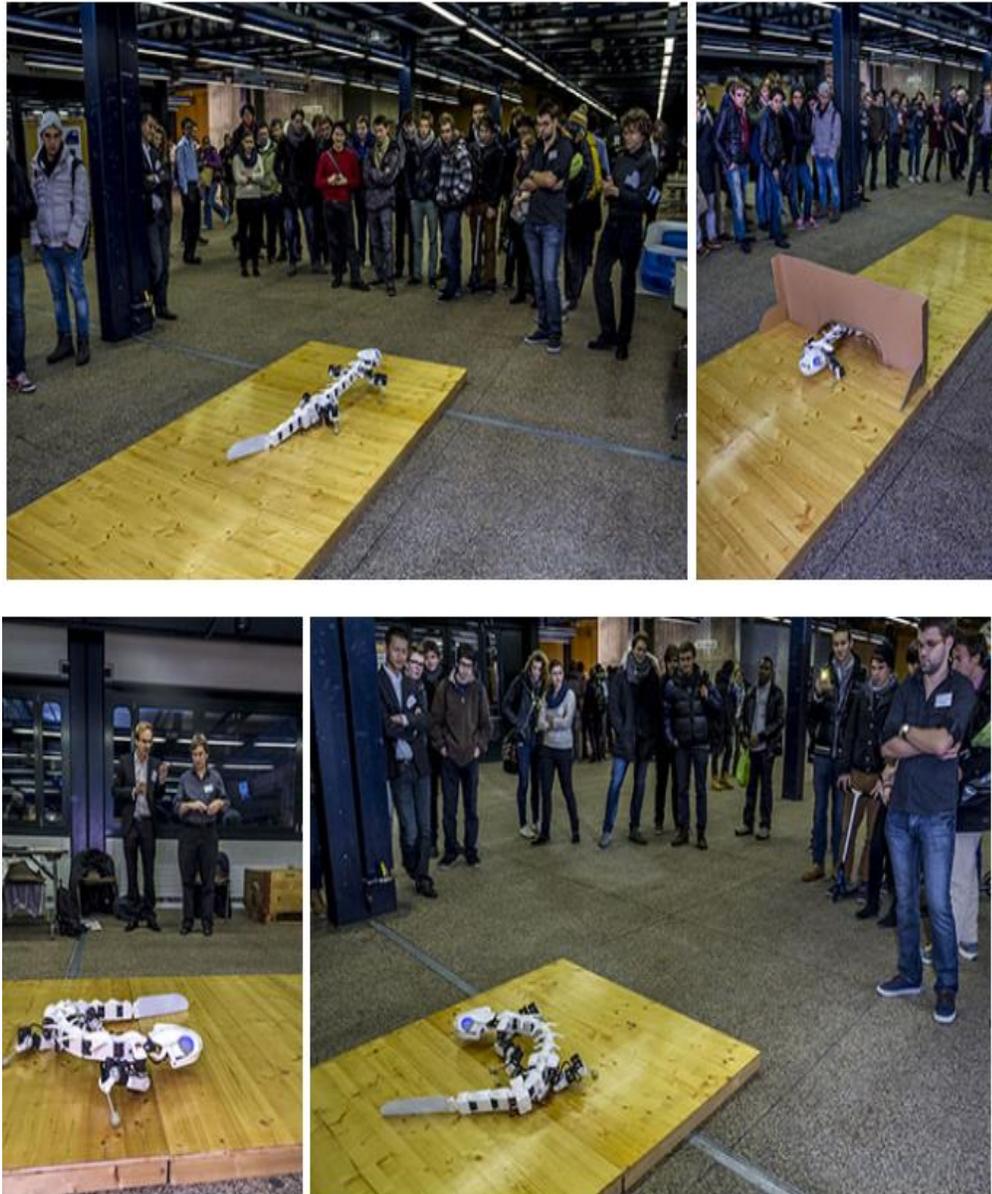
## **2.2. Antecedentes en robótica**

En la actualidad, existen múltiples proyectos de investigación y desarrollo de dispositivos robóticos con estrategias de control biomiméticas que implementan variantes de CPG para crear patrones de locomoción en sus actuadores. A continuación, se describen brevemente algunos de ellos.

### **2.2.1. Pleurobot**

Robot con forma de salamandra. Se desplaza por medio de cuatro extremidades y una columna vertebral articulada. Su controlador de locomoción posee múltiples CPG que se encargan de producir, a partir de la recepción de señales o estímulos externos, tres diferentes modalidades de desplazamiento: caminata sobre tierra, caminata sobre agua poco profunda y nado. Cuenta con 27 grados de libertad, lo que dificulta su análisis cinemático y la implementación de controladores tradicionales. Es desarrollado por un grupo de investigación de la Escuela Politécnica Federal de Lausana, Suiza.

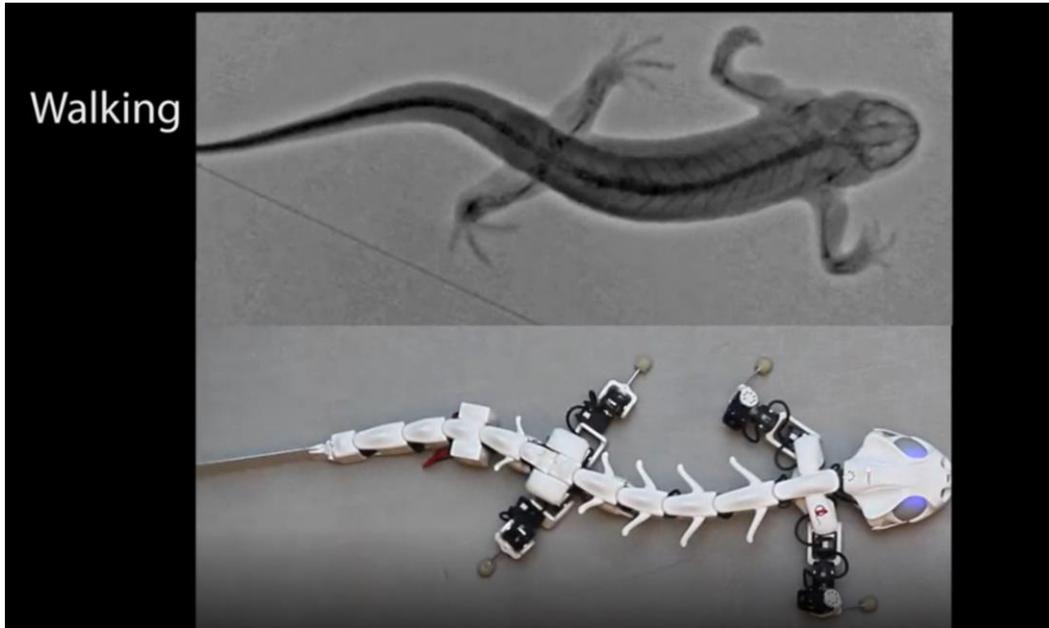
Figura 3.  **Demostración de operación de Pleurobot**



Fuente: EPFL. *Pleurobot*. <https://www.epfl.ch/labs/biorob/research/amphibious/pleurobot/>.

Consulta: 26 de junio de 2021.

Figura 4. **Comparación de movimiento de Pleurobot y una salamandra**



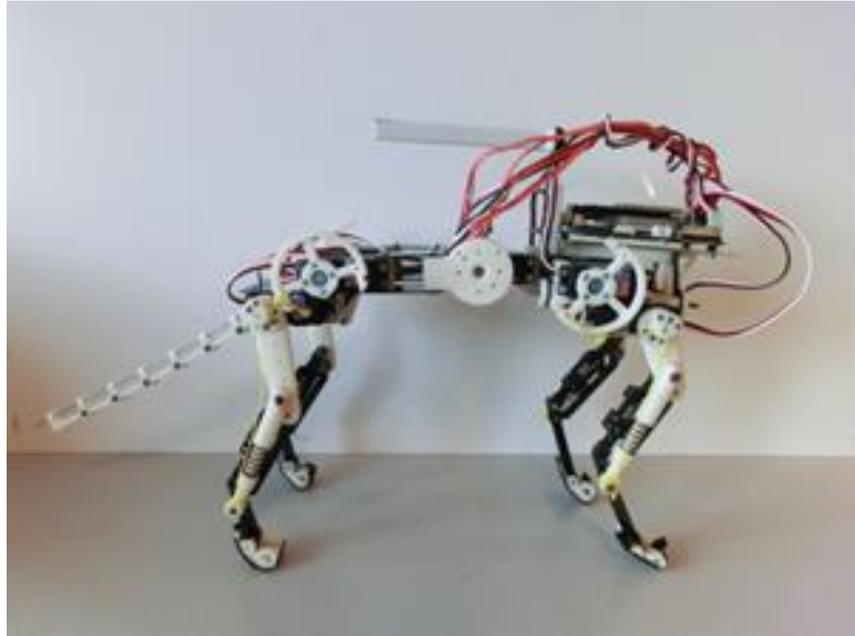
Fuente: EPFL. *Pleurobot walking comparison*.

<https://biorob2.epfl.ch/utills/movieplayer.php?id=288>. Consulta: 26 de junio de 2021.

### 2.2.2. Lynx

Robot con forma felina. Se desplaza por medio de cuatro extremidades y una columna vertebral articulada, además de una cola estabilizadora. Emplea CPG para producir patrones de movimiento que generan desplazamiento. Cuenta con 9 grados de libertad y es desarrollado por un grupo de investigación de la Escuela Politécnica Federal de Lausana, Suiza.

Figura 5. **Vista lateral de robot Lynx**

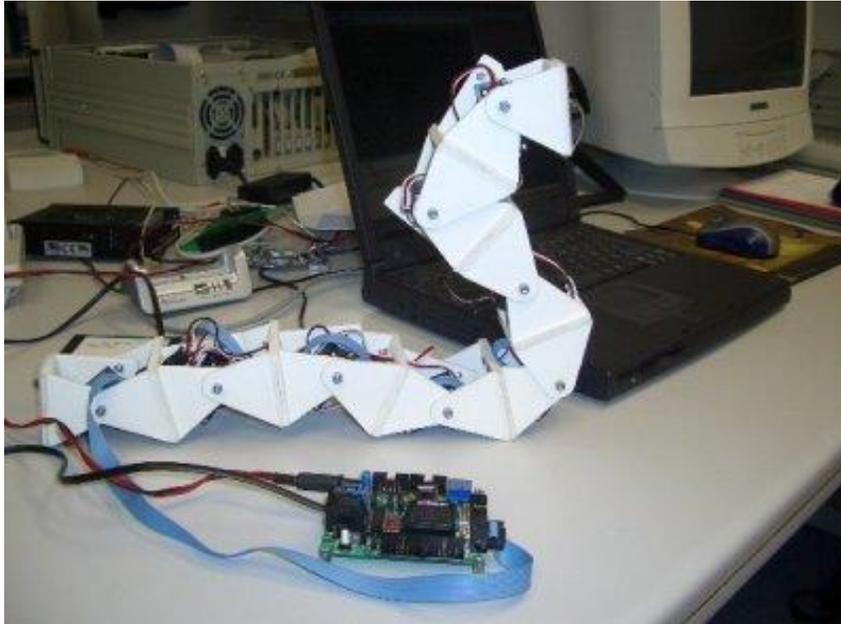


Fuente: EPFL. *Lynx*. <https://www.epfl.ch/labs/biorob/research/quadruped/quadruped-past/lynx-2/>. Consulta: 26 de junio de 2021.

### **2.2.3. Cube Revolutions**

Robot ápodo modular. Se desplaza por medio de una columna vertebral articulada. Su controlador de locomoción se basa en un CPG que produce patrones periódicos propagados a través de sus actuadores, desde la cola hasta la cabeza, lo que se traduce en desplazamientos hacia adelante, atrás y a los lados. Fue desarrollado en la Universidad Autónoma de Madrid, España.

Figura 6. **Vista lateral de robot Cube Revolutions**



Fuente: GONZÁLEZ GÓMEZ, Juan. *Cube revolutions*.

<http://www.learobotics.com/personal/juan/doctorado/cube-revolutions/>. Consulta: 26 de junio de 2021.

Todos los dispositivos descritos anteriormente tienen en común el uso de CPG para la generación de patrones de movimiento que generan desplazamiento y se han diseñados tomando inspiración de organismos biológicos complejos.

### **3. MODELO COMPUTACIONAL DE GENERADORES DE PATRONES CENTRALES**

#### **3.1. Simplificación de modelos**

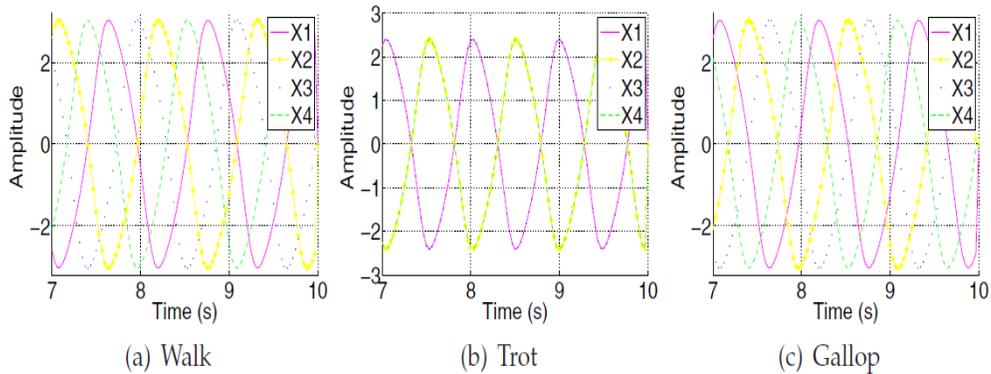
La capacidad de un robot para navegar por sí mismo es aún limitada. Múltiples experimentos e investigaciones se han desarrollado para determinar la factibilidad de diseñar mecanismos de locomoción para robots con extremidades tomando inspiración de los mecanismos de locomoción de humanos y otros animales.

El control de las extremidades se realiza de forma rítmica y sincronizada entre un gran número de grados de libertad para producir movimientos coordinados que generan patrones de desplazamiento.

Estos mecanismos son precisamente los Generadores de Patrones Centrales o CPG. Los CPG son producto de largos procesos de evolución biológica natural, y presentan un alto grado de complejidad. Se han propuesto modelos simplificados que toman ventaja de sus principales características y permiten su implementación a nivel computacional.

Estos modelos describen el comportamiento de sistemas con múltiples osciladores no lineales y la relación de acople entre cada uno de ellos, aproximándose así a la forma de operación de un CPG biológico.

Figura 7. **Gráficas de amplitud vs tiempo de modelo con cuatro CPG para simulación de patrones de desplazamiento. (a) Caminata, (b) Trote, (c) Galope**



Fuente: BARRON-ZAMBRANO, Jose. *CPG implementations for robot locomotion: analysis and design*. p. 177.

En el diseño de sistemas basados en CPG, se utilizan las siguientes convenciones para simplificar:

- Se asume que todos los osciladores no lineales del sistema son idénticos
- El movimiento de paso de cada extremidad es controlado por un solo oscilador mientras que la coordinación de movimiento de los componentes de las mismas es definida por las conexiones entre dichos osciladores.

Existen múltiples modelos de osciladores no lineales acoplados para modelar CPG en extremidades, como el modelo Amari-Hopfield (Amari, 1988), modelo Matsuoka (Billard y Ijspeert, 2000), modelo Van Der Pol (Van Der Pol B, 1928), y el modelo Kuramoto (Yoshiki Kuramoto, 1975).

Este último se ha seleccionado, por su simplicidad, para la fase de implementación de este trabajo. Se detalla en la siguiente sección de este capítulo y se utiliza como base para el capítulo 4.

### 3.2. El modelo de Kuramoto

Es un modelo matemático desarrollado por Yoshiki Kuramoto que Describe la relación de sincronización entre osciladores acoplados por fase. Toma como convención que todos los osciladores del sistema descrito son idénticos o muy similares. Define la siguiente expresión que representa al ángulo instantáneo de cada oscilador (motor, en el caso de la implementación) en un sistema con  $i$  cantidad de osciladores:

$$\theta_i = r_i \text{sen}(\varphi_i) + X_i \quad [\text{Ec. 1}]$$

Donde:

- $\theta_i$  = posición angular instantánea del  $i$ -ésimo oscilador
- $r_i$  = amplitud de movimiento del  $i$ -ésimo oscilador
- $\varphi_i$  = fase del  $i$ -ésimo oscilador
- $X_i$  = valor de compensación (offset) del  $i$ -ésimo oscilador

Para determinar el valor de  $\theta_i$  en un instante específico, se define un conjunto de ecuaciones diferenciales que describen las variaciones con respecto al tiempo de cada uno de los parámetros del modelo Kuramoto:

$$\frac{\partial \varphi_i}{\partial t} = 2\pi f_i + \sum_{j \neq i}^N w_{ij} r_i \text{sen}(\varphi_j - \varphi_i - \psi_i) \quad [\text{Ec. 2}]$$

$$\frac{\partial r_i}{\partial t} = a_i(R_i - r_i) \quad [\text{Ec. 3}]$$

$$\frac{\partial X_i}{\partial t} = b_i(X_i - x_i) \quad [\text{Ec. 4}]$$

Donde:

- $f_i$  = frecuencia de oscilación del i-ésimo oscilador
- $w_{ij}$  =  
coeficiente o peso de acople de fase entre el i-ésimo y el j-ésimo oscilador
- $\psi_i$  = valor de compensación u offset de fase del i-ésimo oscilador
- $a_i$  = coeficiente o peso del error de amplitud del i-ésimo oscilador
- $R_i$  = valor de amplitud deseado en el i-ésimo oscilador
- $r_i$  = valor de amplitud actual del i-ésimo oscilador
- $b_i$  =  
coeficiente o peso del error de compensación u offset del i-ésimo oscilador
- $X_i$  =  
es el valor de compensación u offset deseado en el i-ésimo oscilador
- $x_i$  = es el valor de compensación u offset actual del i-ésimo oscilador

Generalmente, los parámetros que se busca optimizar en el modelo son los pesos sinápticos o coeficientes de acople entre un CPG y cada uno de los otros con los que se encuentra dentro de la misma red.

Para obtener la solución del modelo de Kuramoto y determinar el ángulo de cada oscilador del sistema en un instante específico, es necesario integrar cada una de las ecuaciones diferenciales descritas anteriormente. Se recurre, para tal efecto, al método de integración numérica de Euler, ya que se hace necesario describir el modelo de forma computable para su implementación. Se muestra a

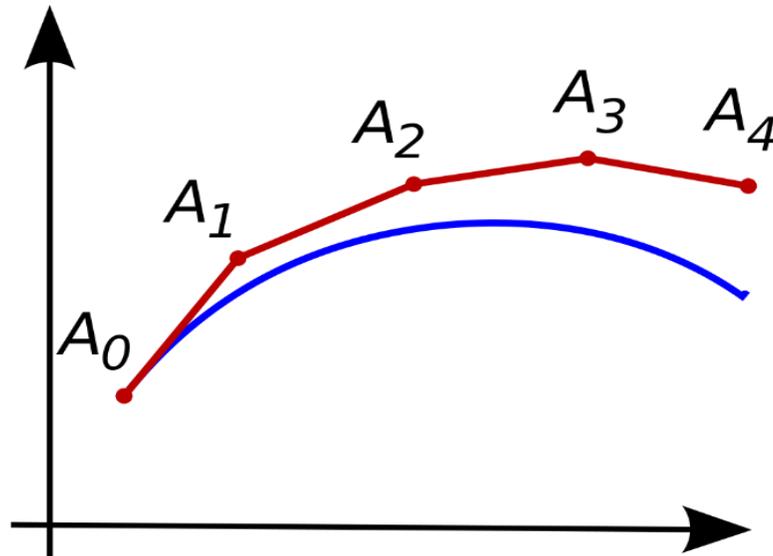
continuación la expresión matemática que describe el método numérico de valor inicial de orden 2 de Euler.

$$\varphi_t = \varphi_{t-1} + \frac{\partial \varphi_{t-1}}{\partial t} \Delta t \quad [\text{Ec. 5}]$$

Donde:

- $\varphi_t$  = valor en instante t
- $\Delta t$  = paso temporal

Figura 8. **Ilustración del método de Euler, la curva desconocida es azul y las aproximaciones poligonales se presentan en rojo, cada punto está separado horizontalmente por el valor del paso temporal**



Fuente: ZAPATA, Fanny. *Método de Euler: para qué sirve, procedimiento y ejercicios*.  
<https://www.lifeder.com/metodo-de-euler/>. Consulta: 29 de junio de 2021.

En el próximo capítulo, se detallarán las consideraciones de diseño para el desarrollo del controlador de locomoción basado en CPG. El mismo, como se mencionó anteriormente, estará basado en el modelo de Kuramoto para emular la generación de patrones que generan movimientos de desplazamiento en la naturaleza.

## **4. MÉTODO PARA EL DISEÑO DEL CONTROLADOR DE LOCOMOCIÓN**

Como en todo diseño de ingeniería, es necesario empezar por enlistar los requerimientos y parámetros bajo los que se regirá el producto final. Estos se eligen, para este trabajo en particular, teniendo en cuenta los objetivos establecidos en el mismo, tomando en consideración simplicidad en la implementación y disponibilidad de componentes y materiales.

En este caso, se trata de un controlador de locomoción basado en CPG con capacidad mínima para generar tres distintos patrones de desplazamiento en un robot móvil de una o más extremidades. Se detalla en las próximas secciones los pasos seguidos para diseñar el controlador.

### **4.1. Selección de características mecánicas del robot controlado**

Se decidió tomar este como el primer paso porque de él depende gran parte de la dificultad para la implementación del controlador, consumo energético y capacidad de procesamiento requerido.

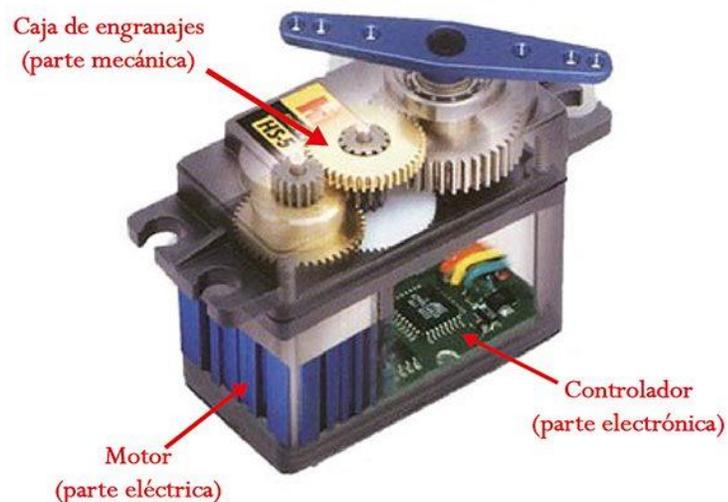
El factor determinante en este paso fue la cantidad de extremidades para la que se diseña el controlador. Se eligió trabajar con cuatro extremidades pues emplear más habría implicado mayor costo de materiales, componentes y energía; emplear menos habría implicado ahorros en los aspectos mencionados, pero también habría limitado la visibilidad y evaluación cualitativa de funcionamiento del controlador.

Se determinó que cada extremidad tendría dos articulaciones, incluyendo la que la uniría al cuerpo del robot. La articulación restante le daría la capacidad al robot de subir y bajar el segmento que tendría contacto directo con la superficie sobre la que se posara el robot.

#### 4.2. Selección de dispositivos para dar movimiento al robot

Se eligió utilizar servomotores, puesto que estos ofrecen control activo de posición angular por medio de un circuito de control PID interno y una caja reductora que le otorga la capacidad de mantener un torque considerable con bajo consumo energético.

Figura 9. Estructura interna de un servomotor



Fuente: GONZÁLEZ GARCÍA, Anthony. *¿Qué es y cómo funciona un servomotor?*  
<http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>. Consulta: 30 de junio de 2021.

Se definió que articulación de la extremidad llevaría un servomotor. El uso de servomotores implicó la inclusión de circuitos de interfaz de control que se detallarán en el próximo capítulo.

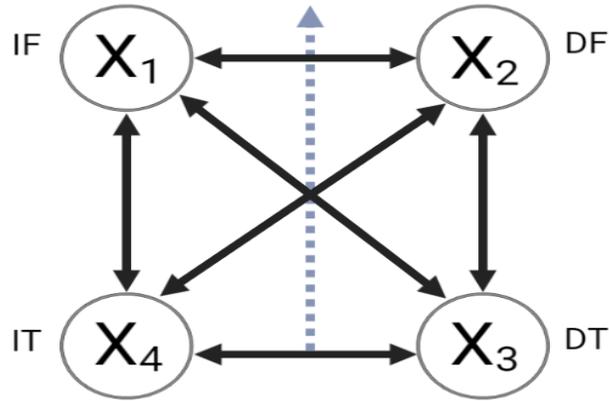
#### **4.3. Selección de la arquitectura del controlador**

Tomando como punto de partida que se controlaría un robot cuadrúpedo, se procedió a seleccionar la forma en que la red de osciladores acoplados que simularían el comportamiento de CPG en la naturaleza estaría establecida. Para tal efecto, se utilizaron como referencia los diagramas de interconexión propuestos en Barrón-Zambrano para determinar las relaciones de excitación e inhibición entre cada oscilador, dependiendo del patrón de desplazamiento presente.

Se observa en la figura 10 el diagrama general de la red de osciladores acoplados que emula de forma simplificada aun CPG. El subíndice de cada etiqueta  $X$  indica la numeración correspondiente a cada oscilador dentro del sistema, esta se asignó, por convención, en dirección horaria, este detalle es de importancia al momento de pretender escalar el controlador para utilizar más osciladores.

Las etiquetas IF, DF, IT y DT indican izquierda-frontal, derecha-frontal, izquierda-trasera y derecha-trasera respectivamente y denotan la posición de la extremidad correspondiente a cada oscilador en el sistema. La línea y flecha punteadas denotan la parte central y frontal del robot.

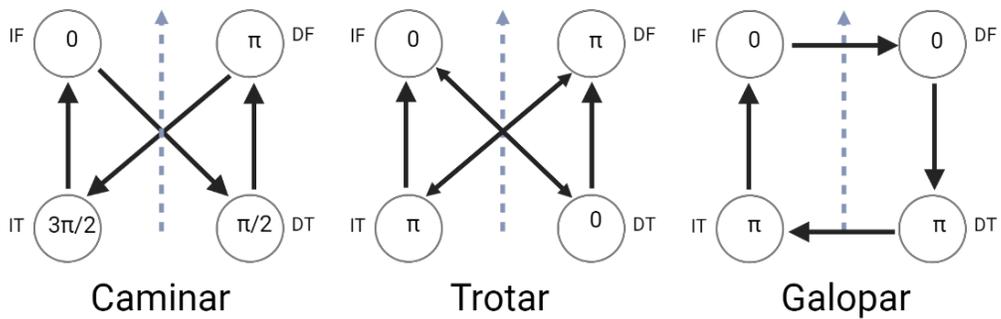
Figura 10. **Diagrama general de red de osciladores acoplados**



Fuente: elaboración propia, empleando Adobe Photoshop 2020.

Cada patrón de desplazamiento implica relaciones de fases diferentes entre los osciladores del sistema.

Figura 11. **Diagramas que representan las relaciones de fase de cada extremidad de un robot cuadrúpedo para distintos patrones de desplazamiento**



Fuente: elaboración propia, empleando Adobe Photoshop 2020.

En la figura 11, se muestran adaptaciones del diagrama general, mostrado en la figura inmediata anterior, correspondientes a las configuraciones necesarias para que la red de osciladores genere cada patrón de desplazamiento indicado.

Tabla II. **Patrones de movimiento típicos y sus respectivas relaciones de fase entre osciladores para un robot cuadrúpedo**

<i>Patrón</i>	Relaciones de fase
<b>Caminar</b>	$(0, \pi, \frac{\pi}{2}, \frac{3\pi}{2})$
<b>Trotar</b>	$(0, \pi, 0, \pi)$
<b>Galopar</b>	$(0, 0, \pi, \pi)$

Fuente: ZENG, Yinquan. *A bio-inspired control strategy for locomotion of a quadruped robot*. p.

7.

#### 4.4. Selección del modelo matemático para la implementación de CPG

Luego de definir la arquitectura del controlador, se procedió a elegir el modelo base para la implementación de CPG. El modelo matemático que se seleccionó, por simplicidad y escalabilidad, fue el modelo de sincronización de osciladores acoplados de Kuramoto. Se establecieron dos categorías para distinguir los parámetros requeridos para la operación del modelo. Se describen a continuación y se indica qué parámetros corresponden a cada una.

##### 4.4.1. Parámetros de definición de patrón

Estos fueron los parámetros necesarios para establecer las relaciones entre los osciladores conectados dentro de una red, ya sean de excitación o inhibición. Dentro de esta categoría se incluyó a todos los coeficientes de acople. Estos se

representan por medio de una matriz de acople  $W$  de dimensiones  $n \times n$ , siendo  $n$  la cantidad de osciladores del sistema, que a su vez representan la cantidad de extremidades controladas.

Figura 12. **Representación algebraica de la matriz de coeficientes de acople para un sistema de cuatro osciladores**

$$W = \begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} & W_{44} \end{pmatrix}$$

Fuente: elaboración propia.

Es notable, en la figura 12, que la diagonal principal corresponde al coeficiente de acople de cada oscilador consigo mismo, por lo que, al introducir valores numéricos a la matriz, los coeficientes sobre dicha diagonal siempre tendrán un valor de 1, como se muestra en la figura 13.

Figura 13. **Matriz de coeficientes de acople para un sistema de cuatro osciladores con valores numéricos para las relaciones de autoacople**

$$W = \begin{pmatrix} 1 & w_{12} & w_{13} & w_{14} \\ w_{21} & 1 & w_{23} & w_{24} \\ w_{31} & w_{32} & 1 & w_{34} \\ w_{41} & w_{42} & w_{43} & 1 \end{pmatrix}$$

Fuente: elaboración propia.

Era necesario determinar los valores restantes para la definición de los tres patrones de movimiento que el trabajo tuvo por objetivo, lo que suponía un extenso trabajo de prueba y error. Existen trabajos previos relacionados a la optimización de estos parámetros, para la generación de cada patrón de desplazamiento. La tabla III describe las matrices de acople de cada uno, de acuerdo al trabajo realizado por el autor citado como fuente de la misma.

Tabla III. **Matrices de acople de diferentes patrones de desplazamiento**

Patrón	Caminata	Trote	Galope
<b>Matriz de acople</b>	$\begin{pmatrix} 1 & -0.2 & -0.2 & -0.2 \\ -0.2 & 1 & -0.2 & -0.2 \\ -0.2 & -0.2 & 1 & -0.2 \\ -0.2 & -0.2 & -0.2 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -0.2 & 0.2 & -0.2 \\ -0.2 & 1 & -0.2 & 0.2 \\ 0.2 & -0.2 & 1 & -0.2 \\ -0.2 & 0.2 & -0.2 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.2 & -0.2 & -0.2 \\ -0.2 & 1 & 0.2 & -0.2 \\ -0.2 & -0.2 & 1 & 0.2 \\ 0.2 & -0.2 & -0.2 & 1 \end{pmatrix}$

Fuente: BARRON-ZAMBRANO, José. *Hardware implementation of a CPG-based locomotion control for quadruped robots*. p. 284.

#### 4.4.2. Parámetros de ajuste de patrón

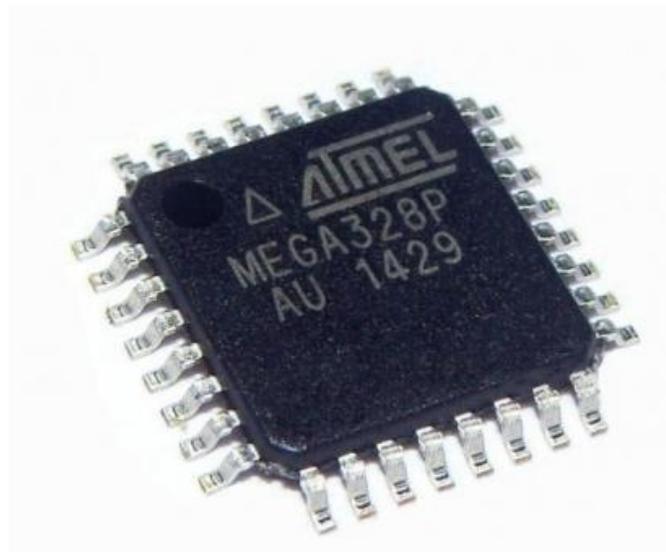
Mientras que los parámetros de definición de patrón deben definirse inicialmente y mantenerse constantes para generar consistentemente los patrones deseados, los parámetros de ajustes de patrón pueden ser variables y no afectan fundamentalmente cada patrón, sino más bien modifican la velocidad y el alcance de cada uno. Se categorizó en ellos a la fase, amplitud y valor de compensación de cada oscilador del sistema.

#### 4.5. Selección de procesador

Este fue un paso primordial para la implementación. De él dependieron parte de los costos, la facilidad para implementar el modelo elegido y las herramientas de programación empleadas.

Basándose en conocimientos previos, disponibilidad, dimensiones y bajo costo de implementación, se seleccionó un microcontrolador, particularmente el ATmega328P, fabricado por la empresa estadounidense Atmel y programado por medio de la popular plataforma de diseño de prototipos Arduino, versión Nano. Las especificaciones técnicas del microcontrolador y la plataforma de diseño empleada se detallarán en el próximo capítulo.

Figura 14. **Chip ATmega328P con encapsulado TQFP**



Fuente: Smart Prototyping. *ATMEGA328P-AU TQFP*. <https://www.smart-prototyping.com/ATMEGA328P-AU-TQFP>. Consulta: 1 de julio de 2021.

## **5. IMPLEMENTACIÓN DE CONTROLADOR**

Se describen en este capítulo los detalles técnicos de la implementación del controlador cuyo diseño se detalló en el capítulo anterior.

### **5.1. Hardware**

En esta sección se dan detalles sobre las especificaciones técnicas de los componentes tangibles del controlador y el robot al que se aplica.

#### **5.1.1. Estructura del robot**

Se eligió una estructura mecánica previamente diseñada y cuyos archivos de diseño se encontraban disponibles públicamente, "Hexapod Robot", del sitio web <https://www.thingiverse.com/>, esta es una plataforma que ofrece archivos de diseño listos para imprimir utilizando tecnologías de impresión 3D ampliamente disponibles en la actualidad. Se eligió este diseño en particular por su relativa simpleza (factor influyente en el tiempo y costo de impresión), y por contar con adaptaciones para hasta seis extremidades con tres servomotores cada una.

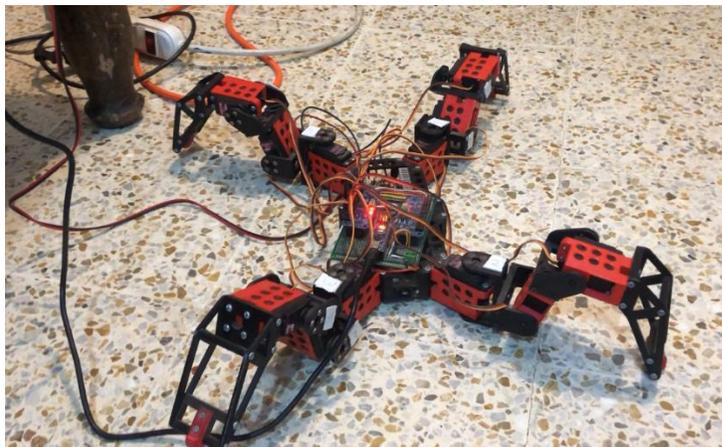
En este trabajo se emplean únicamente cuatro de las seis extremidades posibles, dejando las puertas abiertas para futuras iteraciones con más extremidades.

Figura 15. **Vista simulada del diseño de “Hexapod Robot”**



Fuente: MARKWTECH. *Hexapod robot*. <https://www.thingiverse.com/thing:3463845>. Consulta: 15 de julio de 2021.

Figura 16. **“Hexapod Robot” impreso y adaptado con cuatro extremidades**



Fuente: elaboración propia.

### **5.1.2. Circuito eléctrico del controlador**

Los componentes principales del circuito del controlador fueron el microcontrolador ATmega328P, integrado a la plataforma de desarrollo Arduino Nano, y los servomotores que darían movimiento al robot a partir de las señales generadas por la red de osciladores acoplados.

Fue necesario un circuito intermedio que permitiera separar el sistema en una fase lógica, y una de potencia; para esto se utilizó un circuito especializado de control de servomotores basado en el circuito integrado PCA9685PW.

También se utilizó un regulador de voltaje como intermediario entre la fuente de alimentación principal y todos los sub circuitos posteriores.

En los siguientes incisos se detallan las especificaciones técnicas y características útiles de los componentes utilizados para la construcción del circuito de implementación del controlador.

#### **5.1.2.1. Arduino Nano**

En el capítulo anterior se mencionaron las características por las que fue seleccionado el microcontrolador ATmega328P integrado a la plataforma de desarrollo Arduino Nano. En la tabla IV se muestran sus especificaciones técnicas.

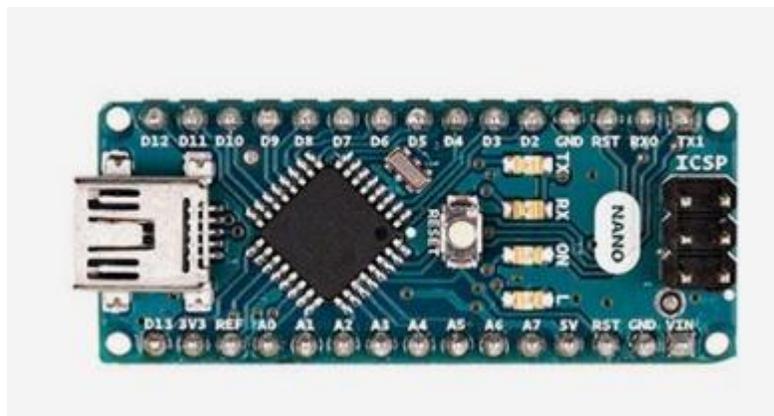
Tabla IV. **Especificaciones técnicas de Arduino Nano**

Concepto	Descripción
Microcontrolador	ATmega328P
Arquitectura	AVR
Tensión de funcionamiento	5 V
Memoria flash	32 KB de los cuales 2 KB son utilizados por el gestor de arranque
SRAM	2 KB
Velocidad de reloj	16 MHz
Pines analógicos IN	8
EEPROM	1 KB
Corriente CC por pines de E / S	40 mA (pines de E / S)
Voltaje de entrada	7-12 V
Pines de E / S digitales	22 (6 de los cuales son PWM)
Salida PWM	6
Consumo de energía	19 mA
Tamaño de PCB	18 x 45 mm
Peso	7 g

Fuente: Arduino. *Arduino Nano*. <https://store.arduino.cc/arduino-nano>. Consulta: 3 de julio de 2021.

Cuenta, además, con un módulo de comunicación serial I2C que fue de utilidad para hacer la interfaz entre el microcontrolador y los servomotores.

Figura 17. **Arduino Nano**



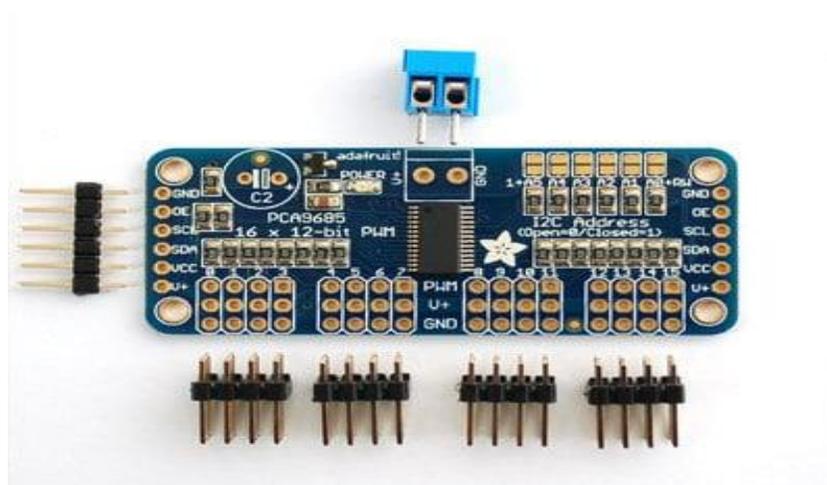
Fuente: Arduino. *Arduino Nano*. <https://store.arduino.cc/arduino-nano>. Consulta: 3 de julio de 2021.

### 5.1.2.2. Adafruit PCA9685 16-Channel Servo Driver

Se utilizó el controlador de servomotores Adafruit PCA9685 16-Channel Servo Driver (controlador de servomotores de 16 canales) por las capacidades que ofrece para controlar múltiples dispositivos utilizando solamente un bus de comunicación serial I2C, que implica únicamente el uso de dos pines del Arduino Nano.

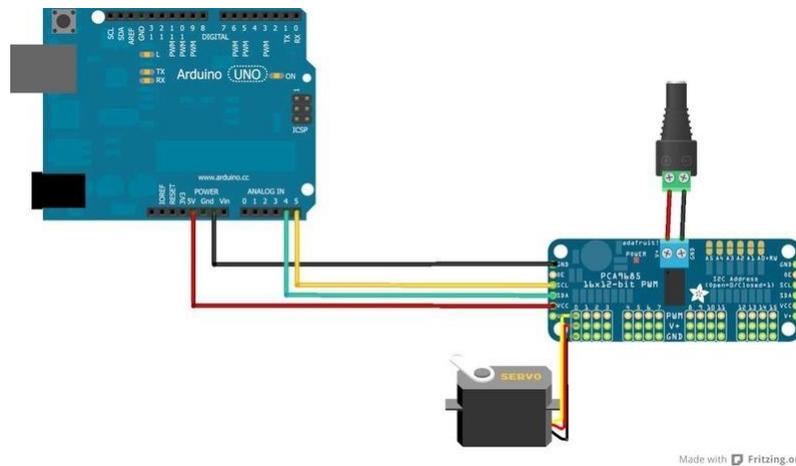
Este controlador puede manejar hasta 16 motores, lo que facilitaría escalar la cantidad de osciladores interconectados dentro del sistema, en caso se deseara agregar más extremidades al mismo. Ofrece también la capacidad para conectarse en cascada con otros módulos del mismo tipo, haciendo posible controlar hasta 992 servomotores, si se compensan las características eléctricas del sistema de forma proporcional. Cada uno de sus canales cuenta con un módulo PWM de 12 bits, lo que hace más suave la transición entre cada posición del servomotor. Opera en un rango de voltaje entre 2.3 V y 5.5 V.

Figura 18. **Adafruit PCA9685 16-Channel Servo Driver**



Fuente: Adafruit. *Hooking it up*. <https://learn.adafruit.com/16-channel-pwm-servo-driver/hooking-it-up>. Consulta: 3 de julio de 2021.

Figura 19. **Diagrama de conexión entre Arduino UNO, Adafruit PCA9685 16-Channel Servo Driver y un servomotor**



Fuente: Adafruit. *Hooking it up*. <https://learn.adafruit.com/16-channel-pwm-servo-driver/hooking-it-up>. Consulta: 3 de julio de 2021.

### 5.1.2.3. Servomotores

Como se mencionó anteriormente, fueron utilizados servomotores para dar al robot la capacidad de mover sus articulaciones. Específicamente, fueron utilizados servomotores MG996R, este modelo es estándar y múltiples fabricantes lo producen con especificaciones similares, descritas en la tabla V. Los utilizados en este trabajo fueron fabricados por Longrunner, número de parte LKY62.

Tabla V. **Especificaciones técnicas de servomotor MG996R**

<b>Concepto</b>	<b>Descripción</b>
Ciclo de trabajo de operación	1 – 2 ms a 50 Hz
Rango de movimiento angular	0 – 180 °
Voltaje de operación	4.8 - 7.2 V
Torque detenido	9.4 - 11 kgf-cm
Velocidad	0.17 – 0.14 s / 60 °

Fuente: Electrónicos Caldas. *MG996R high torque metal gear dual ball bearing servo.*  
[https://www.electronicoscaldas.com/datasheet/MG996R\\_Tower-Pro.pdf](https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf). Consulta: 15 de julio de 2021.

Figura 20. **Imagen de servomotor MG996R**



Fuente: Amazon. *Longrunner 10 x MG996R metal gear torque digital servo motor para rc modelo coche barco helicóptero lky62.* <https://www.amazon.com/-/es/Longrunner-MG996R-Torque-Digital-Helic%C3%B3ptero/dp/B01MTW06IU/>. Consulta: 15 de julio de 2021.

#### 5.1.2.4. Fuente de alimentación

Para facilitar escalabilidad en futuras pruebas y por disponibilidad inmediata, se utilizó una fuente de voltaje DC sobredimensionada. Este sobredimensionamiento garantiza que, en el futuro, podrían ser conectados más motores correspondientes a extremidades adicionales sin que las capacidades eléctricas del sistema sean excedidas. La tabla VI muestra las especificaciones técnicas de la fuente de alimentación marca Eyeboot utilizada.

Tabla VI. **Especificaciones técnicas de fuente Eyeboot DC Universal Regulated Switching Power Supply AC to DC**

<b>Concepto</b>	<b>Descripción</b>
Rango de voltaje de entrada	100 – 120 VAC
Voltaje de salida	36 VDC
Corriente de salida	13.89 A
Potencia nominal	500 W
Características adicionales	Protección contra cortocircuitos y sobrecargas.

Fuente: Eyeboot. *36V 500W DC universal regulated switching power supply ac to dc 13.8 amps.*

<https://www.eyeboot.com/36v-500w-dc-power-supply.html>. Consulta: 5 de julio de 2021.

Figura 21. **Imagen de Eyeboot DC Universal Regulated Switching Power Supply AC to DC**



Fuente: Eyeboot. *36V 500W DC universal regulated switching power supply ac to dc 13.8 amps.*  
<https://www.eyebboot.com/36v-500w-dc-power-supply.html>. Consulta: 5 de julio de 2021.

#### **5.1.2.5. Regulador de voltaje**

Se eligió un regulador de voltaje tomando como base la fuente de alimentación descrita anteriormente. Este componente garantizó la alimentación energética del sistema con los valores de voltaje y la corriente requeridos por todos sus circuitos. Al igual que la fuente, fue sobredimensionado para dejar lugar a futuras pruebas. La tabla VII muestra las especificaciones técnicas del regulador.

Tabla VII. **Especificaciones técnicas del regulador de voltaje utilizado**

<b>Concepto</b>	<b>Descripción</b>
Rango de voltaje de entrada	6 – 40 VDC
Rango de voltaje de salida	1.2 - 36 VDC
Corriente de salida	20 A
Eficiencia de conversión	95 %

Fuente: La Electrónica. *Módulo regulador de voltaje con límite de corriente CC CV - 20A.*  
<https://laelectronica.com.gt/modulo-regulador-de-voltaje-con-limite-de-corriente-cc-cv---20a>.

Consulta: 5 de julio de 2021.

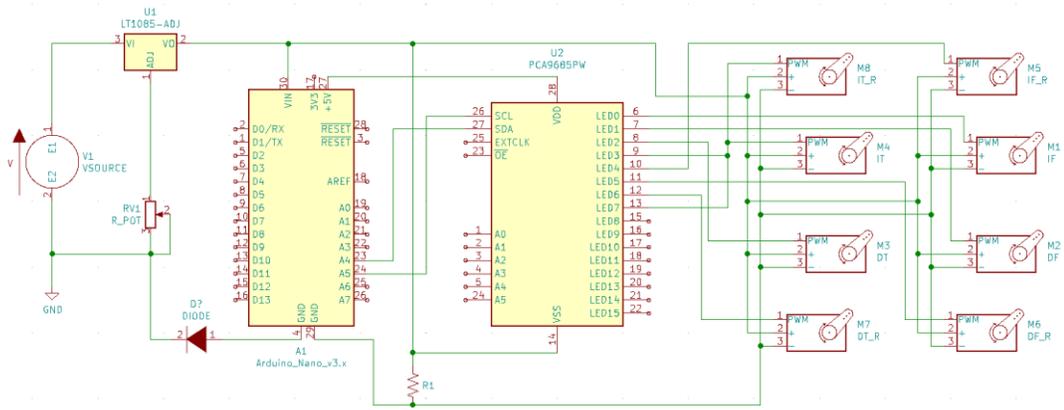
Figura 22. **Imagen del regulador de voltaje utilizado**



Fuente: La Electrónica. *Módulo regulador de voltaje con límite de corriente CC CV - 20A.*  
<https://laelectronica.com.gt/modulo-regulador-de-voltaje-con-limite-de-corriente-cc-cv---20a>.

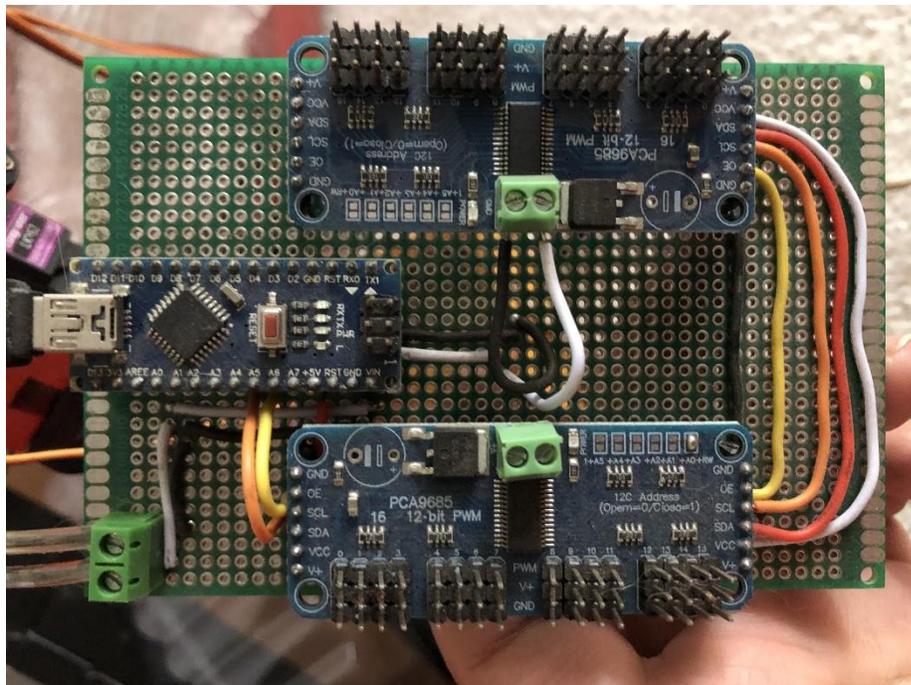
Consulta: 5 de julio de 2021.

Figura 23. Diagrama de conexión del circuito de control del sistema



Fuente: elaboración propia, empleando KiCAD 5.1.5

Figura 24. Placa de prueba montada



Fuente: elaboración propia.

Empleando como referencia el circuito de control del sistema mostrado en la figura 22, se fabricó la placa de prueba mostrada en la figura 23. Puede apreciarse en esta última que fueron instalados dos módulos Adafruit PCA9685 16-Channel Servo Driver, esto con el fin de facilitar la escalabilidad para futuras pruebas.

## **5.2. Programación de firmware**

Una vez definido y especificado el *hardware*, se procedió a recopilar los elementos necesarios para programar el firmware del microcontrolador utilizado. Estos elementos se describen en los próximos incisos. El repositorio del código utilizado para la implementación puede ser encontrado en la página de GitHub del autor de este trabajo.

### **5.2.1. Ambiente Integrado de Desarrollo (IDE)**

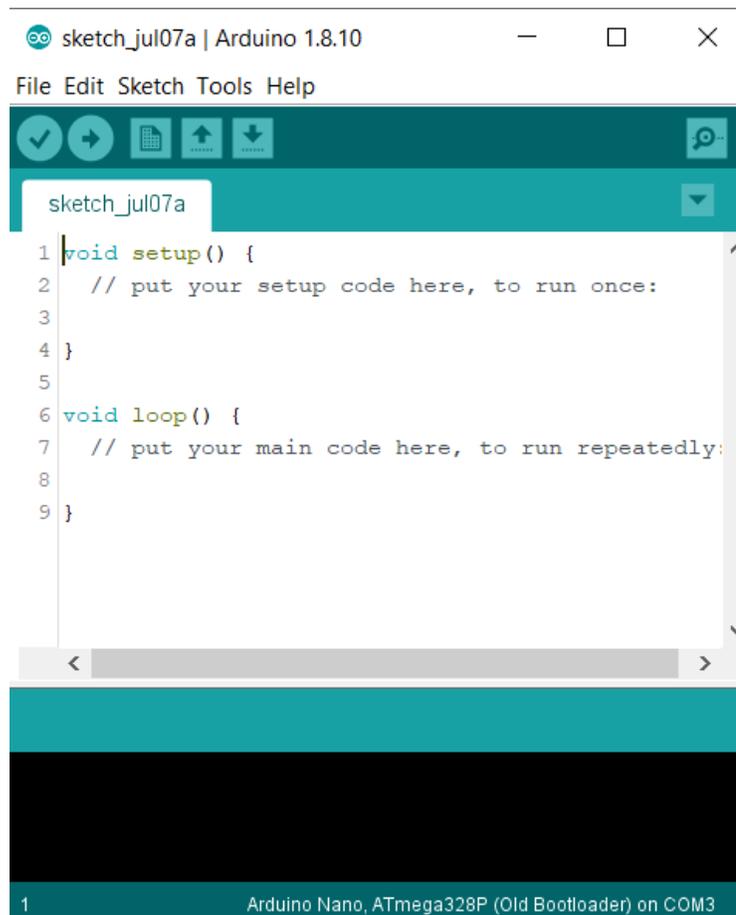
El *Integrated Development Environment* (IDE), o Ambiente Integrado de Desarrollo es un conjunto de herramientas que facilitan la organización de archivos de código de un proyecto. En el caso de este trabajo, se utilizó el IDE Arduino 1.8.10 para el sistema operativo Windows 10.

Arduino 1.8.10 es una herramienta de *software* gratuita y abierta utilizada para programar múltiples modelos de microcontroladores, incluyendo al ATmega328P especificado anteriormente. Provee las herramientas necesarias para escribir, organizar, corregir, monitorear, compilar y cargar código.

Al instalarse, viene por defecto con múltiples librerías de uso general ya integradas. Esto permite acortar de forma significativa el tiempo de desarrollo de prototipos, y en este trabajo, fueron necesarias librerías adicionales para permitir

la interacción entre los distintos componentes del sistema. Estas se describen en el próximo inciso.

Figura 25. **Vista inicial del IDE de Arduino 1.8.10**



Fuente: elaboración propia, empleando imagen de la interfaz.

## **5.2.2. Librerías**

Las librerías son archivos de código que contiene funciones específicas que extienden las capacidades de interacción de componentes al programar. Basta con invocar o incluir una librería en el código que se escribe para tener acceso a sus funciones adicionales. En este trabajo se emplearon dos librerías específicas, una de ellas previamente cargada en el IDE de Arduino 1.8.10 y la otra descargada desde el sitio oficial del fabricante del controlador de servomotores.

### **5.2.2.1. Arduino Wire.h**

Esta librería permite al microcontrolador interactuar con circuitos periféricos por medio del protocolo de comunicación serial I2C. Configura por defecto dos pines de este para funcionar SDA (línea de datos) y SCL (línea de reloj). Fue utilizada para facilitar la comunicación entre el microcontrolador y el controlador de servomotores empleado. La referencia para su uso puede ser encontrada en el sitio web oficial de Arduino.

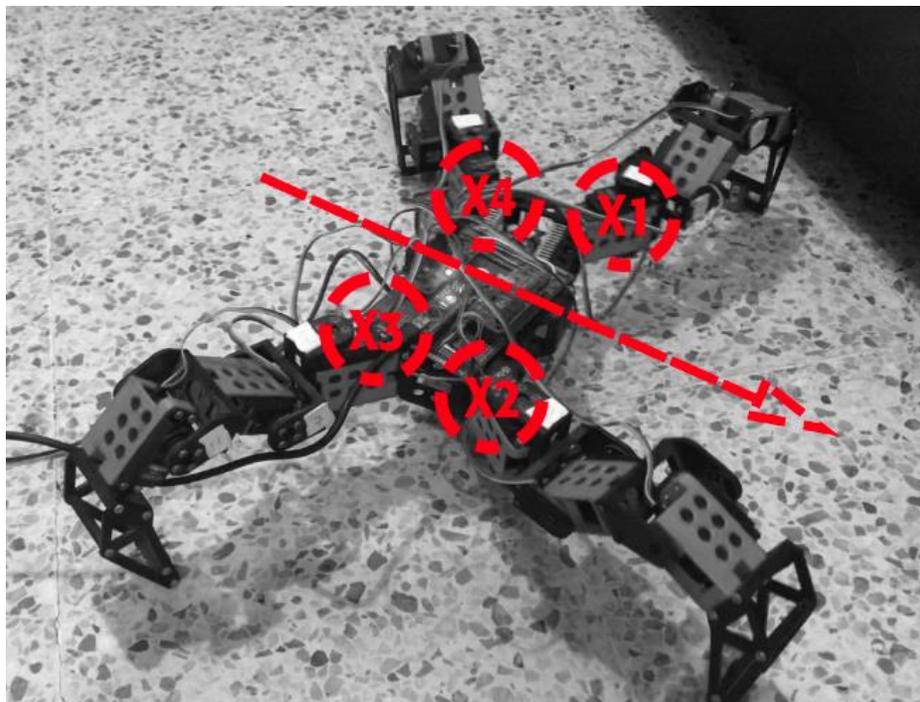
### **5.2.2.2. Adafruit-PWM-Servo-Driver-Library**

Esta librería permite al microcontrolador enviar señales de control de posición a cada uno de los servomotores conectados al módulo Adafruit PCA9685 16-Channel Servo Driver. Referencias para su uso pueden encontrarse en la página de GitHub de Adafruit.

### 5.3. Puesta en operación

Para poner en funcionamiento el código adaptado, se cargó el *firmware* al microcontrolador y se conectaron los servomotores a la placa de pruebas. Los servomotores que unen al cuerpo principal del robot con cada extremidad (uniones de cadera) fueron ordenados como se muestra en la figura 26, tomando como referencia el esquema mostrado en la figura 10.

Figura 26. **Esquema de implementación de osciladores sobre el robot móvil**



Fuente: elaboración propia, empleando Adobe Photoshop 2020.

Los servomotores restantes de cada articulación (uniones de rodilla y pie), fueron configurados para operar en desfase con respecto a las señales generadas por cada oscilador del CPG.

El método principal del programa cargado fue adaptado de García-Saura para funcionar con cuatro osciladores acoplados. Fue necesaria la definición de un método de posicionamiento inicial para las articulaciones, llamado home () en el código, para acelerar la convergencia del sistema y generar patrones de desplazamiento apreciables.

Se agregaron, las matrices de acople para los tres patrones de desplazamiento deseados, siendo posible cargar una a la vez mientras las otras permanecen en el código como comentarios. Fue necesario escalar los valores de las matrices mostradas en la tabla III, exceptuando los correspondientes a autoacoples, para obtener valores significativos como señal de salida de cada oscilador, las relaciones de fase permanecen invariantes. Se aprecia este cambio en la tabla VIII.

Tabla VIII. **Matrices de acople con valores escalados**

<i>Patrón</i>	Caminata	Trote	Galope
<b>Matriz de acople</b>	$\begin{pmatrix} 1 & -5 & -5 & -5 \\ -5 & 1 & -5 & -5 \\ -5 & -5 & 1 & -5 \\ -5 & -5 & -5 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -5 & 5 & -5 \\ -5 & 1 & -5 & 5 \\ 5 & -5 & 1 & -5 \\ -5 & 5 & -5 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 5 & -5 & -5 \\ -5 & 1 & 5 & -5 \\ -5 & -5 & 1 & 5 \\ 5 & -5 & -5 & 1 \end{pmatrix}$

Fuente: elaboración propia.

Se describe en las próximas subsecciones el funcionamiento de cada uno de los patrones definidos. Se utilizan gráficas de datos tomados durante las pruebas de puesta en operación.

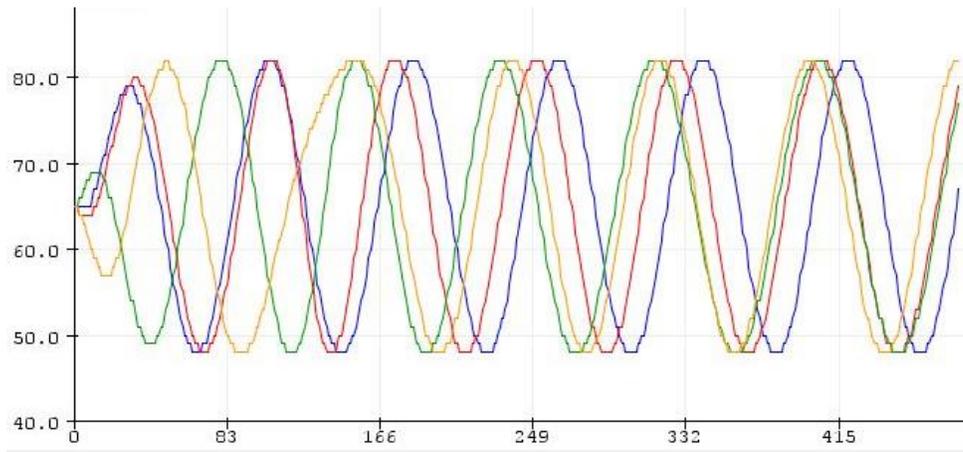
Para interpretar correctamente estas gráficas, deben tomarse en cuenta estas anotaciones:

- El eje horizontal de cada gráfica representa pasos temporales correspondientes al tiempo de muestreo utilizado para la toma de datos.
- El eje vertical de cada gráfica representa el valor del ángulo instantáneo producido por los osciladores.
- Cada oscilador del sistema es representado por una curva de diferente color, correspondiendo azul, rojo, verde y amarillo a  $X_1$ ,  $X_2$ ,  $X_3$  y  $X_4$  respectivamente (ver figura 10 para referencia).
- Se muestran tres gráficas por cada patrón, correspondientes a tres plazos: corto, mediano y largo. Esto para mostrar el comportamiento del sistema de osciladores interconectados en diferentes etapas temporales.

### **5.3.1. Funcionamiento de patrón de caminata**

Las figuras 27, 28 y 29 muestra las señales generadas por los osciladores utilizando la matriz de acople que genera el patrón de caminata (ver tabla VIII).

Figura 27. **Inicio de operación de CPG con matriz de patrón de caminata**

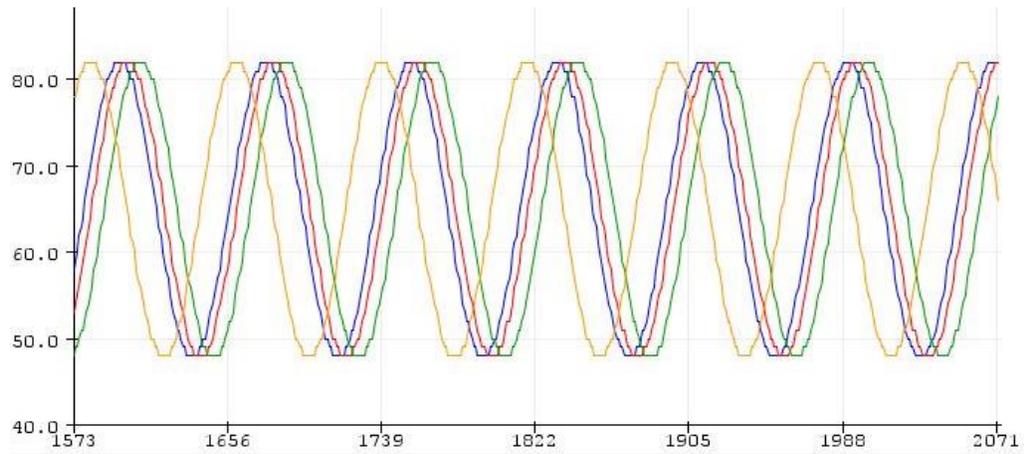


Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

En la figura 27 se observa que cada señal parte de un valor inicial de  $65^\circ$ , correspondiente a la posición angular inicial de todas las articulaciones de cadera al arrancar la operación.

Este valor empieza a variar en cada oscilador, permitiendo observar desfases entre cada uno. Los desfases en las gráficas se traducen en desfases en el movimiento de los servomotores con los que el robot móvil está equipado.

Figura 28. **Comportamiento a mediano plazo de CPG con matriz de patrón de caminata**

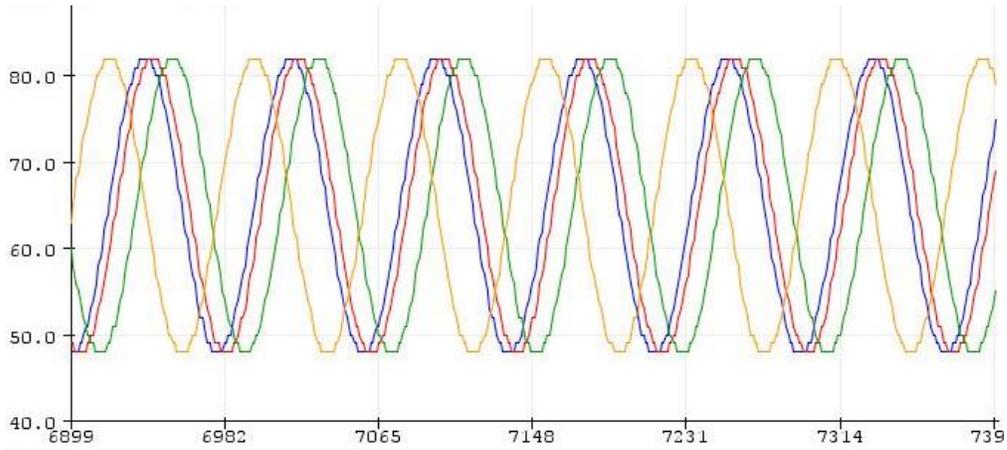


Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

Puede notarse que la generación de un patrón estable no es instantánea, más bien toma tiempo en converger al comportamiento deseado.

Esto se aprecia al comparar la figura 27 con las figuras 28 y 29, en las que las señales parecen comportarse de forma idéntica.

Figura 29. **Comportamiento a largo plazo de CPG con matriz de patrón de caminata**



Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

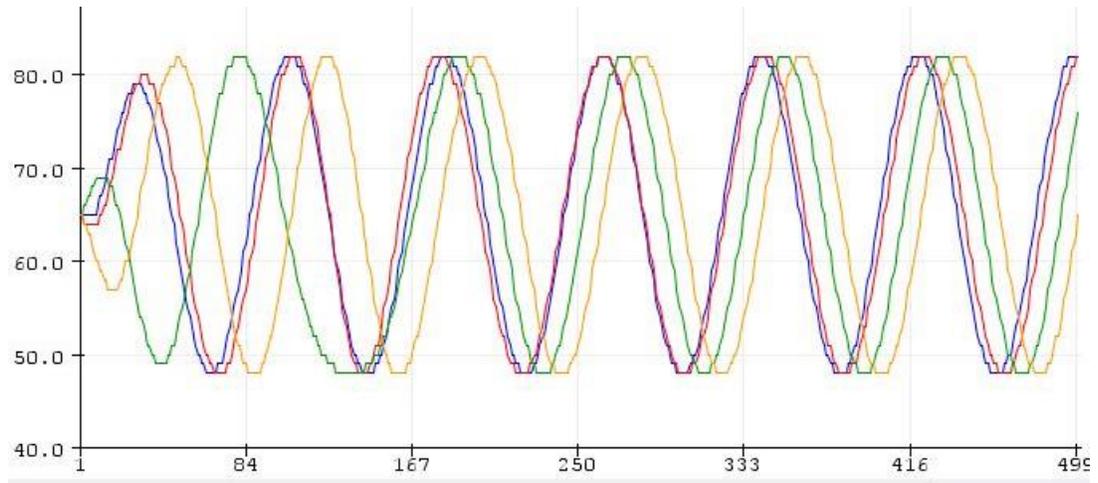
Se observa en el patrón estable (figura 29) que  $X_1$ ,  $X_2$  y  $X_3$  presentan desfases ligeros, mientras  $X_4$  presenta un desfase considerable, que corresponde al último paso en un ciclo del patrón de caminata.

### 5.3.2. **Funcionamiento de patrón de trote**

Las figuras 30, 31 y 32 muestra las señales generadas por los osciladores utilizando la matriz de acople que genera el patrón de trote (ver tabla VIII). En la figura 30 se observa que cada señal parte de un valor inicial de  $65^\circ$ , correspondiente a la posición angular inicial de todas las articulaciones de cadera al arrancar la operación.

Este valor empieza a variar en cada oscilador, permitiendo observar desfases entre cada uno. Los desfases en las gráficas se traducen en desfases en el movimiento de los servomotores con los que el robot móvil está equipado.

Figura 30. **Inicio de operación de CPG con matriz de patrón de trote**

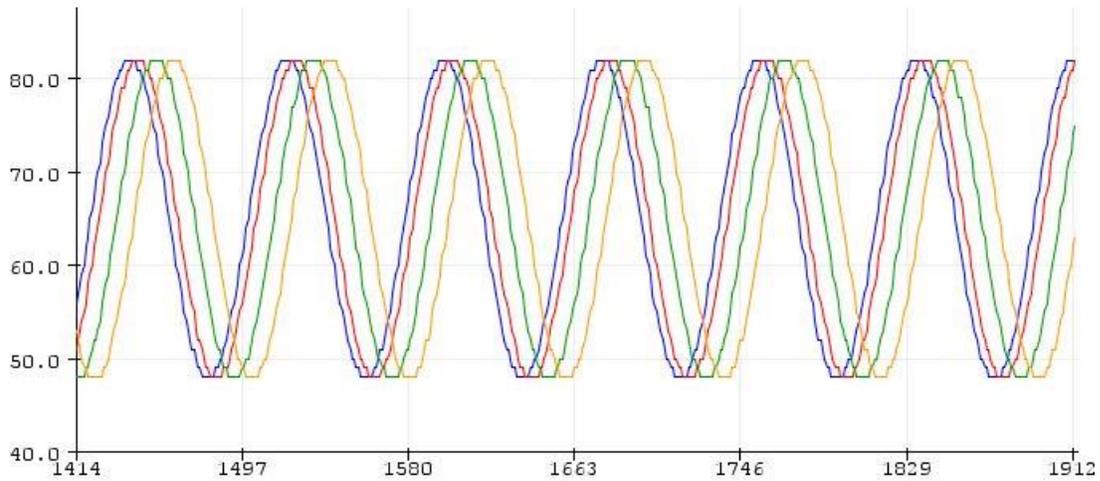


Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

Puede notarse que, como en el patrón de caminata, la generación de un patrón estable no es instantánea. Toma tiempo en converger al comportamiento deseado.

Esto se aprecia al comparar la figura 30 con las figuras 31 y 32, en las que las señales parecen comportarse de forma idéntica. Cabe resaltar que el patrón estable empieza a generarse considerablemente más rápido que utilizando la matriz de acople de caminata; esto es notable observando cerca del extremo derecho de la gráfica mostrada en la figura 30, donde se puede observar ya el comportamiento estable.

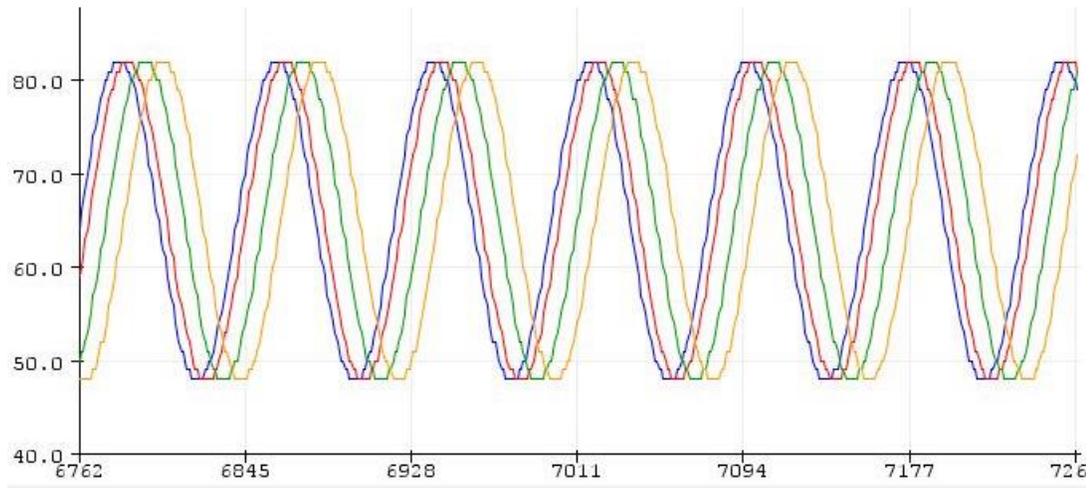
Figura 31. **Comportamiento a mediano plazo de CPG con matriz de patrón de trote**



Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

Se observa en el patrón estable (figura 32) que  $X_1$  y  $X_2$  presentan un desfase minúsculo entre sí y un desfase ligero con respecto a  $X_3$  y  $X_4$ , esto se traduce en que los servomotores de cadera tienen menor diferencia temporal entre el accionamiento de cada uno con respecto al anterior.

Figura 32. **Comportamiento a largo plazo de CPG con matriz de patrón de trote**



Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

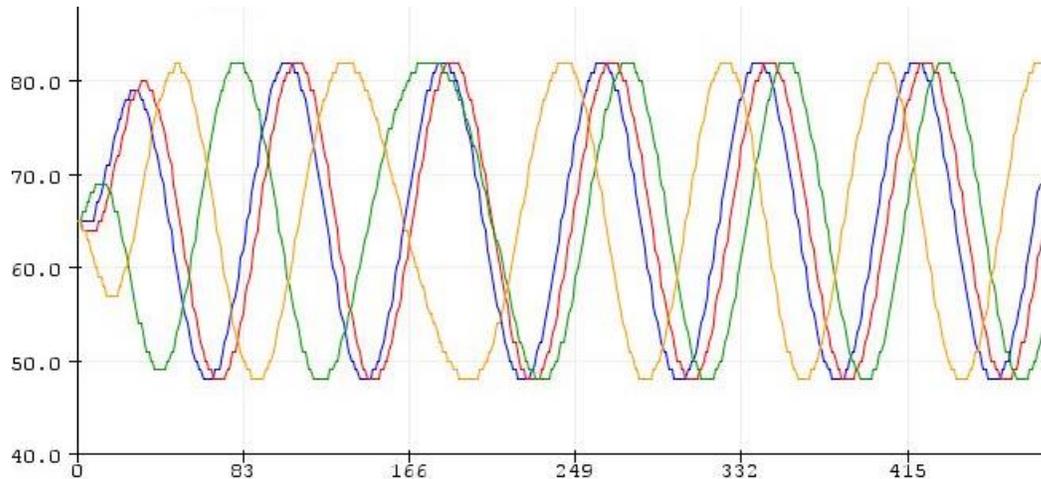
### 5.3.3. **Funcionamiento de patrón de galope**

Las figuras 33, 34 y 35 muestra las señales generadas por los osciladores utilizando la matriz de acople que genera el patrón de galope (ver tabla VIII).

En la figura 33 se observa que cada señal parte de un valor inicial de  $65^\circ$ , correspondiente a la posición angular inicial de todas las articulaciones de cadera al arrancar la operación. Este valor empieza a variar en cada oscilador, permitiendo observar desfases entre cada uno.

Los desfases en las gráficas se traducen en desfases en el movimiento de los servomotores con los que el robot móvil está equipado.

Figura 33. **Inicio de operación de CPG con matriz de patrón de galope**



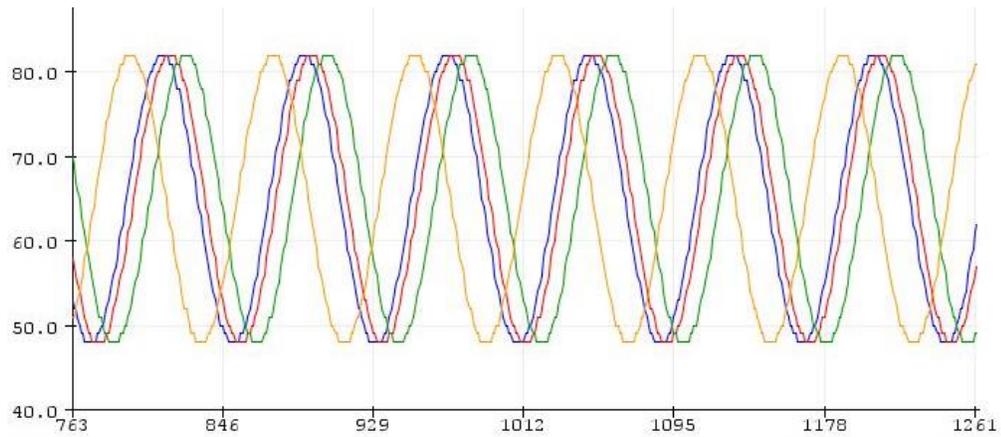
Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

Puede notarse que la generación de un patrón estable no es instantánea, más bien toma tiempo en converger al comportamiento deseado.

Esto se aprecia al comparar la figura 33 con las figuras 34 y 35; sin embargo, a diferencia de los dos patrones observados anteriormente, el patrón estable al utilizar la matriz de galope muestra diferencias de fase significativa entre las gráficas de las figuras 34 y 35, correspondiente al comportamiento a mediano y largo plazo.

Esto implica que el patrón de galope es, de los tres implementados, el que más tarda en estabilizarse.

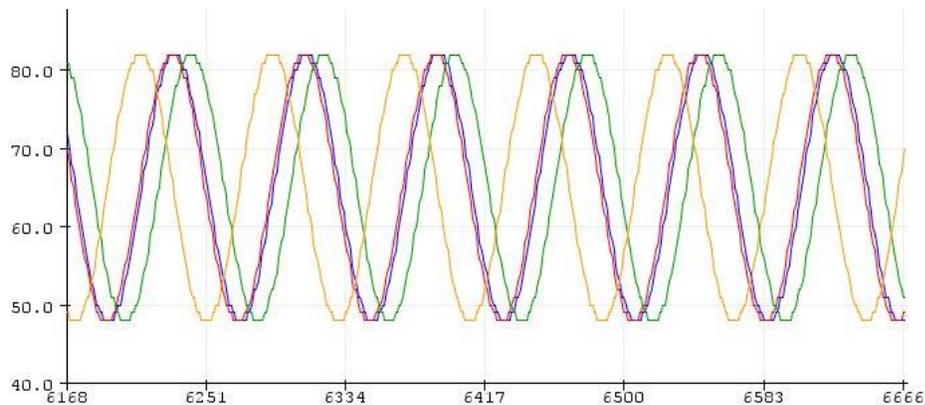
Figura 34. **Comportamiento a mediano plazo de CPG con matriz de patrón de galope**



Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

Se observa en el patrón estable (figura 35), que  $X_1$  y  $X_2$  presentan desfases minúsculos, mientras  $X_3$  y  $X_4$  presenta desfases considerables tanto entre sí como con respecto a los otros dos osciladores.

Figura 35. **Comportamiento a largo plazo de CPG con matriz de patrón de galope**



Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

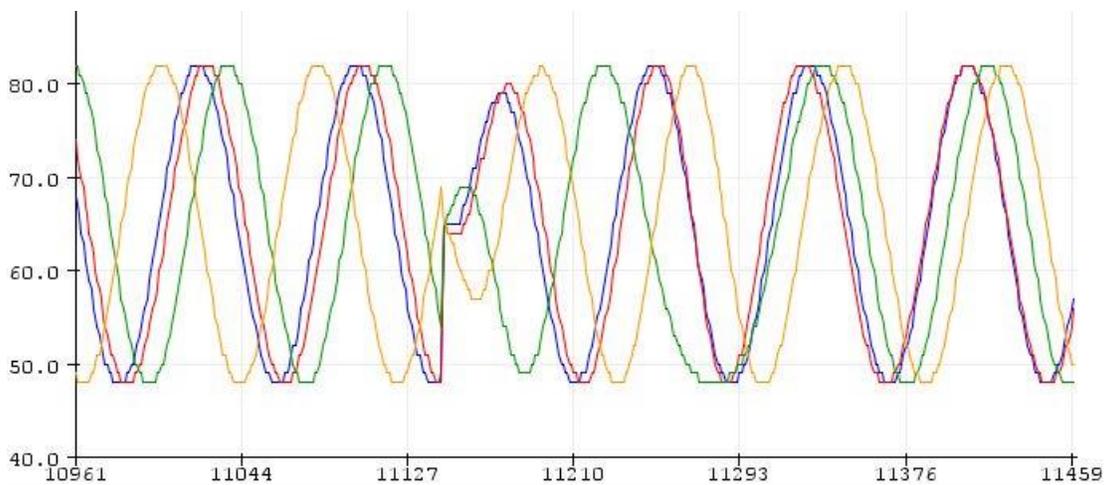
### 5.3.4. Transiciones entre patrones

Como prueba adicional, se observó la variación de señales que ocurría al cambiar el programa cargado al microcontrolador para utilizar, primero, la matriz de acople de caminata y luego la de trote.

En la figura 36 se observa una disrupción notable por el cambio repentino de amplitud de las señales de todos los osciladores, esta corresponde precisamente al cambio mencionado. Es notable también que las señales no tardaron mucho tiempo en reestablecerse y buscar sincronizarse de acuerdo a la nueva matriz de acople utilizada.

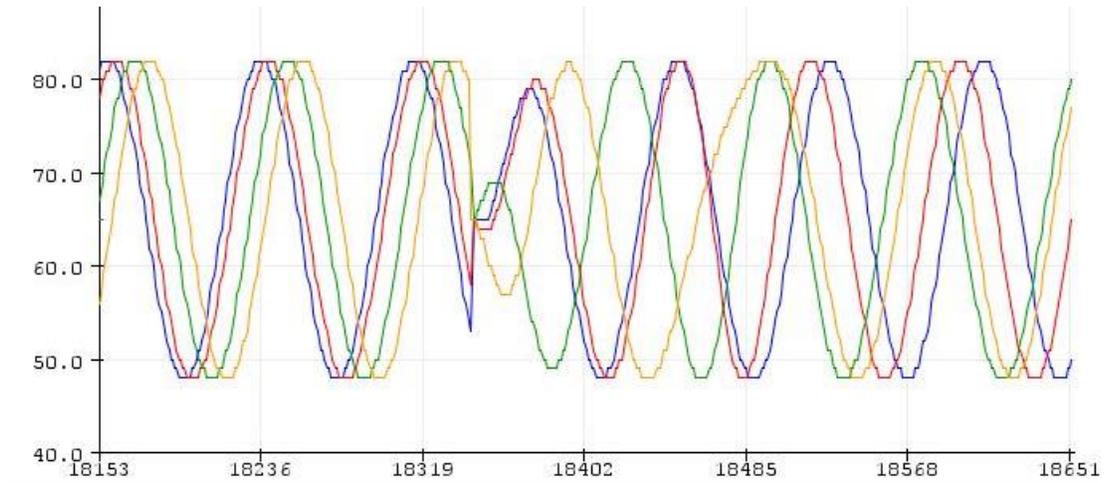
En la figura 37 puede observarse el caso inverso al descrito para la figura 36.

Figura 36. Transición entre patrones de caminata a trote



Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

Figura 37. **Transición entre patrones de trote a caminata**



Fuente: elaboración propia, empleando la herramienta de graficado serial de Arduino 1.8.10.

Esta prueba adicional fue la última realizada. Tomando como base los resultados obtenidos, que completaron a satisfacción los objetivos propuestos. Se procede a describir propuestas para futuros trabajos, que pueden tomar el presente como punto de partida, en el próximo capítulo.



## 6. PROPUESTAS A FUTURO

Como se pudo apreciar en los capítulos anteriores, muchas de las decisiones de diseño y dimensionamiento para el sistema fueron tomadas con escalabilidad y mejoras futuras en mente. Esto se debe a que los controladores basados en CPG ofrecen tanto facilidades como desafíos al agregar componentes a sistemas regidos bajo este paradigma.

En este capítulo, último del presente trabajo, se describen brevemente algunas propuestas de mejoras que, por salirse del enfoque definido inicialmente, no fueron implementadas, pero podrían serlo en futuras iteraciones.

### 6.1. Hardware y firmware

Se propone el desarrollo y montaje de una placa de circuito impreso (PCB), para el sistema. Esta propuesta conlleva ventajas de estabilidad en el montaje, que evitan malas conexiones y demás problemas eléctricos que interrumpen el flujo de pruebas, como la potencial reducción del área requerida para el montaje del controlador.

KiCAD, el *software* con el que se elaboró el diagrama del circuito del controlador, tiene la capacidad de exportar este diagrama y convertirlo, al menos parcialmente, a un formato de PCB montable.

La siguiente propuesta de *hardware* corresponde a la selección de servomotores. Conforme se agregan más extremidades y articulaciones a controlar al sistema, también aumenta, de forma proporcional, el peso de las

partes móviles del robot controlado. Esto hace necesario elegir servomotores de mayor capacidad de torque y velocidad para compensar por la carga mecánica adicional.

La placa de pruebas actual fue diseñada de modo que pudieran serle conectados hasta 32 servomotores de consumo energético aún mayor del requerido por los actuales, por lo que esta propuesta es fácil de implementar en futuros trabajos.

La última propuesta de mejora de *hardware* va de la mano con el aumento de extremidades y partes móviles del robot. El movimiento de cada una de ellas requiere poder de procesamiento proporcional, y eventualmente, sería necesario elegir un microcontrolador (u otro tipo de procesador), con capacidad de procesamiento mayor al actual.

Para que esta propuesta sea funcional, debe verificarse que el procesador elegido sea compatible con las características eléctricas del sistema. De no serlo, deberán implementarse las adaptaciones e interfaces adicionales correspondientes.

De la mano con las propuestas de mejora del *hardware*, particularmente las que están relacionadas con el procesador del sistema, debe adaptarse también el *firmware* a implementarse con el potencial nuevo procesador. Debe verificarse las condiciones de licenciamiento y uso de *software* y *hardware* requeridos para la implementación del *firmware*, así como sus nuevas capacidades, lenguaje de programación y librerías disponibles.

## **6.2. Modelo matemático y computacional**

El modelo matemático de Kuramoto, implementado en este trabajo para modelar osciladores acoplados en una red, ofrece varias ventajas que ya se han mencionado en el mismo. Existen modelos más eficientes que implementan incluso retroalimentación sensorial y cálculos cinemáticos, como los empleados por Yinquan Zeng en una de las fuentes bibliográficas citadas en el presente.



## CONCLUSIONES

1. Se adaptó la arquitectura de control biomimética basada en el modelo computacional de osciladores acoplados de Kuramoto para emular el comportamiento de los CPG.
2. Se implementó la arquitectura de control diseñada en un robot móvil dotado de servomotores como medio de movimiento.
3. Se evaluó cualitativamente el desempeño del controlador, revisando que los movimientos que inducía tuvieran transiciones suaves y similares a las de organismos vivos y cuantitativamente verificando los efectos de las alteraciones en los parámetros de frecuencia, fase y amplitud del modelo de Kuramoto sobre el comportamiento del sistema.
4. Se establecieron los parámetros requeridos para generar tres patrones de desplazamiento frontal diferentes: caminata, trote y galope.



## RECOMENDACIONES

1. Proponer el uso de robots móviles con extremidades como soluciones a tareas que se realicen bajo condiciones de terrenos no controlados. Como se ha referido en este trabajo, el uso de robots móviles con extremidades, en lugar de esquemas tradicionales basados en ruedas, es un área muy prometedora que debería seguir siendo explorada. Los robots que emplean extremidades pueden adaptarse mejor a condiciones no controladas y recuperarse con mayor facilidad de accidentes.
2. Emplear la metodología propuesta en este trabajo para iniciar el desarrollo de arquitecturas de controladores de robots móviles con extremidades. Si bien no existe aún una metodología general para el desarrollo de controladores de robots móviles basados en CPG, la descrita en este documento, especialmente la parte referente al diseño electrónico, puede ser el punto de partida para proyectos más avanzados.
3. Mejorar el controlador diseñado para este trabajo agregando adaptaciones como análisis cinemático para mejorar la exactitud y precisión del modelo para el posicionamiento de partes específicas del cuerpo del robot móvil.
4. Utilizar los parámetros empleados en la implementación descrita, recordando escalarlos según la aplicación lo requiera, y mantener por encima de todo, las relaciones de fase que estos representan para cada CPG.



## BIBLIOGRAFÍA

1. ACEBRÓN, Juan A. *The Kuramoto model: A simple paradigm for synchronization*. Alcalá de Henares, España: Reviews of Modern Physics, 2005. 49 p.
2. ADAFRUIT. Adafruit PCA9685 16-channel servo driver. *Learn Adafruit*. [en línea]. <<https://learn.adafruit.com/16-channel-pwm-servo-driver>>. [Consulta: 3 de julio de 2021].
3. AMAZON. *Longrunner 10 x MG996R metal gear torque digital servo motor para rc modelo coche barco helicóptero lky62*. [en línea]. <<https://www.amazon.com/-/es/Longrunner-MG996R-Torque-Digital-Helic%C3%B3ptero/dp/B01MTW06IU/>>. [Consulta: 15 de julio de 2021].
4. BARRÓN-ZAMBRANO, José; TORRES-HUITZIL, César. *CPG implementations for robot locomotion. Robotic systems-applications, control and programming*. Londres, Inglaterra: InTechOpen, 2012. 24 p.
5. BARRÓN-ZAMBRANO, José; TORRES-HUITZIL, César; GIRAU, Bernard. *Hardware implementation of a CPG-based locomotion control for quadruped robots*. Thessaloniki, Grecia: Artificial Neural Networks - ICANN 2010, 20th International Conference, 2010.

6. CARRANZA, Víctor. *CPG\_project*. *GitHub*. [en línea]. <[https://github.com/vmcarranzam/CPG\\_project](https://github.com/vmcarranzam/CPG_project)>. [Consulta: 9 de julio de 2021].
7. DAVIDSON, Andrew. *Lecture 1: Introduction to robotics*. *Department of Computing, Imperial College London*. [en línea]. <<https://www.doc.ic.ac.uk/~ajd/Robotics/RoboticsResources/lecture1.pdf>>. [Consulta: 2 de julio de 2021].
8. DHOLAKIYA, Dhaivat; BHATTACHARYA, Shounak; GUNALAN Ajay; SINGLA, Abhik. *Design, development and experimental realization of a quadrupedal research platform: stoch*. *arXiv*. [en línea]. <<https://arxiv.org/abs/1901.00697>>. [Consulta: 2 de julio de 2021].
9. ELECTRÓNICOS CALDAS. *MG996R high torque metal gear dual ball bearing servo*. [en línea]. <[https://www.electronicoscaldas.com/datasheet/MG996R\\_Tower-Pro.pdf](https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf)>. [Consulta: 15 de julio de 2021.]
10. EPFL. Lynx. *EPFL biorobotics laboratory*. [en línea]. <<https://www.epfl.ch/labs/biorob/research/quadruped/quadruped-past/lynx-2/>>. [Consulta: 26 de junio de 2021].
11. EPFL. Pleurobot. *EPFL biorobotics laboratory*. [en línea]. <<https://www.epfl.ch/labs/biorob/research/amphibious/pleurobot/>>. [Consulta: 26 de junio de 2021].

12. EPFL. *Pleurobot walking comparison*. EPFL Biorobotics Laboratory. [en línea]. <<https://biorob2.epfl.ch/utills/movieplayer.php?id=288>>. [Consulta: 26 de junio de 2021]. GARCÍA-SAURA, Carlos. *Central Pattern Generators for the control of robotic systems*. Londres, Inglaterra: Cornell University, 2015. 1509.02417. 12 p.
13. EYEBOOT. *36V 500W DC universal regulated switching power supply AC to DC 13.8 Amps*. [en línea]. <<https://www.eyebboot.com/36v-500w-dc-power-supply.html>>. [Consulta: 5 de julio de 2021].
14. GONZÁLEZ GARCÍA, Anthony. *¿Qué es y cómo funciona un servomotor?* Panamahitek. [en línea]. <<http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>>. [Consulta: 30 de junio de 2021].
15. GONZÁLEZ GÓMEZ, Juan. *Cube revolutions*. IEA robotics. [en línea]. <<http://www.iearobotics.com/personal/juan/doctorado/cube-revolutions/>>. [Consulta: 26 de junio de 2021].
16. LA ELECTRÓNICA. *Módulo regulador de voltaje con límite de corriente CC CV - 20A*. [en línea]. <<https://laelectronica.com.gt/modulo-regulador-de-voltaje-con-limite-de-corriente-cc-cv---20a>>. [Consulta: 5 de julio de 2021].
17. MARKWTECH. *Hexapod robot*. Thingiverse. [en línea]. <<https://www.thingiverse.com/thing:3463845>>. [Consulta: 15 de julio de 2021].

18. NASA/JPL-Caltech. *Some differences between Mars 2020 and curiosity. NASA Mars exploration program.* [en línea]. <<https://mars.nasa.gov/resources/24716/some-differences-between-mars-2020-and-curiosity/>>. [Consulta: 26 de junio de 2021].
19. Rosseta Code. *Euler Method.* [en línea]. <[http://rosettacode.org/wiki/Euler\\_method](http://rosettacode.org/wiki/Euler_method)>. [Consulta: 2 de julio de 2021].
20. SIEGWART, Roland; NOURBAKHS, Illah R; SARAMUZZ, Davide. *Introduction to autonomous mobile robots.* 2nd. Cambridge, Massachusetts: The MIT Press, 2011. ISBN 978-0-262-01535-6. 453 p.
21. ZAPATA, Fanny. *Método de Euler: para qué sirve, procedimiento y ejercicios.* [en línea]. <<https://www.lifeder.com/metodo-de-euler/>>. [Consulta: 29 de junio de 2021].
22. ZENG, Yinquan; LI, Junmin; YANG, Simon. REN, Erwei. *A bio-inspired control strategy for locomotion of a quadruped robot 1.* Chengdu, China: Applied Sciences, 2 de enero de 2018, Vol. 8, 56 p.