

*Estimación de una serie de movimientos utilizando un algoritmo de optimización bio-inspirado para la operación de manera autónoma y On-Line de una plataforma Multi-Robot (Caso robot modular)*

HENRY ALBERTO HERNÁNDEZ MARTINEZ  
INGENIERO



UNIVERSIDAD NACIONAL DE COLOMBIA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS E INDUSTRIAL  
BOGOTÁ, D.C.  
AGOSTO DE 2018

*Estimación de una serie de movimientos utilizando un  
algoritmo de optimización bio-inspirado para la operación  
de manera autónoma y On-Line de una plataforma  
Multi-Robot (Caso robot modular)*

HENRY ALBERTO HERNÁNDEZ MARTINEZ  
INGENIERO

PROPUESTA PRESENTADA PARA OPTAR AL TÍTULO DE MAGÍSTER  
MAGÍSTER EN INGENIERÍA DE TELECOMUNICACIONES

DIRECTOR  
JONATAN GOMÉZ PERDOMO, PH.D.  
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

ASESOR  
RODRIGO MORENO GARCÍA, PH.D (C).  
CANDIDATO A DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

LÍNEA DE INVESTIGACIÓN  
INTELIGENCIA Y MENTE ARTIFICIAL

GRUPO DE INVESTIGACIÓN  
GRUPO DE INVESTIGACIÓN EN VIDA ARTIFICIAL (ALIFE)



UNIVERSIDAD NACIONAL DE COLOMBIA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS E INDUSTRIAL  
BOGOTÁ, D.C.  
AGOSTO DE 2018

**Título en español**

Estimación de una serie de movimientos utilizando un algoritmo de optimización bio-inspirado para la operación de manera autónoma y On-Line de una plataforma Multi-Robot (Caso robot modular)

**Title in English**

Estimation of a series of movements using a bio-inspired optimization algorithm for the autonomous operation and ON-LINE of a multi-robot platform (modular robot case)

**Resumen:** Esta tesis presenta una estrategia de adaptación implementable en un controlador local de un robot modular tipo cadena. Este controlador se adapta a diferentes configuraciones del robot y estima conjuntos de movimientos al modular un Generador Central de Patrones (CPG) mediante una técnica de optimización. Esta técnica permite coordinar y controlar los movimientos de los módulos usando la información de los sensores, ya que asignan un valor de aptitud a cada movimiento realizado por el robot. Esta característica le permite al controlador seleccionar movimientos adecuados para que el robot resuelva diferentes tipos de problemas de manera autónoma. Teniendo en cuenta que la interacción entre los módulos y dispositivos depende del mecanismo de comunicación basado en el estándar CAN (Controller Area Network).

**Abstract:** This thesis presents an adaptation strategy that can be implemented in the local controller of a modular chain-type robot. This controller adapts to different robot configurations and estimates movement sets by modulating of a Central Pattern Generator (CPG) using an optimization technique. This technique allows to coordinate and to control the movements of the modules using the information of the sensors, since they assign a fitness value to each movement made by the robot. This feature allows the controller to select appropriate movements for the robot to solve different types of problems autonomously. Bearing in mind that the interaction between modules and devices depends on the communication mechanism based on the CAN (Controller Area Network) standard.

**Palabras clave:** Algoritmo de optimización, Robot modular, Bio-inspirado

**Keywords:** Optimization algorithm, Modular robot, Bio-inspired.

---

---

## Agradecimientos

---

---

En primer lugar, en el aspecto profesional quisiera agradecer a mi director de tesis el profesor Jonatan Gómez Perdomo y a mi asesor el ingeniero Rodrigo Moreno García, ya que, me han instruido adecuadamente durante el desarrollo de esta tesis desde que ingrese al grupo de investigación ALIFE. También a los integrantes del grupo de investigación cuyos aportes y observaciones me ayudaron a mejorar la forma de presentar mi trabajo. Finalmente, al laboratorio de fabricación de circuitos impresos de la Universidad Distrital Francisco José de Caldas, quien me apoyó en la fabricación de los circuitos impresos de prueba durante la fase de desarrollo del prototipo de control del robot.

Finalmente, en el aspecto personal quisiera agradecer a mi familia y amigos quienes han ofrecido su ayuda y apoyado cuando lo he necesitado. Finalmente, a todos muchas gracias.

---

---

## Contribuciones

---

---

Durante el desarrollo de esta tesis se realizaron dos (2) publicaciones que se describen a continuación.

- La primer publicación fue presentada en la modalidad de póster en GECCO (del inglés, Genetic and Evolutionary Computation Conference) en el año 2017. A pesar de que en esta publicación se describe el ensamblaje y funcionamiento del robot EMERGE, el aporte de esta tesis permitió incorporar en los módulos del robot un controlador local que se describe detalladamente en el Capítulo 3. La referencia de esta publicación se presenta a continuación.

Rodrigo Moreno, Ceyue Liu, Andres Faina, Henry Hernandez, and Jonatan Gomez, 2017. *The EMeRGE modular robot, an open platform for quick testing of evolved robot morphologies*. In Proceedings of GECCO '17 Companion, Berlín, Alemania, Julio 15-19, 2017.

- La segunda publicación fue presentada como un artículo en IBERAMIA (del inglés, IberoAmerican Society of Artificial Intelligence) en el año 2018. En esta publicación se describe la forma de implementar tres (3) técnicas de optimización (descritas en el Capítulo 4) en un robot modular y su comportamiento durante el funcionamiento del robot (presentado en el Capítulo 5). La referencia de esta publicación se presenta a continuación.

Henry Hernández, Rodrigo Moreno, Andres Faina, and Jonatan Gomez, 2018. *Design of a bio-inspired controller to operate a modular robot autonomously*. In Proceedings of the 16th Ibero-American Conference on Artificial Intelligence, Trujillo, Peru, Noviembre 13-16, 2018.

Otras de las contribuciones de esta tesis fueron; La fabricación y diseño de los circuitos electrónicos de control para cada modulo y para la estructura (el funcionamiento de los controladores se describe en la sección 3.3.1 y el diagrama de conexiones en el apéndice B). La fabricación de accesorios adicionales para realizar el control del robot (descritos en la sección 3.3.2). El desarrollo de aplicaciones de prueba para la plataforma (basados en los algoritmos del Capítulo 4). Sin embargo, el prototipo de robot modular EMERGE no se construyó totalmente durante el desarrollo de esta tesis, ya que ha sido construido de manera colaborativa entre estudiantes de la Universidad Nacional de Colombia y la Universidad de Copenhagen, más específicamente el equipo de trabajo está conformado por; Rodrigo Moreno, Andres Faina, Ceyue Liu y el autor de esta tesis. Las contribuciones

del equipo de trabajo se encuentran disponibles en una página web<sup>1</sup> en la cual están disponibles; los diagramas eléctricos, los circuitos impresos, entre otras cosas necesarias para la fabricación del robot.

---

<sup>1</sup>Disponible en: <https://sites.google.com/view/emergemodular/>

---

---

# Índice general

---

---

Índice general	II
Índice de tablas	V
Índice de figuras	VI
Introducción	IX
Objetivo general	XII
<b>1. Robots Modulares</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Descripción de los robots modulares . . . . .	1
1.3. Categorías de los robots modulares . . . . .	2
1.3.1. Estructura tipo cadena . . . . .	2
1.3.1.1. ModRED . . . . .	3
1.3.1.2. LR . . . . .	4
1.3.1.3. Mecabot . . . . .	4
1.3.2. Estructura tipo retícula . . . . .	5
1.3.2.1. Odín . . . . .	5
1.3.2.2. Robot modular con ejes magnéticos . . . . .	6
1.3.3. Estructuras híbridas . . . . .	6
1.3.3.1. HyMod . . . . .	7
1.3.3.2. HexaMob . . . . .	7
1.4. Resumen del Capítulo . . . . .	8
<b>2. Control de movimiento en robots modulares</b>	<b>9</b>
2.1. Introducción . . . . .	9

2.2. Locomoción en robots . . . . .	9
2.2.1. Sistemas de control . . . . .	10
2.2.1.1. Control centralizado . . . . .	10
2.2.1.2. Control des-centralizado . . . . .	10
2.2.2. Técnicas de control de movimiento en robots modulares . . . . .	11
2.2.2.1. Tablas de control de movimiento . . . . .	11
2.2.2.2. Cinemática inversa . . . . .	11
2.2.2.3. Generadores Centrales de Patrones . . . . .	12
2.2.2.4. Generadores sinusoidales . . . . .	13
2.3. Aplicaciones de los CPGs en robótica . . . . .	14
2.4. Estrategias de navegación . . . . .	15
2.5. Resumen del capítulo . . . . .	18
<b>3. Robot modular EMeRGE . . . . .</b>	<b>20</b>
3.1. Introducción . . . . .	20
3.2. Descripción general del robot . . . . .	20
3.3. Arquitectura del sistema de control . . . . .	21
3.3.1. Controlador local de cada módulo . . . . .	21
3.3.2. Dispositivos externos o accesorios . . . . .	24
3.4. Funcionamiento del sistema de control . . . . .	25
3.4.1. Comunicación entre dispositivos . . . . .	25
3.4.2. Generación de movimientos . . . . .	27
3.4.3. Detección de obstáculos . . . . .	27
3.5. Coordinación del sistema de control . . . . .	28
3.6. Limitaciones de los módulos . . . . .	29
3.7. Resumen del capítulo . . . . .	30
<b>4. Estrategia de adaptación . . . . .</b>	<b>31</b>
4.1. Introducción . . . . .	31
4.2. Descripción del funcionamiento del CPG . . . . .	31
4.3. Optimización del CPG . . . . .	35
4.3.1. Características de un individuo . . . . .	35
4.3.2. Ascenso a la colina y templado simulado . . . . .	37
4.3.3. Algoritmo genético . . . . .	39
4.4. Resumen del capítulo . . . . .	43



---

<b>5. Resultados obtenidos</b>	<b>44</b>
5.1. Introducción . . . . .	44
5.2. Experimentos con morfologías tipo cadena . . . . .	44
5.2.1. Experimento 1 . . . . .	44
5.2.2. Experimento 2 . . . . .	47
5.3. Resumen del capítulo . . . . .	50
<b>6. Conclusiones</b>	<b>51</b>
<b>7. Trabajo Futuro</b>	<b>53</b>
<b>A. Construcción de los módulos del robot</b>	<b>55</b>
A.1. Circuitos impresos . . . . .	55
A.2. Ensamblaje de un módulo . . . . .	57
<b>B. Circuitos electrónicos.</b>	<b>59</b>
<b>Bibliografía</b>	<b>62</b>

---

---

## Índice de tablas

---

---

2.1. Clasificación y características de algunos robots modulares. . . . .	18
5.1. Resultados de la prueba de suma de rangos de Wilcoxon realizada a las combinaciones de los mejores valores de aptitud registrados en cinco (5) ejecuciones de cada técnica de optimización en el escenario uno (1). . . . .	50
5.2. Resultados de la prueba de suma de rangos de Wilcoxon realizada a las combinaciones de los mejores valores de aptitud registrados en cinco (5) ejecuciones de cada técnica de optimización en el escenario dos (2). . . . .	50

---

---

## Índice de figuras

---

---

1.1. Categorías de los robots modulares organizados de acuerdo con su estructura (Adaptado del trabajo presentado por Satoshi Murata [57]). . . . .	3
1.2. Robot modular ModRED. Tomado de [9]. . . . .	4
1.3. Robot modular LR (Little Red). Tomado de [102]. . . . .	4
1.4. Robot modular Mecabot. Tomado de [39]. . . . .	5
1.5. Robot modular Odín. Tomado de [48]. . . . .	5
1.6. Robot modular con ejes magnéticos. Tomado de [75]. . . . .	6
1.7. Robot Modular HyMod. Tomado de [64]. . . . .	7
1.8. Robot modular HexaMob. Tomado de [16]. . . . .	8
2.1. Sistema de dos (2) masas acopladas mediante resortes. . . . .	12
2.2. Respuesta de un sistema acoplado de masa-resorte con los siguientes parámetros $M_1 = M_2 = 1kg, K_1 = 10N/m, K_2 = 100N/m, K_3 = 10N/m$ . La frecuencia de oscilación del sistema acoplado está marcado por la línea punteada y la frecuencia de cada sistema independiente es la línea continua. . . . .	13
2.3. Respuesta de la estructura robótica EMeRGE durante un ciclo de marcha hacia adelante usando un controlador sinusoidal. . . . .	14
2.4. Representación del robot modular Microtub. Tomado de [14]. . . . .	17
2.5. Ruta trazada para un cuadro-podo usando el algoritmo RRT. Tomado de [85]. . . . .	18
3.1. Morfología de tipo serpiente ensamblada usando la plataforma Robótica EMeRGE. . . . .	21
3.2. Vista frontal de un módulo de la plataforma EMeRGE y distribución de los pines para realizar una conexión con otros módulos. . . . .	21
3.3. Representación de las conexiones eléctricas que alimentan diferentes elementos del módulo. . . . .	22
3.4. Representación de las conexiones eléctricas de control. . . . .	23
3.5. Comportamiento de la señal de control del servomotor. . . . .	23

3.6. Ubicación de los puentes según el modo de funcionamiento y JTAG. . . . .	24
3.7. Dispositivos externos o accesorios compatibles con la plataforma EMeRGE. . . . .	25
3.8. Buzones que tiene definidos en micro-controlador y su respectiva función o variables a almacenar asignadas. . . . .	26
3.9. Características del servomotor (de referencia AX12 fabricado por Dinami-xel), basado en [67]. . . . .	27
3.10. Ejemplo de cómo se asignan las direcciones de cada módulo durante el inicio de la rutina de control. . . . .	28
3.11. Detección de la orientación de los módulos durante la ejecución de tareas de locomoción. . . . .	29
3.12. Posición angular máxima que puede tener un módulo. . . . .	29
3.13. Corte de una sección transversal del módulo que muestra 3 escenarios dife-rentes que puede enfrentar el sensor de proximidad. . . . .	30
4.1. Actualización de los parámetros del CPG. . . . .	32
4.2. Tiempo de estabilización de $r_i$ y $x_i$ para diferentes valores de $a_r$ y $a_x$ . . . . .	33
4.3. Solución de la ecuación de CPG usando el método de Euler para diferentes valores $a_r, a_x, R_i$ y $X_i$ . . . . .	33
4.4. Respuesta de la ecuación de CPG con dos (2) módulos acoplados con dife-rencia de fase. . . . .	34
4.5. Comportamiento del CPG en la estructura robótica EMeRGE. . . . .	34
4.6. Medición del valor de aptitud con los dos (2) tipos de sensores disponibles en el robot EMeRGE. . . . .	36
4.7. Combinaciones de las variables de entrada dependiendo de la orientación del módulo. . . . .	36
4.8. Representación del funcionamiento de los algoritmos de ascenso a la colina y templado simulado durante el funcionamiento del robot EMERGE. . . . .	37
4.9. Estructura de un algoritmo genético, basado en el trabajo presentado por Thomas Weise [87]. . . . .	40
4.10. Representación del funcionamiento de los algoritmos de ascenso a la colina y templado simulado durante el funcionamiento del robot EMERGE. . . . .	40
5.1. Fotogramas de la secuencia de movimientos realizada por una estructura con tres módulos conectados en serie con orientación vertical. . . . .	45
5.2. Promedio de los mejores valores de aptitud (distancia recorrida) encontra-dos en cada evaluación, al ejecutar las técnicas de optimización cinco (5) veces con una morfología del robot de tres módulos conectados en serie en orientación vertical. . . . .	45
5.3. Obstáculo en forma de escalera. . . . .	46
5.4. Fotogramas de la estructura de seis (6) módulos evadiendo un obstáculo en forma de escalera. . . . .	46

---

5.5. Morfología ensamblada para habilitar el giro del robot. . . . .	47
5.6. Escenarios propuestos para comparar las estrategias de optimización. . . . .	47
5.7. Resumen de las cinco (5) ejecuciones de las tres técnicas de optimización usadas. . . . .	48
5.8. Comportamiento del valor de aptitud con una mutación inesperada. . . . .	48
5.9. Cambios repentinos detectados durante la ejecución de las técnicas de op- timización. . . . .	49
7.1. Estructuras arbitrarias construidas con cinco (5) módulos del robot EMeRGE. 54	
A.1. Segmento del plano de conexiones eléctricas del módulo . . . . .	55
A.2. Cálculo aproximado de los parámetros de diseño de un PCB . . . . .	56
A.3. Archivo *.Gerber generado para la fabricación de los PCB. . . . .	56
A.4. Acondicionamiento de las caras del módulo . . . . .	57
A.5. Buses y Conectores de la plataforma robótica . . . . .	57
A.6. Fases del ensamblaje de un módulo . . . . .	58

---

---

## Introducción

---

---

A nivel industrial el desarrollo de actividades repetitivas, peligrosas o imposibles ha impulsado el desarrollo de artefactos, que reduzcan el riesgo del operario durante su ejecución para aumentar su productividad. Con el paso del tiempo, el avance tecnológico ha permitido mejorar dichos artefactos convirtiéndolos en máquinas automatizadas o herramientas especializadas, por ejemplo, los brazos articulados que manipulan o ensamblan piezas en un proceso [10, 99].

Algunas de estos artefactos se denotaron con el nombre de robot<sup>2</sup>, ya que este término se utiliza para designar a una máquina que tiene la capacidad de servir al hombre. Existen varias definiciones de robot, entre ellas se conoce como *“máquinas en las que se integran componentes mecánicos, electrónicos, eléctricos y de comunicaciones, que tienen un sistema de procesamiento para su control en tiempo real, percepción del entorno y programación”*[10]. Debido a la gran cantidad de tareas y aplicaciones que puede tener un robot, se popularizó el término robótica, que se define como *“una rama que se ocupa del diseño, fabricación, control, y aplicación de los robots”* [81].

Actualmente, la robótica permite la manufactura de diversos tipos de robots. Algunos de ellos ofrecen ventajas como minimizar costos de manufactura en un proceso o reducir el margen de error en la ejecución de una tarea, entre otras. Sin embargo, estas ventajas están limitadas por el nivel de autonomía del robot, ya que esta característica determina que tanta ayuda externa necesita. Se puede decir que un robot es completamente autónomo si puede interpretar la información del ambiente, trabajar sin ayuda, moverse a través de su entorno operativo, evitar situaciones de peligro que lo afecten a el mismo o a una persona y aprender nuevas cosas para adaptarse al entorno [10, 99].

El intento de emular un comportamiento autónomo, ha impulsado la manufactura de diferentes tipos de robots y se han propuesto varias formas de clasificarlos. Por ejemplo, con base en su estructura se clasifican en: poli-articulados, móviles, androides, zoo-mórficos e híbridos [99]. Cada uno de los tipos de robot mencionados tienen mecanismos propios que determinan el tipo de actividad que pueden realizar. Sin embargo, entre ellos se destacan los robots modulares, que son sistemas de dos o más robots acoplados. Esta característica le da a este tipo de robot una ventaja funcional sobre los demás, debido a que puede re-configurar o re-agrupar sus módulos dependiendo del tipo de tarea a realizar [45].

La capacidad de re-configurarse le permite al robot modular generar patrones de locomoción para adaptarse a diferentes superficies o clases de terrenos, lo que resulta favorable en tareas de exploración, búsqueda y rescate. El interés de explorar este tipo de aplica-

---

<sup>2</sup>El término *robot* viene de la palabra checha *robota* que se usa para denotar una labor forzada [10].

ciones es mejorar la capacidad de adaptación del robot a tal punto que pueda explorar entornos desconocidos. Sin embargo, coordinar y controlar los movimientos de las articulaciones del robot para reproducir un movimiento sincronizado que le permita desplazarse de un punto a otro, sigue siendo un reto [29, 34, 57].

Una manera de reproducir este tipo de movimientos es incorporar al robot un controlador bio-inspirado, que permita generar movimientos similares a los que realizan los seres vivos en la naturaleza. Un ejemplo de este tipo de controlador son los CPG (del inglés, Central Pattern Generator), que representan a un conjunto de neuronas especializadas que emulan un patrón de movimiento rítmico. Algunos modelos de CPG se representan mediante modelos simplificados de generadores sinusoidales, ya que, al estar en estado estacionario un CPG basado en osciladores acoplados se comporta como un oscilador de frecuencia fija [19, 38, 45]. Una ventaja de este tipo de representaciones es que son versátiles y se pueden materializar usando varios tipos de tecnologías como circuitos electrónicos o software.

Otra manera de generar movimientos en robots modulares es definir las diferentes posiciones de las articulaciones y reproducirlas de manera secuencial, donde cada movimiento individual se conoce como movimiento primitivo. Sin embargo, los CPG y las estrategias basadas en movimientos primitivos están limitadas, ya que, la cantidad de parámetros disponible para generar un movimiento varía en función de la morfología del robot. En consecuencia, determinar las posiciones exactas de las articulaciones que le permitan a un robot modular generar un movimiento coordinado no es una tarea sencilla [83, 84]. Esta limitación se ha solucionado parcialmente usando técnicas de optimización que aproximan los parámetros de control del robot para generar diferentes movimientos y pueden implementarse en diversos controladores, por ejemplo; un algoritmo evolutivo que encuentra combinaciones de movimientos para que un robot modular vaya de un punto a otro al ejecutar varias veces una simulación modificando los parámetros de control de las articulaciones [14, 43, 45, 56].

La mayoría de estas técnicas se han optimizado en un simulador, debido a la gran cantidad de evaluaciones y parámetros que se deben ejecutar para llegar a una solución. Sin embargo, la transferencia al robot real de los resultados obtenidos usando el simulador, no garantizan un funcionamiento adecuado, ya que dependen de la interacción con su entorno. Para reducir el margen de error de la estrategia simulada, se han incorporado técnicas de optimización en el controlador de un robot real (normalmente en un microcontrolador), donde se han propuesto algoritmos de aprendizaje por refuerzo, genéticos, co-evolutivos o combinaciones entre ellos, para que el robot re-configure su estructura y se desplace en entorno sin obstáculos [36, 58, 61]. Aunque las estrategias mencionadas permiten generar esquemas de control modulares implementables en este tipo de robot, en ocasiones, la construcción del controlador no permite que la estructura robótica procese la información de percepción sensorial en tiempo real, limitando su operación autónoma, es decir, el conjunto de movimientos generado no permite a la estructura adaptarse de manera autónoma al entorno.

En esta tesis se propone una solución a esta limitación, mediante la implementación de un algoritmo genético en un controlador centralizado. Este controlador permite que la estructura robótica conformada module de manera automática los parámetros del CPG presentes en cada módulo, para mejorar su patrón de locomoción a partir de la información recolectada por los sensores. El controlador direcciona la información de los sensores hacia él, lo que permite una re-alimentación y respuesta en tiempo real. Este controlador es

---

flexible y se adapta a cualquier tipo de morfología del robot, ya que el intercambio de información entre los módulos y el controlador se realiza mediante un único bus que usa el protocolo de comunicación CAN. Estas características le permiten al robot navegar y adaptarse de manera autónoma a terrenos irregulares o con obstáculos.

El aporte realizado con este trabajo facilitaría las operaciones de exploración, búsqueda y rescate, ya que, el controlador le da al robot la capacidad de coordinar sus movimientos para adaptarse a entornos desconocidos. Dicho aporte se describe de una manera detallada en este documento, el cual se encuentra organizado de la siguiente manera.

- **Capítulo 1:** Se presentan las definiciones de robot modular y módulo. Además, se clasifica a los robots de este tipo de acuerdo con su estructura.
- **Capítulo 2:** Se presenta una definición de locomoción animal y su relación con robots. También, se encuentran diferentes técnicas que se usan para generar patrones de locomoción en robots y se desatacan las características particulares que permiten su implementación en robots. Se hace énfasis en los CPG y en el modelo de osciladores acoplados. Al final del capítulo se muestran algunas estrategias de planeación de trayectorias basadas en algoritmos evolutivos implementables en este tipo de robot.
- **Capítulo 3:** Se realiza una descripción de la plataforma robótica EMeRGE (del inglés Easy Modular Embodied Robot Generator), el diseño del sistema de control presente en cada módulo y, al final, de cómo se realizó la construcción del sistema de control centralizado.
- **Capítulo 4:** Se describe como se adaptaron tres técnicas de optimización diferentes y como se incorporaron en el controlador del robot.
- **Capítulo 5:** Se muestran los escenarios propuestos para medir el rendimiento de las técnicas de optimización incorporadas en el controlador del robot.



---

---

## Objetivo general

---

---

Proponer un algoritmo de optimización bio-inspirado que permita realizar movimientos primitivos apropiados a una plataforma multi-robot, de manera ON-LINE y autónoma, a partir de la información de percepción sensorial.

## Objetivos específicos

1. Diseñar un algoritmo de optimización bio-inspirado para el ajuste de los patrones de movimiento de una plataforma multi-robot, a partir de la información de percepción sensorial.
2. Implementar el algoritmo diseñado en un sistema de procesamiento que permita su interacción con los sensores y actuadores de la plataforma multi-robot.
3. Verificar el funcionamiento de la plataforma multi-robot incorporando el sistema de procesamiento.
4. Comparar los resultados del algoritmo propuesto con otras técnicas de optimización dentro de un ambiente controlado.

---

---

## Robots Modulares

---

---

### 1.1. Introducción

Al re-configurarse, un robot modular adopta diferentes tipos de morfologías que le permiten generar varios tipos de movimiento. Esta capacidad le permite al robot adaptarse a su entorno con cierto grado de autonomía, ya que, el controlador del robot construye conjuntos de movimientos para crear esquemas de locomoción e ir de un punto a otro [52]. Sin embargo, los movimientos dependen de algunas características particulares de cada robot, entre las cuales se encuentran; el diseño de cada módulo, la cantidad de módulos conectados y los grados de libertad que el robot pueda tener [45].

El desarrollo de esta temática ha permitido construir varios tipos de robot modular [57, 94]. Las principales características de este tipo de robot se describen en este capítulo, el cual se encuentra organizado de la siguiente manera; la sección 1.2 presenta una definición de robot modular y las características básicas para su diseño, en la sección 1.3 se presenta una clasificación de los diferentes tipos de robot modular.

### 1.2. Descripción de los robots modulares

El trabajo presentado por Jantapremjt y Austin define a un Robot Modular como “*un sistema construido a partir de un conjunto de bloques, componentes estandarizados o módulos*”<sup>1</sup> [34]. En robots de este tipo la estructura se define uniendo los módulos mediante una interfaz mecánica, que le permite al robot crear varios tipos de cadenas cinemáticas o cambiar su morfología combinando módulos [65]. Estas características particulares en cada robot dependen del diseño de sus módulos, por está razón se han planteado ciertos parámetros, tales como:

- **Las caras:** Determinan la forma del módulo y en algunos casos tienen mecanismos que le permiten al módulo acoplarse con otros módulos. En ocasiones, la forma de las caras se basa en una figura geométrica regular para facilitar su manufactura [23].

---

<sup>1</sup>El sub-rayado es propio

- **Grados de libertad:** Establecen el número de ejes de rotación que tiene el módulo. Este número representa la forma del movimiento que tiene el módulo, entre los cuales se pueden encontrar: lineal, rotacional o combinaciones entre ellos. El tipo de movimiento se relaciona con el actuador implementado en el módulo que puede ser de tipo eléctrico, neumático o hidráulico [8].
- **Escalabilidad:** Ajusta la capacidad de adaptación y respuesta del módulo al acoplarse con otros módulos o dispositivos y está determina el funcionamiento de la estructura conformada, ya que al aumentar la cantidad de módulos conectados se pueden realizar tareas más complejas. Sin embargo, la escalabilidad depende de ciertas características del módulo y esto puede limitar la operación del robot. Por ejemplo, cuando el mecanismo de acople entre módulos no tiene la suficiente fuerza de tracción para levantar el peso de los módulos vecinos, estos se desconectan [57].
- **Comunicación:** Define el mecanismo de un módulo para compartir información con otros módulos o dispositivos. Cada módulo puede incorporar un protocolo para enviar o recibir información y así realizar ciertas tareas, por ejemplo, actualizar un parámetro de funcionamiento del módulo [41].

El diseño de algunos módulos a partir de mecanismos y circuitos eléctricos simples ha permitido que este tipo de robot sea usado con fines educativos [54]. Además, la capacidad de re-configurar su estructura le da a un robot modular la posibilidad de ejecutar diversos tipos de tareas, tareas como [29, 57]:

- **Exploración de áreas de difícil acceso:** El acceso limitado a ciertas zonas debido a las condiciones ambientales o del terreno, pueden convertir esta tarea en una de alto riesgo para un ser humano. Para reducir el riesgo de accidentalidad se propone usar a los robots modulares, ya que su capacidad de re-configuración le permite inspeccionar un terreno desconocido sin supervisión y con la posibilidad de enviar un reporte a la persona encargada.
- **Reparación de ductos:** La escalabilidad de un robot modular le permite incorporar herramientas con las cuales podría reparar tuberías que lleven sustancias nocivas o sean de difícil acceso para un ser humano.
- **Tareas domésticas:** Al re-configurar la estructura de un robot modular con diferentes herramientas es posible programar ciertas tareas repetitivas, como doblar la ropa, barrer o aspirar.

### 1.3. Categorías de los robots modulares

Existe una gran cantidad de robots modulares y maneras de clasificarlos [94]. En este trabajo este tipo de robot se clasifica de acuerdo con su estructura, a partir de la relación que existe entre las características particulares de cada módulo y los diferentes tipos de movimientos que la estructura robótica conformada realiza (Fig. 1.1) [57].

#### 1.3.1. Estructura tipo cadena

La forma de las caras de los módulos en esta sub-categoría se basa en figuras geométricas regulares. Las formas permiten que la conexión entre módulos se realice de una manera

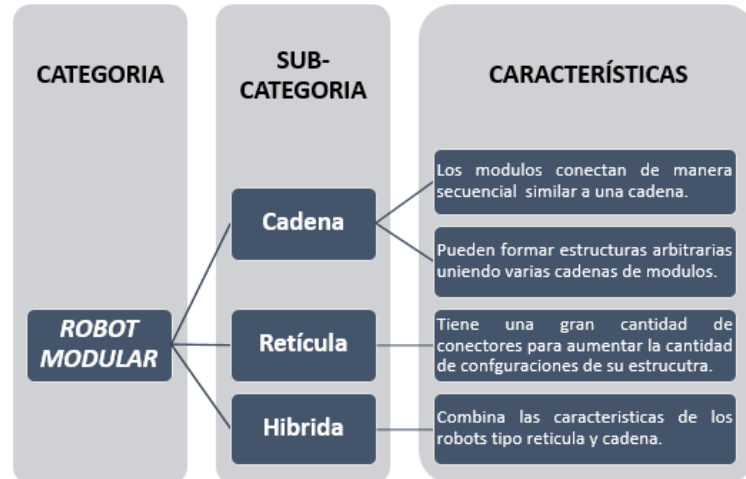


FIGURA 1.1. Categorías de los robots modulares organizados de acuerdo con su estructura (Adaptado del trabajo presentado por Satoshi Murata [57]).

sencilla y las estructuras sean construidas rápidamente. Un ejemplo de este tipo de estructura es el tipo serpiente, que se consigue uniendo cada módulo de la estructura en serie con el anterior [102].

Las estructuras de módulos en serie se denominan cadenas de módulos, dichas cadenas pueden unirse con otras para formar estructuras arbitrarias. Las estructuras arbitrarias, en algunos casos tienen extremidades parecidas a brazos o piernas mediante las cuales puede ejecutar diferentes tipos de tarea. Sin embargo, una de las desventajas de las estructuras arbitrarias es la complejidad para implementar una rutina de control o generación de movimientos, puesto que al aumentar la cantidad de módulos, aumentan los grados de libertad y la cantidad de posibles movimientos que el robot puede llegar a realizar.

A continuación, se presentan tres (3) ejemplos de robot modular con estructura tipo cadena.

#### 1.3.1.1. ModRED

La estructura robótica modular ModRED (Fig. 1.2) está compuesta por módulos que tienen cuatro (4) grados de libertad (tres (3) rotacionales y uno (1) prismático). La intención de los cuatro grados de libertad es aumentar las capacidades de re-configuración y locomoción de la estructura robótica conformada. Cada módulo tiene una placa de acoplamiento que se puede conectar a otra mediante un mecanismo de ocho (8) pines. Estos pines se alinean para permitir que los módulos compartan líneas de alimentación eléctrica y comunicación [9].

Las placas de acoplamiento se encuentran en los extremos de cada módulo. Entre los módulos se encuentran tres (3) segmentos o cajas que se encargan de almacenar: los actuadores, la caja de transmisión y los componentes eléctricos del robot. Los actuadores rotacionales ubicados en las secciones intermedias, tienen un rango de giro de  $\pm 90$  grados; los actuadores rotacionales que se encuentran en los extremos giran 360 grados; y el actuador lineal, con un rango de desplazamiento de una (1) pulgada. Esta combinación de actuadores es capaz de producir movimientos de cabeceo (del inglés *pitch*), de alabeo (del inglés *roll*) y de guiñada (del inglés *yaw*) [9].

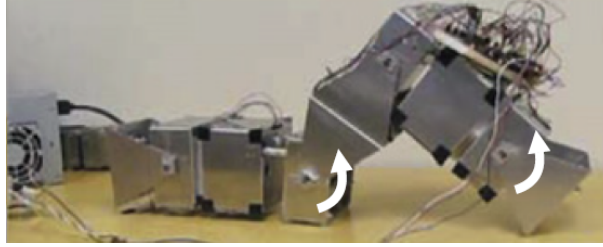


FIGURA 1.2. Robot modular ModRED. Tomado de [9].

### 1.3.1.2. LR

El robot LR (Fig. 1.3) tiene una configuración tipo serpiente compuesta por cinco (5) módulos. El módulo de la cabeza tiene una cámara de vídeo y el módulo de la cola es reservado para agregar más periféricos al robot. En los módulos intermedios hay un servomotor que le permite mover la estructura impulsando las ruedas, además, cada módulo cuenta con dos (2) articulaciones con un rango de movilidad de ciento ochenta (180) grados y tiene cuatro (4) grados de libertad (de los cuales las ruedas tienen dos (2) y las articulaciones dos (2)) para generar diferentes tipos de movimientos.

La estructura robótica es escalable en un rango de uno (1) a cinco (5) módulos y cada componente mecánico de la estructura se puede fabricar usando tecnología de impresión 3D. Cada módulo incorpora un sistema de comunicación inalámbrico con un controlador, basado en la tecnología IoT<sup>2</sup> (del inglés Internet of Things). Dicho controlador le da robustez a la plataforma al permitirle al robot seguir funcionando, a pesar de que alguno de los módulos falle [102].

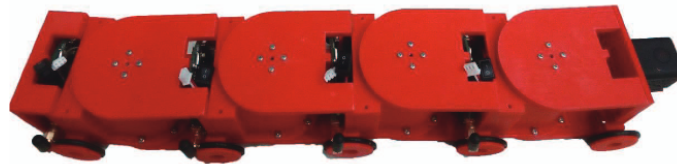


FIGURA 1.3. Robot modular LR (Little Red). Tomado de [102].

### 1.3.1.3. Mecabot

Mecabot 3.0 (Fig 1.4) es un robot modular que tiene la capacidad de configurarse de cuatro (4) maneras: Oruga, Rueda, Serpiente y Hexápodo. Dichas configuraciones se establecen de manera manual conectando los módulos mediante dos (2) tipos de conector. El conector hembra tiene un orificio para encajar al conector macho. Cuando el conector macho encaja, debe girar para terminar de asegurarse al orificio del conector hembra.

Cada módulo incorpora un micro-controlador arduino, un XBee<sup>3</sup> y un acople de potencia para operar al servo-motor. El micro-controlador se encarga de mover el motor de acuerdo a los parámetros de movimiento que llegan al módulo XBee. El módulo XBee establece una conexión entre la estructura robótica conformada y un computador, con el ob-

<sup>2</sup>Es un conjunto de dispositivos que tienen identificadores únicos para conectarse a una red [30].

<sup>3</sup>Los módulos XBee permiten la comunicación inalámbrica entre dispositivos. Estos módulos tienen un protocolo de comunicación por radio frecuencia que permite crear redes punto a punto [46].

jetivo de generar diferentes conjuntos de movimientos usando los programas MATLAB® y Webots. Estos programas permiten estimar los parámetros de control de robot, que una vez calculados, serán transferidos al robot real [39].

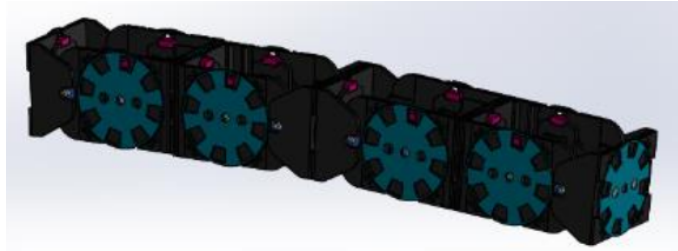


FIGURA 1.4. Robot modular Mecabot. Tomado de [39].

### 1.3.2. Estructura tipo retícula

Este tipo de robot tiene módulos que le permiten conformar estructuras en dos (2) o tres (3) dimensiones. Algunas estructuras siguen un patrón geométrico regular en forma de prismas, moléculas o mallas, como se describe a continuación [12].

#### 1.3.2.1. Odín

El robot modular Odín (Fig. 1.5) se compone de módulos con una jerarquía que depende de la funcionalidad asociada a cada módulo. La funcionalidad de los módulos con nivel jerárquico más bajo, se asocia a los módulos que se usan como articulaciones y se encargan de compartir información únicamente. Los módulos de una jerarquía superior, se usan como soporte en la estructura y se encargan de generar movimientos [48].

La estructura de este robot es re-configurable y varía en función de la distribución de las fuerzas internas al aplicar una fuerza externa. La estructura de los módulos del robot Odín le permiten deformarse para adaptarse a diferentes entornos en vez de cambiar su configuración [48].



FIGURA 1.5. Robot modular Odín. Tomado de [48].

Esta deformación se consigue gracias a que los módulos de soporte son actuadores telescópicos que permiten variar la distancia entre articulaciones. La electrónica incorporada en los módulos de articulación se encarga de distribuir las líneas de alimentación y las señales de control entre los módulos de soporte. El controlador de los módulos de soporte permite manejar un motor sin escobillas (en inglés *brushless*) usando un microcontrolador que, a su vez, se encarga de establecer la comunicación con otros módulos usando el protocolo SPI<sup>4</sup> [48].

### 1.3.2.2. Robot modular con ejes magnéticos

El diseño de este robot modular (Fig. 1.6) incorpora imanes en la estructura de sus módulos. Estos imanes tienen una forma helicoidal y están dispuestos en los bordes del módulo, lo cual le permite unirse con otros formando estructuras 3D. Para lograr diferentes morfologías, el módulo receptor detecta al módulo vecino mediante un sensor de proximidad. Luego, el módulo vecino es desplazado por el módulo receptor hasta la posición deseada. El desplazamiento se consigue usando un mecanismo que desliza al módulo vecino sobre la cara del módulo receptor.

El mecanismo es un conjunto de elementos electro-mecánicos que hace girar los imanes dentro del módulo. El giro de cada imán depende del cálculo de posición angular efectuado por el sistema de control del módulo, cálculo que se realiza teniendo en cuenta el campo magnético detectado por un sensor de efecto hall<sup>5</sup> y la orientación de cada imán. Cuando los módulos se acoplan se establece una comunicación infrarroja, la comunicación permite sincronizar la estructura y planificar movimientos de manera autónoma [75].

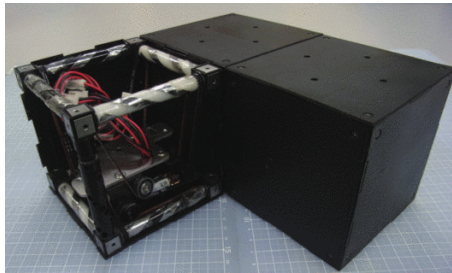


FIGURA 1.6. Robot modular con ejes magnéticos. Tomado de [75].

### 1.3.3. Estructuras híbridas

Los módulos de este tipo de robot combinan características de los robots tipo cadena y retícula, es decir, los módulos pueden agruparse para formar cadenas de módulos o estructuras con un patrón geométrico regular. A continuación, se presentan dos (2) ejemplos de este tipo de robot.

---

<sup>4</sup>SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones que se usa normalmente en dispositivos electrónicos que requieren de una línea de reloj para sincronizar la transmisión de información [88].

<sup>5</sup>Es un efecto que se produce cuando aparece una tensión en un conductor eléctrico al estar expuesto a un campo magnético. En consecuencia, el sensor de efecto hall es un sensor que cambia de estado de acuerdo con el campo magnético detectado [93].

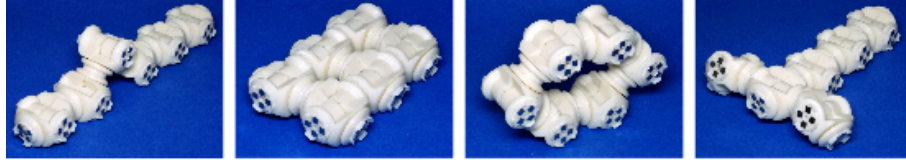


FIGURA 1.7. Robot Modular HyMod. Tomado de [64]

### 1.3.3.1. HyMod

Es un robot modular auto-configurable, en donde cada módulo tiene tres (3) grados de libertad y cuatro (4) conectores hermafroditas. Los conectores le permiten al robot conectar módulos sin importar su orientación para formar estructuras arbitrarias rápidamente. Las estructuras conformadas pueden re-configurarse durante el funcionamiento para cambiar su morfología y el tipo de movimiento realizado (Fig. 1.7).

Cada módulo usa un micro-controlador Teensy 3.2<sup>6</sup> que está conectado a una interfaz CAN<sup>7</sup> (del inglés Controller Area Network) para establecer la comunicación entre módulos. La información procesada por los módulos puede ser enviada a un terminal vía Bluetooth. Para alimentar el circuito electrónico de control y comunicaciones, el robot cuenta con una batería LiPo<sup>8</sup> de 7.4 voltios a 750mAh. Además, posee un sistema de regulación de voltaje para alimentar al controlador de la estructura.

Una variante de este robot integra una serie de herramientas que le permiten manipular objetos. Sin embargo, estas herramientas no funcionan de manera autónoma, sino que son activadas por el usuario desde una aplicación [64].

### 1.3.3.2. HexaMob

El robot modular HexaMob puede formar estructuras en dos (2) y tres (3) dimensiones (Fig. 1.8). Cada módulo de esta plataforma se encuentra dividida en tres (3) secciones llamadas: chasis frontal, chasis trasero y unidad móvil. El chasis frontal, tiene dos (2) garras que giran sobre el mismo eje usando un engranaje. Las garras le dan la posibilidad de unirse con otros módulos y girarlos en ángulos de  $\pm 90$  grados. Para agrupar los módulos, las garras se pueden unir con otras o con los conectores disponibles en las partes laterales del chasis delantero y trasero.

La unidad móvil le permite al chasis frontal girar y el mecanismo para graduar el ángulo de giro se encuentra almacenado en el chasis trasero. La forma de los módulos es hexagonal para reducir la tensión en las garras al conformar una estructura. La estructura cuenta entonces con un sistema de visión artificial que implementa un controlador reactivo, es decir, el robot reconoce y responde únicamente ante los obstáculos del entorno [16].

<sup>6</sup>Es un circuito de control en desarrollo cuyo editor de programas se basa en lenguaje c. Además, el circuito tiene treinta y dos (32) puertos digitales, veintiún (21) entradas análogas y una velocidad de procesamiento de setenta y dos mega-hertz (72 MHz) [66].

<sup>7</sup>Es un estándar de comunicación para la transmisión de mensajes en sistemas de control distribuidos [33].

<sup>8</sup>Es una batería compuesta por un núcleo de iones de litio que puede tener alrededor de mil (1000) ciclos de carga. Además, estas baterías tienen varias celdas para aumentar la capacidad de corriente de descarga [50].



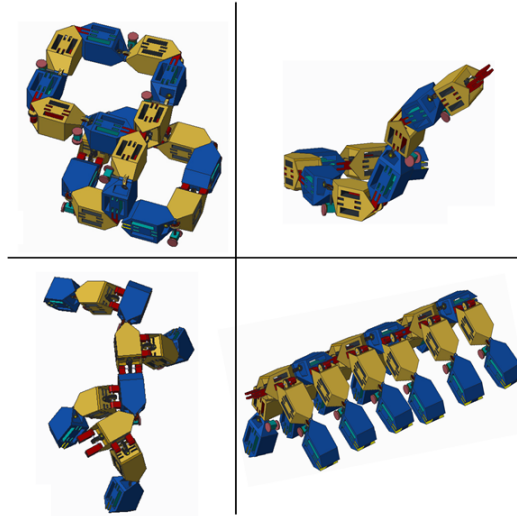


FIGURA 1.8. Robot modular HexaMob. Tomado de [16].

## 1.4. Resumen del Capítulo

En este capítulo se enunció el concepto de robot modular y se clasificó a este tipo de robot de acuerdo con la morfología de su estructura, con ejemplos para cada caso. En los ejemplos se describe el funcionamiento de cada robot y se mencionan sus principales características.

En particular se observa que la morfología de un robot modular puede cambiar al re-configurarse para ejecutar diferentes tipos de movimientos. Sin embargo, En algunos casos la construcción de los módulos no permite que un robot modular sea fácilmente re-configurable, limitando la cantidad de movimientos y morfologías que la estructura robótica puede tener. Esta limitación se ha solucionado parcialmente con el desarrollo del robot modular EMeRGE, que permite construir diversos tipos de estructuras rápidamente. Este robot fue usado durante el desarrollo de este trabajo, por lo tanto, se dedica un capítulo a su descripción (Capítulo 3).

En el siguiente capítulo se complementará la información presentada en este capítulo, mediante la descripción de los sistemas de control y estrategias de adaptación para generar conjuntos de movimientos en robots modulares.

---

---

## Control de movimiento en robots modulares

---

---

### 2.1. Introducción

Existen varias estrategias para controlar los movimientos de un robot modular, estrategias tales como: Tablas de control de movimientos, Cinemática inversa y Generadores sinusoidales [44, 47, 96]. Una manera de implementar dichas estrategias ha sido mediante el uso de simuladores, ya que permiten estimar y transferir conjuntos de parámetros al robot real para ejecutar tareas de locomoción de manera adaptativa [14]. Sin embargo, esta característica limita la autonomía del robot, debido a que la aproximación simulada no tiene en cuenta todas las interacciones físicas del robot con su entorno.

El desarrollo de esta temática ha permitido encontrar soluciones parciales a esta limitación, mediante la construcción de controladores que estiman los conjuntos de movimientos en el robot real. Entre ellos se destacan los controladores basados en Generadores Centrales de Patrones (CPG del inglés, Central Pattern Generator), ya que emulan patrones de movimiento rítmico [18, 72]. Las principales características de las estrategias de control mencionadas y su funcionamiento se describen en este capítulo, el cual está organizado de la siguiente manera; la sección 2.2 presenta características particulares de la locomoción en robots y varias técnicas para generar movimientos usando robots modulares; en la sección 2.3, se presenta cómo se incorporan los CPG en robots modulares; en la sección 2.4, se presentan varias estrategias de navegación y su relación con los robots modulares.

### 2.2. Locomoción en robots

El término *locomoción* se usa para describir la capacidad de los seres vivos para ir de un lugar a otro. Esta capacidad varía dependiendo del medio y puede clasificarse como: acuática, terrestre o aérea. En particular, la locomoción terrestre puede realizarse mediante patas (animales bípedos o cuadrúpedos) o movimientos corporales (caracol, babosa, oruga) [91].

De manera similar, el término locomoción se ha usado en robótica para designar mecanismos que le permitan a un robot desplazarse de un lugar a otro. Algunos de estos mecanismos tienen ruedas, orugas o patas, y se han incorporado en diferentes plataformas

robóticas, pero tienen ciertas limitaciones. Un ejemplo de estas limitaciones se observa en los robots de dos (2) ruedas lisas, los cuales al intentar adaptarse a terrenos irregulares no pueden recuperar una orientación que les permita avanzar en caso de volcarse. Esta limitación ha sido solucionada parcialmente cambiando las ruedas lisas por un mecanismo de desplazamiento basado en orugas o ruedas omnidireccionales. Sin embargo, aún no se ha solucionado en su totalidad [17, 22].

Otra posible solución para esta limitación ha sido explorar morfologías y mecanismos de locomoción para la manufactura de robots. Un caso particular son los robots modulares ápodos<sup>1</sup>, que tratan de imitar la locomoción animal usando movimientos corporales y cambian de forma para adaptarse al terreno en el que se desplazan. Entre los movimientos que realiza este tipo de robot se tienen: movimiento sinusoidal, giratorio, de escalar y en espiral [3, 19, 34]. La generación de este tipo de movimientos dependen del control y la coordinación de la estructura robótica conformada, ya que todas las partes móviles y articulaciones deben sincronizarse para que el robot se desplace [90, 95].

La forma de generar movimientos en robots modulares apodos se ha generalizado, para controlar robots modulares con estructuras arbitrarias. Se han construido dos tipos de sistemas de control y varias técnicas para generar movimientos de manera coordinada usando este tipo de robot, las cuales se presentan a continuación.

### 2.2.1. Sistemas de control

Los sistemas de control o controladores son “un conjunto de dispositivos encargados de administrar, ordenar, dirigir o regular el comportamiento de otro sistema” [60]. En robots modulares, el controlador se encarga de establecer las acciones necesarias para que el robot realice diferentes tareas y se clasifican como centralizados o descentralizados.

#### 2.2.1.1. Control centralizado

En controladores de este tipo todas las acciones que realizan los módulos dependen de un único dispositivo, que puede ser un módulo específico de la estructura o un dispositivo externo. Al tener un solo dispositivo de control es fácil de modificar la estrategia de control cuando sea necesario. Sin embargo, una limitación de este tipo de controlador es que al presentarse un fallo en dicho módulo, la estructura completa falla [59, 71].

#### 2.2.1.2. Control des-centralizado

Este tipo de controlador se caracteriza por definir uno (1) o varios módulos como coordinador o maestro, quienes se encargan de controlar a los demás módulos. Esta característica soluciona parcialmente la limitación de los controladores centralizados, debido a que la estructura puede fallar solamente si todos los módulos maestros fallan a la vez [73, 103].

---

<sup>1</sup>El término ápodo, se usa para describir una forma de vida carente de patas que se desplaza usando movimientos corporales [24].

## 2.2.2. Técnicas de control de movimiento en robots modulares

Las técnicas de control de movimientos se han usado en varios tipos de robot, para generar esquemas de control que le permitan realizar diferentes tareas. Gracias a estas técnicas, existe una gran cantidad de aplicaciones desarrolladas por diferentes autores. La relación entre los robots modulares y los aportes esenciales de estas aplicaciones se describe a continuación.

### 2.2.2.1. Tablas de control de movimiento

Los movimientos generados por un robot articulado usando tablas de control de movimiento (del inglés *gait control tables*) se consigue pre-estableciendo la posición de las articulaciones. Las posiciones de las articulaciones se encuentran almacenadas en vectores, cuyo tamaño depende de la cantidad de módulos que conforman la estructura. Los vectores se anidan en forma de lista para definir una secuencia de movimientos de la estructura robótica, que es recorrida por el controlador con el fin de enviar las posiciones de los actuadores a cada módulo de la estructura. Cuando una posición de la lista es enviada, el controlador espera un tiempo para enviar la siguiente posición y repite esta acción hasta recorrer la lista completa. Al terminar de recorrer la lista vuelve a empezar para generar un movimiento cíclico o repetitivo.

Esta estrategia fue usada con éxito por Mark Yim en su robot Polypod [96]. Al incorporar esta estrategia el robot realizó movimientos con diferentes configuraciones de módulos, tales como: circular, en forma de S y araña. También mostró que las tablas de control de movimiento pueden ser usadas sin importar la cantidad de módulos que tenga la estructura.

Una variante de este tipo de controlador se presenta en el trabajo de José Baca [7]. Este trabajo presenta como incorporar al robot MoDRED un controlador difuso, el cual estima un valor de salida dependiendo del error medido. El error es la diferencia entre la posición actual del módulo (medida por un sensor de orientación) y la posición esperada. El resultado es procesado por el controlador para generar un valor de salida que le permita al robot reducir el margen de error estimado y llegar a la posición deseada.

La salida del controlador es un código que tiene una equivalencia dentro de la tabla de control de movimientos. Cada vez que varía el código el robot genera un movimiento diferente. Cada módulo incorpora su propio controlador difuso y funciona de manera independiente. Una ventaja de esta estrategia es la reducción en el tiempo de búsqueda de los parámetros de movimiento, ya que no se realiza de manera secuencial, sino que el controlador busca el movimiento más adecuado dependiendo de la situación.

### 2.2.2.2. Cinemática inversa

La cinemática inversa se ha usado para determinar la posición de una cadena de articulaciones a partir de una expresión matemática, con el fin de ubicar el actuador final en la posición deseada. Normalmente se utiliza un software para determinar la posición de las articulaciones, debido a la cantidad de configuraciones que puede tener el robot. Por ejemplo, el trabajo realizado por John Vera presenta un sistema de control para un brazo robótico Mitsubishi® de cinco (5) grados de libertad [47]. El brazo es controlado desde una aplicación desarrollada en MATLAB®, que convierte las diferentes rutas tra-

zadas en el espacio de trabajo simulado al sistema de coordenadas por defecto del robot. Las coordenadas son enviadas usando un DSP<sup>2</sup> (del inglés de Digital Signal Processor) programado en Code Composer Studio®.

La cinemática inversa también ha sido usada para estimar el movimiento de las articulaciones de un robot modular. Por ejemplo, Mario Andrei propone un robot tipo serpiente cuyo movimiento depende de un conjunto de coordenadas, que son determinadas y enviadas a cada uno de los módulos usando un computador. Una limitación de esta estrategia es que, al aumentar la cantidad de grados de libertad del robot, aumenta la dificultad para realizar un modelo matemático que permita generar un conjunto de movimientos [20].

### 2.2.2.3. Generadores Centrales de Patrones

Los CPGs emulan el funcionamiento de un conjunto de neuronas biológicas especializadas, para generar diferentes patrones rítmicos, imitando el control de movimiento de los músculos de un animal durante la realización de ciertas actividades como respirar, caminar, masticar, entre otras [72].

Existen varios modelos de CPG: Modelos bio-físicos, Modelos de conexiones o Sistemas de osciladores acoplados. Los modelos bio-físicos se basan en el modelo de neurona de Hodgkin-Huxley, los cuales representan mediante ecuaciones diferenciales ordinarias, no lineales, los potenciales de acción en las neuronas; los modelos de conexiones, tratan de emular la sinapsis entre neuronas y su comportamiento con el medio que la rodea [32].

Por otro lado, los modelos de sistemas de osciladores acoplados son grupos de osciladores presentes en diferentes sistemas, que se sincronizan para reproducir un movimiento coordinado. Un ejemplo de estos es el sistema de dos masas unidas con resortes (Fig. 2.1), cuyo comportamiento se simuló en una aplicación que implementa el método de Runge Kutta<sup>3</sup> [60].

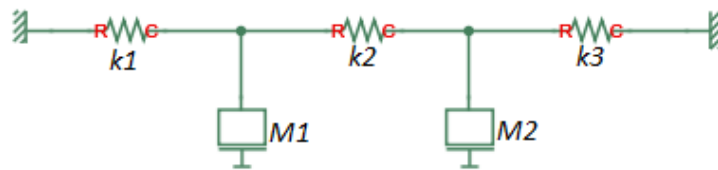


FIGURA 2.1. Sistema de dos (2) masas acopladas mediante resortes.

El sistema en general (Fig. 2.2) tiene una frecuencia característica de oscilación (determinada por el valor de la constante del resorte  $k_2$ ) y las masas ( $M_1$  y  $M_2$ ) pueden oscilar de manera independiente con una frecuencia ajustable en función del valor de la constante de los resortes ( $k_1$  y  $k_3$  respectivamente).

El comportamiento descrito se puede observar en otros tipos de osciladores acoplados, tales como: Van Der Pol [63] y FitzHugh-Nagumo [91]. El modelo simplificado de tipo

<sup>2</sup>Es un termino que se usa para designar mecanismos que se encarguen del procesamiento digital de señales [53].

<sup>3</sup>El método de Runge Kutta y Euler; Son métodos numéricos que se usan para encontrar una solución aproximada a una ecuación diferencial de manera iterativa [28]

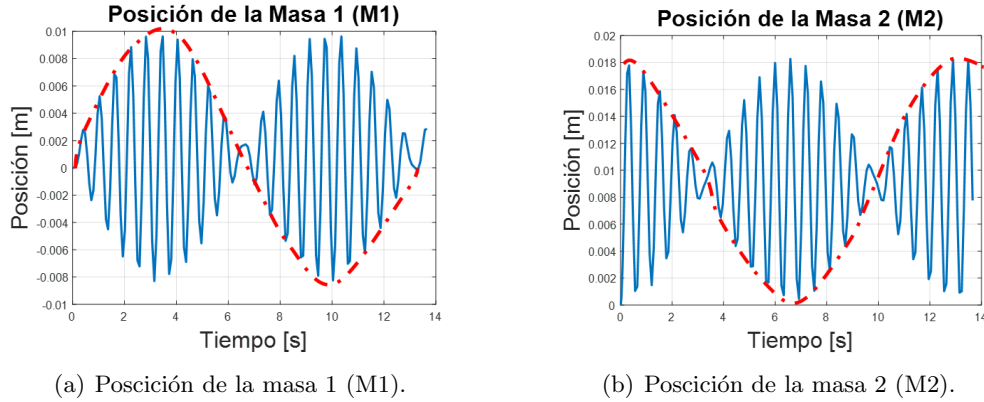


FIGURA 2.2. Respuesta de un sistema acoplado de masa-resorte con los siguientes parámetros  $M_1 = M_2 = 1kg$ ,  $K_1 = 10N/m$ ,  $K_2 = 100N/m$  y  $K_3 = 10N/m$ . La frecuencia de oscilación del sistema acoplado está marcado por la línea punteada y la frecuencia de cada sistema independiente es la línea continua.

FitzHugh-Nagumo está representado por las ecuaciones 2.1 y 2.2.

$$\dot{x}_i = c \left( y_i + x_i + \frac{x_i^3}{3} + f_{ci} \right) \quad (2.1)$$

$$\dot{y}_i = -\frac{x_i - a + by_i}{c} \quad (2.2)$$

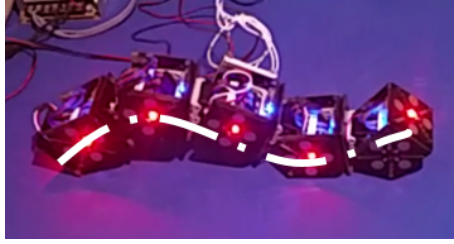
En las expresiones,  $x_i$  es el potencial de membrana de la  $i$ -ésima neurona,  $f_{ci}$  es la señal de conducción e  $i$ ,  $a$ ,  $b$  y  $c$  son parámetros que dependen de la cantidad de osciladores y del tipo de sistema. En algunos casos, la salida de este oscilador se calcula usando métodos de integración numérica [40].

Los rasgos biológicos de este tipo de oscilador se han tomado de algunos modelos experimentales. Uno (1) de ellos se obtuvo a partir de la experimentación con una lamprea<sup>4</sup> [18]. En este estudio se encontró un modelo simplificado de neurona con osciladores acoplados en diferencia de fase, que explica la relación entre la coordinación entre neuronas y la locomoción de la lamprea. Este modelo permite realizar un análisis de coordinación de los movimientos generados por un CPG, a partir del efecto que produce el cambio en la frecuencia de oscilación y la diferencia de fase [89].

#### 2.2.2.4. Generadores sinusoidales

Los generadores sinusoidales no son osciladores acoplados. El controlador es un oscilador que controla toda la estructura robótica y determina la posición de sus articulaciones directamente. Se puede decir que estos generadores son una generalización de los osciladores acoplados cuando se sincronizan y tienen un comportamiento similar al de un oscilador de frecuencia fija. Una desventaja de esta estrategia es que no es fácilmente implementable en robots modulares con estructuras arbitrarias. Sin embargo, ha sido usado con éxito en robot tipo serpiente para generar movimientos locomotores y

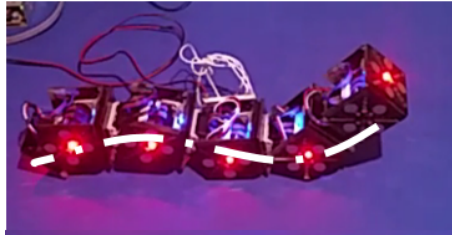
<sup>4</sup>La lamprea es un tipo de pez con una forma similar a la de una anguila, no tienen escamas y tienen un cuerpo resbaloso [18].



(a) Fase del ciclo de movimiento 1.



(b) Fase del ciclo de movimiento 2.



(c) Fase del ciclo de movimiento 3.

FIGURA 2.3. Respuesta de la estructura robótica EMeRGE durante un ciclo de marcha hacia adelante usando un controlador sinusoidal.

adaptarse a entornos con obstáculos [44].

Esta técnica también se ha usado para controlar las patas de un robot hexápodo mediante un generador sinusoidal. El generador calcula la posición de los extremos de las patas y el resultado se envía al controlador. El controlador incorpora un PID<sup>5</sup> determina la posición de cada actuador, al reducir el margen de error entre la posición actual y el valor calculado [76]. Este generador se implementó en la plataforma robótica EMeRGE (del inglés Easy Modular Embodied Robot Generator), donde el primer módulo de la plataforma es el coordinador y es el encargado de estimar las posiciones de los demás módulos (Fig. 2.3). La expresión matemática incorporada en el controlador se presenta en la ecuación 2.3, donde  $x$  es la salida de cada módulo,  $\theta$  es un ángulo que varía con respecto al tiempo y  $\phi$ , es la diferencia de fase que se asigna a cada módulo.

$$x = A * \sin(\theta * t + \phi) \quad (2.3)$$

A pesar de que este generador de movimientos funcionó correctamente, se optó por implementar una técnica basada en CPG para que el robot tenga la posibilidad de moverse al conformar cualquier tipo de morfología.

### 2.3. Aplicaciones de los CPGs en robótica

El controlador basado en CPGs tiene una ventaja sobre los movimientos realizados usando cinemática inversa, ya que los CPGs no necesitan del modelo aproximado del sistema a controlar para estimar un valor de salida. Al reducir la cantidad de expresiones matemáticas necesarias para incorporar un mecanismo de control en un robot, es posible ajustar el valor de salida del CPG para que funcione en diversas estructuras robóticas.

<sup>5</sup>PID (Controlador Proporcional Integral Derivativo) es una técnica de control re-alimentado que corrige el margen de error a la entrada al variar la salida [60].

Los CPG se han aplicado en robots con patas y su influencia ha permitido el desarrollo de controladores para robots modulares [90]. Algunos ejemplos de robots que usan un CPG para desplazarse son:

- **Bípedos:** Este tipo de robot imita los movimientos corporales de las piernas durante un ciclo de marcha. El movimiento de las piernas este ligado a dos (2) articulaciones: una a nivel de la cadera y la otra a nivel de la rodilla. Para iniciar una caminata, el robot detecta la pose corporal actual y sincroniza los CPG de acuerdo con esta información [74].
- **Cuadrúpedo:** Este tipo de robot incorpora un sistema de osciladores acoplados para controlar cada una de sus patas y el movimiento de cada pata se sincroniza de acuerdo con la posición actual del robot. Esta ventaja le permite recorrer terrenos irregulares, aumentando o disminuyendo la longitud del paso [4].
- **Hexápodo:** Este tipo de robot tiene seis (6) patas para aumentar su capacidad de adaptarse a terrenos irregulares. Un ejemplo de robot hexápodo es el robot Hexapod, que incorpora un oscilador Van Der Pol para generar diferentes movimientos en las patas bi-articuladas. Los osciladores se sincronizan para realizar un movimiento coordinado. Los movimientos que la estructura puede realizar son: girar, avanzar o retroceder [98].
- **Serpiente:** Este tipo de robot puede generar movimientos rítmicos usando un CPG o un generador sinusoidal, en ambos casos, se sincronizan los módulos a partir de una diferencia de fase preestablecida. En el caso de los CPG, cada módulo comparte su posición actual y los módulos vecinos estiman su salida a partir de este valor. En el caso del generador sinusoidal, el controlador envía un conjunto de coordenadas a cada módulo. Dichos controladores permiten realizar un movimiento periódico en forma de “S” para que la estructura se desplace de un punto a otro [19, 100].
- **Modular de estructura arbitraria:** Este tipo de robot normalmente incorpora un CPG basado en osciladores acoplados para generar diferentes movimientos. En algunos casos debido a la complejidad de estas estructuras, la posición de cada módulo se determina usando un computador y no el controlador propio de la estructura robótica [21].

## 2.4. Estrategias de navegación

Una estrategia de navegación es un conjunto de operaciones que permiten guiar a un robot a través de su entorno. Algunas tareas que se ejecutan durante la navegación de un robot son: la percepción, la planificación y la operación. La percepción, es la forma del robot para crear una representación del mundo que lo rodea; la planificación, es la capacidad del controlador para estimar una trayectoria en el entorno del robot; y la operación, es el conjunto de acciones que le permiten al robot recorrer la trayectoria estimada [22]. Algunos ejemplos en los cuales se implementa este tipo de estrategias son:

- **Control reactivo:** Se basa en el esquema de percepción-acción, que le permite al robot tomar la información de percepción sensorial y realizar movimientos según la información recolectada. Una característica de este tipo de controlador es que la



información de percepción sensorial no requiere un procesamiento complejo, lo que reduce el tiempo de procesamiento para acelerar la velocidad de respuesta del robot frente a un obstáculo [17].

- **Grafos de visibilidad:** Representa los obstáculos del entorno mediante polígonos, en entornos planos bi-dimensionales. En estos entornos el concepto de visibilidad se utiliza para trazar una ruta que no intercepte con obstáculos, usando segmentos rectilíneos. La ruta se estima usando un algoritmo de búsqueda que permite encontrar el recorrido más corto entre dos (2) puntos [78].
- **Diagramas de Voronoi:** Representa el entorno de la misma forma que los grafos, sin embargo, busca la ruta más larga entre dos (2) puntos y el trazo se realiza usando líneas curvas y rectas [31].
- **Descomposición en celdas:** Divide el espacio libre entre los obstáculos del entorno en celdas e incorpora un algoritmo para encontrar un conjunto de celdas que una dos (2) puntos sin discontinuidades [37].
- **Campos potenciales:** Traza una ruta asumiendo que el robot es una partícula atraída por la posición deseada y repelida por los obstáculos [42].

Las tareas de navegación en robots móviles se pueden ejecutar de manera global o local [31]. Se denomina global cuando un robot sigue una ruta trazada por otro dispositivo. Se denomina local cuando el controlador tiene la capacidad de trazar la ruta que el robot debe seguir. Estas estrategias de navegación se han implementado en sistemas de múltiples robots [86] y en robots modulares [101], sin embargo, esta temática aún se encuentra en exploración.

Los simuladores son una opción para la planificación de trayectorias en robots modulares. Una característica particular de los simuladores es que permiten recrear las condiciones del entorno, la configuración del robot y el comportamiento de ciertos sensores. Los resultados proporcionados por el simulador muestran la posición de los módulos, los diferentes movimientos que pueden generar dependiendo de la configuración y registros del comportamiento de los sensores [43].

Una variante incorporada a los simuladores enlaza al robot modular con el simulador, transfiriéndole al controlador del robot, los parámetros necesarios para que la estructura robótica tenga la capacidad de realizar los movimientos estimados en el simulador [45]. Otra característica de los simuladores es la posibilidad de modelar estructuras robóticas modulares re-configurables con morfología cambiante durante su funcionamiento, aumentando la cantidad de comportamientos disponibles para analizar los movimientos de tipo de robot. Una ventaja de los entornos simulados, es que permiten visualizar el comportamiento del robot antes de implementarlo y pueden llegar a ahorrar costos durante su manufactura [56].

Una limitación de la estrategia simulada es que no le permite al robot adaptarse a entornos cambiantes pues el simulador estima un conjunto de movimientos y el robot los ejecuta sin tener una re-alimentación sobre las condiciones ambientales. Dicha limitación se ha solucionado parcialmente incorporando al controlador del robot un sistema de visión artificial que le permita re-alimentar al computador y actualizar los parámetros de configuración del robot, de acuerdo con la posición en la que se encuentre [55].

La configuración o cantidad de grados de libertad de un robot modular depende de los módulos asociados a su estructura. Esta característica hace que el espacio de búsqueda de movimientos no tenga un tamaño determinado, sino que, las dimensiones del espacio de

búsqueda varían en función de la cantidad de módulos del robot. Debido a la variabilidad del espacio de búsqueda se han propuesto estrategias para encontrar configuraciones adecuadas del robot al realizar una tarea, o conjuntos de movimientos para que el robot se adapte a su entorno [83, 92].

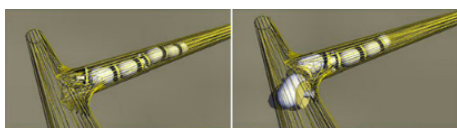
Dichas estrategias se pueden ejecutar por el controlador de un robot fuera de línea (en inglés *Off-line*) o en línea (en inglés *On-line*). Se dice en línea, cuando el controlador está incorporado en la estructura robótica y tiene la capacidad de ejecutar la estrategia de búsqueda y se dice fuera de línea, en el caso contrario. Es decir, es fuera de línea cuando la estrategia de búsqueda se ejecuta en un dispositivo externo al robot que normalmente es un computador, el cual envía los parámetros de control al robot [36]. Algunos ejemplos de la incorporación de estas estrategias son:

- El robot ATRON usa un algoritmo genético que funciona fuera de línea, para determinar la combinación de módulos más adecuada para realizar una tarea asignada [61].
- El robot M-TRAN incorpora un controlador fuera de línea para realizar la re-configuración de su estructura, a partir de los parámetros estimados usando un computador [58].

Los ejemplos mencionados anteriormente permiten re-configurar la estructura del robot únicamente. Esta característica le permite al robot moverse, a pesar de que no tenga en cuenta la interacción con su entorno. Sin embargo, existen simuladores que permiten encontrar algunos parámetros de control del robot, asumiendo un entorno virtual como el entorno que el robot real va a recorrer. En particular se destaca el trabajo propuesto por Brunete para optimizar los parámetros de control del robot Microtub (Fig. 2.4(a)) fuera de línea. El robot Microtub es un robot modular tipo cadena con dos (2) grados de libertad en cada módulo. Este robot se simuló en un programa de virtualización de entornos 3D (Fig. 2.4(b)). El entorno y el robot fueron digitalizados por los autores, con el fin de encontrar conjuntos de parámetros para modular el CPG usando un algoritmo genético y así lograr que el robot se desplace de un punto a otro [14].

El algoritmo genético toma agrupa las variables de entrada: amplitud, fase, desplazamiento pre-definido (conocido como offset), velocidad de desplazamiento y diferencia de fase, para formar un individuo. El valor de aptitud (variable de salida) asociado a cada individuo es la distancia recorrida durante una simulación, para cada individuo se ejecuta una simulación diferente con la misma configuración del robot y del entorno.

Las características particulares del algoritmo genético implementado son: cada individuo de la población esta codificado en binario, se realizó una selección de individuos usando el método ruleta, un cruce de punto simple y la mutación varía un gen del cromosoma de cada individuo. El algoritmo genético requiere veinticinco (25) generaciones para



(a) Robot Modular Microtub.



(b) Entorno virtual del robot Microtub.

FIGURA 2.4. Representación del robot modular Microtub. Tomado de [14].

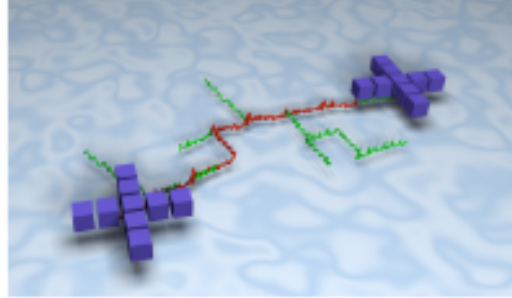


FIGURA 2.5. Ruta trazada para un cuadro-podo usando el algoritmo RRT. Tomado de [85].

encontrar una solución a cada problema propuesto y la mejor de todas las soluciones es la combinación de parámetros que se envían al controlador del robot [14].

Los algoritmos genéticos no son los únicos usados para optimizar los parámetros de movimiento de robots modulares de manera simulada, también se han usado otras técnicas de optimización tales como: optimización multi-objetivo [92] y enjambres de partículas [35]. Un ejemplo de este tipo de técnicas implementada se puede observar en el trabajo presentado por Vonasek, en el cual se propone el algoritmo llamado RRT (del inglés *Rapidly Exploring Random Tree with Motion primitives*) para crear conjuntos de movimientos a partir de movimientos predefinidos. Los conjuntos de movimientos le dan al robot la posibilidad de seguir una ruta trazada previamente usando un simulador (Fig. 2.5) [85]. Recapitulando, en la Tabla 2.1 se presenta una síntesis de las características mostradas hasta ahora de los robots modulares y sus posibles formas de control.

TABLA 2.1. Clasificación y características de algunos robots modulares.

Estructura	Nombre	Tipo	Controlador	Estrategia de adaptación	Cita
Cadena	Millibot	No Auto-configurable	Descentralizado	Cinemática inversa	[13]
Cadena	Polybot	Auto-configurable	Descentralizado	Tablas de Movimiento	[95]
Cadena	Robot Serpiente	No Auto-configurable	Centralizado	Generador Sinusoidal	[69]
Cadena	OSMOS	No Auto-configurable	Centralizado	Cinemática inversa	[70]
Retícula	ATRON	Auto-configurable	Descentralizado	Cinemática inversa	[11]
Retícula	EM-cube	No Auto-configurable	Centralizado	CPG	[3]
Retícula	Telecubes	Auto-configurable	Descentralizado	CPG	[73]
Retícula	Mu-cubes	Auto-configurable	Descentralizado	CPG	[103]
Híbrida	ULGEN	Auto-configurable	Descentralizado	Cinemática inversa	[25]
Híbrida	Neurobot	No Auto-configurable	Centralizado	Generador Sinusoidal	[26]
Híbrida	Roombot	Auto-configurable	Centralizado	CPG	[59, 71]

## 2.5. Resumen del capítulo

A partir del concepto de locomoción se enunciaron varias técnicas para controlar y coordinar las articulaciones de un robot modular. Dentro de esas técnicas existen algunas que intentan imitar las características propias de la locomoción animal, aun así, en este trabajo se hizo énfasis en los controladores de robots que incorporan CPGs. Como se mencionó antes, el controlador basado en CPG tiene ciertas ventajas y una de ellas es que funciona en diversos tipos de robot sin importar su configuración.

Los parámetros de configuración de los CPGs se han modulado usando varios tipos de algoritmos de búsqueda. Algunos de ellos se ejecutan fuera de línea, debido a la cantidad de

---

evaluaciones que se deben realizar para encontrar un conjunto de movimientos apropiado. Sin embargo, esta característica limita la autonomía del robot, ya que no se re-alimenta al controlador del robot en tiempo real sobre su interacción con el entorno, lo que impide que se adapte a un entorno cambiante.

En el siguiente capítulo se describe plataforma robótica EMeRGE y se presenta el diseño electrónico de un controlador implementado en este trabajo, el cual funciona en línea y encuentra conjuntos de movimientos para que el robot se desplace en entornos con obstáculos fijos o variables.

---

---

## Robot modular EMeRGE

---

---

### 3.1. Introducción

Como se mencionó en el Capítulo 1, existen diversos tipos de robot modulares, sin embargo, algunos no adoptan morfologías arbitrarias fácilmente, debido a que no tienen una estructura flexible. Una iniciativa que intenta solucionar esta limitación se propone mediante el desarrollo del robot modular EMeRGE (del inglés Easy Modular Embodied Robot Generator), el cual permite formar diferentes morfologías usando módulos homogéneos. Cada módulo de la estructura tiene su propio controlador, que se sincroniza con otros módulos o dispositivos usando el protocolo de comunicación CAN.

Las principales características del robot modular EMeRGE serán descritas en este capítulo, que se estructura de la siguiente forma: En la sección 3.2 se hace una descripción general del robot. En la sección 3.3 se describe prototipo del sistema de control. En la sección 3.4 se muestran los principales componentes electrónicos del sistema de control. En la sección 3.5 se describe la conectividad entre módulos y en la sección 3.6 se presentan algunas limitaciones del robot EMeRGE.

### 3.2. Descripción general del robot

EMeRGE (Fig. 3.1) es un robot modular tipo cadena, con el cual se construyen estructuras arbitrarias rápidamente gracias al diseño de sus módulos. Los módulos tienen un (1) grado de libertad y cuatro (4) caras para conectarse con otros módulos o dispositivos. Tres (3) caras tienen un conector hembra y se ensamblan usando un soporte (del inglés, *bracket*) en forma de "U" que está ajustado al eje del servomotor<sup>1</sup>. La cara macho se une a la carcasa del servomotor con un soporte plano que se encuentra al extremo contrario de su eje [45].

Cada conector tiene cuatro (4) imanes de neodimio, que varían su polaridad dependiendo del tipo de conector. Bajo cada conector se encuentra un circuito impreso que está conectado a los demás para compartir señales eléctricas. Estas señales recorren toda la

---

<sup>1</sup>Es un tipo de motor que tiene su propio controlador para ajustar sus valores de posición y velocidad [97].

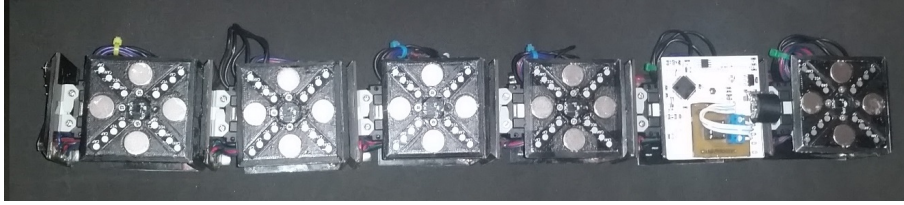


FIGURA 3.1. Morfología de tipo serpiente ensamblada usando la plataforma Robótica EMERGE.

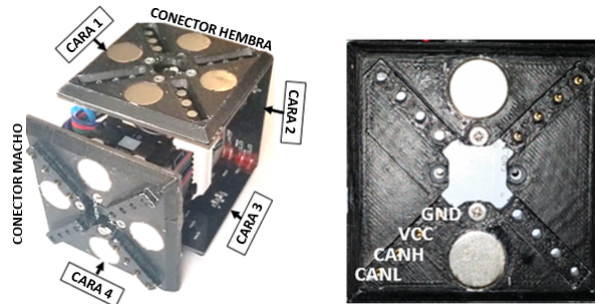


FIGURA 3.2. Vista frontal de un módulo de la plataforma EMERGE y distribución de los pines para realizar una conexión con otros módulos.

estructura robótica con ayuda cuatro (4) de pines con resorte que se encuentran en la cara macho y conectan los módulos al entrar en contacto con los parches de cobre (superficie de forma circular de 3 mili-metros de radio) de la cara hembra.

Los parches y los pines están distribuidos de manera simétrica, de tal manera que los módulos se pueden conectar sin tener en cuenta su orientación. Además, la forma del conector y el diseño de los circuitos impresos permite conectar dispositivos diferentes a los módulos en la estructura robótica conformada. Teniendo en cuenta esta característica se incorporó al robot un sistema de control, el cual se describe detalladamente a continuación.

### 3.3. Arquitectura del sistema de control

Como se mencionó antes, cada módulo (Fig. 3.2) tiene un circuito impreso, el cual se diseñó con diferentes componentes electrónicos para controlar los movimientos del módulo de manera local. El diseño de este circuito se generalizó para construir varios dispositivos externos, que se acoplan a los módulos y determinan características del ambiente o generan esquemas de movimiento para la estructura robótica conformada. Es decir, el sistema de control implementado se divide en dos partes las cuales se describen a continuación.

#### 3.3.1. Controlador local de cada módulo

El controlador de cada módulo es un circuito electrónico (Fig. 3.3) que se compone de varios grupos de circuitos independientes, los cuales están dispuestos para funcionar de manera coordinada y se alimentan con una fuente externa. Estos circuitos independien-

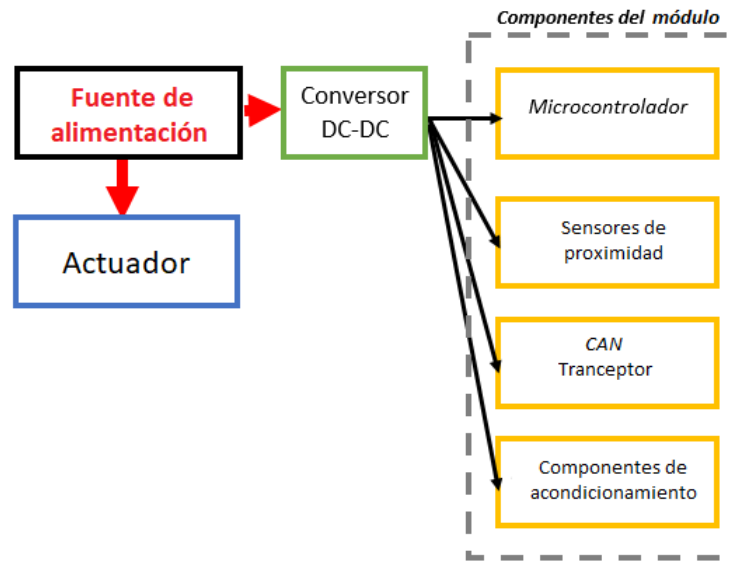


FIGURA 3.3. Representación de las conexiones eléctricas que alimentan diferentes elementos del módulo.

tes están compuestos por conjuntos de componentes electrónicos pasivos<sup>2</sup> (indicadores luminosos, resistencias, condensadores y bobinas) que son ensamblados para asegurar el funcionamiento de los componentes principales del módulo, que son; el micro-controlador, los sensores de proximidad, el tranceptor<sup>3</sup> (del inglés *transceiver*) de CAN y el servomotor. El servomotor es el único componente que requiere una tensión eléctrica de doce voltios (12 V) para funcionar, por lo tanto, se conecta directamente a la fuente externa en paralelo al circuito impreso. Los demás componentes necesitan una tensión de trabajo de 3.3 voltios que se consigue implementando un convertidor DC-DC.

Este tipo de convertidor es una fuente conmutada de tipo reductor, que se encarga de regular la tensión eléctrica de la fuente de alimentación combinando diferentes componentes electrónicos pasivos. Frente a una fuente de tipo lineal, la fuente conmutada tiene ciertas ventajas funcionales, ya que, entrega la misma cantidad de corriente que una fuente lineal con una eficiencia mayor al setenta por ciento (70 %) y puede mantenerse en la región estable sin sobrecalentamiento [77].

Estas características son convenientes teniendo en cuenta que, el circuito de control local de un módulo consume un máximo de cuatrocientos mili-amperios (400 mA) durante la ejecución de una tarea. Estos valores se encuentran en el rango de operación normal del convertidor (el valor máximo de operación es de 2.5A), lo que garantiza un funcionamiento continuo del módulo, ya que, la tensión de trabajo se mantiene estable.

Otro aspecto importante de los componentes principales es que las conexiones entre ellos y el micro-controlador (de referencia PSOC 4200M) se establecen directamente, es decir, no hay componentes electrónicos de acople entre ellos (Fig. 3.4). Esta característica y la arquitectura flexible del micro-controlador permite que cada componente principal sea definido mediante un conjunto de funciones dentro de su rutina de control. Al definir

<sup>2</sup>Un componente electrónico pasivo es un elemento que necesita de una fuente de alimentación para funcionar, por ejemplo; resistencias, condensadores y bobinas [51].

<sup>3</sup>Un tranceptor es un dispositivo que se usa para acoplar las señales de comunicación entre dispositivos, es decir, sirve como interfaz electrónica entre el micro-controlador y el bus de transmisión de datos [77].

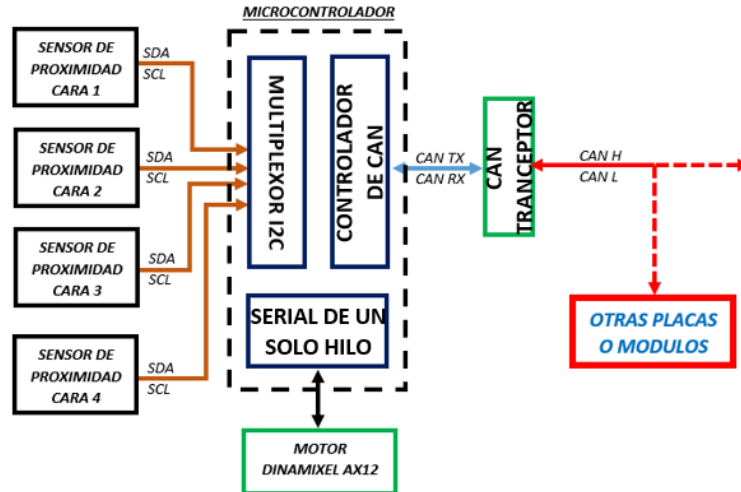
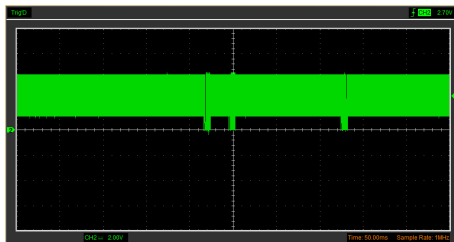


FIGURA 3.4. Representación de las conexiones eléctricas de control.

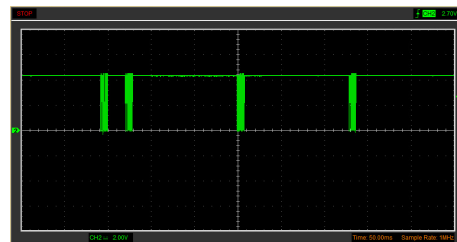
el conjunto de funciones se gestiona el intercambio de información entre los componentes principales y el micro-controlador usando varios protocolos de comunicación, que incorporen comandos de lectura y escritura pre-definidos en la información técnica de cada uno de ellos.

Además, de la información técnica se tomaron algunas sugerencias de los fabricantes para establecer los parámetros de conexión de los circuitos independientes de cada componente principal. Sin embargo, se realizaron algunos ajustes los cuales se describen a continuación:

1. Incorporar un condensador y un diodo en paralelo a la fuente de alimentación, para filtrar los armónicos que se producen durante el funcionamiento del servomotor, ya que, estos armónicos modifican el valor de posición angular del servomotor generado por el micro-controlador (Fig. 3.5).



(a) Señal de control sin filtro.



(b) Señal de control con filtro.

FIGURA 3.5. Comportamiento de la señal de control del servomotor.

2. Instalar un conmutador para cambiar el modo de funcionamiento del módulo y en consecuencia de la estructura robótica (Fig. 3.6). El modo A 3.6(a) se utiliza para que la estructura sea controlada desde un computador usando un adaptador



USB2Dynamixel<sup>4</sup>. El modo B 3.6(b) permite el intercambio de información entre los micro-controladores de la estructura robótica conformada.

3. Eliminar el circuito de programación del micro-controlador establecido por defecto y se acondiciono uno (1) usando un JTAG (del inglés *Joint Test Action Group*, Fig. 3.6(c)) para la programar los módulos vía USB<sup>5</sup>.

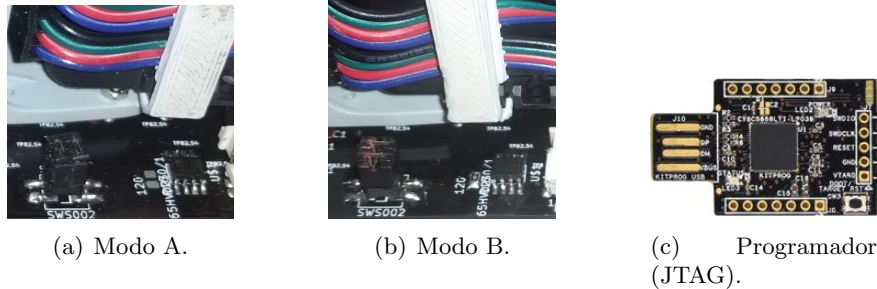


FIGURA 3.6. Ubicación de los puentes según el modo de funcionamiento y JTAG.

### 3.3.2. Dispositivos externos o accesorios

Como se mencionó antes, a partir de una modificación hecha del circuito de control de cada módulo y su ensamblaje sobre un conector macho, se crearon varios accesorios que se pueden conectar a la estructura robótica conformada. Estos accesorios le permiten al robot; generar esquemas de movimiento, medir distancia entre él y un obstáculo, conocer su orientación o comunicarse con un computador a través de una interfaz inalámbrica. En total se realizaron tres (3) dispositivos los cuales se describen a continuación:

- Dispositivo de comunicaciones: Este dispositivo (Fig. 3.7(a)) funciona como un analizador de protocolos o sniffer<sup>6</sup> de CAN para enviar la información de control de los módulos del robot a un computador. Esto se logra usando el micro-controlador como interfaz entre el bus de datos de los módulos y el XBee (módulo de comunicaciones), ya que, el micro-controlador almacena temporalmente la información que está presente en el bus de datos y posteriormente la envía al computador mediante el XBee.
- Dispositivo de medición de orientación: Este dispositivo (Fig. 3.7(b)) incorpora un acelerómetro (de referencia Adxl335 fabricado por ANALOG) que está conectado a un micro-controlador, el cual determina la orientación del robot a partir de los valores proporcionados por el sensor.
- Dispositivo de medición de distancia: Este dispositivo (Fig. 3.7(c)) mide la distancia entre un obstáculo y el robot usando un sensor de ultrasonido (de referencia SRF01

<sup>4</sup>Es una interface electrónica que permite operar los servomotores desde un computador usando la aplicación de configuración de DINAMIXEL [67].

<sup>5</sup>Se conoce como Bus Universal en Serie (en inglés Universal Serial Bus) que define el orden de los cables, la forma de los conectores y protocolos que se usan para conectar periféricos o dispositivos electrónicos a una computadora [6].

<sup>6</sup>El término sniffer se usa para describir aplicaciones que capturan datos que viajan a través de una red [5].

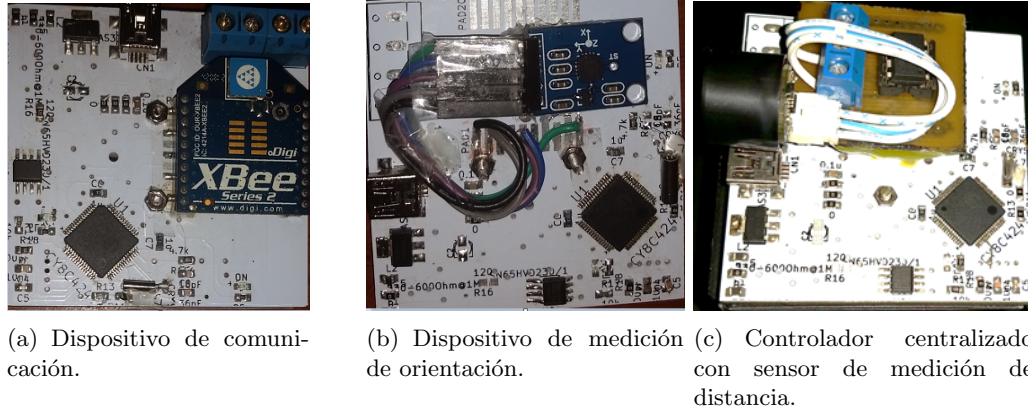


FIGURA 3.7. Dispositivos externos o accesorios compatibles con la plataforma EMERGE.

fabricado por Devantech). Dicho sensor tiene un rango de medición de dos (2) metros y está conectado al micro-controlador que procesa las diferentes mediciones realizadas por el sensor. Además, este micro-controlador funciona como un controlador centralizado, ya que, permite modular los parámetros de movimiento del robot a partir de la información medida por el sensor.

Los dispositivos descritos anteriormente<sup>7</sup> utilizan la misma configuración de pines con resorte que los módulos, de esta forma al conectar cualquier dispositivo al robot se puede establecer una comunicación con otro dispositivo o cualquier módulo, ya que, utilizan el mismo bus de datos.

### 3.4. Funcionamiento del sistema de control

Como se afirmó en el numeral anterior, los módulos tienen un controlador local el cual se encarga de ejecutar ciertas tareas básicas y existe un controlador centralizado que se encarga de modular los parámetros de movimiento del robot. Estos controladores trabajan en conjunto para generar esquemas de movimiento que le permitan al robot desplazarse de un punto a otro. Debido a esto se establecieron ciertas funciones básicas que permiten coordinar los módulos y dispositivos que conforman la estructura robótica, dichas funciones se describen a continuación.

#### 3.4.1. Comunicación entre dispositivos

La comunicación entre dispositivos se realiza usando el protocolo CAN (del inglés, Controller Area Network). Este protocolo es estándar y se usa normalmente en sistemas distribuidos, ya que, facilita la transmisión de la información entre dispositivos. Además, este protocolo se basa en el encapsulamiento de mensajes, es decir, se le asigna un identificador al mensaje y se empaqueta para realizar su transmisión. De esta manera, se asegura que el dispositivo con el identificador sea el único que pueda leer la información del men-

<sup>7</sup>Hay una descripción más detallada del ensamblaje de los accesorios y los módulos en el apéndice. Además, los circuitos y planos con sus respectivas instrucciones se encuentran disponibles para su descarga en el siguiente enlace: <https://sites.google.com/view/emergemodular/home>

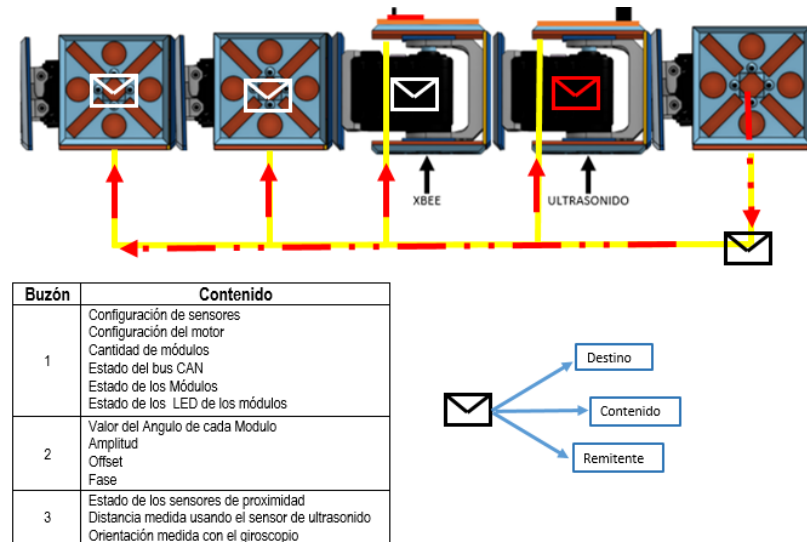


FIGURA 3.8. Buzones que tiene definidos en micro-controlador y su respectiva función o variables a almacenar asignadas.

saje, a pesar de que el mismo mensaje llegue a todos los dispositivos conectados a la red [33].

El protocolo CAN se incorpora en cada módulo y dispositivo usando un controlador que se encarga de almacenar, empaquetar, transmitir y recibir la información. Este controlador se define como una función en el micro-controlador. Sin embargo, la tensión de trabajo del micro-controlador no es compatible con la tensión estándar establecida para dispositivos que usen el protocolo CAN [33], debido a esto se incorporó un circuito integrado de referencia VP230.

El protocolo de comunicación implementado en el micro-controlador se compone de funciones que codifican la información que viaja a través del bus de datos. Dichas funciones permiten encapsular la información en determinadas zonas de memoria del micro-controlador denominadas buzón (del inglés, *mailbox*). El controlador dispone de dieciséis (16) buzónes para recibir información y ocho (8) para enviar, cada buzón almacena temporalmente la información de llegada o envío (antes de enviarse) y tienen un identificador diferente. La información que circula en el bus CAN se ha clasificado en tres (3) grupos de la siguiente forma:

1. El buzón con un número identificador igual a uno (1), recolecta la información relacionada con la configuración de cada sensor, configuración del servomotor, cantidad de módulos y estado de los leds de cada cara del módulo.
2. El buzón con un número identificador igual a dos (2), recolecta la información relacionada con los parámetros al ángulo de salida, amplitud, fase y offset del CPG.
3. El buzón con un número identificador igual a tres (3), recolecta la información relacionada con el estado de los sensores de proximidad, orientación y ultrasonido.

Se puede decir, que la información se empaqueta antes de ser enviada asignando a cada paquete, datos relacionados al identificador del destinatario y el remitente. Cuando un paquete va a ser enviado a un módulo se clasifica de acuerdo al tipo de contenido y

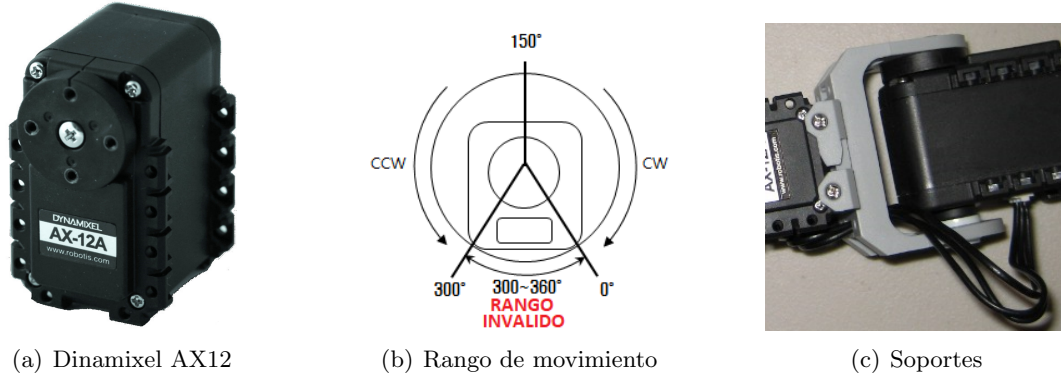


FIGURA 3.9. Características del servomotor (de referencia AX12 fabricado por Dinamixel), basado en [67].

se envía a través del buzón correspondiente. Dicho paquete llega a todos los módulos, sin embargo, el protocolo filtra usando la dirección del destinatario como parámetro de selección del paquete (Fig. 3.8).

### 3.4.2. Generación de movimientos

Los movimientos de la estructura robótica conformada dependen del movimiento individual de cada módulo, el cual se genera usando un servomotor (Fig. 3.9). Este servomotor se comunica con el controlador local mediante un protocolo serial de un solo hilo<sup>8</sup>, el cual permite crear funciones para modificar algunos parámetros del servomotor, tales como: velocidad, posición y torque.

Otra características del servomotor es que su rango de movimiento va desde cero (0) hasta treientos (300) grados (Fig. 3.9(b)), si detecta un movimiento por fuera de este rango el servomotor lo indicara mediante una alarma y se detendrá. Además, El servomotor tiene ciertos elementos (Fig. 3.9(c)) que le permiten acoplarse mecánicamente a las caras del módulo.

Los módulos modifican la posición angular del servomotor usando un CPG implementado en cada controlador local. El CPG presente en cada módulo se coordina con los módulos vecinos y el controlador centralizado, el cual se encarga de modular los parámetros de entrada del CPG para generar diferentes tipos de movimientos.

### 3.4.3. Detección de obstáculos

La detección de obstáculos se realiza mediante cuatro (4) sensores de proximidad incorporados en cada módulo o el dispositivo externo que incorpora el sensor de ultrasonido. Los sensores de proximidad son de tipo infrarrojo y miden magnitudes de luz ambiente y proximidad a un obstáculo. El umbral de detección de un obstáculo es configurable en un rango de veinte (20) centímetros y se comunican con el micro-controlador usando el protocolo I2C<sup>9</sup> [82].

<sup>8</sup>Este protocolo permite la transmisión de datos de manera alternada, es decir, el dispositivo puede enviar información si no esta recibiendo información de otro dispositivo [51].

<sup>9</sup>I2C (del inglés Inter-Integrated Circuit) es un bus de datos creado principalmente para establecer una comunicación entre diferentes partes de un circuito [62].

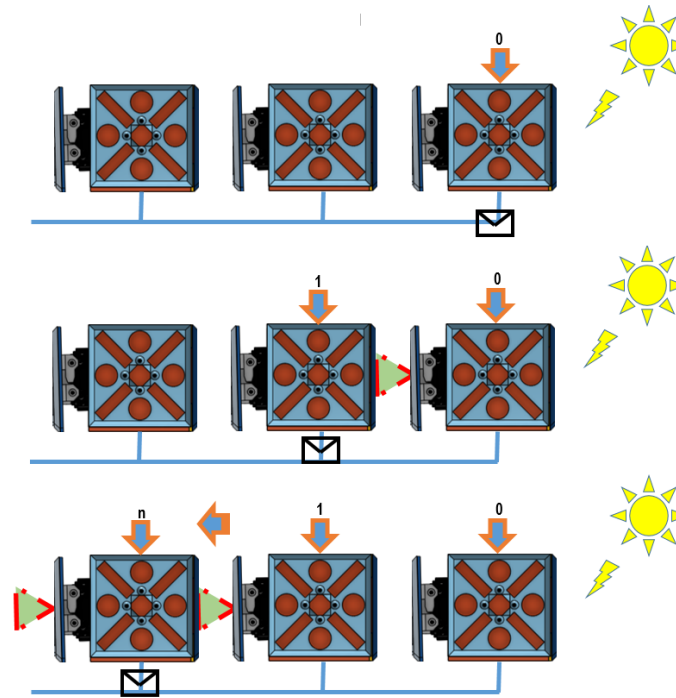


FIGURA 3.10. Ejemplo de cómo se asignan las direcciones de cada módulo durante el inicio de la rutina de control.

A diferencia de los sensores de proximidad que establecen una comunicación directa con el controlador local, el sensor de ultra sonido se conecta con el controlador centralizado y este almacena los diferentes valores de distancia al obstáculo proporcionados por el sensor. En ambos casos, el controlador centralizado se encarga de procesar la información recolectada para modular los parámetros del CPG.

### 3.5. Coordinación del sistema de control

Como se mencionó antes, el controlador centralizado y los controladores locales funcionan de manera coordinada. La coordinación entre ellos se garantiza mediante un protocolo de inicio que se compone de tres (3) funciones, las cuales se describen a continuación;

1. Inicialización: Al ejecutar esta rutina se asignan valores iniciales a los parámetros del CPG, se realiza un test e inician los componentes principales de cada módulo (sensores, módulos de comunicación y el motor). En caso de que algún componente falle, se detiene la operación del robot interrumpiendo la comunicación CAN entre módulos.
2. Descubrimiento: Al finalizar la rutina de inicialización se asigna una dirección a cada módulo (Fig. 3.10). Las direcciones se asignan de manera incremental en valores de una unidad empezando por el número cero (0), dicha dirección se asigna al módulo que detecte la luz del ambiente con el sensor de proximidad de la cara dos (2).

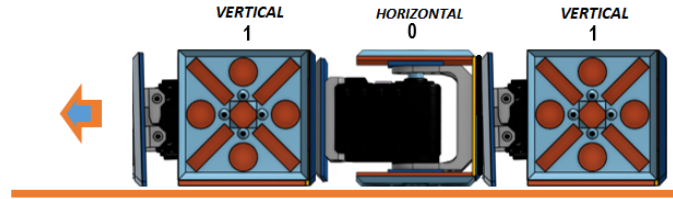


FIGURA 3.11. Detección de la orientación de los módulos durante la ejecución de tareas de locomoción.

Una vez asignada la dirección cero (0) a un módulo, este envía un mensaje mediante el bus CAN al resto de la estructura y enciende el LED de la cara cuatro (4). Todos los módulos reciben el mensaje, sin embargo, el que detecte luz producida por el LED se auto asigna la dirección uno (1). Este módulo envía un mensaje y se repite la misma rutina hasta que todos los módulos tengan una dirección asignada.

3. Ejecución: Al asignar una dirección a cada módulo se detecta su orientación, la cual depende de la activación del sensor de la cara tres (3) (Fig. 3.11), si este se encuentra inactivo significa que la orientación es vertical (codificado como uno (1)) de lo contrario es horizontal (codificado como cero (0)). El valor de orientación se almacena por el controlador local, ya que, el controlador centralizado especifica la orientación del módulo para modificar los parámetros del CPG. Al finalizar, el controlador centralizado ejecuta la función de CPG y el controlador centralizado puede enviar los parámetros de entrada.

### 3.6. Limitaciones de los módulos

Los módulos tienen dos restricciones asociadas a los componentes mecánicos que lo conforman, las cuales se describen a continuación;

- La primera es el rango de movimiento del servomotor, ya que, se ajustó en un rango menor que se encuentra entre sesenta (60) y doscientos cuarenta (240) grados. Este rango se establece con el fin de evitar colisiones entre los circuitos impresos y la carcasa del servomotor (Fig. 3.12).

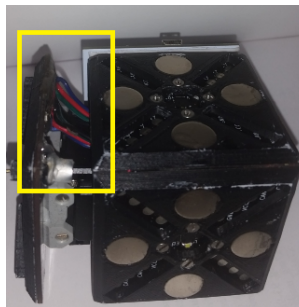


FIGURA 3.12. Posición angular máxima que puede tener un módulo.

- La segunda está relacionada con los sensores de proximidad, porque estos no deben estar rodeados por una superficie con punta (Fig. 3.13(b)) o de color blanco (Fig.

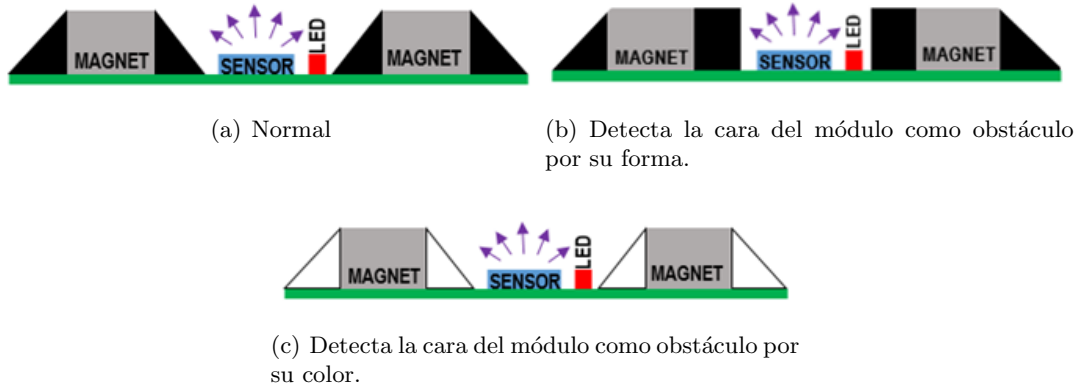


FIGURA 3.13. Corte de una sección transversal del módulo que muestra 3 escenarios diferentes que puede enfrentar el sensor de proximidad.

3.13(c)). Debido a que este tipo de superficies ocasiona un efecto fantasma, es decir, detecta el conector como un obstáculo del entorno (Fig. 3.13). La forma correcta del conector se muestra en la Fig. 3.13(a).

### 3.7. Resumen del capítulo

En este capítulo se presentaron las características básicas de los componentes usados para la fabricación de los módulos y accesorios del robot modular EMERGE. Además, se realizó una descripción de las limitaciones de cada módulo y los diferentes mecanismos de comunicación que usa el robot para coordinar y controlar la estructura conformada.

Como se mencionó antes, los parámetros del CPG se modulan en el controlador centralizado y se envían a los controladores locales para estimar la posición angular de cada módulo. Esta característica se amplía en el siguiente capítulo, ya que, para maximizar la distancia recorrida por el robot al modular los parámetros del CPG se implementaron tres (3) técnicas de optimización que le permiten al robot generar esquemas de movimiento de manera autónoma, a partir de la información de percepción sensorial.

---

---

## Estrategia de adaptación

---

---

### 4.1. Introducción

Como se describió en el Capítulo 3, cada módulo del robot EMERGE cuenta con un controlador local, cuya función principal es ejecutar el modelo aproximado de CPG. Este modelo se basa en un sistema de osciladores acoplados que le permite al robot realizar movimientos rítmicos. Sin embargo, coordinar los movimientos que genera un controlador al implementar un CPG no es una tarea sencilla, ya que la forma del movimiento varía en función de la morfología del robot.

En esta parte del trabajo aborda esta temática mediante tres (3) técnicas de optimización, las cuales se incorporan en un controlador centralizado para coordinar los movimientos del robot. La relación entre las técnicas de optimización, el controlador centralizado y el CPG se presentan en este capítulo, el cual está organizado de la siguiente manera; la sección 4.2 describe el funcionamiento del CPG y la sección 4.3 presenta las técnicas de optimización empleadas para modular los parámetros del CPG.

### 4.2. Descripción del funcionamiento del CPG

El circuito electrónico (descrito en el numeral 3.4) implementado en cada módulo funciona como un controlador local, que se comunica con otros módulos o dispositivos a través del bus CAN. Esta capacidad permite incorporar en el robot EMERGE un modelo de CPG basado en osciladores acoplados, que estima el valor de la posición angular de cada módulo a partir de una diferencia de fase establecida, la posición de los módulos y la modulación de tres (3) parámetros (Fig. 4.1).

Los osciladores independientes estiman su valor de salida a partir de una función sinusoidal, es decir, cada módulo tiene un oscilador y genera un movimiento diferente. Sin embargo, cuando los osciladores se acoplan, el robot realiza movimientos coordinados que dependen de los parámetros de amplitud, fase y desplazamiento fijo (en ocasiones se denota como offset). Estos parámetros se ajustan usando un controlador centralizado (Fig. 4.1), que se comunica con el robot usando el protocolo CAN y permite asignar los valores de los parámetros de dos (2) maneras. La primera permite que los valores se envíen



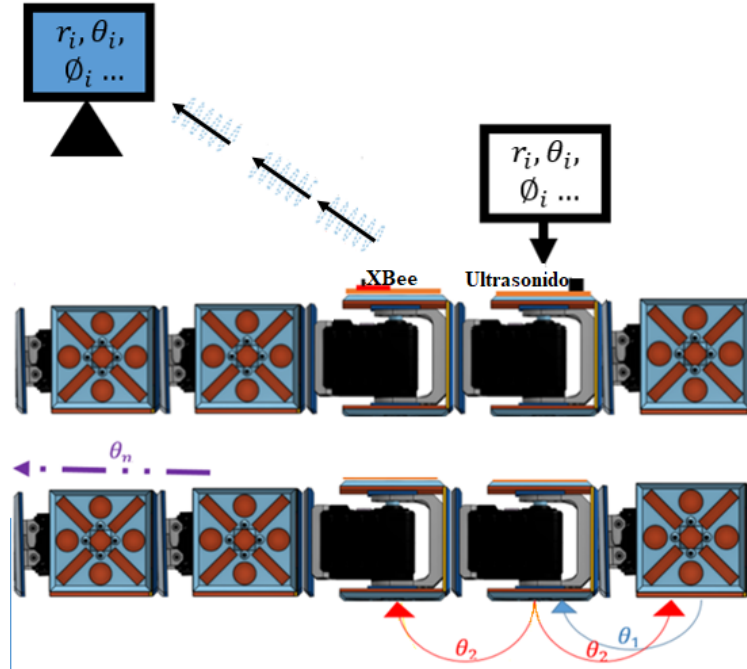


FIGURA 4.1. Actualización de los parámetros del CPG.

desde un computador, mediante una red inalámbrica establecida con un módulo XBee. La segunda permite al controlador centralizado estimar dichos valores usando una técnica de optimización.

Como se dijo, un CPG es un modelo que genera patrones de movimiento rítmicos y tiene varios tipos de representación. En este caso, el controlador local estima el valor de posición angular del  $i$ -ésimo módulo mediante las ecuaciones 4.1, 4.2 y 4.3 [19, 38, 45].

$$\ddot{r}_i = a_r \left( \frac{a_r}{4} (R_i - r_i) - \dot{r}_i \right) \quad (4.1)$$

$$\ddot{x}_i = a_x \left( \frac{a_x}{4} (X_i - x_i) - \dot{x}_i \right) \quad (4.2)$$

$$\theta_i = x_i + r_i \cos(\phi_i) \quad (4.3)$$

La posición angular del  $i$ -ésimo módulo  $\theta_i$  se determina resolviendo la ecuación 4.3, donde  $r_i$ ,  $x_i$  y  $\phi_i$  representan los parámetros de amplitud, desplazamiento fijo y fase del  $i$ -ésimo módulo. La respuesta de las variables  $r_i$  y  $x_i$  se estima mediante las ecuaciones 4.1 y 4.2, ya que su velocidad de convergencia a un valor de amplitud  $R_i$  o de desplazamiento fijo  $X_i$  se ajusta graduando las constantes  $a_r$  y  $a_x$  respectivamente. Al establecer los valores de las constantes  $R_i$  y  $X_i$  usando el controlador centralizado se espera que la salida  $r_i$  y  $x_i$  sea amortiguada y sin oscilaciones, para que el actuador del módulo llegue a una posición fija rápidamente (Fig. 4.2).

En el modelo de osciladores acoplados [19], la ecuación 4.4 representa el valor de la fase  $\dot{\phi}_i$  del  $i$ -ésimo módulo, que depende de: el valor de  $\theta_i$  representado como  $\phi_i$ , la posición angular de sus vecinos más próximos  $\phi_j$  (donde a  $j$  representa a los módulos conectados al  $i$ -ésimo módulo) y una diferencia de fase  $\varphi_{ij}$  que es un valor fijo asignado por el controlador

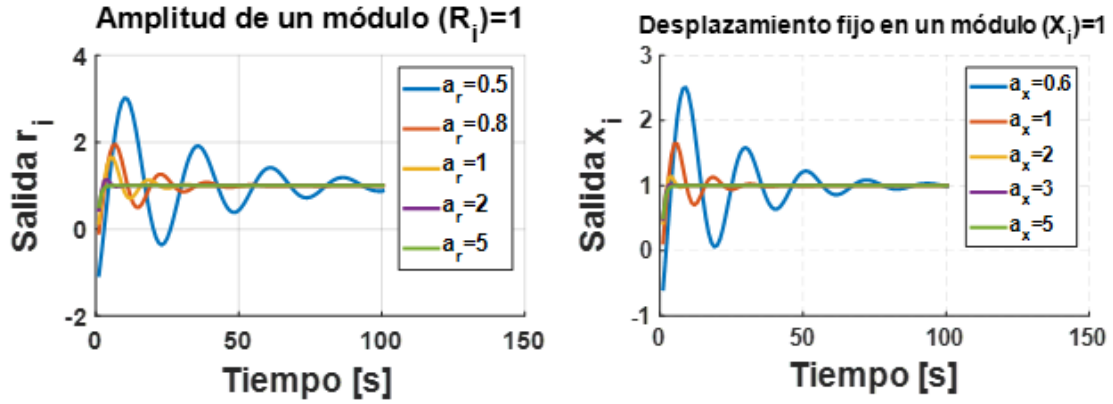


FIGURA 4.2. Tiempo de estabilización de  $r_i$  y  $x_i$  para diferentes valores de  $a_r$  y  $a_x$

centralizado. Además en la ecuación 4.4, el valor de salida  $\dot{\phi}_i$  varía en función de los pesos  $w_i$  y  $w_{ij}$ , los cuales determinan la influencia de los vecinos más cercanos sobre el valor de  $\theta_i$ . Es decir, cuando  $w_i \ll w_{ij}$  el valor de  $\theta_i$  tratará de mantenerse fijo, ya que la salida de cada oscilador individual no mostrará cambios significativos. En el otro caso cuando  $w_i \gg w_{ij}$ , el sistema no generará movimientos coordinados, debido a que se omite la diferencia de fase al estimar el valor de  $\theta_i$ . En este trabajo se propone un valor para  $w_i$  y  $w_{ij}$  de 0.2 y 0.8 respectivamente (estos valores se determinaron de manera experimental), ya que al fijar estos valores el robot genera un movimiento que le permite avanzar o girar.

$$\dot{\phi}_i = w_i + \sum_{ij} (w_{ij} \text{Sin}(\phi_j - \phi_i + \varphi_{ij})) \quad (4.4)$$

El  $i$ -ésimo módulo estima el valor de  $\theta_i$  (Fig. 4.5(a)) usando el método de Euler con un tiempo de paso  $\Delta t$  de trescientos mili-segundos (300 ms) [49]. En otras palabras, este método encuentra valores aproximados para los términos  $r_i$ ,  $x_i$  y  $\phi_i$  actualizando las ecuaciones del CPG realizando incrementos con longitud  $\Delta t$  (Fig. 4.5(b)). Por ejemplo, el valor de la fase en cada iteración del algoritmo se estima de la siguiente forma  $\phi_t \leftarrow \dot{\phi}_t \Delta t + \phi_t$ .

Además, el método de Euler se usó para observar el comportamiento de  $\theta_i$  con diferentes configuraciones de  $a_r$  y  $a_x$  (Fig. 4.3). Entre las configuraciones probadas se eligió un valor de cinco (5) para cada variable, ya que al usar asignar estos valores se observó una

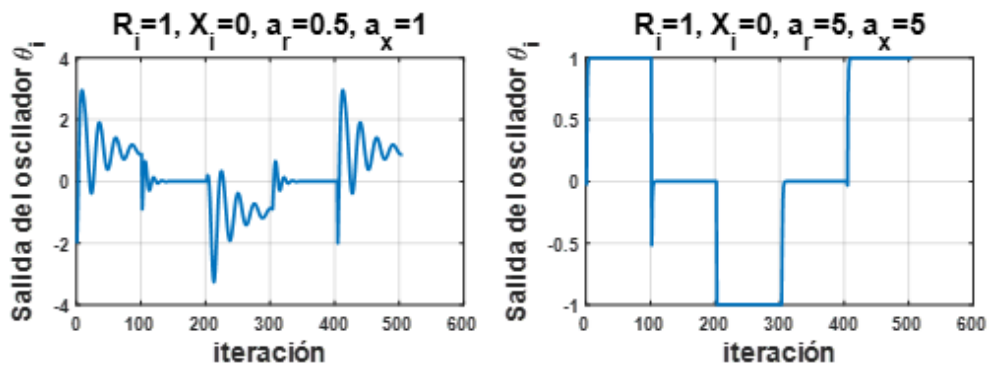


FIGURA 4.3. Solución de la ecuación de CPG usando el método de Euler para diferentes valores  $a_r$ ,  $a_x$ ,  $R_i$  y  $X_i$ .

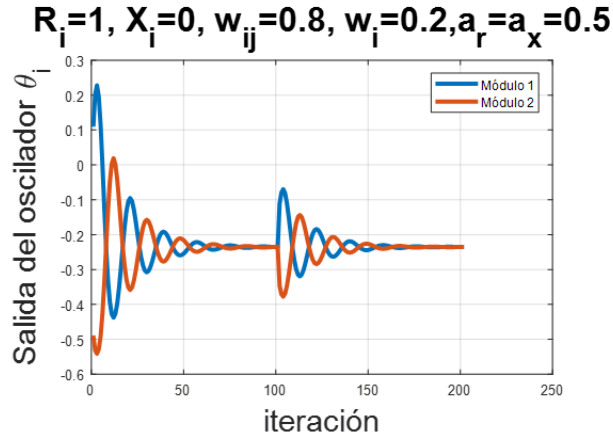
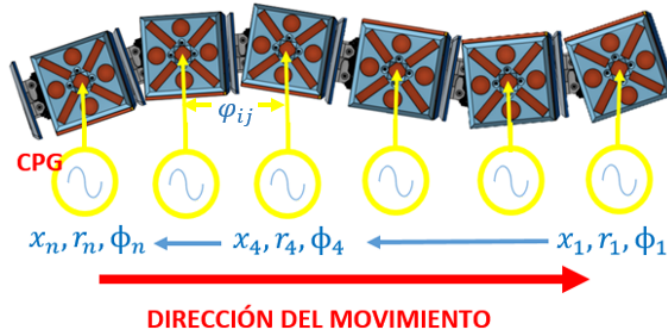
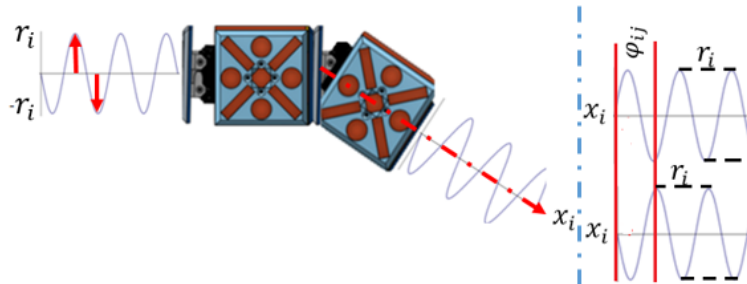


FIGURA 4.4. Respuesta de la ecuación de CPG con dos (2) módulos acoplados con diferencia de fase.

transición entre valores de  $\theta_i$  amortiguada, sin oscilaciones y con un tiempo de estabilización de tres (3) segundos. Finalmente, al determinar todos los valores de las constantes de las expresiones 4.1-4.4 se realizó una simulación del comportamiento de dos (2) módulos acoplados (Fig. 4.4), para determinar si el valor de  $\theta_i$  de cada módulo tiende a una diferencia de fase  $\varphi_{ij}$ .



(a) Distribución de los CPG.



(b) Parámetros del controlador.

FIGURA 4.5. Comportamiento del CPG en la estructura robótica EMeRGE.

### 4.3. Optimización del CPG

En relación con la sección anterior, para modular los parámetros del CPG se implementaron tres técnicas de optimización<sup>1</sup> en el controlador centralizado. La idea es que el controlador maximice la distancia recorrida por el robot, para que se desplace en su entorno de manera adaptativa.

En este caso, el espacio de búsqueda o conjunto de soluciones es determinado por los rangos de valores que toman los parámetros del CPG (variables de entrada). Las posibles combinaciones de las variables de entrada convierten en un individuo, cuyo rendimiento o valor de aptitud (variable de salida) se determina a partir de la información de percepción sensorial. El valor de aptitud de los individuos evaluados se estima de la misma manera en cada técnica. Sin embargo, la forma de recorrer el espacio de búsqueda para optimizar los movimientos del robot cambia dependiendo de la técnica usada. Las técnicas de optimización, las características particulares de los individuos y el espacio de búsqueda se describen de forma detallada a continuación.

#### 4.3.1. Características de un individuo

Como se mencionó, un individuo está compuesto por tres (3) parámetros del CPG: amplitud, fase y desplazamiento fijo. Al enviar estos parámetros a cada módulo y modificar el estado del CPG, se asigna un valor a la variable de salida de cada individuo dependiendo del movimiento realizado. Este valor se mide usando los sensores del robot, los cuales generan un valor numérico que se almacena temporalmente en el controlador centralizado (Fig. 4.6). Este valor numérico corresponde a la distancia recorrida por el robot y se determina de dos (2) maneras.

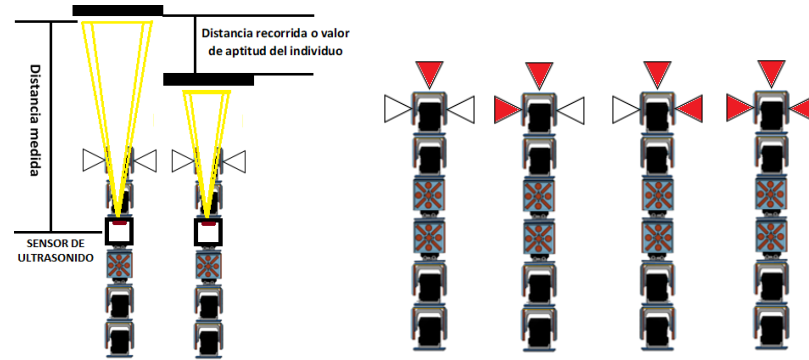
1. La distancia medida ( $d_1$ ) por el sensor de ultrasonido (Fig. 4.6(a)) se almacena antes de enviar los valores de las variables de entrada al robot. Una vez el robot ha realizado el movimiento, se toma una nueva medición ( $d_2$ ) y se calcula la diferencia entre ambos valores medidos ( $F$ ) para estimar la distancia recorrida, como se muestra en la Ecuación 4.5. Cada vez que el robot va a realizar una medición con el sensor de ultrasonido, no se ejecuta ningún movimiento, es decir, el robot se encuentra en reposo para que el sensor se estabilice. El tiempo de reposo del robot es de diez (10) segundos y el tiempo de ejecución del movimiento es de treinta (30) segundos.

$$F = |d_1 - d_2| \quad (4.5)$$

2. La distancia recorrida es igual a cero (0) cuando los sensores de proximidad se activan en diferentes combinaciones (Fig. 4.6(b)), ya que puede haber un obstáculo que por su tamaño no es detectado por el sensor de ultrasonido.

El espacio de búsqueda de las variables de entrada se encuentra limitado de acuerdo con la orientación de los módulos (Fig. 4.7). Al establecer esta limitación se reduce la cantidad de movimientos que ocasionan daños a los accesorios conectados, interrupciones en la comunicación entre módulos o des-conexiones de la fuente de alimentación. A pesar de

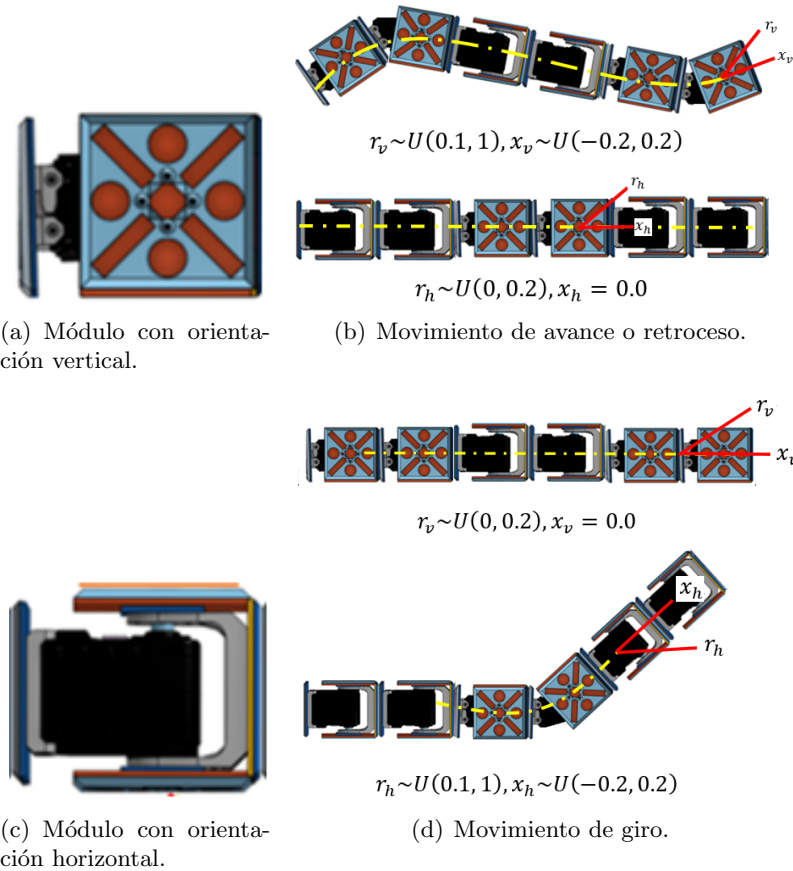
<sup>1</sup>El término optimización se usa para designar mecanismos que ajustan las entradas de un sistema para determinar, idealmente, al mejor valor de salida [1, 2].



(a) Medición de la distancia recorrida con el sensor de ultrasonido. (b) Combinaciones en las cuales los sensores de proximidad asignan un valor de cero (0) a la distancia recorrida.

FIGURA 4.6. Medición del valor de aptitud con los dos (2) tipos de sensores disponibles en el robot EMeRGE.

que el CPG tiene tres (3) parámetros, para indicar si el robot gira, avanza o retrocede, en cada individuo se especifica el valor de las variables de entrada según la orientación del módulo. Es decir, cada individuo tiene cinco (5) variables de entrada que son; amplitud V ( $r_v$ ), amplitud H ( $r_h$ ), fase ( $\phi$ ), desplazamiento fijo V ( $x_v$ ) y desplazamiento fijo H ( $x_h$ ).



(a) Módulo con orientación vertical.

(b) Movimiento de avance o retroceso.

(c) Módulo con orientación horizontal.

(d) Movimiento de giro.

FIGURA 4.7. Combinaciones de las variables de entrada dependiendo de la orientación del módulo.

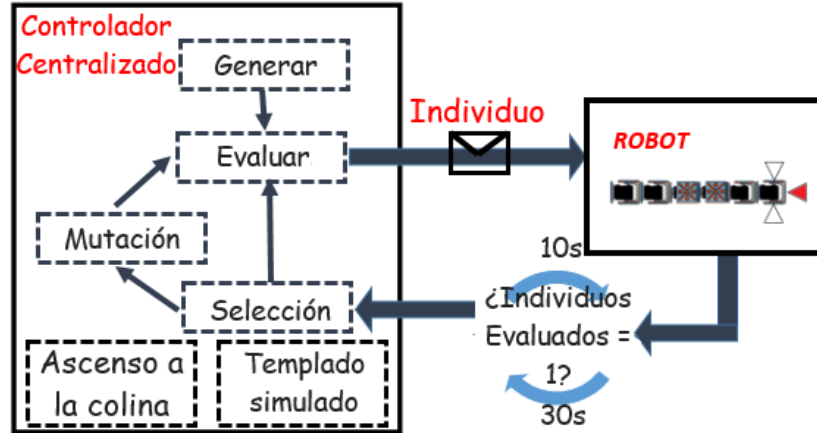


FIGURA 4.8. Representación del funcionamiento de los algoritmos de ascenso a la colina y templado simulado durante el funcionamiento del robot EMERGE.

Al generar un individuo, los valores que se asignan a las variables de entrada dependen de un número aleatorio que sigue una distribución de Bernoulli<sup>2</sup>. Cuando el número es uno, las variables de entrada tendrán un espacio de búsqueda en los rangos definidos para un movimiento de avance o retroceso (Fig. 4.7(b)), en el caso contrario, las variables de entrada tendrán un espacio de búsqueda en los rangos definidos para un movimiento de giro (Fig. 4.7(d)). En ambos casos, la diferencia de fase ( $0 < \phi < 2\pi$ ) es la misma para todos los módulos. Sin embargo, la amplitud y el desplazamiento fijo cambian dependiendo del tipo de movimiento, es decir, cuando el robot avanza o retrocede el rango de movimiento de los módulos con orientación horizontal ( $0 < r_h < 0.2$ ,  $x_h = 0.0$ ) se reduce con respecto a los módulos con orientación vertical ( $0.1 < r_v < 1.0$ ,  $-0.2 < x_v < 0.2$ ) y cuando el robot gira el rango de los módulos con orientación vertical ( $0 < r_v < 0.2$ ,  $x_v = 0.0$ ) se reduce con respecto a los de orientación horizontal ( $0.1 < r_h < 1.0$ ,  $-0.2 < x_h < 0.2$ ).

### 4.3.2. Ascenso a la colina y templado simulado

Las técnicas de optimización de ascenso a la colina y templado simulado modulan los parámetros del CPG, actualizando el valor de aptitud (distancia recorrida  $d$ ) de un solo individuo. A pesar de que, ambas técnicas se basan en el esquema que propone Weise [87], el método de evaluación del valor de aptitud cambia (Fig. 4.8), ya que, en este método el valor de aptitud del individuo a evaluar depende de la evaluación anterior. Es decir, cada vez que el robot realiza un movimiento su posición cambia y al intentar realizar un nuevo movimiento parte de la posición anterior. Además, cada técnica tiene sus características particulares las cuales se describen detalladamente a continuación.

*Ascenso a la colina:* Es una técnica de búsqueda local funciona de manera iterativa, como se muestra en el Algoritmo 1. En cada evaluación de la ecuación 4.5, el mejor individuo conocido  $P_1$  se usa para generar uno (1) nuevo  $P_2$ , si el valor de aptitud del individuo generado es mejor que el conocido, el individuo generado se almacena temporalmente hasta encontrar uno (1) mejor. En relación con lo anterior, para generar un nuevo individuo se aplica una mutación al mejor individuo conocido. La operación de mutación consiste en seleccionar uno (1) de los componentes del mejor individuo conocido de manera aleatoria

<sup>2</sup>Esta distribución de probabilidad genera dos posibles valores para identificar sucesos de éxito (uno (1)) o fracaso (cero (0)) [68].

y sumarle un valor. Este valor es un número generado de manera aleatoria con límites entre  $-0.1$  y  $0.1$ . Sin embargo, cuando la distancia recorrida es cero ( $0$ ) el controlador genera un nuevo individuo con nuevos componentes, ya que, el robot puede estar frente a un obstáculo y necesita cambiar de posición para evadirlo.

---

**Algoritmo 1:** Técnica de optimización de ascenso a la colina incorporada en el controlador centralizado. Basado en [87].

---

```

1 Función Colina()
2   mientras Criterio de terminación no activo hacer
3     si  $d(P_1) > d(P_2)$  entonces
4        $P_2 \leftarrow P_1$ 
5     fin
6     si  $f = 1$  o  $d = 0$  entonces
7        $P_1 \leftarrow$  Generar individuo
8     fin
9     en otro caso
10       $M \leftarrow$  seleccionar parámetro( $P_2$ )
11       $S \sim U(-0.1, 0.1)$ 
12       $M \leftarrow M+S$ 
13       $P_1 \leftarrow$  reemplazar parámetro seleccionado( $M$ )
14    fin
15  fin
  /* f es igual a uno (1), cuando una de las posibles combinaciones de
  los sensores de proximidad se activa. En el otro caso f es igual a
  cero (0). */
  /* d es la distancia recorrida. */

```

---

*Templado simulado:* Es una técnica de optimización que trata de imitar el proceso de recocido de los aceros, el cual consiste en calentar el material y luego enfriarlo para aumentar su rigidez y resistencia [79]. De manera similar, al reproducir este fenómeno físico con el Algoritmo 2 se busca mejorar los movimientos generados por el robot. Estos movimientos se estiman usando un método iterativo que inicialmente genera dos (2) individuos  $P_1$  y  $P_2$ , los cuales son evaluados para determinar su valor de aptitud.

Al estimar dichos valores se escoge el individuo con mejor valor de aptitud  $F$  y se almacena temporalmente en  $P_2$ , luego se aplica una mutación al individuo  $P_1$  de la misma forma empleada en la técnica de ascenso a la colina. Finalmente, se evalúa el nuevo individuo para establecer si es mejor que el individuo almacenado, de acuerdo con un criterio de selección basado en una comparación de los valores de aptitud y un valor estimado por las ecuaciones 4.6 y 4.7.

De las ecuaciones,  $\tau$  representa la temperatura actual del sistema que aumenta a medida que avanza el proceso. Esta característica establece un umbral para modificar el valor de  $P_2$ , al comparar dicho umbral con un valor generado de manera aleatoria [80].

$$\Delta D = d(P_1) - d(P_2) \quad (4.6)$$

$$P(\Delta D, \tau) = e^{-\Delta D/\tau} \quad (4.7)$$

---

**Algoritmo 2:** Técnica de optimización de templado simulado incorporada en el controlador centralizado. Basado en [87].

---

```

1 Función Templado()
2    $\tau = 1;$ 
3   mientras Criterio de terminación no activo hacer
4      $P(\Delta D, \tau) \leftarrow \text{Actualizar}(P(\Delta D, \tau));$ 
5      $r \leftarrow$  número aleatorio (0, 1);
6     si  $d(P_1) > d(P_2)$  O  $r < P(\Delta D, \tau)$  entonces
7        $P_2 \leftarrow P_1;$ 
8     fin
9     si  $f=1$  o  $d = 0$  entonces
10       $P_1 \leftarrow$  Generar individuo;
11    fin
12    en otro caso
13       $M \leftarrow$  seleccionar parámetro( $P_2$ );
14       $S \sim U(-0.1, 0.1);$ 
15       $M \leftarrow M+S;$ 
16       $P_1 \leftarrow$  reemplazar parámetro seleccionado( $M$ );
17    fin
18    si  $\tau \geq 50$  entonces
19       $\tau = 1;$ 
20    fin
21    en otro caso
22       $\tau = \tau + 1;$ 
23    fin
24  fin
  /*  $P(\Delta D, \tau)$  se actualiza de acuerdo con las ecuaciones 4.6 y 4.7. */
  /* f es igual a uno (1), cuando una de las posibles combinaciones de
     los sensores de proximidad se activa. En el otro caso f es igual a
     cero (0). */
  /* d es la distancia recorrida. */

```

---

### 4.3.3. Algoritmo genético

Los algoritmos genéticos se basan en el principio de selección natural de Darwin (Fig. 4.9). Este principio se emula realizando operaciones de selección, cruce y mutación a una población (conjunto de individuos), para determinar la supervivencia de los individuos más aptos. En este caso, la probabilidad de supervivencia de cada individuo está ligada a su valor de aptitud y en cada iteración del algoritmo se asigna un valor de aptitud a todos los individuos de la población, los cuales se actualizan de acuerdo con las operaciones realizadas para mejorar a la población en cada generación [87].

Una ventaja de los algoritmos genéticos es que no requieren un conocimiento específico del problema a resolver. En este caso, dicha característica le permite al robot ajustar sus movimientos para adaptarse a entornos desconocidos. Estos movimientos dependen de un conjunto de diez (10) individuos, el cual se optimiza con un algoritmo genético (Fig. 4.10) implementado en el controlador centralizado. De manera similar a las técnicas descritas en el numeral anterior, el algoritmo genético se modificó para que el controlador



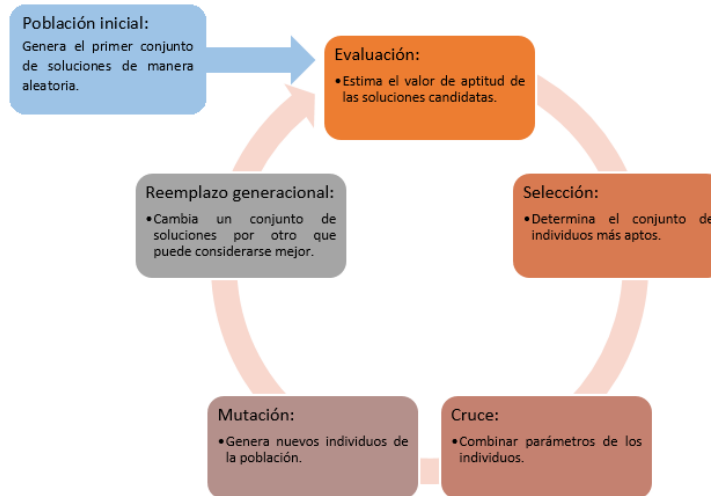


FIGURA 4.9. Estructura de un algoritmo genético, basado en el trabajo presentado por Thomas Weise [87].

centralizado almacene temporalmente el valor de aptitud de cada individuo de la población durante el proceso de evaluación y una vez finaliza se realiza el proceso de evolución. La combinación de ambos procesos mediante un algoritmo genético se describe detalladamente a continuación.

1. *Población inicial:* Es un conjunto de diez (10) individuos que tiene las características mostradas en el numeral 4.3.1.
2. *Selección:* Es un mecanismo que determina el conjunto de individuos más aptos a partir de su valor de aptitud. En este trabajo se implementó la selección por el método ruleta, que consiste en asignar un valor de probabilidad proporcional a cada individuo de la población, de tal manera que al sumar todos los valores asignados se obtenga un resultado igual a uno (1). Luego se genera un número aleatorio que sigue una distribución uniforme entre cero (0) y uno (1), para que la función retorne

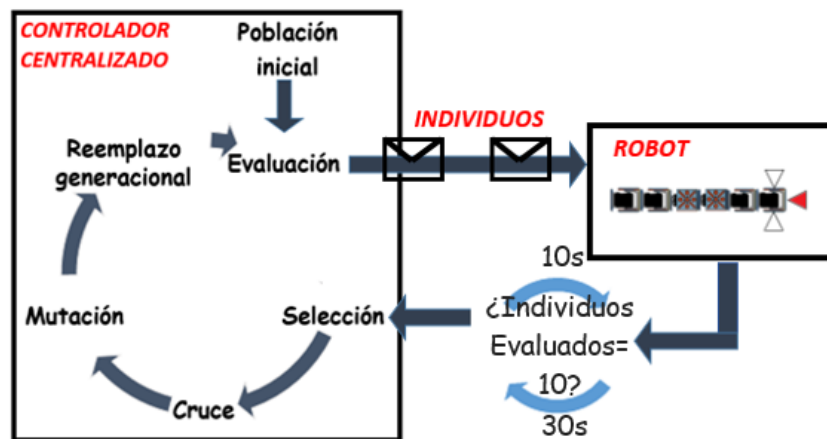


FIGURA 4.10. Representación del funcionamiento de los algoritmos de ascenso a la colina y templado simulado durante el funcionamiento del robot EMERGE.

al individuo que se encuentre más cercano al valor del número generado. Como se muestra de las líneas cinco (5) a la quince (15) en el Algoritmo 3.

3. *Cruce*: Es una estrategia que se usa para combinar características de los individuos de la población. En este trabajo se definió un valor umbral de cruce de 0.7, es decir, si un número generado de manera aleatoria con distribución uniforme no supera el umbral se ejecuta la función de cruce. Esta función permite que dos individuos de la población seleccionados de manera aleatoria se combinen mediante un método de combinación de factores lineales, que consiste en una suma de los pares de componentes de cada individuo multiplicados entre sí, como se muestra de las líneas dieciséis (16) a la veinticinco (25) del Algoritmo 3.
4. *Mutación*: Es una variación que se realiza de manera similar a la mostrada en el numeral 4.3.1. Sin embargo, la ejecución de la función está sujeta a un umbral de 0.1, es decir, depende de un número generado de manera aleatoria con distribución uniforme. Cuando la función es ejecutada cambia las características de un individuo de la población seleccionado de manera aleatoria. Este cambio puede darse de la siguiente manera; sumando un valor generado de manera aleatoria con una función de distribución uniforme a alguno de los componentes del individuo o cambiando el individuo por otro, si está activo el sensor de proximidad, como se muestra de las líneas veintiséis (26) a la treinta y ocho (38) del Algoritmo 3.
5. *Reemplazo generacional*: Es un mecanismo que se usa para sustituir un conjunto de individuos por otro bajo cierto criterio. En particular, en este trabajo se realizó un reemplazo generacional que cambia la población actual por la modificada al aplicar los operadores de selección, mutación y cruce. Como se muestra en la línea treinta y nueve (39) del Algoritmo 3

Como se mencionó anteriormente, la ejecución de los operadores de cruce y mutación depende de un umbral constante, cuyo valor se fijó de manera experimental. Durante la realización de los experimentos, se modificaban los valores de las constantes, hasta encontrar una combinación que le permitiera al robot ir de un lugar a otro rápidamente. Los valores encontrados son similares a los reportados por Alberto Brunete [15]. En ese trabajo se simuló el recorrido de un robot modular con módulos de tres grados de libertad, que genera movimientos usando un algoritmo genético para recorrer entornos cerrados. El algoritmo se ejecutó con varias combinaciones de umbrales de cruce y mutación, en las cuales los valores de 0.7 (cruce) y 0.02 (mutación) fueron los que le permitieron al robot recorrer su entorno en la menor cantidad de generaciones.

A diferencia del trabajo presentado por Brunete, en este trabajo se fijó el umbral de mutación en 0.1, ya que se observó que al aumentar este valor el robot actúa rápidamente frente a un obstáculo de manera adaptativa. Sin embargo, no se tomaron valores mayores a 0.1 a pesar de que el robot evadiera el obstáculo, porque los movimientos encontrados no son coordinados.

---

**Algoritmo 3:** Estructura del algoritmo genético implementado en el robot EMeR-GE.

---

```

1 Función AG( $t$ )
2    $P_0 \leftarrow$  Generar población inicial(10)
3   mientras Condición de parada no activa hacer
4      $d \leftarrow$  Evaluar( $P_0$ )
5     /* Operador de selección */
6      $d \leftarrow$  Normalizar( $d$ )
7     para  $i = 1$  hasta 10 hacer
8        $b \sim U(0, 1)$ 
9        $a = 0$ 
10       $j = 0$ 
11      mientras  $a < b$  hacer
12         $a = a + d[j]$ 
13         $P_1[i] \leftarrow P_0[j]$ 
14         $j = j + 1$ 
15      fin
16    fin
17    /* Operador de cruce */
18    si  $U(0, 1) < 0.7$  entonces
19       $[S_1, S_2] \leftarrow$  seleccionar dos individuos ( $P_1$ )
20      para  $i = 1$  hasta 5 hacer
21         $a_1 \sim U(0, 1)$ 
22         $a_2 \sim U(0, 1)$ 
23         $S_1[i] \leftarrow a_1 * S_1[i] + (1 - a_1) * S_2[i]$ 
24         $S_2[i] \leftarrow a_2 * S_1[i] + (1 - a_2) * S_2[i]$ 
25      fin
26       $P_1 \leftarrow$  reemplazar individuos seleccionados( $S_1, S_2$ )
27    fin
28    /* Operador de mutación */
29    si  $U(0, 1) < 0.1$  entonces
30       $S_1 \leftarrow$  seleccionar un individuo ( $P_1$ )
31      si  $f=1$  entonces
32         $S_1 \leftarrow$  Generar individuo
33      fin
34      en otro caso
35         $M \leftarrow$  seleccionar parámetro( $S_1$ )
36         $R \sim U(-0.1, 0.1)$ 
37         $M \leftarrow M+R$ 
38         $S_1 \leftarrow$  reemplazar parámetro seleccionado( $M$ )
39      fin
40       $P_1 \leftarrow$  reemplazar individuo seleccionado( $S_1$ )
41    fin
42    /* Reemplazo generacional */
43     $P_0 \leftarrow P_1$ 
44     $t \leftarrow t + 1$ 
45  fin

```

---

## 4.4. Resumen del capítulo

En este capítulo se presentó una descripción de tres (3) técnicas de optimización usadas para modular los parámetros del CPG, las expresiones matemáticas características del CPG que se incorporaron en el robot y su método de solución. En particular, en el caso del algoritmo genético se asignó una población de diez (10) individuos, para realizar un cambio generacional rápidamente. Estos cambios generacionales le permiten al robot realizar mutaciones y cruces para ajustar sus movimientos al entorno y así resolver varios tipos de problemas. En el siguiente capítulo se presentan los resultados obtenidos al resolver dichos problemas usando las tres (3) técnicas de optimización descritas anteriormente.

---

---

## Resultados obtenidos

---

---

### 5.1. Introducción

El hardware del controlador y las técnicas de optimización descritas en el Capítulo 4, se combinaron para crear una estrategia de adaptación. Esta estrategia le permite al robot EMeRGE recorrer su entorno de manera autónoma y adaptativa, lo cual se logra en dos (2) pruebas con varios tipos de morfologías.

Los datos experimentales del funcionamiento del robot registrados durante las pruebas en varios escenarios se presentan en este capítulo. En la sección 5.2 se realiza una descripción de los experimentos y resultados obtenidos al ensamblar el robot en morfologías de tipo cadena.

### 5.2. Experimentos con morfologías tipo cadena

En esta sección se presentan dos (2) experimentos realizados para comparar el rendimiento del robot al optimizar la distancia recorrida (Ecuación 4.5) usando las tres (3) técnicas de optimización descritas en el Capítulo 4. Los resultados obtenidos durante el funcionamiento del robot en configuraciones tipo cadena se almacenaron en archivos de texto plano, los cuales se generaron con una aplicación de computadora y una red establecida usando el sniffer de CAN (descrito en el numeral 3.3.2). Finalmente, la información recolectada en estos archivos se usó para representar gráficamente la distancia recorrida por el robot en cada experimento que se describe a continuación.

#### 5.2.1. Experimento 1

En relación con las tres (3) técnicas de optimización mencionadas en el Capítulo 4, se planteó una prueba inicial para determinar si es viable maximizar la distancia recorrida a partir de la modulación de los parámetros del CPG usando el controlador centralizado. En este caso, el robot ensamblado tenía una morfología tipo cadena de tres (3) módulos conectados en serie en orientación vertical (Fig. 5.1) y recorrió un entorno sin obstáculos.



FIGURA 5.1. Fotogramas de la secuencia de movimientos realizada por una estructura con tres módulos conectados en serie con orientación vertical.

Cada técnica se ejecutó cinco (5) veces y el valor de aptitud de diferentes individuos se evaluó ciento cincuenta (150) veces. Los resultados obtenidos se presentan en cuatro (4) gráficas diferentes (Fig. 5.2). Dos (2) de las gráficas muestran el promedio del mejor valor de aptitud (distancia recorrida) encontrado por el robot al implementar las técnicas de ascenso a la colina (Fig. 5.2(a)) y templado simulado (Fig. 5.2(b)). En cada gráfica, se presenta el promedio de los cinco (5) mejores individuos encontrados (línea de color rojo) en cada evaluación y su desviación estándar (línea de color azul).

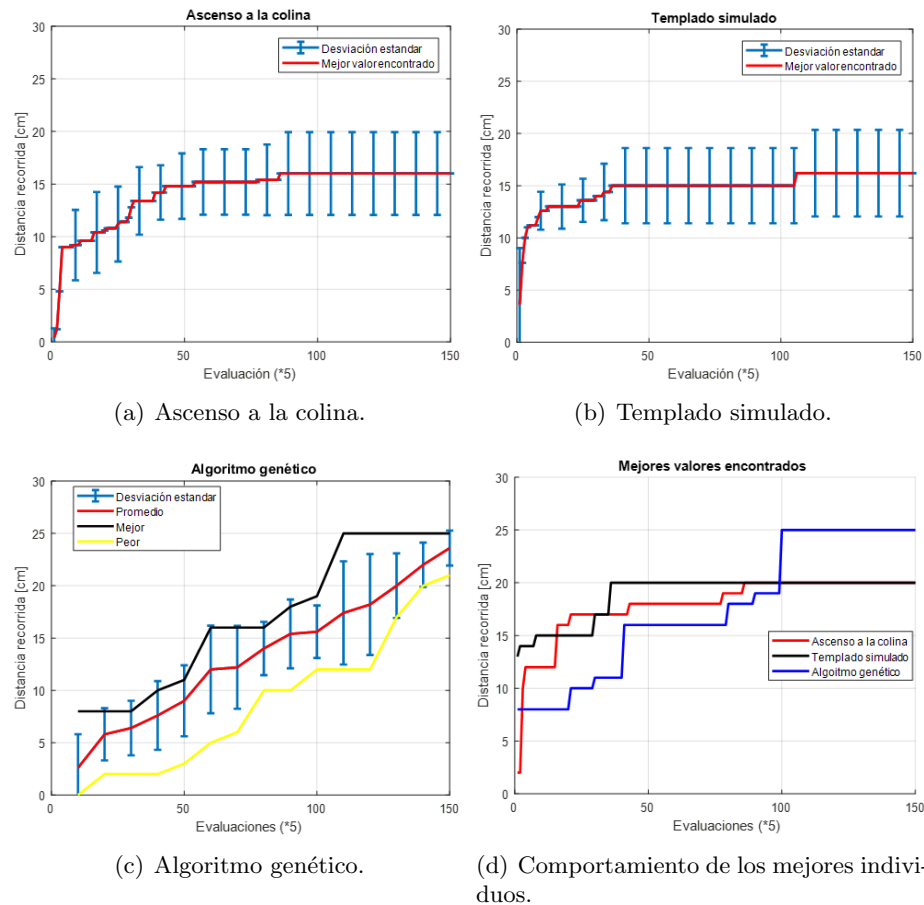


FIGURA 5.2. Promedio de los mejores valores de aptitud (distancia recorrida) encontrados en cada evaluación, al ejecutar las técnicas de optimización cinco (5) veces con una morfología del robot de tres módulos conectados en serie en orientación vertical.

Las dos (2) técnicas de optimización mencionadas le permitieron al robot recorrer una distancia máxima de entre diecinueve (19) y veinte centímetros (20). A diferencia de estas técnicas, el algoritmo genético le permitió al robot recorrer una distancia máxima



FIGURA 5.3. Obstáculo en forma de escalera.

de veinticinco (25) centímetros, lo cual se muestra gráficamente en la Fig. 5.2(c) junto al comportamiento de los mejores valores de aptitud encontrados. En esta gráfica, cada diez (10) evaluaciones representa una (1) generación del algoritmo genético, de la cual se toma el mejor valor de aptitud encontrado en cada ejecución. Al agrupar los valores se determinan características en los mejores valores de aptitud encontrados, tales como el mejor (línea negra), el peor (línea amarilla), el valor promedio (línea roja) y su desviación estándar (línea azul).

Como se puede observar, el eje de las abscisas de las gráficas Fig. 5.2(a) y Fig. 5.2(b) se presenta en número de evaluaciones y el eje de la Fig. 5.2(c) agrupa la información en aumentos de diez (10), ya que es la cantidad de evaluaciones que corresponde a una (1) generación. Esta característica no permite que los algoritmos se puedan comparar fácilmente, por lo que se planteó la gráfica de la Fig. 5.2(d), la cual muestra los mejores valores de aptitud encontrados durante la ejecución de las tres (3) técnicas. Conviene subrayar que la línea que representa el algoritmo genético muestra el mejor valor de aptitud del total de ejecuciones en cada generación.

Una de las mejoras que mostró el algoritmo genético frente a las otras técnicas de optimización fue que el valor de aptitud encontrado es significativamente mayor a los de las otras técnicas. Además, el comportamiento de los individuos encontrados le permite al robot desplazarse de manera adaptativa reduciendo el margen de interrupciones. En otras palabras, los conjuntos de movimientos después de la generación doce (12) permiten desplazamientos continuos con un promedio cercano a los quince centímetros (15 cm), ya que los individuos encontrados tienen cierto grado de similaridad, lo cual no se logró con las otras técnicas. Teniendo en cuenta que, algunos de los movimientos típicos que realizó el robot fueron el de forma sinusoidal ( $r_v = 0.7 \pm 0.1$ ,  $r_h = 0.1 \pm 0.01$ ,  $\phi = 0 \pm 0.01$ ,  $x_v = 0 \pm 0.1$  y  $x_h = 0$ ), en forma de  $\cup$  ( $r_v = 0.5 \pm 0.25$ ,  $r_h = 0 \pm 0.0$ ,  $\phi = 0.3 \pm 0.1$ ,  $x_v = 1.5 \pm 0.27$  y  $x_h = 0$ ) y  $\cap$  ( $r_v = -0.5 \pm 0.25$ ,  $r_h = 0 \pm 0.0$ ,  $\phi = 0.3 \pm 0.1$ ,  $x_v = -1.5 \pm 0.27$  y  $x_h = 0$ ).

Finalmente, se realizó una prueba con un obstáculo en forma de escalera (Fig. 5.3) para saber si el controlador centralizado generaba conjuntos de movimientos que le permitieran al robot evadirlo. Esto se logró ejecutando el algoritmo genético cincuenta (50) generaciones en una estructura de seis (6) módulos conectados en serie en orientación vertical (Fig. 5.4).



FIGURA 5.4. Fotogramas de la estructura de seis (6) módulos evadiendo un obstáculo en forma de escalera.

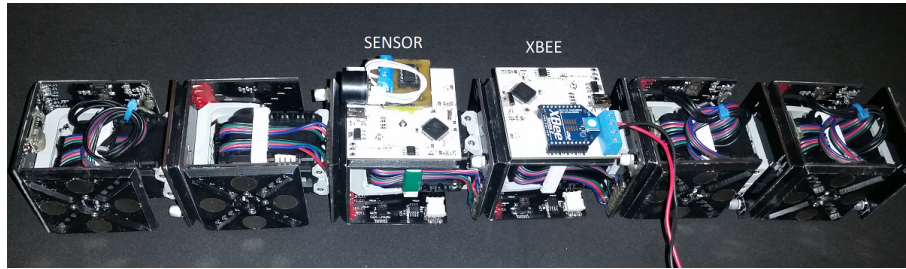


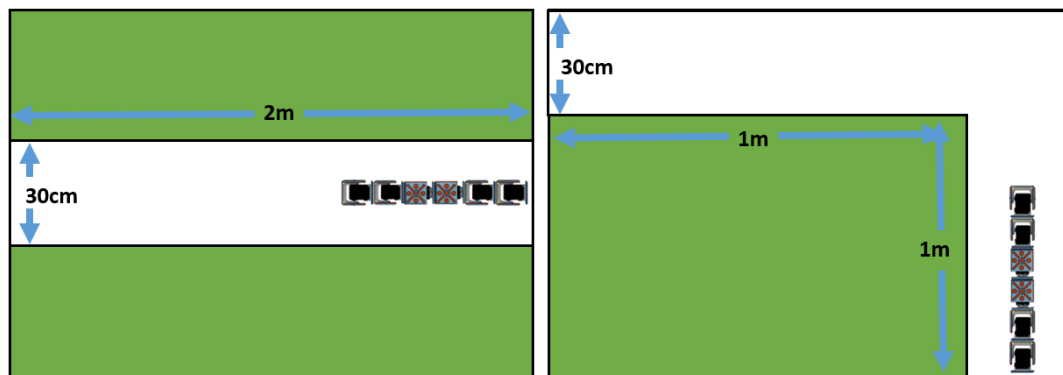
FIGURA 5.5. Morfología ensamblada para habilitar el giro del robot.

### 5.2.2. Experimento 2

El objetivo de estas pruebas es verificar si al modificar la morfología del robot, las técnicas de optimización encuentran combinaciones de movimientos que le permitan desplazarse. En este caso, se ensambló una morfología similar a la de seis (6) módulos conectados en serie con orientación vertical. Sin embargo, la estructura ensamblada para estas pruebas tiene dos (2) módulos en el medio conectados en posición horizontal, para que el robot gire (Fig. 5.5).

Los escenarios propuestos (Fig. 5.6) para realizar estas pruebas son dos (2): el primero es un pasillo recto (Fig. 5.6(a)) y el segundo es un pasillo en forma de ele (Fig. 5.6(b)). La posición de inicio puede ser cualquiera de los extremos, ya que el objetivo es que el robot vaya de un extremo a otro. En cada escenario, se ejecutó cada algoritmo de optimización cinco (5) veces. En el escenario uno (1) se realizaron ciento cincuenta (150) evaluaciones de individuos y en el escenario dos (2) se realizaron doscientas cincuenta (250) evaluaciones de individuos. Los resultados obtenidos se presentan en dos (2) diagramas de cajas y bigotes (Fig. 5.7(a) y 5.7(b)) que presentan el comportamiento de los mejores valores de aptitud encontrados por cada técnica.

Este comportamiento muestra que las técnicas de ascenso a la colina y templado simulado no encuentran conjuntos de individuos que le permitan al robot mantener un movimiento continuo y estable, lo cual produce una variabilidad del valor de aptitud mayor comparada con el algoritmo genético. Es decir, los conjuntos de movimientos que encuen-

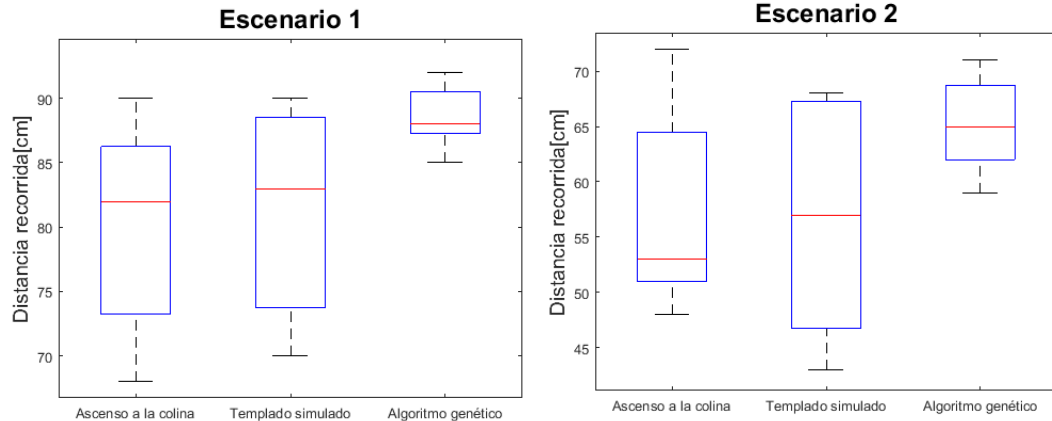


(a) Escenario uno (1): pasillo recto.

(b) Escenario dos (2): pasillo en ele.

FIGURA 5.6. Escenarios propuestos para comparar las estrategias de optimización.





(a) Diagrama de cajas del escenario uno (1). (b) Diagrama de cajas del escenario dos (2).

FIGURA 5.7. Resumen de las cinco (5) ejecuciones de las tres técnicas de optimización usadas.

tra el algoritmo genético reducen la variabilidad del valor de aptitud, ya que en algunos casos los conjuntos de individuos producen movimientos coordinados, lo que indica que el robot recorre su entorno de una manera adaptativa a partir de la información de los sensores.

Otro aspecto que se quiere resaltar es la capacidad de las técnicas de optimización, para encontrar conjuntos de parámetros que le permiten al robot girar en el escenario dos (2). Estos conjuntos dependen del estado de los sensores de proximidad, ya que cuando el robot no se mueve el estado de los sensores genera una mutación que produce un cambio repentino (Fig. 5.8). Este cambio tiene un alto valor de aptitud, debido a que la distancia estimada depende de los puntos de referencia donde se encuentre el robot.

En otras palabras, cuando se produce una mutación que se asocia a un obstáculo detectado por los sensores, el controlador asume que el robot recorrió una gran distancia. Este efecto se observa como un cambio repentino del valor de aptitud durante las ejecuciones de

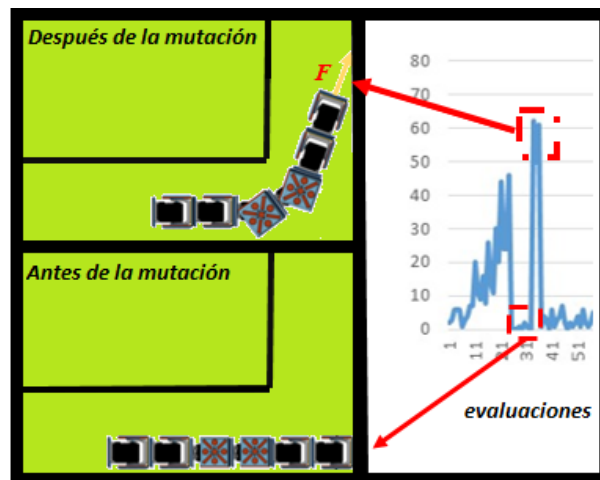
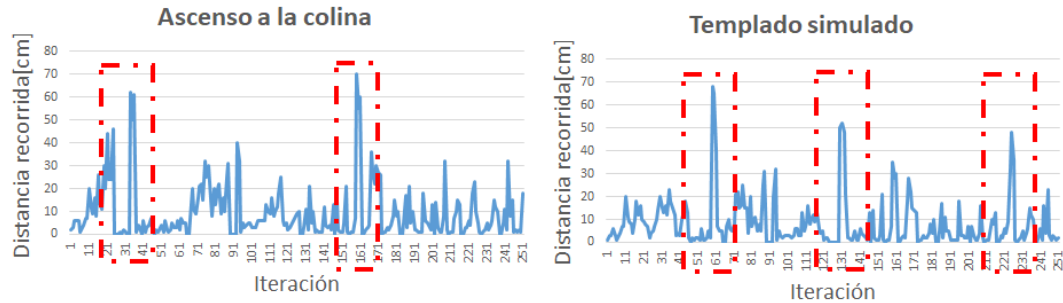


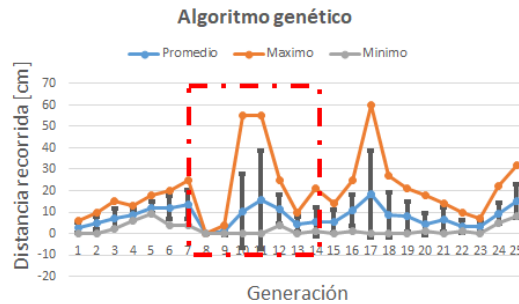
FIGURA 5.8. Comportamiento del valor de aptitud con una mutación inesperada.

las técnicas de optimización (indicados con un cuadro rojo en la Fig. 5.9). En las técnicas de ascenso a la colina (Fig. 5.9(a)) y templado simulado (Fig. 5.9(b)), este cambio no tiene un efecto significativo durante su ejecución, ya que estos algoritmos tienen la mayor variabilidad registrada aún sin aplicar mutaciones continuas al individuo como ocurre cuando hay cambios bruscos del valor de aptitud. Sin embargo, el algoritmo genético (Fig. 5.9(c)) aprovecha esta variación para que los individuos de la población se ajusten y el robot genere conjuntos de movimientos continuos para girar. En otras palabras, como se genera un individuo con un alto valor de aptitud, el proceso de selección trata de descartar a los individuos que no siguen la trayectoria del giro. Sin embargo, cuando el giro termina nuevamente, se produce un cambio repentino, que le permite al controlador encontrar una configuración de parámetros que le permiten al robot avanzar.



(a) Cambios repentinos detectados usando ascenso a la colina.

(b) Cambios repentinos detectados usando templado simulado.



(c) Cambios repentinos detectados usando el algoritmo genético.

FIGURA 5.9. Cambios repentinos detectados durante la ejecución de las técnicas de optimización.

Finalmente, se comparan los resultados obtenidos durante la realización de este experimento usando la prueba de suma de rangos de Wilcoxon. Esta prueba es no paramétrica<sup>1</sup> y se usa para comparar dos (2) poblaciones de muestras que son independientes. En este caso para realizar la prueba se asume que la población de muestras son los mejores valores de aptitud registrados durante el funcionamiento del robot en los dos (2) escenarios propuestos en las cinco (5) ejecuciones de cada técnica de optimización. Las poblaciones de muestras tienen el nombre de las técnicas de optimización y se agrupan en pares de

<sup>1</sup>Esta prueba se utiliza cuando no se conocen ciertos parámetros de la población o no se tiene una gran cantidad de muestras [27].

la siguiente forma; ascenso a la colina-algoritmo genético, templado simulado-algoritmo genético y ascenso a la colina-templado simulado. Los valores obtenidos en la prueba ( $p$ ) se presentan en las Tablas 5.1 y 5.2.

TABLA 5.1. Resultados de la prueba de suma de rangos de Wilcoxon realizada a las combinaciones de los mejores valores de aptitud registrados en cinco (5) ejecuciones de cada técnica de optimización en el escenario uno (1).

<b>Combinación</b>	$p$
Ascenso a la colina-algoritmo genético	0.2857
Templado simulado-algoritmo genético	0.2857
Ascenso a la colina-templado simulado	0.8333

TABLA 5.2. Resultados de la prueba de suma de rangos de Wilcoxon realizada a las combinaciones de los mejores valores de aptitud registrados en cinco (5) ejecuciones de cada técnica de optimización en el escenario dos (2).

<b>Combinación</b>	$p$
Ascenso a la colina-algoritmo genético	0.2778
Templado simulado-algoritmo genético	0.1746
Ascenso a la colina-templado simulado	0.6905

Como se observa en los resultados de las pruebas, la combinación de ascenso a la colina-templado simulado tiene un coeficiente mayor en relación con los estimados para las demás combinaciones. Es decir que en los dos (2) escenarios, las técnicas de ascenso a la colina y templado simulado se comportan de manera similar durante el funcionamiento del robot, ya que el controlador no encuentra movimientos coordinados y adaptativos rápidamente. A diferencia del resultado obtenido con la combinación descrita, las combinaciones que involucran al algoritmo genético obtienen un coeficiente menor, ya que al generar movimientos coordinados se reduce la variabilidad del valor de aptitud y los conjuntos de muestras tienden a agruparse alrededor de un único valor. Esta característica se ve reflejada en la diferencia de aproximadamente 0.5 que se obtiene al comparar los resultados de la prueba de Wilcoxon en cada una de las combinaciones.

### 5.3. Resumen del capítulo

En este capítulo se presentaron los resultados obtenidos al ejecutar tres (3) técnicas de optimización en un controlador centralizado. Se puede decir que estas técnicas le permiten al robot encontrar conjuntos de movimientos para desplazarse en su entorno a partir de la información de percepción sensorial. Esta capacidad aumenta el nivel de autonomía del robot, ya que los parámetros de los CPG son modulados para generar movimientos sin una simulación previa. Las ventajas de la estrategia de adaptación se presentan en el siguiente capítulo, junto con las conclusiones y el aporte de la realización de este trabajo.

---

---

## Conclusiones

---

---

- En esta tesis de maestría, se propuso una estrategia de adaptación que combina características de técnicas bio-inspiradas como los CPG y los algoritmos genéticos, para que el robot EMERGE genere movimientos coordinados de manera adaptativa a partir de la información de los sensores. Este fenómeno se puede visualizar en las gráficas (Fig. 5.7(a) y 5.7(b)) del numeral 5.2.2, ya que al comparar la variabilidad del valor de aptitud obtenida con el algoritmo genético con las otras técnicas de optimización se observa que al disminuir la variabilidad aumenta la posibilidad de que el robot genere conjuntos de movimientos coordinados y estables. Esta característica le da a la estrategia propuesta una ventaja frente a las técnicas basadas en una simulación, ya que el algoritmo de optimización se ejecuta en línea, lo cual le permite al robot adaptarse a su entorno incluso cuando no tenga información previa de este. Este efecto se produce debido a que el simulador genera conjuntos de movimientos predefinidos, sin tener en cuenta que la forma del entorno, la ubicación de los obstáculos o las condiciones ambientales pueden cambiar de un momento a otro.
- Los conjuntos de movimientos adaptativos se forman a partir de una interacción local entre los módulos del robot, que se consigue al agrupar osciladores independientes mediante un modelo de osciladores acoplados con diferencia de fase. Este modelo permite controlar y coordinar los movimientos de un robot modular y es fácilmente implementable en un micro-controlador, ya que tiene una representación matemática simple comparada con otros modelos, como los de Hodgkin-Huxley o FitzHugh-Nagumo. La ventaja de este método es que no requiere de un entrenamiento previo, lo que reduce el tiempo de implementación en cada módulo de la estructura.

El intercambio de información entre los módulos y dispositivos del robot se establece mediante la transmisión de mensajes usando el bus CAN, lo cual le da al controlador centralizado la capacidad de generar varios tipos de movimientos al modular los parámetros del CPG usando una (1) técnica de optimización. Esta característica permite ajustar de manera automática los parámetros del CPG para que el robot realice movimientos de locomoción, entre los que se encuentran el de forma sinusoidal, en forma de  $\cup$  y  $\cap$  (cuyos parámetros se describen en la sección 5.2.1). Sin embargo, el robot no realiza movimientos que impliquen una rotación sobre su eje longitudinal, ya que puede desconectarse o cortar los cables de la fuente de alimentación.

- Las restricciones en los movimientos del robot están ligadas a su funcionamiento mecánico, ya que el mecanismo de acople entre módulos no asegura una conexión continua de la estructura. Esto se observó durante la realización de pruebas con cadenas de ocho (8) y diez (10) módulos conectadas en serie, debido a que, cuando el controlador establece una amplitud y desplazamiento fijo mayores a 0.8 y 0.3 respectivamente, normalmente se desconecta una sección del robot.

Otros factores que influyen en el funcionamiento del robot son la velocidad del servo motor y la fuente de alimentación. Por un lado, al incrementar la velocidad de los motores de los módulos por encima de 59 rpm (revoluciones por minuto) los movimientos realizados pueden desconectar secciones de la estructura ensamblada. Por el otro, se observa que el robot no es totalmente autónomo, ya que depende de una fuente de alimentación externa. Al depender de una fuente se deben evitar los movimientos que desconecten al robot de la fuente o los que produzcan un cortocircuito.

- Durante la realización de los experimentos se observó que el valor de aptitud puede cambiar rápidamente de un momento a otro, debido a que la evaluación del valor de aptitud actual depende de la evaluación anterior. A pesar de que esta característica produce una alta variabilidad entre individuos, se probó que usando una técnica de optimización es posible encontrar esquemas de movimiento para que un robot modular se adapte a su entorno con diferentes morfologías de tipo cadena. Sin embargo, las técnicas de ascenso a la colina y templado simulado no encontraron conjuntos de movimientos para que el robot evadiera el obstáculo en forma de escalera.
- Como se dijo, las pruebas realizadas en este trabajo contemplan diferentes morfologías del robot en configuraciones de tipo cadena. Sin embargo, el sistema de control propuesto se puede generalizar para controlar robots modulares con morfologías arbitrarias o con módulos con más de un grado de libertad, ya que al modificar la morfología del robot se puede seguir usando el CPG basado en osciladores acoplados y modificando sus parámetros desde un controlador centralizado.

---

---

## Trabajo Futuro

---

---

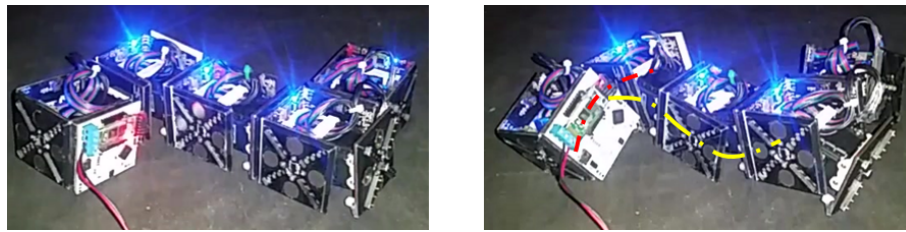
Aunque el robot modular EMERGE funcionó correctamente en conjunto con el controlador centralizado propuesto, el prototipo está sujeto a mejoras. Algunos temas para continuar con este trabajo son:

- *Actualizar el diseño del sistema de control:* El diseño del controlador centralizado se puede extender, para generar una arquitectura de control descentralizada usando los controladores locales de cada módulo. Esto le daría una ventaja funcional al robot, ya que cada módulo tendría su propio controlador y no sería susceptible a fallar cuando se desconecta un módulo o accesorio. Se propone simplificar el diseño del circuito electrónico eliminando el bus de datos CAN e incorporando un único bus basado en el protocolo serial de un solo hilo. Esto permitiría que todos los componentes estuvieran conectados al mismo bus, lo cual le daría al motor la posibilidad de funcionar así no esté conectado el controlador del módulo.
- *Desarrollo de nuevos accesorios o herramientas:* La cantidad de accesorios puede limitar el movimiento del robot. Una solución parcial a esta limitación puede ser la creación de accesorios o herramientas que se almacenen dentro de un módulo o con formas que no interfieran con el movimiento del robot. Algunos de estos accesorios pueden ser herramientas para interactuar con otros elementos o robots, otro tipo de sensores para medir más variables del ambiente o una cámara para implementar un sistema de visión artificial que le permita tener más información de la forma del entorno.
- *Fuentes de energía o gasto energético:* Como se mencionó en el Capítulo 2, el robot modular EMERGE depende de una fuente de alimentación externa, lo cual reduce su autonomía. Este aspecto se puede mejorar de manera parcial implementando baterías o sistemas foto voltaicos que funcionen como una fuente de alimentación propia. Sin embargo, estos sistemas no garantizan un funcionamiento continuo, por lo tanto, se propone maximizar el tiempo de funcionamiento del robot, a partir de un estudio de eficiencia energética que relacione los movimientos generados y su efecto en el consumo energético o implementar un mecanismo que facilite la realización de los movimientos para reducir su gasto energético.

- *Implementar nuevas técnicas de control:* La versatilidad del controlador centralizado le permite al usuario probar diferentes mecanismos de adaptación. Entre ellos, se propone mejorar las técnicas de optimización presentadas en este trabajo mediante la implementación de nuevos operadores genéticos o técnicas que ajusten automáticamente los operadores usados dependiendo del tipo terreno.

Además, al desarrollar este tipo de técnica se debe tener en cuenta que el robot EMeRGE puede tener morfologías arbitrarias, lo cual aumenta la dificultad para diseñar un único controlador. Teniendo en cuenta que el alcance de esta tesis no contempla controlar el movimiento de robots con estructuras arbitrarias, se realizó un avance en la temática propuesta mostrando que el controlador centralizado tiene la capacidad de permitir al robot generar movimientos oscilatorios con morfologías arbitrarias.

En este caso se, establecieron los parámetros del CPG para morfologías de cinco módulos que forman robots con dos (2) (7.1(a)), tres (3) (7.1(b)) y cuatro (4) extremidades (7.1(c)). Al ensamblar dichas estructuras, se conectó el controlador centralizado en un módulo para fijar los parámetros de amplitud (0.8), fase ( $\pi/4$ ) y offset (0.0) del CPG de cada módulo.



(a) Estructura arbitraria de dos (2) extremidades.



(b) Estructura arbitraria de tres (3) extremidades.



(c) Cuadrópodo

FIGURA 7.1. Estructuras arbitrarias construidas con cinco (5) módulos del robot EMeRGE.

## Construcción de los módulos del robot

### A.1. Circuitos impresos

Los circuitos impresos fueron diseñados a partir del esquema de conexiones presentado en el Capítulo 3. Este esquema se representa mediante un circuito eléctrico (Fig. A.1) que permite generar el arte del circuito impreso en el programa EAGLE versión 7.5. Este programa se usó debido a que es de uso libre y tiene una gran cantidad de componentes disponible para diseñar circuitos impresos.

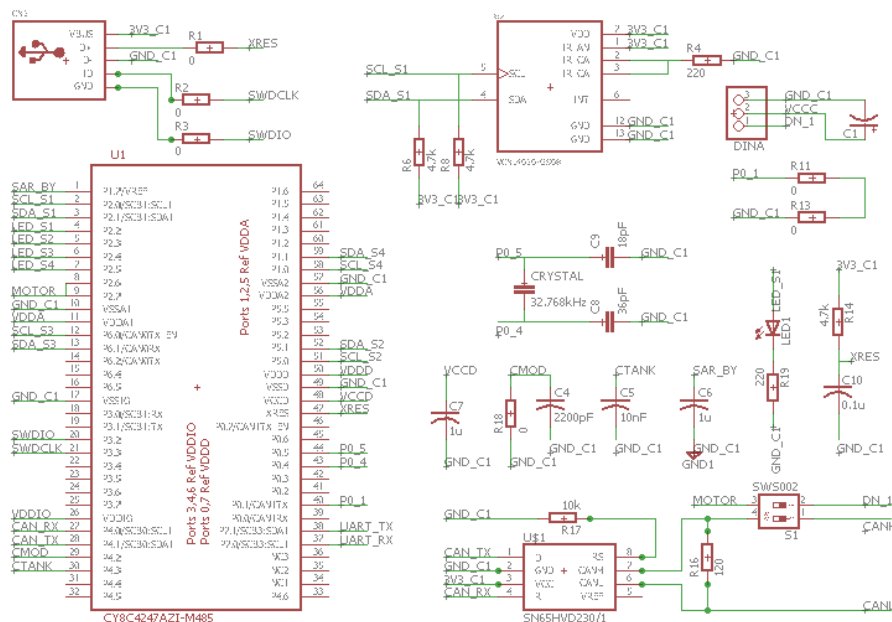


FIGURA A.1. Segmento del plano de conexiones eléctricas del módulo

Una variable que se debe tener en cuenta al diseñar un circuito impreso es el tamaño de las pistas que conectan los diferentes elementos electrónicos, ya que, dicha variable es calculada a partir de una corriente aproximada que circulará a través de cada pista. En los circuitos impresos hay dos tipos de pistas; las de control y las de alimentación. La



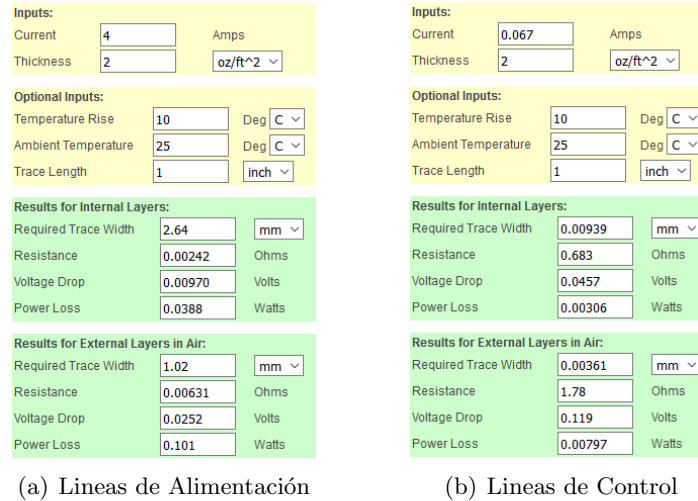


FIGURA A.2. Cálculo aproximado de los parámetros de diseño de un PCB

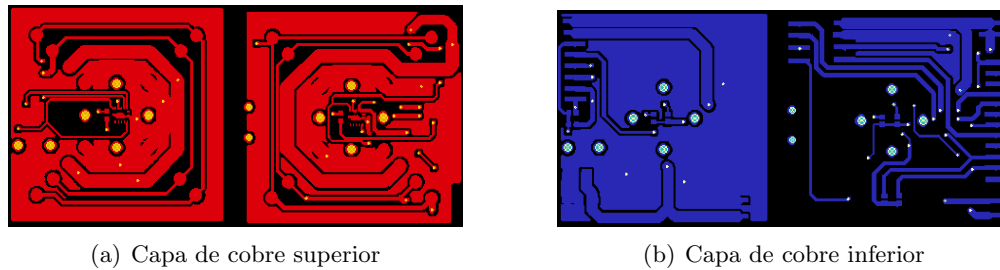


FIGURA A.3. Archivo \*.Gerber generado para la fabricación de los PCB

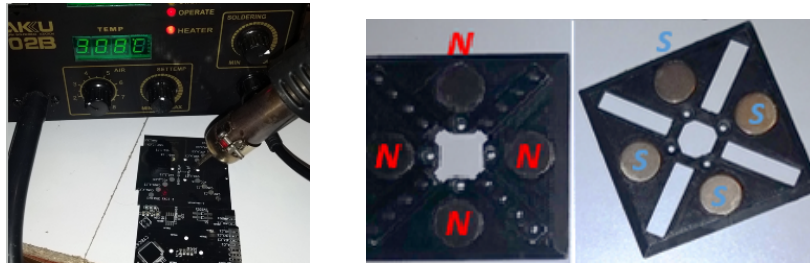
corriente máxima que circulará a través de las pistas de alimentación se estimó en cuatro amperios, para conectar una cadena de máximo veinte (20) módulos en serie. La corriente máxima que circulara a través de las pistas de control se estimó a partir de la información técnica de los componentes, cuyo valor de consumo promedio es de 67mA.

Una vez estimado el valor de la corriente se usó una aplicación llamada *Circuit Calculator* para estimar el grosor de las pistas (Fig. A.2) y las características propias del diseño del circuito impreso. Estas características son incluidas durante el diseño del arte en el programa Eagle, el cual se compone de dos capas. En la capa superior (Fig. A.3(a)) van las pistas de alimentación de cada módulo las cuales son compartidas a través de los pads y en la capa inferior (Fig. A.3(b)) se encuentran las líneas de conexión entre los diferentes elementos de control.

Se diseñaron un total de cuatro circuitos impresos por módulo y tres para los accesorios de la estructura robótica, la fabricación de los circuitos impresos se realizó por una empresa llamada PCBWAY, a partir de un archivo con extensión .Gerber que se genera desde el programa Eagle. Todos los archivos se encuentran disponibles para su descarga en el repositorio mencionado en el Capítulo 3.

## A.2. Ensamblaje de un módulo

El ensamblaje del módulo se realiza usando como caras conectores macho y hembra fabricados en una impresora 3D, los conectores llevan imanes con diferente orientación (Fig. A.4(b)) para que se conecten entre sí. Además, el tamaño de los conectores va de acuerdo con el tamaño de los circuitos impresos para no realizar ningún ajuste mecánico.

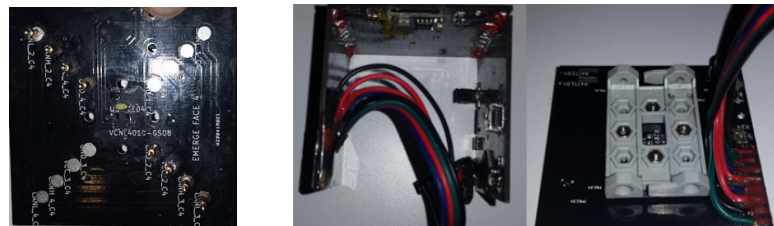


(a) Montaje de Componentes Electrónicos

(b) Orientación de los imanes

FIGURA A.4. Acondicionamiento de las caras del módulo

Previamente se deben soldar los componentes electrónicos a los circuitos impresos con una estación de soldadura de aire caliente (Fig. A.4(a)), la estación de soldadura debe regularse entre trescientos (300) y trescientos cincuenta (350) grados celcius porque es la temperatura promedio de calentamiento recomendada en las hojas técnicas de los componentes.



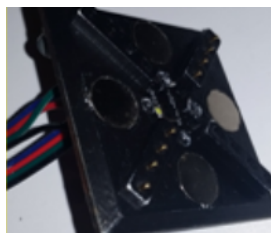
(a) Conectores de Resorte

(b) Buses de datos

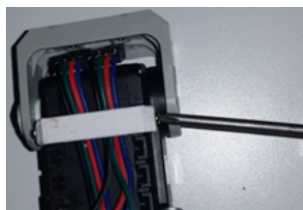
FIGURA A.5. Buses y Conectores de la plataforma robótica

Al terminar de soldar los componentes electrónicos se acoplan los conectores, buses y pads para comunicar los circuitos impresos (Fig. A.5(a) y A.5(b)) y se ensambla la estructura mecánica interna del módulo (Fig. A.6(a)). La base de la estructura (Fig. A.6(e)) esta compuesta por el motor, el cual se acopla usando soportes (Fig. A.6(b) y Fig. A.6(c)). Los soportes se sujetan a las caras usando tornillos y tuercas (Fig. A.6(f)), teniendo en cuenta que en la cara uno se debe colocar un buje para reducir el rozamiento con el tornillo del motor.

Esta metodología permitió el ensamblaje de diez (10) módulos que se probaron con algunos programas de inicio rápido, los cuales se encuentran disponibles para su descarga con las instrucciones de uso en el siguiente repositorio; <https://github.com/HenryHdez/MODULARROBOT>.



(a) Cara macho ensamblada



(b) Soporte superior



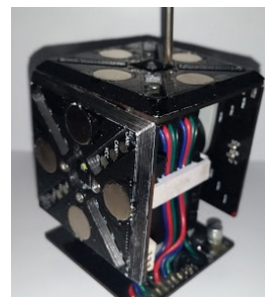
(c) Soporte inferior



(d) Arandela



(e) Cuerpo del módulo

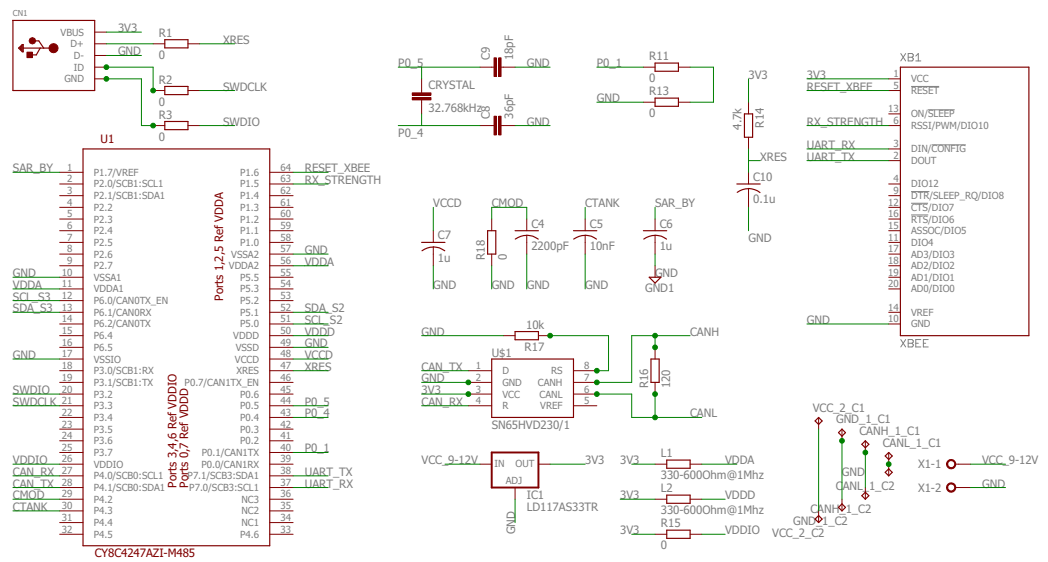


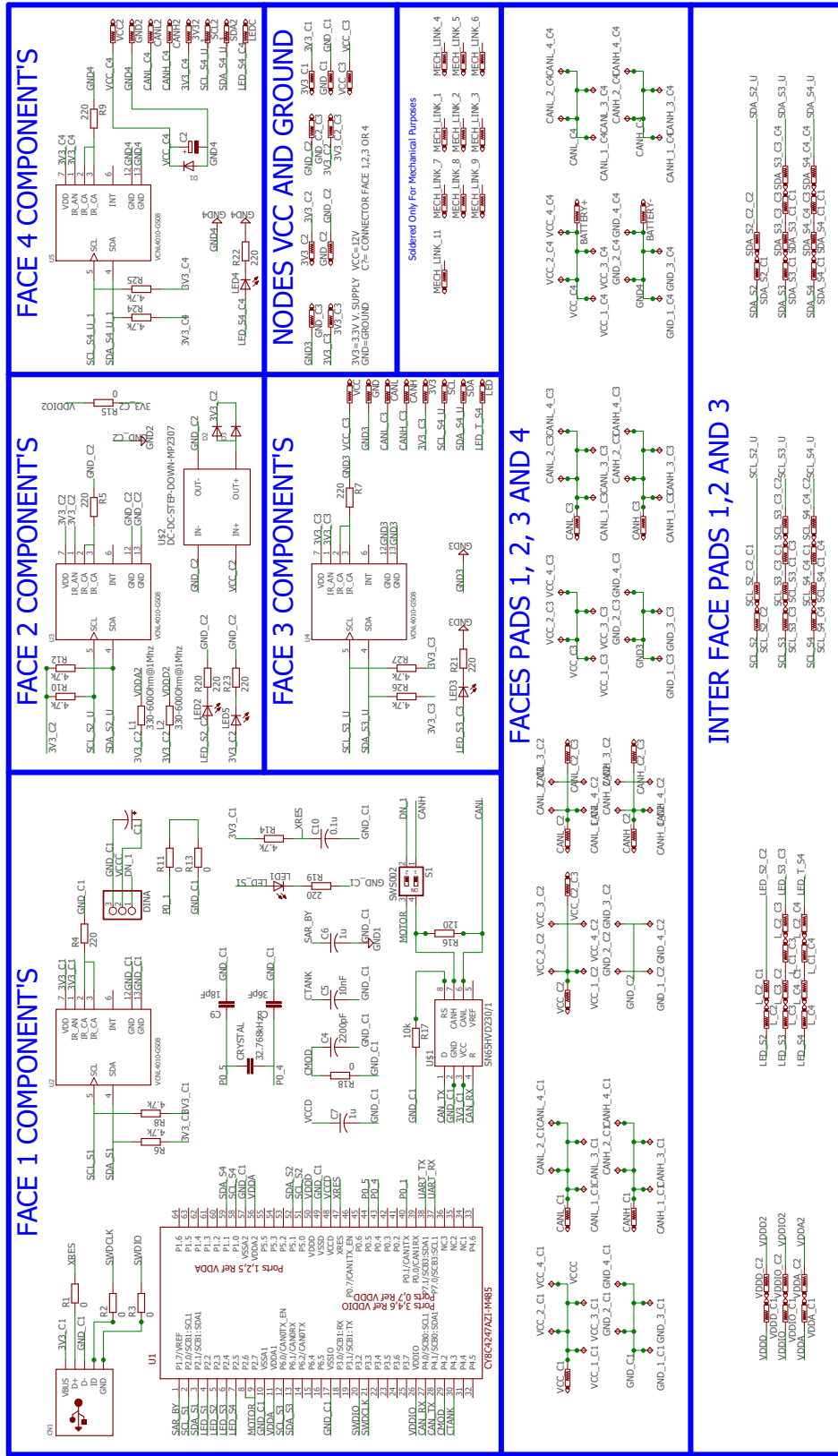
(f) Módulo ensamblado

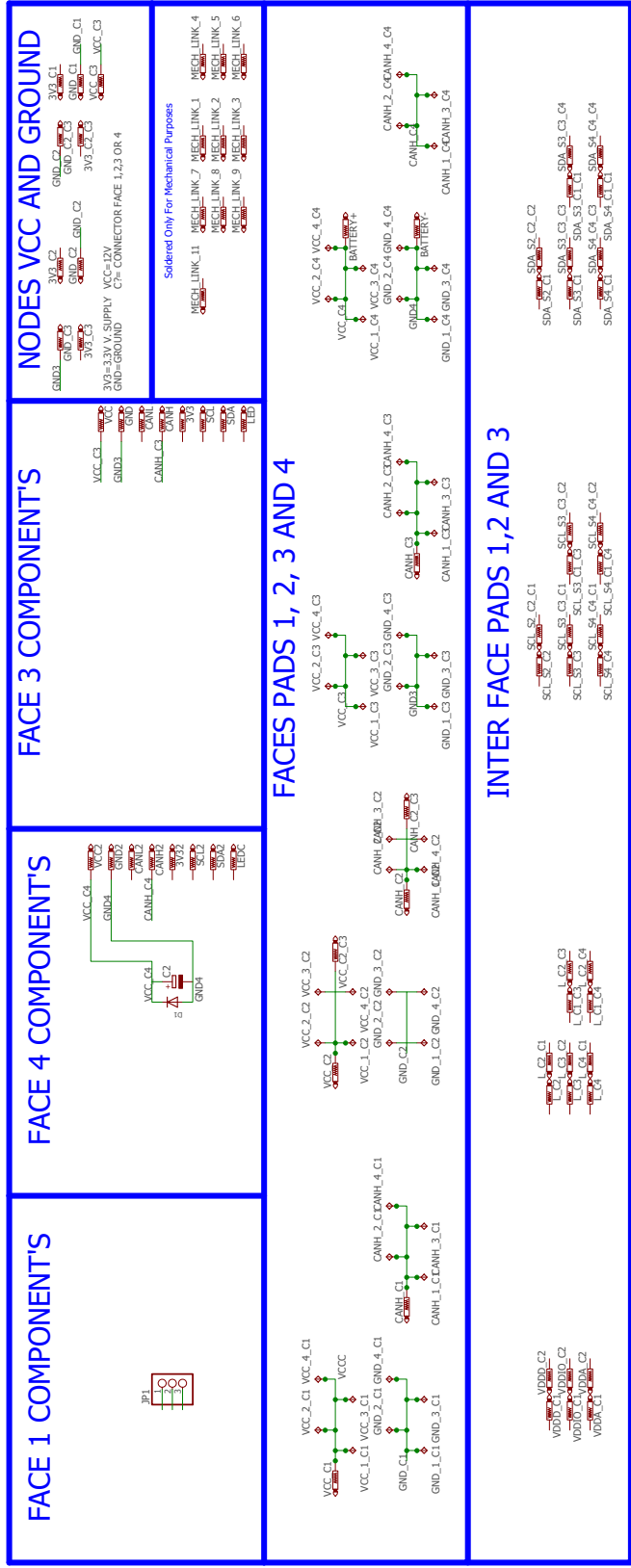
FIGURA A.6. Fases del ensamblaje de un módulo

# APÉNDICE B

## Circuitos electrónicos.





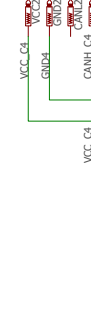


**NODES VCC AND GROUND**

**FACE 3 COMPONENT'S**

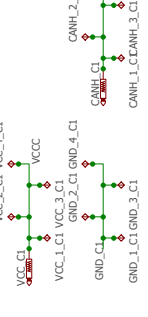
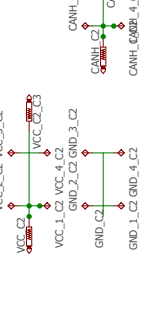
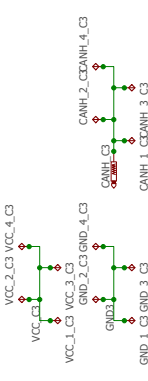
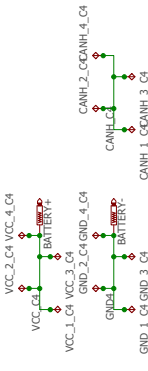
**FACE 4 COMPONENT'S**

**FACE 1 COMPONENT'S**

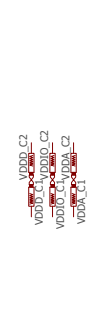
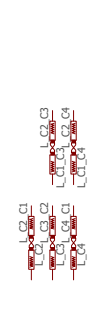
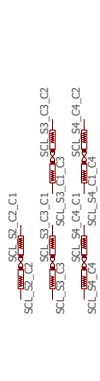
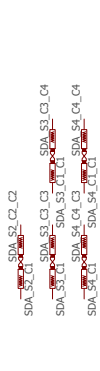


Soldered Only For Mechanical Purposes  
MECH\_LINK\_11 MECH\_LINK\_7 MECH\_LINK\_4  
MECH\_LINK\_9 MECH\_LINK\_5  
MECH\_LINK\_8 MECH\_LINK\_2  
MECH\_LINK\_6 MECH\_LINK\_3  
MECH\_LINK\_10

**FACES PADS 1, 2, 3 AND 4**



**INTER FACE PADS 1, 2 AND 3**



---

---

## Bibliografía

---

---

- [1] E. Alba and J. F. Chicano, *Evolutionary algorithms in telecommunications*, MELECON 2006 - 2006 IEEE Mediterranean Electrotechnical Conference, May 2006, pp. 795–798.
- [2] Enrique Alba and Bernabé Dorronsoro, *Solving the vehicle routing problem by using cellular genetic algorithms*, Evolutionary Computation in Combinatorial Optimization (Berlin, Heidelberg) (Jens Gottlieb and Günther R. Raidl, eds.), Springer Berlin Heidelberg, 2004, pp. 11–20.
- [3] Byoung Kwon An, *Em-cube: cube-shaped, self-reconfigurable robots sliding on structure surfaces*, 2008 IEEE International Conference on Robotics and Automation, May 2008, pp. 3149–3155.
- [4] F. Asadi, M. Khorram, and S. A. A. Moosavian, *Cpg-based gait transition of a quadruped robot*, 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), Oct 2015, pp. 210–215.
- [5] Pallavi Asrodia and Hemlata Patel, *Network traffic analysis using packet sniffer*.
- [6] Jan Axelson, *Usb complete: The developer's guide*, 4th ed., Lakeview Research, 2009.
- [7] J. Baca, P. Dasgupta, S. G. M. Hossain, and C. Nelson, *Modular robot locomotion based on a distributed fuzzy controller: The combination of modred's basic module motions*, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nov 2013, pp. 4302–4307.
- [8] José Baca, Manuel Ferre, and Rafael Aracil, *A heterogeneous modular robotic design for fast response to a diversity of tasks*, Robotics and Autonomous Systems **60** (2012), no. 4, 522 – 531.
- [9] José Baca, S.G.M. Hossain, Prithviraj Dasgupta, Carl A. Nelson, and Ayan Dutta, *Modred: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration*, Robotics and Autonomous Systems **62** (2014), no. 7, 1002 – 1015, Reconfigurable Modular Robotics.
- [10] A.O. Baturone, *Robotica: manipuladores y robots moviles*, ACCESO RAPIDO, Marcombo, 2005.
- [11] D. Brandt, D. J. Christensen, and H. H. Lund, *Atron robots: Versatility from self-reconfigurable modules*, 2007 International Conference on Mechatronics and Automation, Aug 2007, pp. 26–32.

- 
- [12] N. Brener, F. Ben Amar, and P. Bidaud, *Characterization of lattice modular robots by discrete displacement groups*, 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2010, pp. 1133–1139.
- [13] H. B. Brown, J. M. Vande Weghe, C. A. Bererton, and P. K. Khosla, *Millibot trains for enhanced mobility*, IEEE/ASME Transactions on Mechatronics **7** (2002), no. 4, 452–461.
- [14] A. Brunete, M. Hernando, and E. Gambao, *Offline ga-based optimization for heterogeneous modular multiconfigurably chained microrobots*, IEEE/ASME Transactions on Mechatronics **18** (2013), no. 2, 578–585.
- [15] Alberto Brunete, Miguel Hernando, and Ernesto Gambao, *A simulation environment for bio-inspired heterogeneous chained modular robots*, International Journal of Advanced Robotic Systems **11** (2014), no. 2, 17.
- [16] Sasanka Sankhar Reddy CH., Sharath Patlolla, Anita Agrawal, and Anupama K. R., *Hexamob—a hybrid modular robotic design for implementing biomimetic structures*, Robotics **6** (2017), no. 4.
- [17] P. Chand, *Fuzzy reactive control for wheeled mobile robots*, 2015 6th International Conference on Automation, Robotics and Applications (ICARA), Feb 2015, pp. 167–172.
- [18] Avis H. Cohen, Philip J. Holmes, and Richard H. Rand, *The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: A mathematical model*, Journal of Mathematical Biology **13** (1982), no. 3, 345–369.
- [19] A. Crespi and A. J. Ijspeert, *Online optimization of swimming and crawling in an amphibious snake robot*, IEEE Transactions on Robotics **24** (2008), no. 1, 75–87.
- [20] Antonio Barrientos Cruz, *Estrategias bio-inspiradas para locomoción de robots Ápodos*, Tesis de Maestría, Universidad Politécnica de Madrid (2011).
- [21] X. Cui, Y. Zhu, X. Wang, S. Tang, and J. Zhao, *A homogenous cpg-network for multimode locomotion control of modular self-reconfigurable robot*, 2012 IEEE International Conference on Mechatronics and Automation, Aug 2012, pp. 2526–2530.
- [22] M. d. R. de Pinho, Z. Foroozandeh, and A. Matos, *Optimal control problems for path planning of auv using simplified models*, 2016 IEEE 55th Conference on Decision and Control (CDC), Dec 2016, pp. 210–215.
- [23] N. Eckenstein and M. Yim, *The x-face: An improved planar passive mechanical connector for modular self-reconfigurable robots*, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2012, pp. 3073–3078.
- [24] S. edizioni, *La gran enciclopedia de los animales - volumi singoli*, Script, 2012.
- [25] H. Ercan and P. Boyraz, *Design of a modular mobile multi robot system: Ulgen (universal-generative robot)*, 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), July 2016, pp. 8–15.
- [26] Z. Fang, Y. Fu, and T. Chai, *A low-cost modular robot for research and education of control systems, mechatronics and robotics*, 2009 4th IEEE Conference on Industrial Electronics and Applications, May 2009, pp. 2828–2833.



- 
- [27] Jean Dickinson Gibbons and Subhabrata Chakraborti, *Nonparametric statistical inference*, pp. 977–979, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [28] David F. Griffiths and Desmond J. Higham, *Euler’s method*, pp. 19–31, Springer London, London, 2010.
- [29] M. Guarnieri, I. Takao, E. F. Fukushima, and S. Hirose, *Helios viii search and rescue robot: Design of an adaptive gripper and system improvements*, 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2007, pp. 1775–1780.
- [30] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, *Internet of things (iot): A vision, architectural elements, and future directions*, *Future Gener. Comput. Syst.* **29** (2013), no. 7, 1645–1660.
- [31] N. Habib, D. Purwanto, and A. Soeprijanto, *Mobile robot motion planning by point to point based on modified ant colony optimization and voronoi diagram*, 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), July 2016, pp. 613–618.
- [32] Auke Jan Ijspeert, *Central pattern generators for locomotion control in animals and robots: A review*, *Neural Networks* **21** (2008), no. 4, 642 – 653, Robotics and Neuroscience.
- [33] ISO, *Controller area network (can)*, 2003.
- [34] Austin D. Jantapremjt P, *Design of a modular self-reconfigurable robot.*, Proceedings of Australian Conference on Robotics and Automation (2001), 38–43.
- [35] Q. Jia, J. Zhang, G. Chen, H. Sun, and T. Li, *Trajectory optimization of modular robot based on the fault-tolerant performance functions*, 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), June 2015, pp. 580–585.
- [36] A. Kamimura, H. Kurokawa, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata, *Distributed adaptive locomotion by a modular robotic system, m-tran ii*, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), vol. 3, Sept 2004, pp. 2370–2377 vol.3.
- [37] M. Kloetzer, C. Mahulea, and R. Gonzalez, *Optimizing cell decomposition path planning for mobile robots using different metrics*, 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), Oct 2015, pp. 565–570.
- [38] D. Lachat, A. Crespi, and Auke Jan Ijspeert, *Boxybot: a swimming and crawling fish robot controlled by a central pattern generator*, The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006., Feb 2006, pp. 643–648.
- [39] P. N. Lancheros, L. B. Sanabria, and R. A. Castillo, *Simulation of modular robotic system mecabot in caterpillar and snake configurations using webots software*, 2016 IEEE Colombian Conference on Robotics and Automation (CCRA), Sept 2016, pp. 1–6.
- [40] Khanh Chau Le and Lu Trong Khiem Nguyen, *Coupled oscillators*, pp. 297–342, Springer International Publishing, Cham, 2014.

- 
- [41] V. T. Le and T. D. Ngo, *morebots: System development and integration of an educational and entertainment modular robot*, 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Oct 2017, pp. 74–80.
- [42] D. Lee, J. Jeong, Y. H. Kim, and J. B. Park, *An improved artificial potential field method with a new point of attractive force for a mobile robot*, 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Dec 2017, pp. 63–67.
- [43] G. Li, R. Urbina, H. Zhang, and J. G. Gomez, *Concept design and simulation of a water proofing modular robot for amphibious locomotion*, 2017 International Conference on Advanced Mechatronic Systems (ICAMechS), Dec 2017, pp. 145–150.
- [44] K. Lipkin, I. Brown, A. Peck, H. Choset, J. Rembisz, P. Gianfortoni, and A. Naaktgeboren, *Differentiable and piecewise differentiable gaits for snake robots*, 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2007, pp. 1864–1869.
- [45] C. Liu, J. Liu, R. Moreno, F. Veenstra, and A. Faina, *The impact of module morphologies on modular robots*, 2017 18th International Conference on Advanced Robotics (ICAR), July 2017, pp. 237–243.
- [46] E. Lupu, A. Tigauan, G. P. Pop, and R. Arsinte, *Evaluation of zigbee technology for a low-cost video surveillance system*, International Conference on Advancements of Medicine and Health Care through Technology; 5th – 7th June 2014, Cluj-Napoca, Romania (Cham) (Simona Vlad and Radu V. Ciupa, eds.), Springer International Publishing, 2014, pp. 233–236.
- [47] John Germán Vera Luzuriaga, *Control de trayectoria de la simulación de un brazo robot de 5 grados de libertad, controlado mediante la plataforma c2000 piccolo launchxl -f28027f*, Tesis de Maestría, Universidad Politécnica de Valencia (2017).
- [48] A. Lyder, R. F. M. Garcia, and K. Stoy, *Mechanical design of odin, an extendable heterogeneous deformable modular robot*, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sept 2008, pp. 883–888.
- [49] T. Maeba, G. Wang, F. Yu, M. Minami, and A. Yanou, *Motion representation of walking/slipping/turnover for humanoid robot by newton-euler method*, SICE Annual Conference 2011, Sept 2011, pp. 255–260.
- [50] N Majid, S Hafiz, S Arianto, R Y Yuono, E T Astuti, and B Prihandoko, *Analysis of effective pulse current charging method for lithium ion battery*, Journal of Physics: Conference Series **817** (2017), no. 1, 012008.
- [51] natanael medina, *Electronica basica: Curso de electronica basica*, 01, natanael medina.
- [52] Y. Meng, Y. Zhang, and Y. Jin, *Autonomous self-reconfiguration of modular robots by evolving a hierarchical mechanochemical model*, IEEE Computational Intelligence Magazine **6** (2011), no. 1, 43–54.
- [53] Sanjit K. Mitra, *Digital signal processing: A computer-based approach*, 2nd ed., McGraw-Hill School Education Group, 2001.

- 
- [54] K. Mitsuhashi and Y. Ohyama, *Suggestion and verification of the modular robot education*, 2017 11th Asian Control Conference (ASCC), Dec 2017, pp. 1379–1383.
- [55] R. Moeckel, Y. N. Perov, A. T. Nguyen, M. Vespignani, S. Bonardi, S. Pouya, A. Sproewitz, J. van den Kieboom, F. Wilhelm, and A. J. Ijspeert, *Gait optimization for roombots modular robots 2014; matching simulation and reality*, Nov 2013, pp. 3265–3272.
- [56] J. Monsalve, J. Leon, and K. Melo, *Modular snake robot oriented open simulation software*, The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent, June 2014, pp. 546–550.
- [57] S. Murata and H. Kurokawa, *Self-reconfigurable robots*, IEEE Robotics Automation Magazine **14** (2007), no. 1, 71–78.
- [58] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, *M-tran: self-reconfigurable modular robotic system*, IEEE/ASME Transactions on Mechatronics **7** (2002), no. 4, 431–441.
- [59] V. Nigolian, M. Mutlu, S. Hauser, A. Bernardino, and A. Ijspeert, *Self-reconfigurable modular robot interface using virtual reality: Arrangement of furniture made out of roombots modules*, 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Aug 2017, pp. 772–778.
- [60] K. Ogata and B.F. Frankel, *Ingeniería de control moderna*, Prentice Hall, 1980.
- [61] E. H. Ostergaard and H. H. Lund, *Evolving control for modular robotic units*, Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694), vol. 2, July 2003, pp. 886–892 vol.2.
- [62] Dominique Paret and Carl Fenger, *The i2c bus: From theory to practice*, John Wiley & Sons, Inc., New York, NY, USA, 1997.
- [63] Ulrich Parlitz and Werner Lauterborn, *Period-doubling cascades and devil’s staircases of the driven van der pol oscillator*, Phys. Rev. A **36** (1987), 1428–1434.
- [64] C Parrott, T.J. Dodd, and Roderich Groß, *Hymod: A 3-dof hybrid mobile and self-reconfigurable modular robot and its extensions*, 09 2016.
- [65] Pinhas Ben-Tzvi Paul Moubarak, *Modular and reconfigurable mobile robotics*, ELSEVIER-Robotics and Autonomous Systems **1** (2012), no. 60, 1648–1663.
- [66] PJRC, <https://www.pjrc.com/teensy/teensy31.html>, 12 de agosto de 2018, 2018.
- [67] Robotis, <http://www.crustcrawler.com/products/bioloid/docs/ax-12.pdf>, 2018.
- [68] Neil Salkind, *Encyclopedia of research design*, (2018).
- [69] P. Shi, Q. Shao, and D. Liang, *Design and improved serpentine curve locomotion control of a planar modular snake robot*, 2016 IEEE International Conference on Information and Automation (ICIA), Aug 2016, pp. 1398–1402.

- 
- [70] A. Singh, A. Paigwar, S. T. Manchukanti, M. Saroya, M. Maurya, and S. Chiddarwar, *Design and implementation of omni-directional spherical modular snake robot (osmos)*, 2017 IEEE International Conference on Mechatronics (ICM), Feb 2017, pp. 79–84.
- [71] A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert, *Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture*, 2009 IEEE International Conference on Robotics and Automation, May 2009, pp. 4259–4264.
- [72] Volko A. Straub, *Central pattern generator*, pp. 650–654, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [73] J. W. Suh, S. B. Homans, and M. Yim, *Telecubes: mechanical design of a module for self-reconfigurable robotics*, Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), vol. 4, 2002, pp. 4095–4101 vol.4.
- [74] H. Suzuki, J. H. Lee, and S. Okamoto, *Development of semi-passive biped walking robot embedded with cpg-based locomotion control*, 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), June 2017, pp. 75–78.
- [75] Y. Suzuki, Y. Tsutsui, M. Yaegashi, and S. Kobayashi, *Modular robot using helical magnet for bonding and transformation*, 2017 IEEE International Conference on Robotics and Automation (ICRA), May 2017, pp. 2131–2137.
- [76] Franco Tedeschi and Giuseppe Carbone, *Design issues for hexapod walking robots*, Robotics **3** (2014), no. 2, 181–206.
- [77] Texas Instrument, <http://www.ti.com/lit/ds/symlink/lm3248.pdf>, 2018.
- [78] T. Q. Toan, A. A. Sorokin, and V. T. H. Trang, *Using modification of visibility-graph in solving the problem of finding shortest path for robot*, 2017 International Siberian Conference on Control and Communications (SIBCON), June 2017, pp. 1–6.
- [79] Irfan Ullah, *Optimal synthesis of mechanisms for path generation using fourier descriptors and global search methods*, Journal of Mechanical Design, Decem 1997, pp. 504–510.
- [80] P.J. van Laarhoven and E.H. Aarts, *Simulated annealing: Theory and applications*, Mathematics and Its Applications, Springer Netherlands, 1987.
- [81] Wendy Paola Villca Tapia, *Robótica*, Revista de Información superior, Tecnología y Sociedad (2008), 132 – 136 (es).
- [82] Vishay, <https://www.vishay.com/docs/83462/vcnl4010.pdf>, 22 de enero de 2018, 2018.
- [83] V. Vonásek and J. Faigl, *Evolution of multiple gaits for modular robots*, 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Dec 2016, pp. 1–8.
- [84] V. Vonásek, S. Neumann, D. Oertel, and H. Wörn, *Online motion planning for failure recovery of modular robotic systems*, 2015 IEEE International Conference on Robotics and Automation (ICRA), May 2015, pp. 1905–1910.

- 
- [85] V. Vonäsek, L. Winkler, J. Liedke, M. Saska, K. Kosnar, and L. Preucil, *Fast on-board motion planning for modular robots*, 2014 IEEE International Conference on Robotics and Automation (ICRA), May 2014, pp. 1215–1220.
- [86] W. Wang and Z. Lei, *Trajectory planning for a reconfigurable mobile robot in mobile manipulating mode*, Proceeding of the 11th World Congress on Intelligent Control and Automation, June 2014, pp. 42–47.
- [87] Thomas Weise, *Global optimization algorithms-theory and application*, Self-published **2** (2009).
- [88] Wikipedia contributors, *Serial peripheral interface*, 2018, [Online; accessed 5-April-2018].
- [89] Thelma L. Williams and Karen A. Sigvardt, *Intersegmental phase lags in the lamprey spinal cord: Experimental confirmation of the existence of a boundary region*, Journal of Computational Neuroscience **1** (1994), no. 1, 61–67.
- [90] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, *Design of a modular snake robot*, 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2007, pp. 2609–2614.
- [91] QiDi Wu, ChengJu Liu, JiaQi Zhang, and QiJun Chen, *Survey of locomotion control of legged robots inspired by biological concept*, Science in China Series F: Information Sciences **52** (2009), no. 10, 1715–1729.
- [92] W. Wu, Y. Guan, Y. Yang, and B. Dong, *Multi-objective configuration optimization of assembly-level reconfigurable modular robots*, 2016 IEEE International Conference on Information and Automation (ICIA), Aug 2016, pp. 528–533.
- [93] A.U. Yeregui and Universidad Complutense de Madrid. Facultad de Ciencias Físicas. Departamento de Física de Materiales, *El efecto hall cuántico y las oscilaciones shubnikov-de haas en el gas de electrones bidimensional*, Universidad Complutense de Madrid, 1996.
- [94] M. Yim, W. m. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, *Modular self-reconfigurable robot systems [grand challenges of robotics]*, IEEE Robotics Automation Magazine **14** (2007), no. 1, 43–52.
- [95] M. Yim, Ying Zhang, K. Roufas, D. Duff, and C. Eldershaw, *Connecting and disconnecting for chain self-reconfiguration with polybot*, IEEE/ASME Transactions on Mechatronics **7** (2002), no. 4, 442–451.
- [96] Mark Yim, *Locomotion of a unit modular reconfigurable robot*, Tesis for a degree of a doctor (1994).
- [97] Jin yong Ju, Wei Li, Xue-Feng Yang, Yu-Qiao Wang, and Yu-Fei Liu, *Electromechanical coupling vibration characteristics of an ac servomotor-driven translational flexible manipulator*, International Journal of Advanced Robotic Systems **13** (2016), no. 6, 1729881416662792.
- [98] H. Yu, W. Guo, J. Deng, M. Li, and H. Cai, *A cpg-based locomotion control architecture for hexapod robot*, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nov 2013, pp. 5615–5621.

- 
- [99] G. Zabala, *Robotica*, Gradi, 2007.
- [100] D. Zhang, Q. Xiao, Z. Cao, R. Huang, and Y. Fu, *Smooth transition of the cpg-based controller for snake-like robots*, 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dec 2017, pp. 2716–2721.
- [101] N. Zhang, Y. Zhang, C. Ma, and B. Wang, *Path planning of six-dof serial robots based on improved artificial potential field method*, 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dec 2017, pp. 617–621.
- [102] W. Zhu, X. Guo, and Y. Fang, *Design of a modular snake robot and control with internet of things*, 2017 Chinese Automation Congress (CAC), Oct 2017, pp. 850–854.
- [103] A. B. Zia, S. O. Ejaz, S. Abbas, U. Ikram, and S. ur Rehman, *Mu-cubes: Modular, cube shaped, and self-reconfigurable robots*, 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), Nov 2017, pp. 1–4.