

Desarrollo de una aplicación para simular los movimientos del robot modular

EMERGE en configuración tipo cadena

Development of an application for simulating the movements of the modular robot

EMERGE in chain-type configuration

Angie Alexandra Arenas Morantes* Angie Paola Rincón Cevallos Luis Fernando**

Pedraza* Henry Hernández Martínez******

Para citar: A. A. Arenas-Morantes, A. P. Rincón Cevallos, L. F. Pedraza, H. Hernández-Martínez “Desarrollo de una aplicación para simular los movimientos del robot modular EMERGE en configuración tipo cadena”. *Revista Vínculos*, vol. 17, no. 2, julio-diciembre de 2020, pp. XX-XX. DOI: <https://doi.org/10.14483/2322939X.17352>

Resumen: Se presenta una aplicación basada en software de simulación, la cual renderiza los movimientos del robot modular EMERGE en configuraciones de tipo cadena de una dimensión. Además, implementa una estrategia de control de movimiento que estima las posiciones apropiadas de cada una de las articulaciones del robot, a fin de reproducir un movimiento coordinado y continuo, teniendo en cuenta un ambiente y morfología previamente establecidos y representados en un entorno virtual 3D. Dicha propuesta permite minimizar el desgaste de la estructura mecánica del robot y evitar posibles daños cuando se desee experimentar con el robot, los resultados muestran que el simulador tiene un margen de error menor al 6% en comparación con el tiempo y distancia recorridos por el robot real.

□ Tecnóloga en electrónica, Universidad Distrital Francisco José de Caldas. Correo electrónico: aaarenasm@correodistrital.edu.co ORCID: <https://orcid.org/0000-0001-7608-0332>

** Tecnóloga en electrónica, Universidad Distrital Francisco José de Caldas. Correo electrónico: aprinconc@correodistrital.edu.co ORCID: <https://orcid.org/0000-0001-8156-6097>

*** Doctor en ciencias de la computación; Ingeniero electrónico, Universidad Distrital Francisco José de Caldas Profesor titular, Universidad Distrital Francisco José de Caldas. Correo electrónico: lfpedrazam@udistrital.edu.co ORCID: <https://orcid.org/0000-0002-0997-6478>

**** Magister en ingeniería de telecomunicaciones, Universidad Nacional de Colombia; Ingeniero en Control, Universidad Distrital Francisco José de Caldas; Profesor asociado, Universidad Distrital Francisco José de Caldas. Correo electrónico: heahernandezm@correo.udistrital.edu.co ORCID: <https://orcid.org/0000-0002-2323-0242>

Palabras clave: locomoción, robot modular, simulador, tablas de movimiento.

Abstract: This document presents an application based in software of simulation, which renders modular robot EMERGE movements in configuration string type of one dimension. In addition, implemented a strategy of control of movements that estimate the position suitable of each articulation of robot and producing continued and coordinated movements, taking into account environment and morphology previously established and represented in a 3D virtual environment. This proposal allows minimizing wear to mechanical structure of itself and prevent possible damage when experimenting with the robot, the results indicate that the simulator has a margin of error of less than 6% compared to the time and distance travelled by the real robot.

Key Words: Locomotion, modular robot, simulator, tables of movement.

1. Introducción

Un software de simulación imita procesos de diferentes tipos con el fin de proporcionar una representación de señales de sistemas reales bajo ciertas condiciones, cumpliendo así con un objetivo [1-2]. Este tipo de herramienta es implementada para: predecir, evitar desgaste y experimentar con diseños de hardware a un costo más bajo [3].

Algunos casos particulares de simuladores de robots son [4], [5] y [6]. En el primer caso, [4] permite al usuario diseñar modelos de robots y programar el control para su locomoción; en el segundo caso [5], la plataforma es capaz de simular en tiempo real grandes enjambres de robots; en el último caso [6], además del diseño de robots admite pruebas de regresión en distintos escenarios [7-8].

Los simuladores mencionados anteriormente (entre otros) son usados para refinar completamente la cinemática de un robot y construir conjuntos de movimientos que tengan cierto grado de precisión y estabilidad. Más específicamente en el caso de los robots modulares, los simuladores generan rutinas de movimiento colectivas coordinando las articulaciones de este, a fin de que se mueva. Aunque los simuladores facilitan la tarea de sincronizar las articulaciones del robot aún es un tema de estudio, debido a que al aumentar los grados de libertad del robot también aumenta la dificultad para encontrar conjuntos de movimientos coordinados [9-10]. Cabe mencionar que los simuladores de robots modulares

no contemplan todas las condiciones operativas de estos (estado de los sensores cuando interactúa el robot y su entorno) o físicas del entorno (temperatura y humedad del ambiente, ubicación de los obstáculos y forma del terreno); por ejemplo, en caso de presentarse un obstáculo en el entorno, el controlador del robot no es capaz de implementar una estrategia con el fin de evadirlo o sobrepasarlo.

Ahora bien, entre los robots modulares se encuentra el robot EMERGE el cual admite configuraciones de tipo cadena, este robot se basa en movimientos adaptativos que requieren de una etapa de autoaprendizaje permitiéndole generar un movimiento sincronizado en determinado entorno [11]. A medida que se mueve es posible presentar una desconexión, por esta razón pierde el avance de aprendizaje alcanzado y el robot se ve obligado a empezar de nuevo, por lo tanto, el tiempo empleado en el desplazamiento es extenso, ya que debe evaluar nuevamente su entorno y poder desplazarse. Una forma de solucionar esta limitación es hacer uso de un simulador, el cual se adapta a la morfología del robot, estima y envía los parámetros de control que requiere cada módulo individual de aquél. Es decir, dicho simulador reducirá el tiempo de desplazamiento al estimar un conjunto de movimientos adaptativo con una expresión matemática que se ajusta a la morfología actual del robot o configuración del hardware, lo cual no se consigue con un simulador convencional; en caso de desconexión dichos parámetros seguirán almacenados en el sistema de control del simulador.

El siguiente artículo muestra la solución propuesta organizada de la siguiente manera: en la sección 2 se explicará las condiciones experimentales y descripción del simulador de forma detallada; y en la sección 3 se mostrarán los resultados obtenidos al aplicar el simulador al robot real.

2. Desarrollo del sistema

En esta sección se presentan las condiciones experimentales y desarrollo del simulador del robot modular EMERGE en configuración de tipo cadena de una dimensión, además de los conceptos básicos necesarios para entender el funcionamiento del robot.

2.1. Descripción del robot EMERGE

El robot EMERGE (Figura 1) es un robot modular que incorpora módulos homogéneos acoplados entre sí, los cuales comparten información, alimentación y datos a través de un bus de cuatro hilos. En este caso, el robot se encuentra conectado en una configuración de tipo cadena de una dimensión y cada módulo tiene un controlador propio encargado de controlar

los movimientos del módulo, así como un controlador centralizado que se encarga de los parámetros de movimiento del robot; los dispositivos se comunican a través del protocolo Controller Area Network (CAN, por sus siglas en inglés: Se usa en sistemas distribuidos lo cual facilita la comunicación entre dispositivos basado en el encapsulamiento de mensajes y empaquetado), para mantener coordinación y así obtener un desplazamiento [11-14]. En la Figura 2 se evidencia el diagrama de bloques de funcionamiento del robot, el controlador del simulador se comunica con el robot EMERGE usando tecnología inalámbrica LoRa.

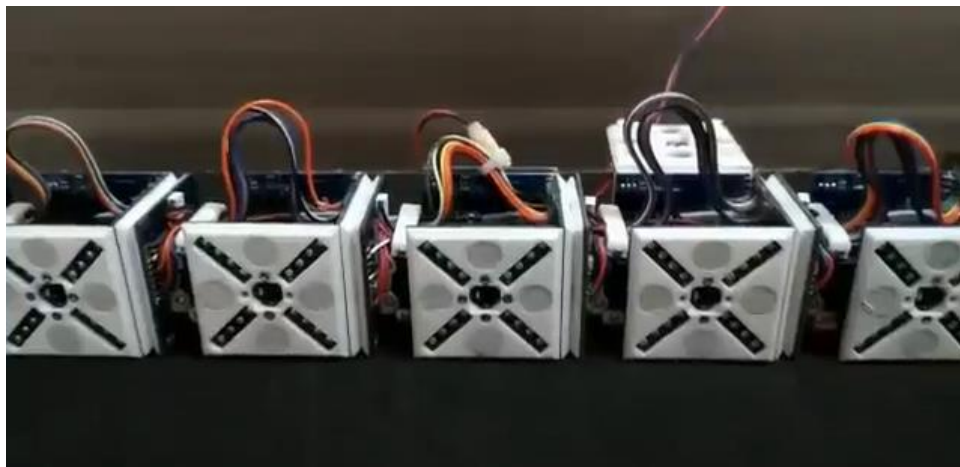


Figura 1. Robot modular EMERGE

Fuente: elaboración propia.

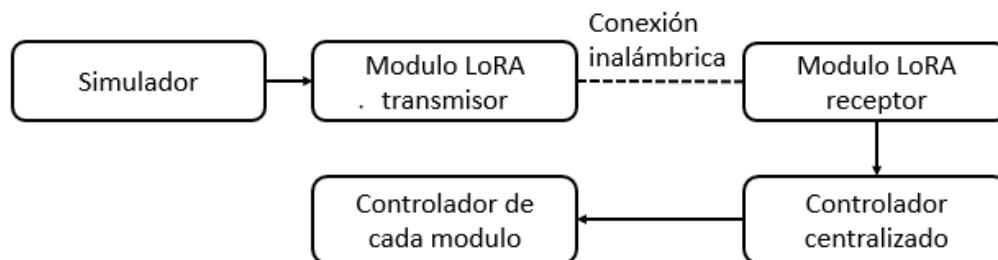


Figura 2. Diagrama de bloques de funcionamiento del robot EMERGE

Fuente: elaboración propia.

2.2 Descripción del simulador

Existen varias plataformas para realizar aplicaciones que visualizan los posibles movimientos de objetos en 3D, en este caso, Matlab es capaz de proporcionar dichos ambientes con la aplicación “3D World Editor”, en donde se pueden implementar distintas formas y especificar Virtual Reality Modeling Language (VRML, por sus siglas en inglés) o X3D, creando mundos virtuales que permiten ensamblar nodos y darles distintos aspectos; es compatible con varios programas, como por ejemplo “Solidworks” del cual es posible exportar los diseños en 3D y manipularlos según los requerimientos de cada proyecto [15].

El entorno de simulación se realizó en la aplicación “3D World Editor” de Matlab en un computador con memoria RAM de 8GB y procesador AMD A10-8700p. Esta parte es la encargada de la visualización de las interacciones dinámicas de las cadenas de módulos con el entorno y los obstáculos simulados. Tal característica le permite al usuario saber anticipadamente cómo serán los movimientos del robot real; en la Figura 3(a) se observa un módulo virtual del robot implementado para la simulación en el entorno 3D, este módulo fue modelado en el software CAD (Diseño asistido por computadora para modelar piezas en 3D y planos en 2D) “SolidWorks”; para realizar la simulación se acoplan estos módulos virtuales dando forma a la morfología del robot, como se muestra en la figura 3(b) y el movimiento se produce de acuerdo con los datos de rotación y translación recibidos del sistema de control.

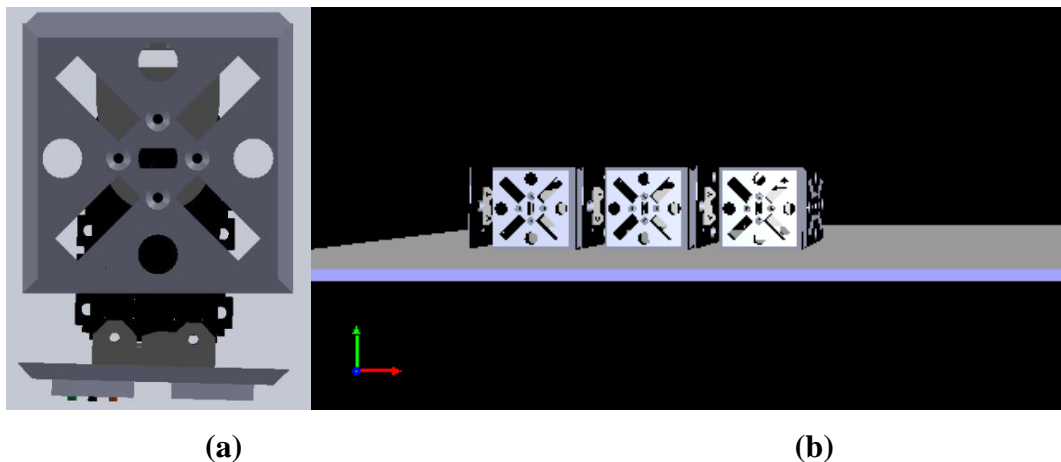


Figura 3. Diseño del mundo virtual (a) modulo modelado en SolidWorks (b) diseño y acople del entorno en 3DWorld Editor

Fuente: elaboración propia.

La aplicación desarrollada transmite las posiciones apropiadas a cada una de sus articulaciones dependiendo su morfología y el entorno en que se va a encontrar, también se puede evidenciar cómo será el movimiento del robot antes de implementarlo, lo cual ayudará a conservar la estructura mecánica del robot. La Figura 4 presenta el diagrama de bloques de la propuesta.

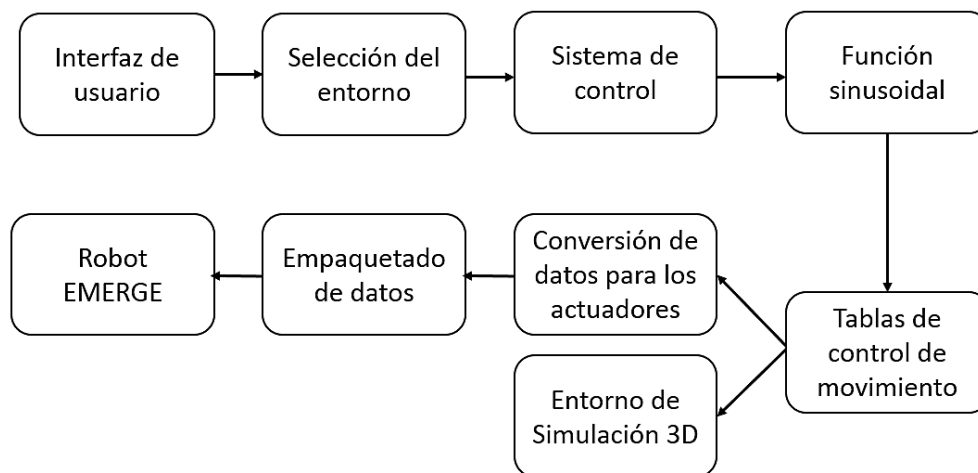


Figura 4. Diagrama de bloques del simulador

Fuente: elaboración propia.

El método utilizado para generar la secuencia de movimiento en este simulador son las tablas de control de movimiento, diseñadas a partir de una función seno modificando su amplitud, frecuencia y desfase para cada uno de los módulos. Como resultado de esta función se obtienen valores entre 0 y π (0° a 180°) y se debe realizar la conversión para estimar el respectivo dato de movimiento del actuador de cada módulo, el cual recibe valores entre 0 y 1024, como se muestra en la Figura 5. Cabe mencionar que la conversión está limitada en un rango de 300 a 700 (53° a 123°), ya que valores menores o superiores a este rango podrían comprometer la estructura mecánica del robot.

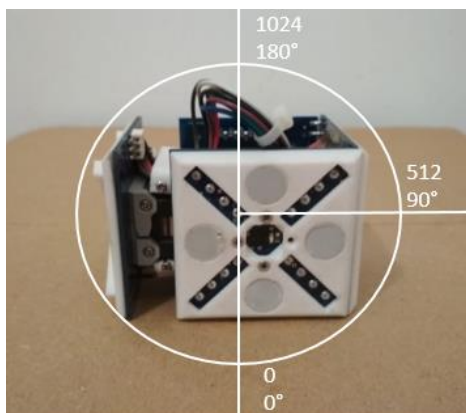


Figura 5. Rangos de movimiento del actuador de un módulo

Fuente: elaboración propia.

Las tablas de control son usadas para dar el movimiento a cada una de las articulaciones implementadas en el robot, estos movimientos son determinados previamente y se almacenan en un vector formando una secuencia. Como se muestra en el Algoritmo 1, esta rutina se guarda en el sistema de control y es la que envía la posición adecuada en que deben estar los actuadores en determinado momento; cuando finalice se ejecutará nuevamente, generando un movimiento repetitivo [16-17]. Los parámetros de control de movimiento son empaquetados colocando el nombre asignado a cada módulo, seguido de la posición del actuador con cuatro cifras (0000 hasta 1024) y finalizando con numeral. Los datos son transmitidos al robot real cuando se requiera y el robot EMERGE efectuará la locomoción a partir de las posiciones recibidas a través del módulo LoRa.

Selección de Entorno

Iniciar variables

Iniciar puerto serie

Tiempo = 0

Valores_S = [vector de posición módulos para el robot real]

Valor función seno = [valores módulos realidad virtual]

Función Seno ()

Dependiendo de la selección de entorno se genera la función seno ()

Valor_módulo1=Sen (x+ θ_1)

$Valor_módulo2 = Sen(x + \theta_2)$
 $Valor_módulo3 = Sen(x + \theta_3)$
 $Valor_módulo4 = Sen(x + \theta_4)$
 $Valor_módulo5 = Sen(x + \theta_5)$
Llenar vector [Valor función seno]
Exportar valor función seno ()

Función entorno ()

Cargar entorno virtual

Cargar valor función seno

Mientras tiempo < 5000 haga

Enviar valores de realidad virtual [valor de la función seno]

Conversión de valores realidad virtual a módulos = $(1024 * Valor_módulo_{1,2,3,4,5}) / \pi$

Enviar valores realidad virtual al módulo LoRa por puerto serial [Valores_S]

Ejecutar la rutina de movimiento durante 500 milisegundos

Tiempo ++

Fin Mientras

Algoritmo 1. Sistema de control

Fuente: elaboración propia.

La interfaz de usuario del simulador, como se muestra en la Figura 6, tiene una pantalla de inicio en la que se puede seleccionar la morfología del robot y el entorno en el cual hará su recorrido. Están disponibles tres morfologías prediseñadas conformadas entre tres y cinco módulos, y se cuenta con tres entornos: una superficie plana, obstáculo tipo escalera y camino en forma de L. Teniendo en cuenta la opción que seleccione el usuario, el software procede a reproducir la rutina de movimiento predefinida en el sistema de control con las posiciones apropiadas para los actuadores del robot, estos datos son representados gráficamente en un entorno virtual y en caso de que el usuario desee, tendrá la opción de enviarlos al robot real por medio de un dispositivo que implementa tecnología LoRa.

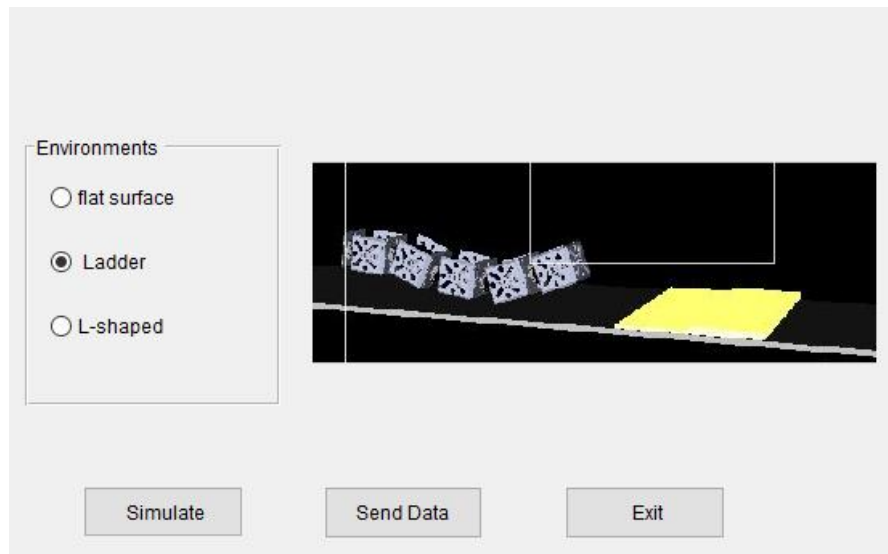


Figura 6. Interfaz de usuario del simulador

Fuente: elaboración propia.

3. Resultados y discusión

La estrategia de movimiento implementada permitió estimar las posiciones de los actuadores del robot EMERGE, de tal forma que este lograra desplazarse con un movimiento coordinado en los entornos según las morfologías predefinidas. En cada uno de los entornos propuestos el robot debe recorrer una distancia de 1m; en el caso del entorno escalera el robot logra superar satisfactoriamente el obstáculo y en el camino en forma L el robot realiza un giro de 45 grados hacia la derecha cuando ha avanzado 0,5m. El giro se consigue cuando los módulos que están de forma vertical realizan tracción sobre el terreno (Figura 9) y el módulo de la mitad gira sobre su propio eje para que le permita a la estructura del robot rotar. En el entorno de simulación se asumen las condiciones de los entornos reales sin tener en cuenta un coeficiente de fricción. En las tablas 1 y 2 se presentan los tiempos y distancias de recorrido del robot en cada entorno y en las figuras 7, 8 y 9 se muestran las morfologías mencionadas y la representación gráfica de la interacción del robot en el entorno de simulación 3D, la cual fue similar a los movimientos reales del robot.

Tabla 1. Tiempos de recorrido en cada entorno

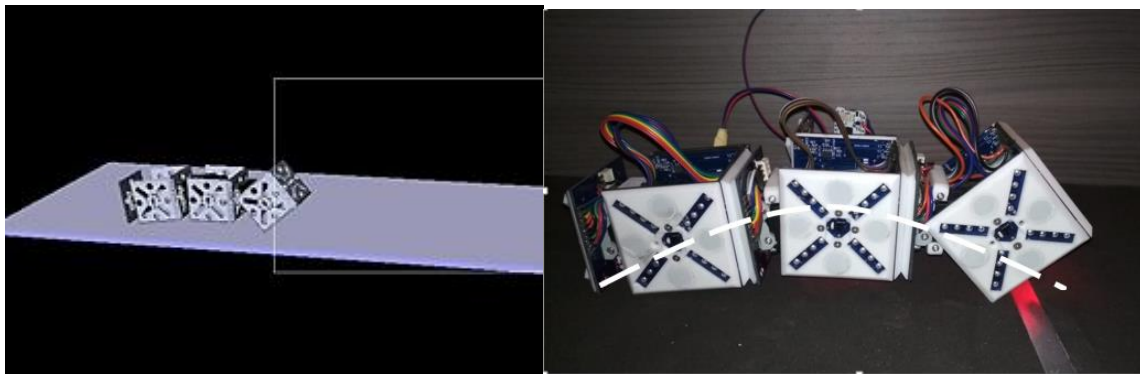
Entorno	Tiempo estimado por el simulador	Tiempo en entorno real	Error
Superficie plana	11min	11,6min	5,45%
Obstáculo escalera	13min	13,75min	5,76%
Camino en forma L	13min	13,7min	5,38%

Fuente: elaboración propia.

Tabla 2. Distancia de recorrido en cada entorno

Entorno	Distancia estimada por el simulador	Distancia en entorno real	Error
Superficie plana	1m	0,98m	2%
Obstáculo escalera	1m	0,94m	6%
Camino en forma L	1m	0,97m	3%

Fuente: elaboración propia.



(a)

(b)

Figura 7. Configuración del robot con tres módulos: (a) Entorno de simulación superficie plana y robot (b) Robot real

Fuente: elaboración propia.

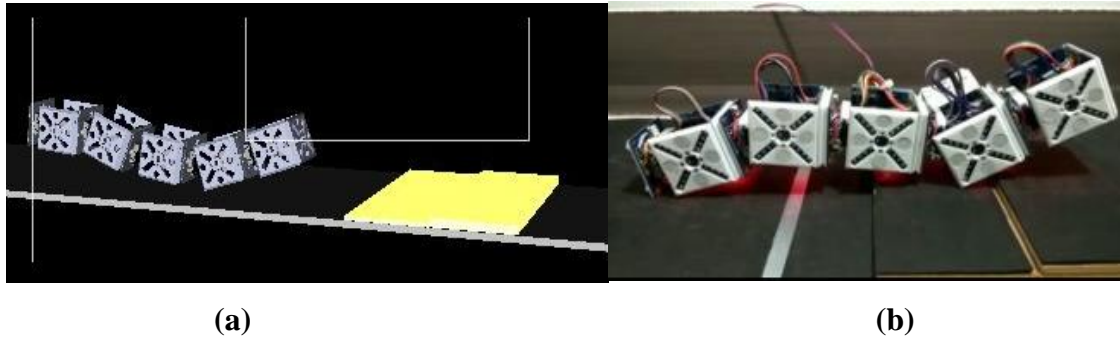


Figura 8. Configuración del robot con 5 módulos: (a) Entorno de simulación escalera y robot (b) Robot real

Fuente: elaboración propia.

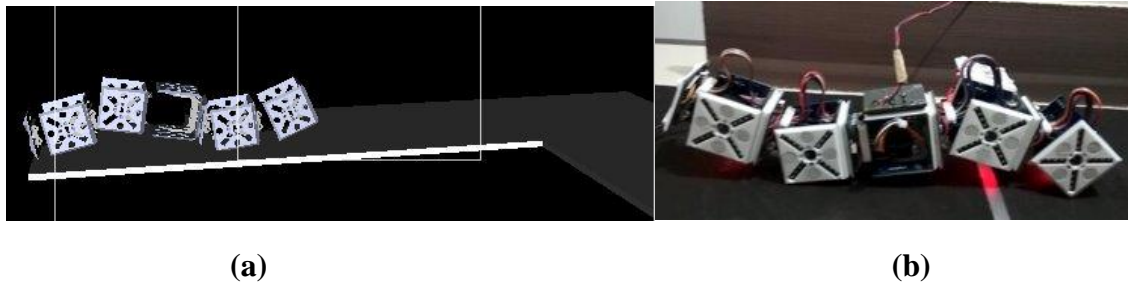


Figura 9. Configuración del robot con 5 módulos: (a) Entorno de simulación camino L y robot (b) Robot real

Fuente: elaboración propia.

En la Figura 10 se muestra la función seno implementada a los módulos con un desfase de $2\pi/5$ cada una. A partir de los resultados obtenidos de esta función se realizó la conversión de los datos para enviar el valor correspondiente a cada actuador del robot con un retardo de 100ms.

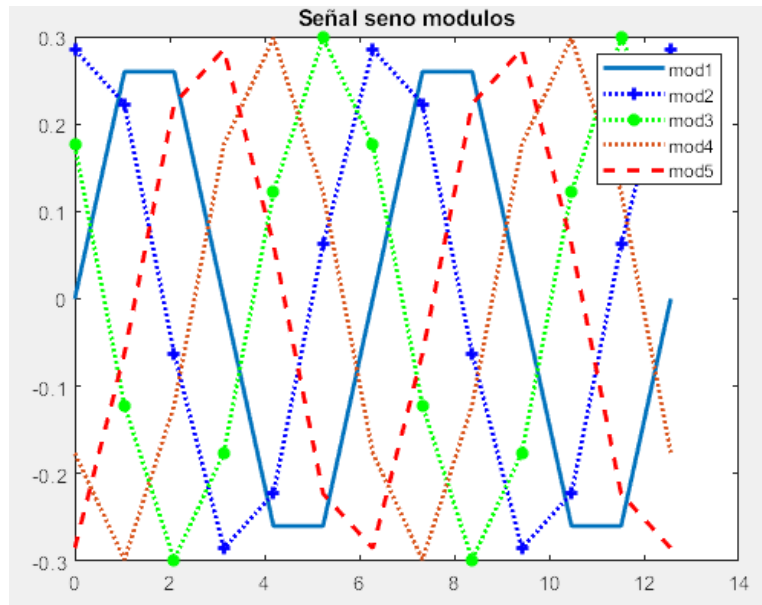


Figura 10. Segmento de la señal sinusoidal enviada a la morfología de cinco módulos

Fuente: elaboración propia.

4. Conclusiones

El software de simulación es una herramienta que prevé el comportamiento del robot en el entorno en que se encuentre sin necesidad de conectar el robot real, como se observa en las figuras 7, 8 y 9, los movimientos del robot en el simulador son similares a los movimientos que realiza el robot real, lo cual ayuda a minimizar el desgaste de la estructura mecánica del robot y evita posibles daños de este. En las tablas 1 y 2 se evidencia que la distancia y el tiempo de recorrido estimado por el simulador no superan el 6% de margen de error, comparados con el recorrido del robot real.

El método utilizado para estimar los parámetros de movimiento (Figura 10) se acondicionó de tal forma que se obtiene un rango de valores aceptables según las características mecánicas del actuador de los módulos del robot. Los parámetros de movimiento son almacenados en el sistema de control, por tanto, aunque el robot se desconecte o falle la comunicación, los datos se podrán enviar nuevamente para que este termine su recorrido.

Dicha estrategia facilita el trabajo con el robot modular EMERGE en configuración tipo cadena de una dimensión y abre camino para mejorar sus capacidades de locomoción empleando distintos tipos de morfologías en diferentes entornos. Además, implementa

tecnología LoRA, que permite el envío y recepción de información, de largo alcance con alta tolerancia a interferencias, lo cual aumentará la productividad en las tareas que el robot realice.

Referencias

- [1] J. P. Rojas, “Entorno virtual para la simulación de un robot móvil sobre el Framework ROS”, Universidad piloto de Colombia, 2015.
- [2] L. C. Florez, “Simulador de robots moviles RMOV”, *Metall. Mater. Trans. A*, vol. 30, no. 8, p. 2221, 2004.
- [3] D. Christensen, D. Brandt, K. Stoy, and U. P. Schultz, “A unified simulator for self-reconfigurable robots”, *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 870-876, 2008. <https://doi.org/10.1109/iros.2008.4650757>
- [4] “Webots: simulador de robot”. [En línea]. Disponible en: <https://cyberbotics.com/>
- [5] “The ARGoS Website”. [En línea]. Disponible en: <https://www.argos-sim.info/>
- [6] “Gazebo”. [En línea]. Disponible en: <http://gazebosim.org/>
- [7] J. Seguido, “Análisis y comparación de las principales plataformas de simulación robótica y su integración con ROS”, Universidad Politécnica de Madrid, España, 2019.
- [8] C. Pinciroli, V. Trianni, R. O’Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, T. Stirling, Á. Gutiérrez, L. Gambardella, M. Dorigo “ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics”. *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 5027-5034, 2011.
- [9] H. S.G.M., C. A. Nelson, and P. Dasgupta, “ModRED: A modular self-reconfigurable robot for autonomous extra-terrestrial exploration and discovery”. *Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, pp. 747-754, 2011.
- [10] “DRL-The Self-Reconfiguring Robotic Molecule”. [En línea]. Disponible en: http://groups.csail.mit.edu/drl/wiki/index.php?title=The_Self-Reconfiguring_Robotic_Molecule

- [11] M. Henry and A. Hern, “Estimación de una serie de movimientos utilizando un algoritmo de optimización bio-inspirado para la operación de manera autó noma y On-Line de una plataforma Multi-Robot (Caso robot modular)”. 2018.
- [12] H. Hernández, R. Moreno, A. Faina, and J. Gómez, “Design of a bio-inspired controller to operate a modular robot autonomously”. In *Proceedings of the 16th Ibero-American Conference on Artificial Intelligence*, Peru,, 2018. https://doi.org/10.1007/978-3-030-03928-8_26
- [13] R. Moreno, C. Liu, A. Faina, H. Hernández, and J. Gómez, “The EMERGE modular robot, an open platform for quick testing of evolved robot morphologies”. In *Proceedings of GECCO '17 Companion*, Berlin, Germany, July 15-19, 2017. <https://doi.org/10.1145/3067695.3075616>
- [14] C. Liu, J. Liu, R. Moreno, F. Veenstra and A. Faina,, “The impact of module morphologies on modular robots”. *18th International Conference on Advanced Robotics*, 2017. <https://doi.org/10.1109/icar.2017.8023524>
- [15] “3D World Editor-MATLAB y Simulink”. [En línea]. Disponible en: <https://www.mathworks.com/help/sl3d/the-3d-world-editor.html>
- [16] R. Moreno, “Diseño y Simulación de un Algoritmo para el Control de un Robot Modular tipo Cadena/Design and Simulation of an Algorithm to Control a Chain Type Modular Robot”, MsC, Universidad Nacional de Colombia, 2010.
- [17] J. González Gómez “Robótica Modular y locomoción: aplicación a robots ápodos”. Tesis doctoral, Universidad Autónoma Madrid Esc. Politécnica Super, 2008.