

# Implementación de un sistema de mapeo y localización a un robot hexápodo enfocado en la exploración del entorno y monitoreo de temperatura

Cristina **Alvarado-Torres**  
Esteban **Velarde-Garcés**  
Orlando **Barcia-Ayala**

Universidad Politécnica Salesiana  
Carrera de Ingeniería Electrónica  
ECUADOR

Teléfono: (+593)984237624

correos electrónicos (email):  
malvaradot@est.ups.edu.ec  
evelarde@est.ups.edu.ec  
obarcia@ups.edu.ec

Recibido 28-02-2019, aceptado 30-04-2019.

## Resumen

Las diversas investigaciones en el campo de la robótica han permitido lograr avances significativos en el desarrollo de métodos que permitan eficientemente determinar la posición de un robot dentro de un entorno que no siempre es conocido. La localización y mapeo simultáneo (SLAM) permite determinar la posición actual de un robot y realizar un mapeo de la ruta a seguir, en el cual se detectan incluso la presencia de elementos sólidos presentes en los entornos reales (*landmarks*), que le representen al robot un cambio en su dirección durante la marcha. Esta propuesta provee el análisis de implementación del método probabilístico, filtro de partículas, que bajo condiciones determinadas asegura el correcto desempeño en escenarios reales controlados, la adquisición de información del entorno sin conexión a red por medio del muestreo de valores de temperatura a través del almacenamiento de los datos en archivos CVS y el monitoreo de la temperatura por medio de la visualización de un mapa térmico. Para el éxito de este análisis se debe asegurar la robustez de los resultados obtenidos al aplicar

estos sistemas y tener en cuenta la viabilidad de la aplicación de este trabajo para los objetivos propuestos.

**Palabras clave:** filtro de partículas, temperatura, posición, muestreo, mapa, robot hexápodo.

## Abstract

**(Implementation of a Mapping and Location System for a Hexapod Robot Focused on the Exploration of the Environment and Temperature Monitoring)**

The various investigations in the field of robotics have allowed achieving significant advances in the development of methods that efficiently allow determining the position of a robot in an environment that is not always known. Simultaneous localization and mapping (SLAM) allow determining the current position of a robot and mapping the route to be followed, in which even the presence of solid elements present in real environments (*landmarks*), which represent the robot a change in its direction during the march. This proposal provides the analysis of the implementation of the probabilistic method, particle filter, which under certain conditions ensures the correct performance in real controlled scenarios, the acquisition of information from the environment without connection to the network by means of the sampling of temperature values through storage of the data in CVS files and the monitoring of the temperature by means of the visualization of a thermal map. For the success of this analysis, it is necessary to ensure the robustness of the results obtained when applying these systems, and to consider the feasibility of the application of this work for the proposed objectives.

**Index terms:** particles filter, temperature, position, sampling, map, hexapod robot.

## 1. Introducción

La exploración ha sido una actividad necesaria para poder determinar las características del entorno que nos rodea lo cual incluso, requiere de acceso a zonas cuyo ambiente puede hacer peligroso el trabajo de muestreo a un ser humano [1]; ahora

la robótica nos permite alcanzar ese objetivo a niveles tan avanzados que tenemos la posibilidad de desarrollar algoritmos de navegación autónoma, así como realizar muestreos de diferentes variables del entorno por medio de sensores [2] como: temperatura, humedad, presión, etc.; la localización y el mapeo para la aplicación en la robótica móvil ha sido de gran ayuda para determinar la ubicación dentro de una zona en algunos casos desconocida, esto es posible por medio del desarrollo de métodos heurísticos aplicando la probabilidad para lograr este propósito [3], lo cual ha permitido mirar hacia un nuevo horizonte el problema de navegación autónoma con el desarrollo del SLAM (localización y mapeo simultáneo).

Con las investigaciones y aplicaciones de diversos métodos de mapeo y localización [4], podemos obtener robots autónomos que puedan lograr realizar trabajos que resulten de cierta manera más complicados para el hombre reduciendo el tiempo de ejecución y aumentando la eficiencia de este [5], sin embargo, la aplicación de este sistema requiere el uso de sensores que permitan interactuar con su entorno [2] los cuales generan ruido durante las mediciones ocasionando un cierto nivel de error en los resultados emitidos, de la misma manera para obtener datos del entorno se debe almacenar los valores extraídos por el sensor de temperatura y humedad por medio de una adquisición de datos offline, de esta manera, se puedan extraer dicha información de forma remota una vez el robot disponga de una conexión de red confiable y se realicen la gráfica temperatura vs. posición en 2D que permite analizar mejor los resultados de la exploración.

Es condición necesaria que el robot aparte de ser autónomo pueda explorar zonas de difícil acceso lo cual requiere de una tecnología capaz de identificar las zonas de exploración y crear mapas que lo ayuden a determinar rutas apropiadas durante la marcha, esto es posible con la implementación de algoritmos de localización y mapeo simultáneos (SLAM) [4] que ayudan a desarrollar aplicaciones diversas, en este caso en particular se utiliza el método de filtro de partículas para crear los mapas del entorno y disminuir la posibilidad de extravío del dispositivo, así como recolectar la mayor información posible de la zona explorada. En la Fig. 1 se muestra la detección de landmarks o puntos de referencia los cuales son representados por los puntos naranjas captados por el robot para desviar su marcha evitando ese obstáculo, por medio de la dispersión de las partículas de remuestreo en color verde y las partículas originales de color magenta, el robot es representado por el círculo azul, el cual muestra su orientación por medio de un vector de posición.

Es importante considerar que la estabilidad del robot usado [6] debe contar con una robustez adecuada en su estructura, así

como la estrategia adecuada para integrar los sistemas necesarios para la exploración, la cantidad de memoria de almacenamiento como el tipo de dispositivo de procesamiento para el sistema. Por esa razón, el propósito de este análisis es establecer los parámetros apropiados del filtro de partículas para aplicarlos al robot hexápodo y determinar la estrategia usada para generar mapas de temperatura por medio del uso del sistema de almacenamiento de datos de temperatura y posición y así realizar un muestreo de la zona de exploración usando dispositivos de bajo costo como el sensor DHT-11.

## 2. Descripción general del sistema de exploración

El sistema de exploración propuesto en este documento consta de dos etapas importantes con las cuales se desea hacer un análisis de la aplicación del SLAM [7] y la adquisición de datos en el modelo de robot six-hexapod de ezrobot [6], a continuación se detallan las siguientes etapas:

- Localización y mapeo simultáneo. Esta etapa se encarga de guiar al robot a través de un entorno de prueba limitado cuyos landmarks han sido previamente establecidos, ese proceso se lo realizó desde Python usando una tarjeta Raspberry Pi 3 modelo B [8].
- Adquisición de datos de temperatura. Por medio de un sensor DHT-11 se realiza el muestreo de variables de temperatura recolectadas durante todo el proceso de exploración, las cuales son almacenadas en archivos CVS para su posterior lectura y descarga a un ordenador que mantiene una conexión inalámbrica con la tarjeta Raspberry Pi instalada en el robot.

En la Fig. 2 se muestra la estructura básica de la implementación realizada en el robot hexápodo y su funcionamiento.

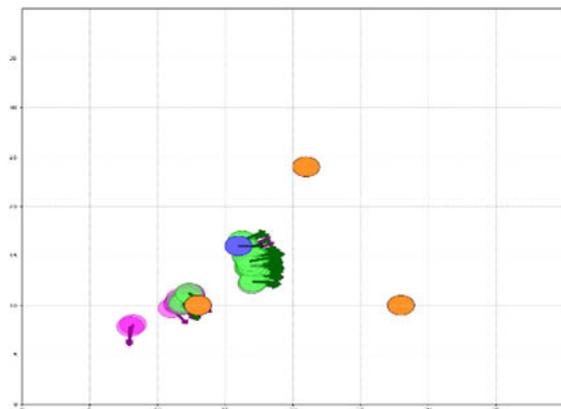


Fig. 1. Detección de landmarks por parte del robot.

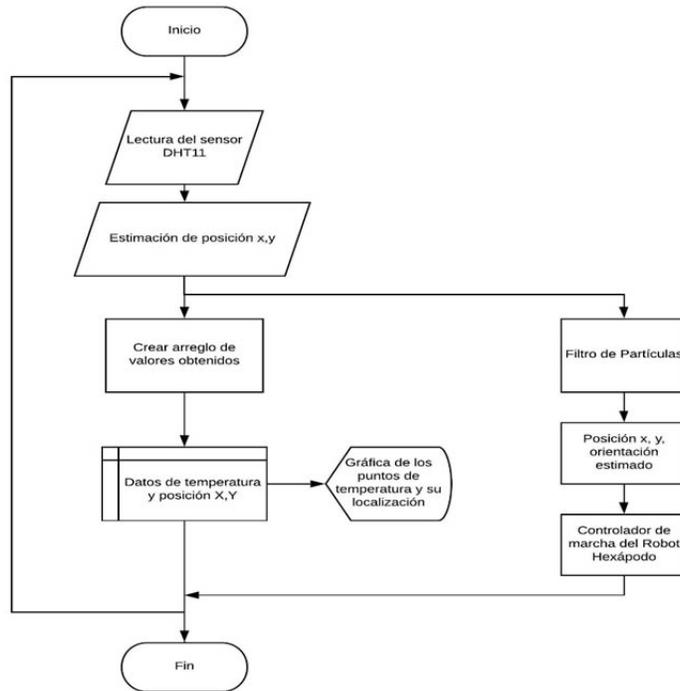


Fig. 2. Diagrama de flujo general

## 2.1. Filtro de partículas

Este filtro es obtenido por sustitución el movimiento probabilístico apropiado y los modelos perceptivos en el algoritmo de filtros de partículas, este representa la creencia  $bel(x_t)$  por un conjunto de partículas  $Mx_t = \{x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}\}$ .

El modelo de medición se aplica en línea para determinar el peso de la partícula. La creencia inicial  $bel(x_0)$  se obtiene al generar  $M$  partículas al azar de la distribución previa  $p(x_0)$ , y asignado el factor de importancia uniforme  $M - 1$  a cada partícula.

Es sencillo aplicar el algoritmo para el escenario de localización basado en puntos de referencia.

Una estrategia común para establecer  $M$  es mantener el muestreo hasta que lleguen el siguiente para  $ut$  y  $zt$ . De esta manera, la implementación es adaptativa con respecto a los recursos informáticos: cuanto más rápido sea el procesador subyacente, mejor será el algoritmo de localización. Sin embargo, se debe tener cuidado de que el número de partículas permanezca lo suficientemente alto como para evitar la divergencia del filtro. Este filtro tiene una característica no paramétrica en sus aproximaciones, puede representar distri-

buciones de probabilidad multimodal complejas y combinarlas perfectamente con distribuciones centradas al estilo gaussiano. A medida que se adquiere la posición, las partículas en otros lugares probablemente no desaparezcan prácticamente. En algún momento, las partículas solo "sobreviven" cerca de una pose única, y el algoritmo no puede recuperarse si esta postura resulta ser incorrecta.

Este problema puede resolverse mediante una heurística que consiste en una inyección de partículas aleatorias que se pueden ajustar matemáticamente suponiendo que el robot podría ser secuestrado con una pequeña probabilidad, generando así una fracción de estados aleatorios en el modelo de movimiento. Sin embargo, incluso si el robot no es secuestrado, las partículas aleatorias agregan un nivel adicional de robustez. Una forma de implementar esta idea es monitorear la probabilidad de las mediciones del sensor:

$$p(Z_t | Z_{1:t-1}, u_{1:t}, m) \quad (1)$$

y relacionar cada una de ellas con la probabilidad de medición (que se ha facilitado a partir de los datos). En los filtros de partículas, una aproximación a esta cantidad se obtiene fácilmente a partir del factor de importancia, ya que, por definición, el peso importante es una estimación estocástica de esta probabilidad. El valor medio:

$$\frac{1}{M} \sum_{m=1}^M \omega_t^{[m]} \sim p(Z_t | Z_{1:t-1}, u_{1:t}, m) \quad (2)$$

Por lo general, es una buena idea suavizar esta estimación promediando varios pasos de tiempo. Existen múltiples razones por las cuales la probabilidad de medición puede ser baja, además de una falla de localización. La cantidad de ruido del sensor puede ser anormalmente alta, o las partículas pueden dispersarse durante una fase de localización global. Por estas razones, es una buena idea mantener un promedio a corto plazo de la probabilidad de medición y relacionarlo con el promedio a largo plazo al determinar el número de muestras aleatorias [3].

## 2.2. Mapeo de temperatura

Los sistemas de adquisición de datos obtenidos del entorno por robots de exploración representan a una de las aplicaciones más necesarias para realizar muestreos en diversos campos de estudio. Diversos trabajos [9], [10] relacionados a este tipo de aplicación se enfocan en la comunicación con la

cual realizan las visualizaciones de los datos en tiempo real, cuyos datos se muestran en relación con el tiempo de muestreo, sin embargo, en situaciones en donde la fase explorativa se la realiza en zonas con presencia de algún tipo de interferencia en la señal resulta complicado la comunicación entre dispositivos. Por medio de este trabajo se realiza una investigación experimental con la cual se analizará esta metodología de adquisición de datos offline de sensores y representación gráfica de la localización de los puntos de muestreo por medio de un sensor de temperatura DHT-11 corregir nombre del sensor con una tarjeta de procesamiento Raspberry Pi 3 modelo B.

Para realizar el muestreo debe considerarse como condición primordial los valores mínimos y máximos de temperatura a la que pueden estar expuestos los materiales del robot hexápodo, la estructura del mismo está compuesto por un material copolímero llamado ABS el cual soporta temperaturas entre  $-20^{\circ}\text{C}$  y  $70^{\circ}\text{C}$  como se indica en [11], de la misma manera los rangos de temperatura soportados por los servomotores y demás componentes electrónicos oscilan en valores similares al mencionado anteriormente. Debido a la dificultad que representa el aumento de temperatura ambiental para el robot, se ha delimitado su uso a guardar valores de temperatura desde  $-10^{\circ}\text{C}$  a  $55^{\circ}\text{C}$ .

Al disponer de una tarjeta electrónica de alta robustez las oportunidades de almacenamiento y procesamiento de datos son superiores a otros dispositivos programables. Para este caso en particular en el que se debe almacenar los datos adquiridos durante la exploración se programa en Python una función que permita guardar todos los datos en un archivo CVS (*comma-separated values*), son archivos de texto cuyos valores están separados por un delimitador. Generalmente, en la primera línea se encuentra los encabezados. Estos incluyen el nombre de cada "columna" de datos o "campos". La separación de los valores se la señala por medio de una coma [12].

### 3. Resultados

El presente sistema se desarrolló con componentes y tecnologías de uso libre y de bajo costo, para asegurar que a esta aplicación tecnológica desarrollada se le puedan realizar mantenimientos sin costos elevados y así pueda ser usado por la comunidad académica para futuras investigaciones. El terreno determinado para realizar las pruebas sobre el robot hexápodo fue de  $175\text{ cm}^2$ , el modelo original del robot six-

hexapod [6] fue ligeramente modificado debido a la necesidad de usar un procesamiento más robusto incorporando la tarjeta Raspberry Pi 3 modelo B. sobre una plataforma adaptada.

#### 3.1. Análisis de la dinámica del filtro de partícula

Por medio del algoritmo de filtro de partículas se pudo determinar una ruta real del robot durante el recorrido, en el cual se posicionaron diferentes landmarks o puntos de referencia distribuidos en todo el entorno, lo cual ha permitido que el robot logre localizarse, siempre y cuando se encuentre dentro de la zona delimitada y pueda obtener una posición de referencia con respecto a los landmarks. En la Fig. 3 se puede observar una vista de la ruta establecida por medio de valores aleatorios deposición  $x$ ,  $y$  (representada con un trazo continuo) y el ajuste de partículas que crean el mapa de recorrido del robot a partir de sus valores originales de la ruta preestablecida en la cual se ha contemplado el error del robot en su movimiento que se representa con un trazo discontinuo. La cantidad de ruido que se ha introducido en el algoritmo es de la misma magnitud que el ruido que aparece típicamente en los experimentos con robots reales sugerido en [3]. Con este gráfico se compara la posición en  $x$  real con la posición en  $x$  estimado por el filtro de partículas para evaluar la calidad de los resultados, de la misma se determinó las gráficas con la posición en  $y$  (véase en la Fig. 4) con la que se puede apreciar que a pesar del nivel de error medio calculado las estimaciones de posición del filtro son bastante sobresalientes.

Una condición necesaria para determinar el óptimo desempeño del filtro es por medio del cálculo de las partículas, de esta manera aseguramos que la estimación de posición mues-

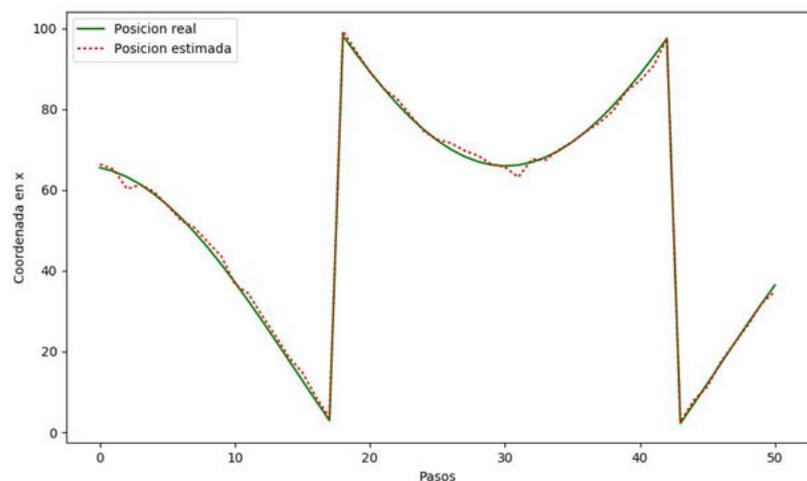


Fig. 3. Posición real en  $x$  del robot vs. Posición estimada en  $x$  por el filtro.

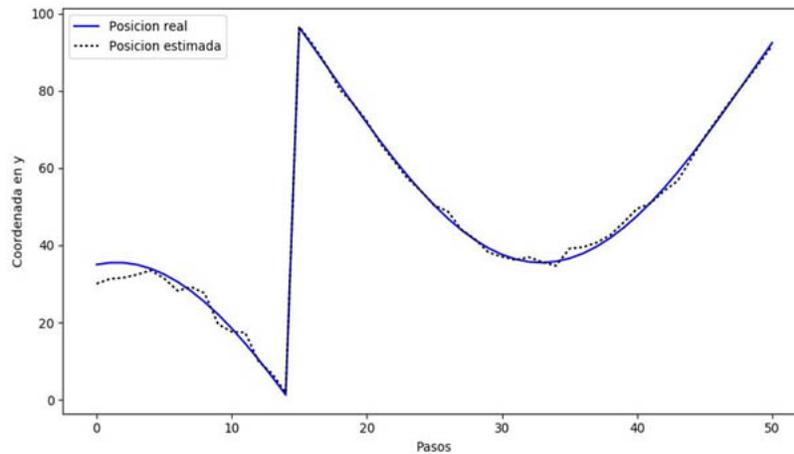


Fig. 4. Posición real en y del robot vs. Posición estimada en y por el filtro.

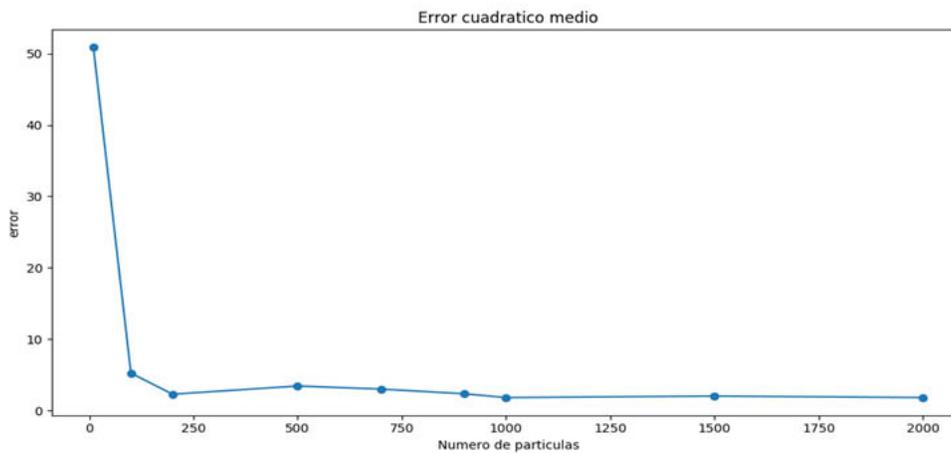


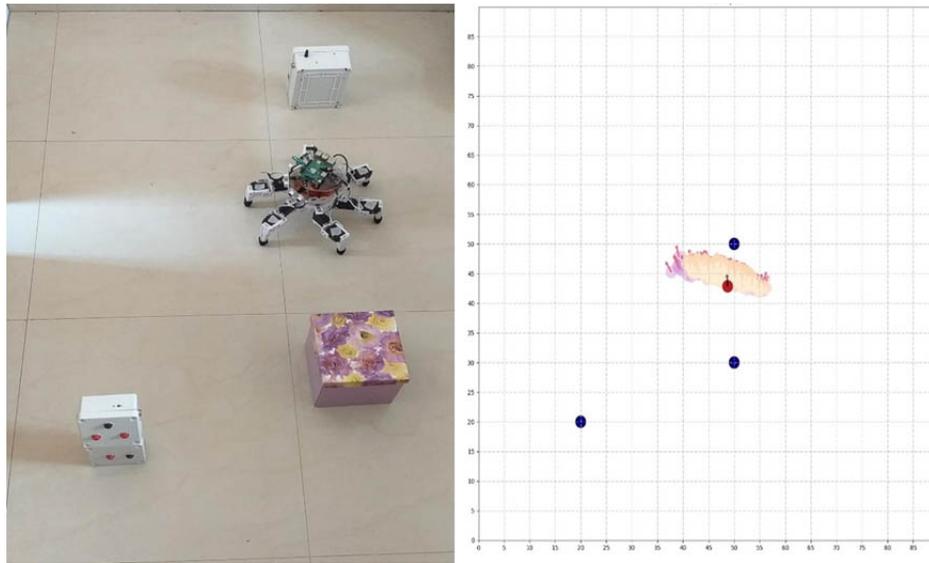
Fig. 5. Gráfica comparativa del error por número de partículas.

tre una disminución considerable del error cuadrático medio calculado a partir de la ecuación (1); se ha realizado una gráfica comparativa del error por cada cambio del valor  $M$  (número de partículas) mostrada en la Fig. 5. En esta gráfica, se puede observar claramente como decrece el error en la estimación según aumenta el número de partículas empleadas. Observando la gráfica se puede ver que a partir de 800 partículas mejora en la localización y demuestra que se pueden obtener unos buenos resultados utilizando 1000-1500 partículas.

Debido al coste computacional alto requerido por el algoritmo de SLAM, es necesario realizar cálculos del tiempo de procesamiento de la tarjeta Raspberry Pi 3 modelo B para determinar el tiempo que requiere el sistema (véase en la Tabla 1) para estimar las posiciones de acuerdo a la cantidad de partículas establecidas para su mejor desempeño.

Tabla 1. Tiempo de procesamiento de la tarjeta Raspberry Pi.

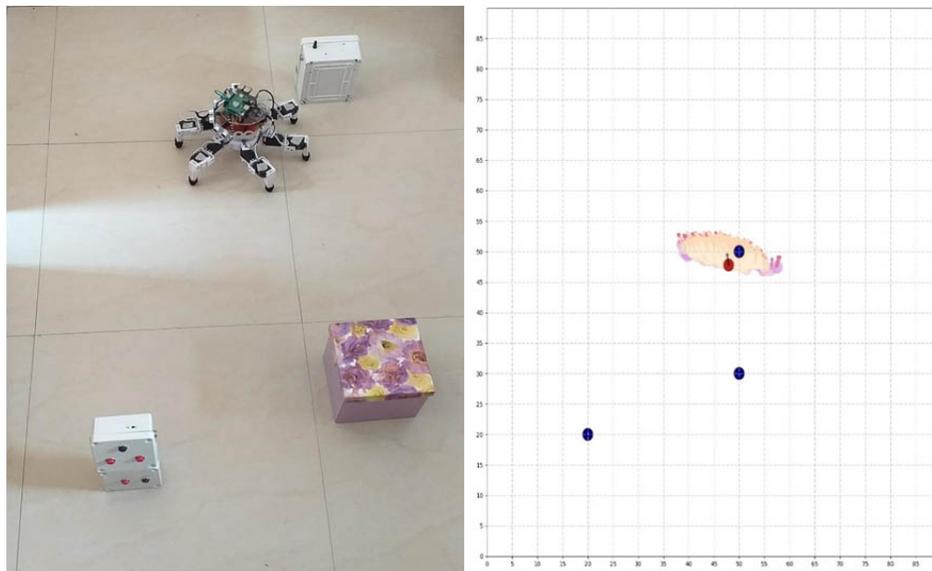
Número de partículas	Tiempo de procesamiento (s)
10	0.1737992
100	1.2880270
200	4.1797537
500	6.0745110
700	8.4064073
900	11.2124450
1000	11.5560207
1500	19.7074775
2000	28.3417411



**Fig. 6a.** Robot hexápodo antes de la actualización de posición respecto al landmark.

En las Fig. 6a y 6b se visualiza al robot en el espacio de prueba, el cual logró reconocer perfectamente el terreno al determinar su posición con respecto a los landmarks que se encontraron en el entorno, la posición de cada landmark ha sido establecido en el algoritmo como un dato constante mientras no cambie las condiciones físicas de su área de exploración, sin embargo, debido al tiempo de respuesta del sistema (véase en la Tabla 1) y a las limitaciones causadas por el ruido en la medición de los sensores que determinan

la posición del robot con respecto sistema de referencia fijo, ocasionaba una demora en su marcha principalmente durante el cálculo de las nuevas partículas, de la misma manera se visualiza los resultados de implementar el filtro de partículas en una zona pequeña de exploración ante un escenario específico y se contrasta con el comportamiento del mismo durante la simulación. La reagrupación de partículas muestra una mayor densidad cuando el robot representado por el círculo rojo se encuentra localizado con respecto a



**Fig. 6b.** Robot hexápodo después de la actualización de posición respecto al landmark.

los landmarks, los cuales se muestran con círculos de color melón alrededor del robot.

Para obtener la actualización de la posición del robot se debe considerar que la naturaleza del algoritmo requiere de al menos dos iteraciones para calcular las futuras posiciones de este, por esta razón existen ciertos márgenes de error entre la posición real y la estimada, por lo tanto, a mayor número de iteraciones o pasos que da el robot hay mayor probabilidad de error en la posición actual en la que se encuentra el mismo.

### 3.2. Análisis de mapa de temperatura

Durante la exploración el sistema de adquisición de datos de temperatura, por medio del sensor DHT-11 fue configurado de tal manera que los valores sean recopilados cada 35 segundos de acuerdo a las variables ingresadas (posición y temperatura) de forma automática en el archivo de almacenamiento CVS de manera interna. Por medio de un muestreo de prueba utilizando este sensor se comprobó que el tiempo de estabilización y climatización para determinar la temperatura correcta varía entre 35 y 40 segundos.

Cuando el robot terminó su recorrido se realizó una conexión de red con la computadora para realizar la gráfica de temperatura con los datos adquiridos, los valores de temperatura están representados por medio de una gama de colores que varían de acuerdo a la intensidad de la misma los cuales se detallan en la Tabla 2 que indican los rangos con los cuales el gráfico define el color en cada punto del recorrido como se muestra en las Fig. 7 y 8, estos datos fueron tomados con el sensor a 7 cm de la fuente de calor y con una temperatura

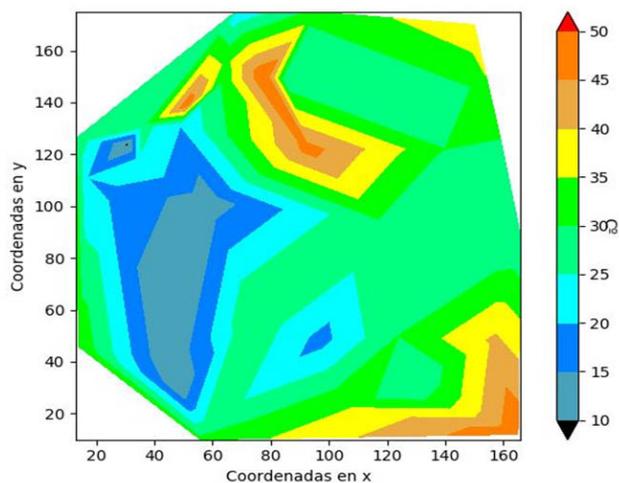


Fig. 7. Mapa de temperatura.

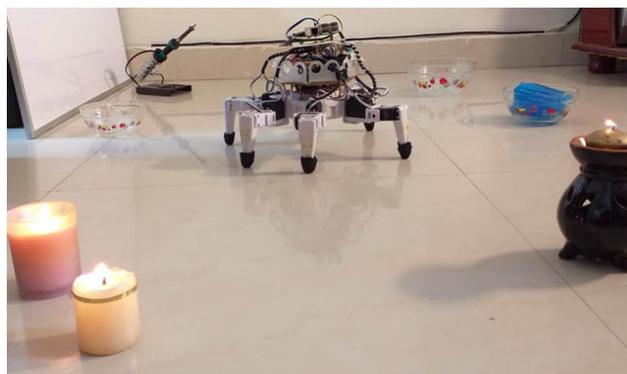


Fig. 8. Fuentes de temperatura en el entorno real.

ambiental de 30°C aproximadamente. El método utilizado para determinar las zonas de temperatura es la triangulación de superficies con la cual se trazan triángulos de acuerdo a los puntos de temperatura cuyas medidas están dentro del mismo rango de temperatura en concordancia al código de colores especificado anteriormente.

### 4. Discusión

La localización juega un papel muy importante en la navegación autónoma de robots móviles por lo cual analizar entre varias posibles soluciones a este problema demanda un estudio de las respuestas del algoritmo sobre el sistema aplicado al robot y su precisión al obtener resultados.

Con el filtro de partículas se mejora la estimación de posición del robot a partir de una posición con respecto al sistema de referencia fijo, sin embargo, el algoritmo no contempla la posi-

Tabla 2. Distribución de colores de acuerdo al nivel de temperatura.

Colores	Temperatura
Negro	$\leq 10^{\circ}\text{C}$
Celeste	$10^{\circ}\text{C} < t \leq 15^{\circ}\text{C}$
Azul	$15^{\circ}\text{C} < t \leq 20^{\circ}\text{C}$
Turquesa	$20^{\circ}\text{C} < t \leq 25^{\circ}\text{C}$
Verde Esmeralda	$25^{\circ}\text{C} < t \leq 30^{\circ}\text{C}$
Verde Limón	$30^{\circ}\text{C} < t \leq 35^{\circ}\text{C}$
Amarillo	$35^{\circ}\text{C} < t \leq 40^{\circ}\text{C}$
Mostaza	$40^{\circ}\text{C} < t \leq 45^{\circ}\text{C}$
Naranja	$45^{\circ}\text{C} < t \leq 50^{\circ}\text{C}$
Rojo	$> 50^{\circ}\text{C}$

bilidad de encontrar objetos u obstáculos que puedan no sólo impedir el movimiento del robot durante la exploración, sino que crea incertidumbre ante el posible cambio de orientación de la marcha que causaría una actualización inmediata de la posición. Debido a su aplicación en el campo de la navegación autónoma este algoritmo calcula varias posiciones o pasos que usará el robot para moverse en el entorno y actualizar las partículas, ante este escenario el algoritmo se torna un tanto rígido que no permite realizar cambios repentinos hasta concluir todo el cálculo de las iteraciones.

Considerando esta funcionalidad, el filtro de partículas, no asegura un éxito de localización del 100% debido a errores propios del redondeo como de estimaciones de sensores externos y de tolerancia a fallas de marcha propios del tipo de locomoción del robot, por lo cual para reducir el nivel de error de la posición del robot se debe añadir un factor de corrección de posición calculado a partir de las fallas mencionadas anteriormente con el fin de que el algoritmo se convierta en un supervisor avanzado de localización que tenga la característica de actualización constante.

Para poder realizar una supervisión o muestreo de elementos propios del clima como: temperatura, humedad, etc. en la zona de exploración se debe considerar que para obtener un mejor estudio de las variables que afectan la zona muestreada se requiere de sensores con una exactitud de media a alta y cuyo tiempo de estabilización en lectura de datos sea <10 segundos, de esta manera podremos obtener mejores mapas de temperatura con mayor exactitud lo cuales para diferentes aplicaciones representan una forma fácil y rápida de detectar cualquier cambio o anomalía durante su análisis. Por motivos de seguridad cuando el robot detecta la intensidad de los marcadores superior a los 55°C o inferior a los 10°C cambia la dirección de marcha a una distancia de 25 cm de los landmarks.

## 5. Conclusiones

Se determinó que con la aplicación de este sistema se ha logrado obtener resultados satisfactorios con respecto al uso del filtro de partículas para determinar el mapeo y localización (posiciones) durante recorrido del robot hexápodo con un nivel de error de estimación de posición menor al 6% y la aplicación de un número mayor o igual a 1000 partículas que permitió que se pueda la realizar la exploración del robot localizándolo en posiciones aproximadas a las calculadas por el algoritmo.

El uso del sensor DHT-11 a pesar de que logra muestrear datos de temperatura de manera correcta y su error de medición es bajo, debido al inconveniente que representa el tiempo de estabilización para obtener una nueva medida y el fallo en la

adquisición de datos que suele presentarse, es recomendable el uso de otro tipo de sensor para realizar mapas más precisos y con mayor cantidad de variables muestreadas.

Dentro de la investigación se observó que en consecuencia a la ausencia de problemas en la interferencia de señal se pudo mantener una comunicación inalámbrica entre el robot y una computadora para poder así visualizar los datos recolectados por medio del sensor DHT-11; los datos obtenidos de temperatura fueron almacenados correctamente por la tarjeta Raspberry Pi 3 modelo B y mostró una aproximación bastante buena del conjunto de valores de temperatura en todo el entorno muestreado. Con eso se determina que el uso de registros generando archivos CVS es un soporte necesario para muestreos realizados sin conexión a red. Esta propuesta genera una solución al problema de ahorro de energía permitiendo una mayor cantidad de horas de exploración. Este entorno sin conexión de red es una solución ideal para aplicarlo en ambiente de condiciones extremas como inaccesibilidad a ciertas zonas donde se necesite explorar y muestrear datos o inclusive para desarrollar un modelo de robot de rescate de bajo costo que ayude en condiciones de desastres naturales presentes en el país.

## Referencias

- [1] M. M. H. Polash *et al.*, "Explorer-0100: An autonomous next generation Mars rover," in *2017 20th International Conference of Computer and Information Technology (ICIT)*, 2017, pp. 1-7.
- [2] S. Russell, P. Norvig, *Artificial intelligence a modern approach*. Pearson, 2010.
- [3] S. Thrun, W. Burgard, D. Fox, *Probabilistic robotics*, MIT Press, 2005.
- [4] V. Narváez *et al.*, "Diseño e Implementación de un Sistema de Localización y Mapeo Simultáneos (SLAM) para la Plataforma Robótica ROBOTINO®," *Rev. Politécnica*, vol. 33, núm. 1, febrero 2014.
- [5] J. L. Li, J. H. Bao, Y. Yu, "Graph SLAM for Rescue Robots," *Appl. Mech. Mater.*, vol. 433-435, pp. 134-137, octubre 2013.
- [6] Ezrobot, "Six Hexapod - EZ-Robot" [en línea]. Disponible en: <https://www.ez-robot.com/Shop/Accessories/Details.aspx?productNumber=30>. [Consultado el: 15-11-2018].
- [7] H. Durrant-Whyte, T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robot. Autom. Mag.*, vol. 13, núm. 2, pp. 99-110, junio 2006.
- [8] E. Upton, G. Halfacree, *Raspberry Pi User Guide*. Wiley, 2012.
- [9] A. Syed, H. A. ElMaraghy, N. Chagneux, "Real-time monitoring and diagnosing of robotic assembly with

- self-organizing neural maps," in *Proceedings Real-Time Systems Symposium*, pp. 271-274, 1992.
- [10] A. Aloisio, P. Branchini, F. Cevenini, "Real-time diagnostic and performance monitoring in a DAQ environment," in *1999 IEEE Conference on Real-Time Computer Applications in Nuclear Particle and Plasma Physics. 11th IEEE NPSS Real Time Conference*. Conference Record (Cat. No.99EX295), 1999, pp. 239-242.
- [11] L. Bilurbina, F. Liesa, *Materiales no metálicos resistentes a la corrosión*, Marcombo, 1990.
- [12] R. Lemesle, A. Petitjean, *Windows PowerShell?: los fundamentos del lenguaje*, ENI, 2015.