

Original Research Article

The Implementation of Hexagonal Robot Mapping and Positioning System Focuses on Environmental Scanning and Temperature Monitoring

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

Carrera de Ingeniería Electrónica, Universidad Politécnica Salesiana, Ecuador

ABSTRACT

Various researches in the field of robotics have made great progress in developing methods to effectively determine the position of robots in unknown environments. The simultaneous localization and mapping (SLAM) task make determining the current position of the robot and performing path mapping possible. In this mapping, solid elements (landmarks) existing in the actual environment are even detected, which indicate that the direction of the robot changes during walking. This scheme provides the implementation analysis of the probabilistic particle filter method, which ensures the correct performance in the controlled actual scene under specific conditions, obtains the non-network connection environment information by storing the data in the temperature value sampling in the CVS file, and monitors the temperature measurement by displaying the heat map. Successful analysis must ensure the robustness of the results obtained when implementing these systems and take into account the feasibility of applying this work to the proposed objectives.

Keywords: Particle Filter; Temperature; Position; Sampling; Mapping; Hexagonal Robot

ARTICLE INFO

Received: Mar 22, 2021
Accepted: May 27, 2021
Available online: Jun 2, 2021

*CORRESPONDING AUTHOR

Cristina Alvarado-Torres
malvaradot@est.ups.edu.ec ;

CITATION

Alvarado-Torres C, Velarde-Garcés E, Barcia-Ayala O. The implementation of hexagonal robot mapping and positioning system focuses on environmental scanning and temperature monitoring. Journal of Autonomous Intelligence 2021; 4(1): 41-49. doi: 10.32629/jai.v4i1.493

COPYRIGHT

Copyright © 2021 by author(s) and Frontier Scientific Publishing. This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0). <https://creativecommons.org/licenses/by-nc/4.0/>

1. Introduction

Exploration has always been a necessary activity in order to be able to learn the characteristics of our surrounding environment, which even needs to enter areas where the environment may pose a risk to humans^[1]; now, robot technology enables us to achieve this goal at such an advanced level that it is possible for us to develop autonomous navigation algorithms and sample different environmental variables through temperature, humidity, pressure and other sensors^[2]; the localization and mapping of mobile robot applications can help to determine the location in the region in some unknown cases, which can be realized by developing heuristic methods^[3], which makes the autonomous navigation problem have a new vision with the development of SLAM (simultaneous localization and mapping).

Through the research and application of various mapping and positioning methods^[4], we can obtain automata robots, which can complete more complex work for human beings by reducing execution time and improving execution efficiency^[5]. The application of the system requires the use of sensors^[2] that can interact with the surrounding environment to generate noise in the measurement process, resulting in a certain degree of error in the measurement results. Similarly, in order to obtain environmental data, the values extracted by the temperature and humid-

ity sensors must be stored through the offline data acquisition of the robot once the robot establishes a reliable network connection, it can extract this information remotely and draws a two-dimensional temperature and location map to better analyze the scanning results.

Apart from the automation of the robot, another premise is the fact that the autonomous robot can explore inaccessible areas which requires a technology that can identify the scanned area and create a map to help it determine the appropriate route during walking. It is possible through the use of simultaneous positioning and mapping (SLAM)^[4] algorithm, which is helpful to develop various applications. In this special case, the particle filter method is used to create the environment map, reduce the possibility of equipment crossing the boundary, and collect as much information as possible from the scanning area. **Figure 1** shows the detection of landmarks or reference points which are represented by orange points captured by the robot to bypass the obstacle. Green represents the scattering of the resampled particles, magenta the original particles, and the robot is represented by a blue circle that displays its direction through the position vector.

It is important to consider that the stability of the robot used^[6] which must have sufficient robustness in its structure, as well as the appropriate strategy of the system required for integrated scanning and the amount of memory stored and the type of processing equipment of the system. Therefore, the purpose of this analysis is to establish the appropriate parameters of the particle filter suitable for the hexagonal robot, and to determine the strategy used to generate the temperature map by using the temperature and position data storage system, so as to re sample the scanning area using low-cost equipment such as DHT-11.

2. Overview of scanning system

The scanning system proposed in this paper consists of two important stages with which we want to make an analysis of the application of SLAM^[7] and data acquisition on the ezrobot^[6] hexapod robot model. The specific steps are as follows:

Simultaneous positioning and mapping. The task of this stage is to guide the robot through a limited test environment whose landmarks have been preset. This process is completed in Python using Raspberry Pi 3B card^[8].

Temperature data acquisition. Through the DHT-11

sensor, the temperature variables collected during the whole scanning process are sampled. These temperature variables are stored in the CVS file for subsequent reading and downloading to the computer, which maintains a wireless connection with the Raspberry Pi card installed on the robot. **Figure 2** shows the basic structure and operation realized on the hexagonal robot.

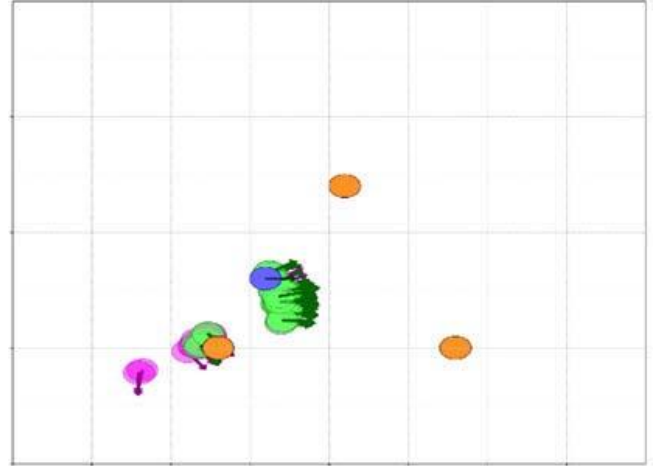


Figure 1. Robots detect landmarks.

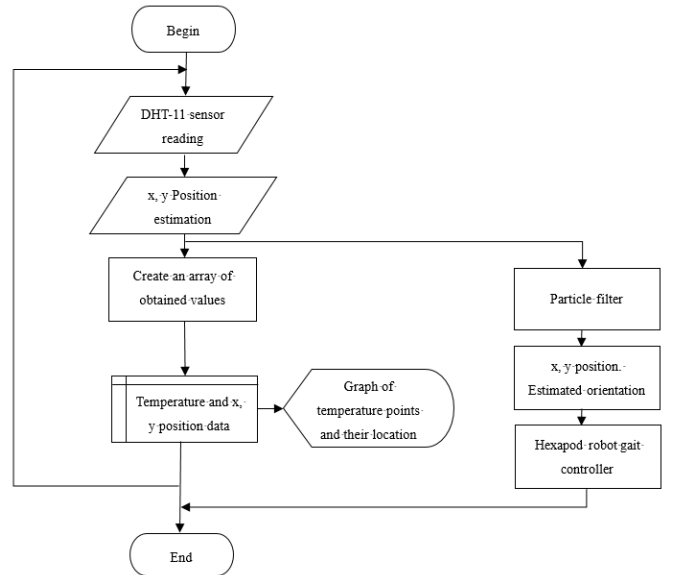


Figure 2. Overall flow chart

2.1 Particulate filter

This filter is obtained by replacing the appropriate probabilistic motion and perception model in the particle filter algorithm. It represents that the belief consists of a group of particles

$$Mxt = \{x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}\}. \quad (1)$$

The measurement model is applied to on-line measurement of particle weight. The initial belief $bel(x_0)$ is obtained by randomly generating M particles from the previous distribution $p(x_0)$ and assigning a uniform importance factor M^{-1} to each particle.

The algorithm is simple to implement and suitable for scene location based on reference points.

A common strategy for establishing M is to keep sampling until the next one reaches ut and zt . In this way, the implementation is adaptive to computing resources: the faster the underlying processor, the better the positioning algorithm. However, care should be taken to ensure that the number of particles remains high enough to avoid filter divergence. The filter has the characteristics of non-parametric approximation that can represent complex multimode probability distribution, and is perfectly combined with Gaussian central distribution. When you get this position, particles elsewhere may not disappear. In some cases, particles only “survive” near a single pose. If this pose is proved to be wrong, the algorithm cannot recover.

This problem can be solved by a heuristic algorithm, which includes injecting random particles, which can be mathematically adjusted under the assumption that the robot may be hijacked with a small probability, so as to generate the score of random state in the motion model. However, even if the robot is not hijacked, random particles will add additional robustness. One way to realize this idea is to monitor the probability of sensor measurement:

$$p(Z_t | Z_{1:t-1}, u_{1:t}, m) \quad (2)$$

And link each method to the possibility of measurement (provided according to the data). In the particle filter, the approximate value of this quantity is easily obtained from the importance factor, because according to the definition, the importance weight is a random estimation of this probability. Average:

$$\frac{1}{M} \sum_{m=1}^M \omega_t^{[m]} \sim p(Z_t | Z_{1:t-1}, u_{1:t}, m) \quad (3)$$

Usually, it's a good idea to smooth this estimate by averaging a few time steps. In addition to fault location, there are many reasons why the measurement probability

may be very low. The noise level of the sensor may be abnormally high, or the particles may be dispersed in the global positioning stage. For these reasons, when determining the number of random samples, it is best to maintain the short-term average of the measurement probability and link it with the long-term average^[3].

2.2 Temperature mapping

The data acquisition system obtained by the scanning robot from the environment represents one of the most important applications for sampling in various research fields. Various works related to this application^[9,10] focus on communication and real-time data visualization. The data is related to the sampling time. However, in the scanning stage, when it is carried out in the area with some signal interference, the communication between devices becomes complex. Through experimental research, this paper analyzes the method of correcting the sensor name by using DHT-11 temperature sensor and Raspberry Pi 3B processing card to realize off-line data acquisition and graphical representation of sampling point position.

When sampling, the lowest and highest temperature values that the hexagonal robot material may be exposed to shall be regarded as the main conditions. Its structure is composed of a copolymer material called ABS, and the temperature of the copolymer material is between -20°C and 70°C , as shown in^[11], similarly, the temperature range supported by servo motors and other electronic components is similar to the above values. Robot has been used to store temperature values from -10°C to 55°C due to its difficulties in raising ambient temperature.

Due to a highly reliable electronic card, the performance of data storage and processing is better than other programmable devices. In this special case, the data obtained during scanning must be stored in Python. A function allows all data to be saved in a CVS file (comma separated values). They are text files, and their values are separated by separators. Generally speaking, the first line is the title which includes the name of each data column or field. Values are separated by commas^[12].

3. Result

The system is developed with free and low-cost components and technologies to ensure that the developed technical applications can be maintained without increasing costs, and can be used by academia for further research. The ground tested on the hexagonal robot is 175

cm². The original model of the hexagonal robot^[6] is slightly modified because it needs to use more robust processing to embed the Raspberry Pi 3 model B card into an adaptive platform.

3.1 Dynamic analysis of particulate filter

Through the particle filter algorithm, the actual path of the robot in the moving process can be determined. In this path, different landmarks or scattered reference points are placed in the whole environment, so that the robot can locate, provided that it is located in the designated area and can obtain the reference position related to landmarks. In **Figure 3** the view of the path set can be observed through random value deposition x , y (represented by continuous strokes) and particle adjustment. These particle adjustments create the path map of the robot from the initial preset path value of the robot, in which the motion error of the robot is regarded as represented by discontinuous strokes. The amount of noise introduced into the algorithm is of the same magnitude as the noise

typically appears in experiments with real robots suggested in^[3]. The figure compares the actual x position with the x position estimated by the particle filter to evaluate the quality of the results and determine the figure of y position (see **Figure 4**). It can be seen that although the average error level is calculated, the position estimation of the filter is quite prominent.

Through the calculation of particles, a necessary condition for determining the optimal design of filter is given, which ensures the estimation of sampling point position and greatly reduces the mean square error of equation (1); the error comparison diagram is made for each change of m value (particle number) shown in the **Figure 5**. In this figure, it can be clearly observed that the estimation error decreases with the increase of the number of particles used. It can be seen from the figure that the localization of 800 particles is enhanced, and it shows that good results can be obtained by using 1,000-1,500 particles.

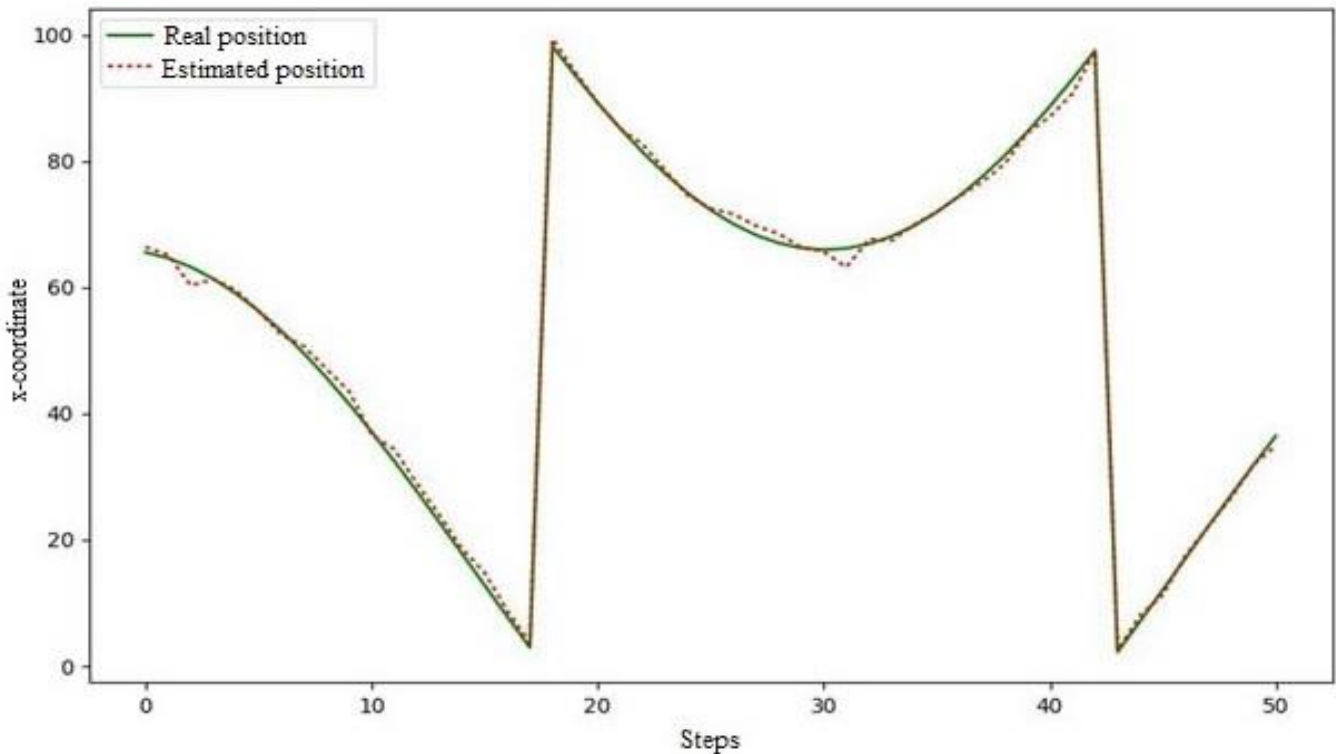


Figure 3. The actual x position of the robot vs. The position of x estimated by filter.

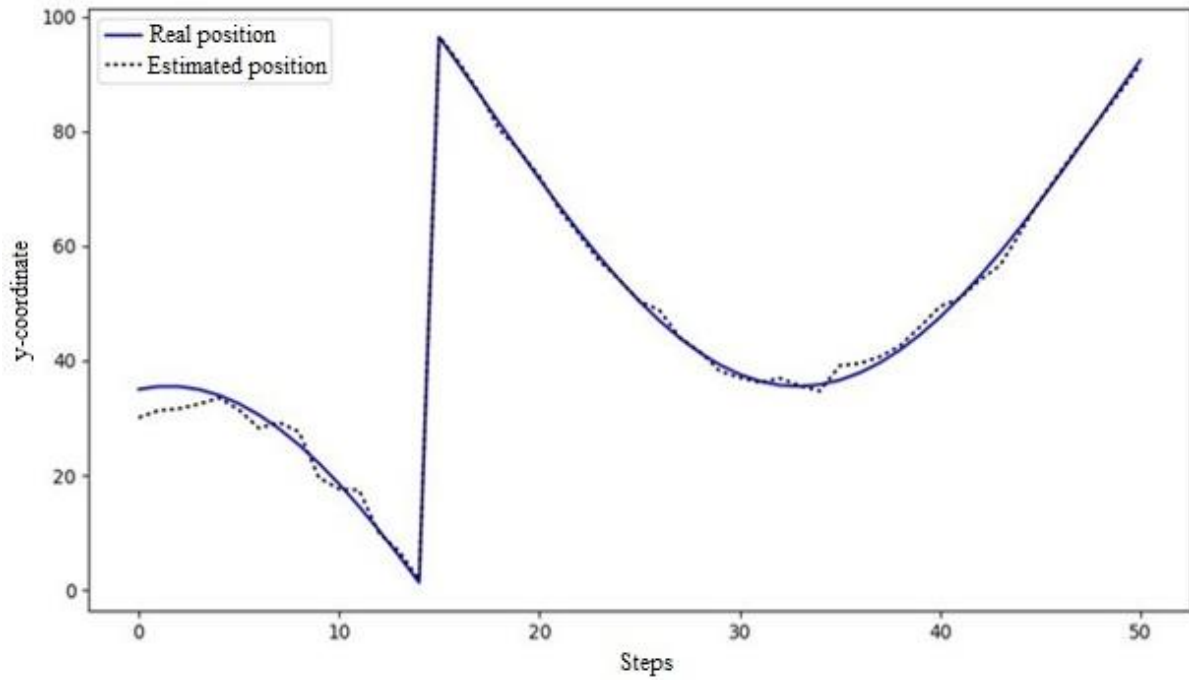


Figure 4. Actual y position of robot vs. The position of y estimated by filter.

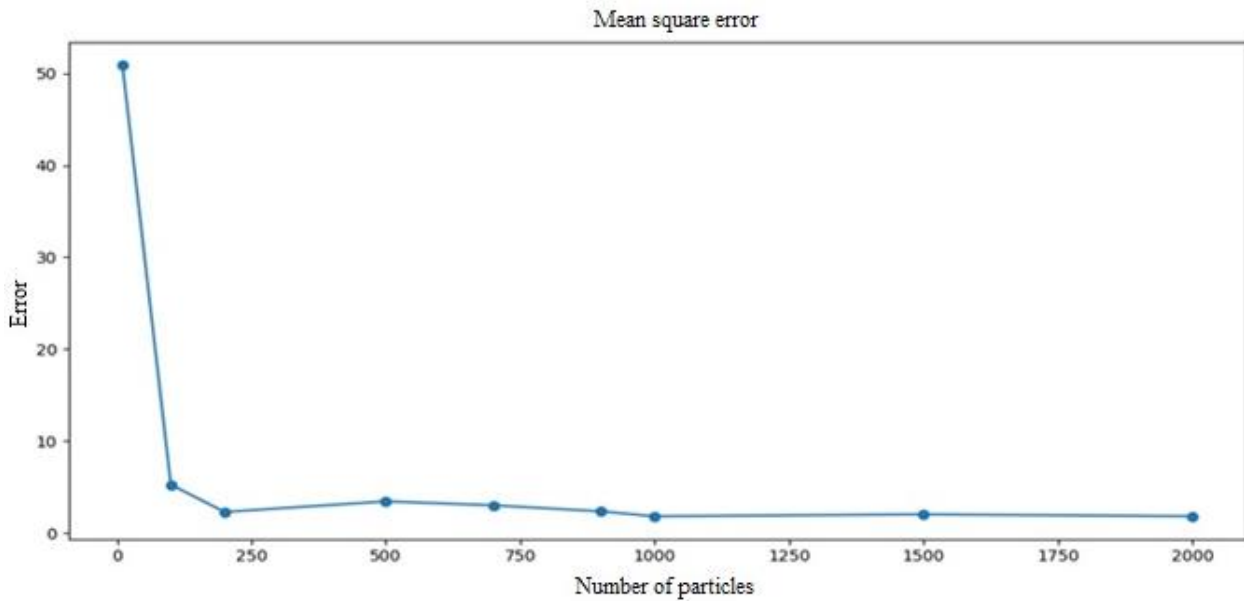


Figure 5. Comparison diagram of particle number error.

Due to the high computational cost of SLAM algorithm, it is necessary to calculate the processing time of Raspberry Pi 3 model B card to determine the time required for the system (Table 1) to estimate the position according to the number of particles set for its best performance.

Figure 6A and 6B show the robot in the test space. The robot perfectly recognizes the terrain by determining its position relative to the landmarks found in the environment. However, as long as the physical conditions of the scanning area remain unchanged, the position of each landmark in the algorithm is set to a constant data.

Table 1. Processing time of Raspberry Pi card

Number of particles	Processing time
10	0.1737992
100	1.2880270
200	4.1797537
500	6.0745110
700	8.4064073
900	11.2124450
1,000	11.5560207
1,500	19.7074775
2,000	28.3417411

Due to the limitation of the response time of the system (see Table 1) and the measurement noise of the sensor determining the position of the robot relative to the

fixed reference system, the robot walking delay is mainly due to the calculation of new particles. Similarly, the results of implementing part filter in a small scanning area of a specific scene are also visualized and compared with the behavior in the simulation process. When the robot

represented by the red circle is in the position corresponding to the landmark, the particle recombination shows a higher density, and the landmark shows a melon colored circle around the robot.

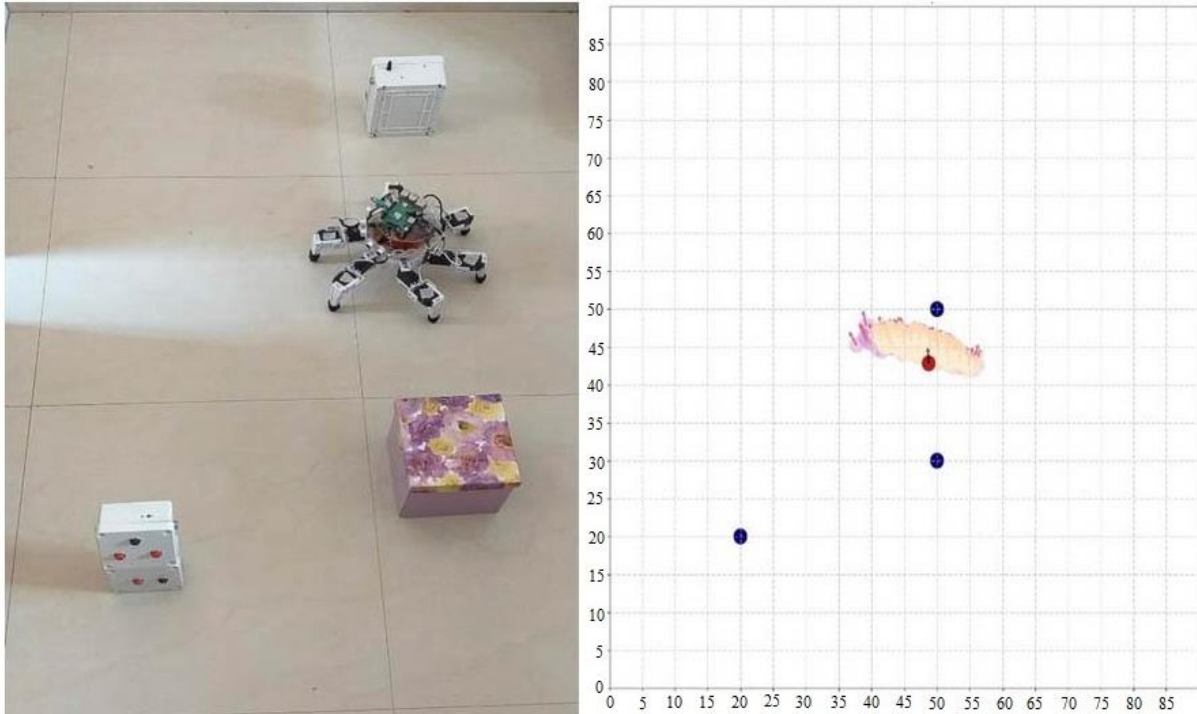


Figure 6A. The hexagonal robot updates its position relative to the landmark.

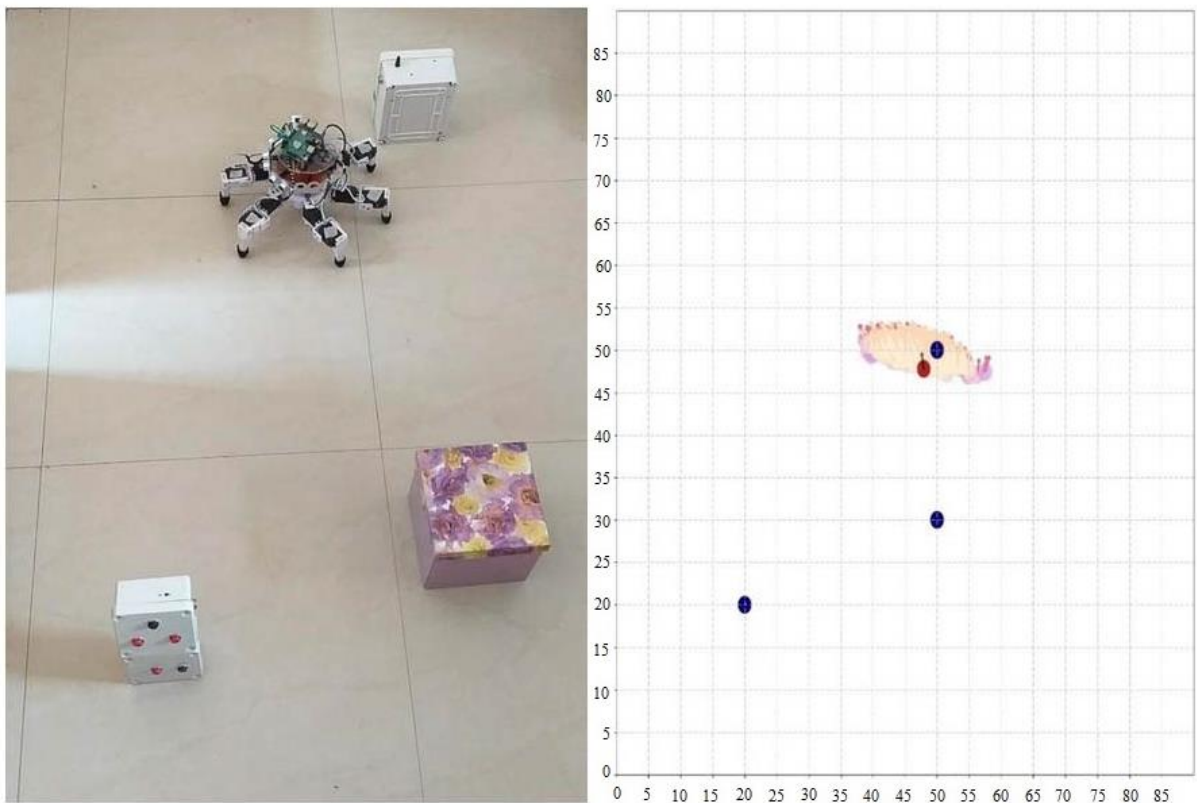


Figure 6B. The position of the hexagonal robot is updated relative to the landmark.

In order to update the position of the robot, the nature of the algorithm must be considered. Two iterations are needed to calculate the future position of the robot. Therefore, there is a certain error range between the actual position and the estimated position. Therefore, the more iterations or steps the robot gives, the greater the possibility of error in its current position.

3.2 Temperature analysis diagram

During the scanning process, through the DHT-11 sensor, the temperature data acquisition system is configured to automatically collect the value according to the input variables (position and temperature) every 35 seconds and save it in the CVS storage file. Through the sampling test using this sensor, it is found that the stabilization and acclimatization time to determine the correct temperature varies between 35 and 40 seconds.

When the robot completes its journey, it establishes a network connection with the computer and draws a temperature diagram according to the obtained data. The temperature value is represented by the color range, which changes according to its intensity, as shown in **Table 2**. These color ranges represent the range of colors defined at each point in the journey, as shown in the Figure. 7 and 8, these data are collected on the sensor 7 cm away from the heat source, and the ambient temperature is about 30 °C. The method used to determine the temperature region is to triangulate the surface and draw a triangle according to the temperature points within the same temperature range according to the color code specified above.

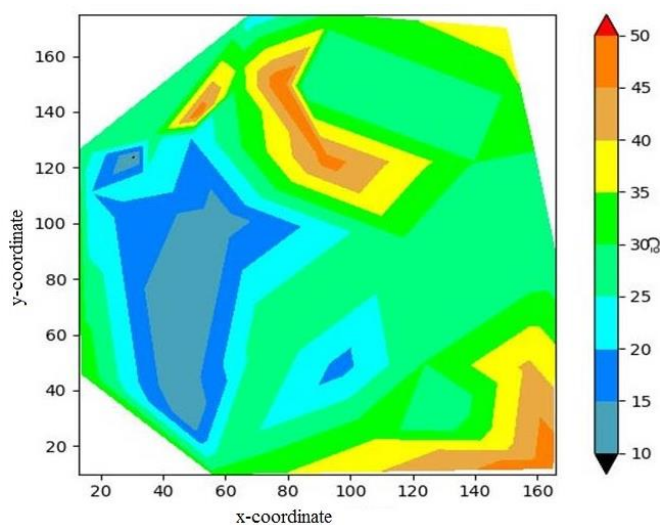


Figure 7. Temperature diagram.

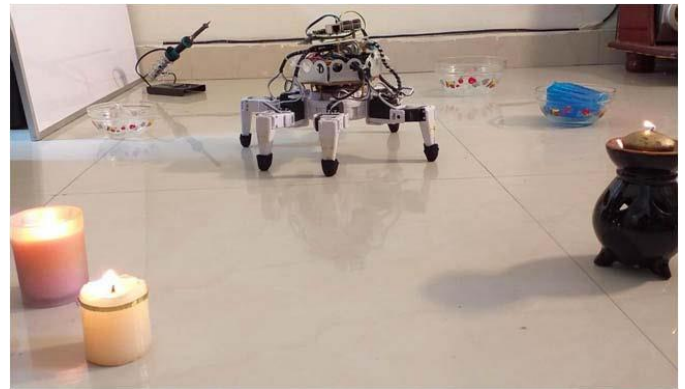


Figure 8. Temperature source in actual environment.

4. Discussion

Localization plays a very important role in the autonomous navigation of mobile robot. Therefore, to analyze several possible methods to solve this problem, it is necessary to study the application of the algorithm in robot system and the accuracy of the results.

Particle filter improves the position estimation of the robot relative to the fixed reference system, but the algorithm does not consider the discovery of objects or obstacles that may not only hinder the robot from moving in the scanning process, but also produce uncertainty for the change of gait direction that may lead to immediate position update. Due to the application of the algorithm in the field of autonomous navigation, the algorithm calculates various positions or steps of the robot moving and updating parts in the environment. In this case, the algorithm becomes a little rigid, and sudden changes are not allowed before all iterative calculations are completed.

Table 2. Color distribution according to temperature level

Colour	Temperature
Black	<10 °C
Celeste	10 °C < T < 15 °C
Blue	15 °C < T < 20 °C
Turquoise	20 °C < T < 25 °C
Emerald green	25 °C < T < 30 °C
Lemon green	30 °C < T < 35 °C
Yellow	35 °C < T < 40 °C
Mustard	40 °C < T < 45 °C
Orange	45 °C < T < 50 °C
Red	>50 °C

Considering this function, particle filter cannot guarantee 100% successful positioning, because the rounding error is the inherent error of external sensors and gait

fault-tolerant estimation of robot motion types. Therefore, in order to reduce the position error level of the robot, a position correction factor calculated from the above fault must be added to make the algorithm an advanced position monitor with continuous updating characteristics.

In order to monitor or sample the temperature, humidity and other climatic characteristics in the exploration area, it must be considered that in order to better study the variables affecting the sampling area, sensors with high average accuracy and data reading stability time less than 10 seconds are required. In this way, we can obtain better temperature RA diagram more accurately. For different applications, this represents a simple and fast method to detect any changes or abnormality in the analysis process. For safety reasons, when the robot detects the strength of the upper mark at a temperature of 55 °C or below 10 °C, it will change the driving direction within 25 cm from the ground mark.

5. Conclusion

The results show that the application of the system has achieved satisfactory results in using particle filter to determine the mapping and positioning (position) in the travel of hexagonal robot, the position estimation error level is less than 6%, and 1000 or more particles are applied to make the robot approach the location calculated by Algorithm.

Although the DHT-11 sensor can correctly collect temperature data and the measurement error is very low, it is inconvenient to obtain the stability time of new measurement and data acquisition failure usually occurs. Therefore, it is recommended to use another sensor to draw a more accurate map and use more sampling variables.

It is found that since there is no problem with signal interference, the wireless communication between robot and computer can be maintained, so the data collected by DHT-11 sensor can be displayed; Raspberry Pi3B card is used to correctly store the collected temperature data, and the temperature value of the whole lathe is well fitted. This determines that using the registry to generate CVS files is a necessary support for sampling without a network connection. The proposal solves the problem of energy saving and allows more scanning time. This network free environment is an ideal solution for applications under extreme conditions, such as lack of access to some areas that need scanning and data sampling, or including

the development of low-cost rescue robot models to help the country in case of natural disasters.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Polash MMH, Tumpa SN, Saumik SS, *et al* (editors). Explorer-0100: An autonomous next generation Mars rover. 2017 20th International conference of computer and information technology (ICIT); 2017 Dec 22–24. IEEE; 2017. p. 1–7.
2. Russell S, Norvig P. Artificial Intelligence-A modern approach. Pearson; 2010.
3. Burgard W, Fox D, Thrun S. Probabilistic robotics. The MIT Press; 2005.
4. Narváez V, Yandún F, Pozo D, *et al*. Design and implementation of simultaneous positioning and mapping system (SLAM) for ROBOTINO® robotic platform, Revista Politécnica 2014; 33(1).
5. Li JL, Bao JH, Yu Y. Graph SLAM for rescue robots. Applied Mechanics and Materials 2013: 134–137.
6. Ezrobot, “Six Hexapod-EZ-robots” [Internet]. Available from: <https://www.ez-robot.com/Shop/AccessoriesDetails.aspx?productNumber=30>.
7. Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I. IEEE Robotics & Automation Magazine 2006; 13(2): 99–110.
8. Upton E, Halfacree G. Raspberry Pi user guide. Wiley; 2014.
9. Syed A, ElMaraghy HA, Chagneux N. Real-time monitoring and diagnosing of robotic assembly with self-organizing neural maps. Proceedings of real time systems symposium; 1992 Dec 2–4. IEEE; 1992. p. 271–274.
10. Aloisio A, Branchini P, Cevenini F (editors). Real-time diagnostic and performance monitoring in a DAQ environment. 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 Jun 14–18. IEEE; 1999. p. 239–242.
11. Bilurbina L, Alter LB, Liesa F. Nonmetallic corrosion resistant materials. Marcombo; 1990.

12. Lemesle R, Petitjean A. Windows powershell? The fundamentals of language. ENI; 2015.