Optimizing Cleaning Paths for Robots in Domestic Settings

Muhammad Azeem Haider
Dhanani School of Science and Engineering
Habib University
Karachi, Pakistan
mh06858@st.habib.edu.pk

Ali Asghar Yousuf
Dhanani School of Science and Engineering
Habib University
Karachi, Pakistan
ay06993@st.habib.edu.pk

***Abstract*—The integration of robots into human lives has significantly reduced manual labor, particularly in menial tasks. Among these, cleaning robots represent a vital category, simplifying routine and labor-intensive chores. Despite being available in the commercial market for over fifteen years, some notable limitations of current cleaning robots are their sub optimal obstacle avoidance capabilities and their inability to achieve maximum coverage of the area they are present in. This paper proposes the development of an advanced cleaning robot designed for domestic environments. The primary focus is to enhance the robot's ability to navigate and avoid obstacles without relying on pre-existing maps of the area. Obstacle Avoidance is carried out using the VFH+ algorithm, and the movement is carried out using random spiral walk which been implemented previously in different research projects to solve the coverage problem.**

***Index Terms*—Path-Planning, Obstacle-Avoidance, Cleaning robots, Random Walk, Coverage Path-Planning**

I. Introduction

Robotics and automation have provided relief in every facet of human life. With over 3.4 million robots active in manufacturing industries and saving lives every day in the healthcare sector, robots are becoming increasingly common and extremely important. Mobile robots, in particular, can go a step further by assisting humans with repetitive, mundane, and dirty tasks. One such task is cleaning floors in domestic settings.

The widespread adoption of cleaning robots in recent years has revolutionized floor cleaning, offering convenience and efficiency. However, robots were not an instant success. The early versions of floor cleaning robots were rudimentary and offered limited functionality, requiring constant human intervention to perform optimally. Even today, a significant limitation of these robots lie in their effectiveness within spaces that are not sparsely populated. Their design seems tailored to minimalist environments, often overlooking the intricate layouts and diverse interior designs of typical homes. In living spaces like bedrooms and living rooms, filled with furniture and decor, these robots frequently find their cleaning paths hindered, necessitating manual intervention.

Despite their promising capabilities, cleaning robots in domestic settings are often described as behaving like "drunk" entities, frequently getting stuck under furniture or ceasing operation upon encountering obstacles. This issue often stems from sensor limitations or algorithmic constraints. Moreover, the complete coverage problem (CCP) - defined as the task of visiting all points in a space while avoiding obstacles and minimizing coverage time - remains a challenge in robotics. These challenges considered, there is still a lot left to be desired when it comes to cleaning robots in domestic settings.

This research project aims to delve into the algorithms governing cleaning and obstacle avoidance in these robots within home environments. The primary objective is to explore ways to enhance the efficiency of their path planning, ensuring optimal cleaning performance while navigating around common household obstacles like furniture. The research will attempt to employ an amalgamated strategy involving coverage path planning and random walk methods while avoiding obstacles, to solve the coverage problem and ensure the robot does not become stuck.

II. Literature Review

Over the years, algorithms and different strategies have been introduced for the successful implementation of floor cleaning robots, while some of these strategies have been successful instantly, other have take a few iterations to be as good as a floor cleaning robot should be.

(Hasan & Reza, 2014) [1] introduces different path planning algorithms for autonomous vacuum cleaning robots. The paper divided the movement of their robot into autonomous and manual mode. Autonomous mode was implemented using four distinct algorithms. The four algorithms were Random Walk, Spiral, 'S' shaped pathway, and Wall follow. The paper concludes that the 'S' shaped algorithm covered the most area in the room in the least amount of time. Similarly, (Ravindran, 2014) [2] employs random walk and standard walk to implement a cleaning robot. Random walk was defined as the robot moving in one of the four directions (left, right, up, and down) and the position of the robot re-calibrated at each time-step, whereas, in standard walk the re-calibration only took place after a certain condition i.e. encountering the boundary of a room. The paper concluded that standard walk gave better results for covering more area in less amount of time when compared to random walk.

Pivoting to different algorithms for floor cleaning than just random walk, many research studies explored the possibilities of getting the A-star algorithm to work for room coverage and path planning. (Suryani, Agustriana, Rakhmatsyah, & Pahlevi, 2023) [3] implemented the A-star algorithm along with the imperfect maze for the shortest path planning of a robot to implement room cleaning. An imperfect maze is defined as a maze that does not always use paths that are unique, thus making the occurrence of a loop highly likely. The paper concluded claiming that the robot could perform floor cleaning and pass through all floor blocks.

(Liu & Zhang, 2022) [4] further explored the improved A-star algorithm and its feasibility in indoor cleaning robots. The paper first introduced the limitations to the original A-star algorithm stating, 'when the A-star algorithm is used for path planning in a complex indoor environment, more search nodes are required, and polyline paths are generated, which reduces the efficiency of path planning.' The research further introduces the global path yaw angle which characterizes the relative relationship between the robot's real-time attitude and the global path. The combination of Dynamic window algorithm (DWA) and global yaw angle improves obstacle avoidance and path planning of the robot.

(Yakoubi & Laskri, 2016) [5] used genetic algorithms to plan the path for their robots. These algorithms were employed to continually evolve and refine the path, ultimately yielding an optimal route for movement. The fusion of genetic algorithms with robotics is a relatively under-explored area in the literature, making this approach to addressing the coverage problem particularly intriguing. The genetic algorithm functioned by segmenting the global path into smaller, more manageable paths.

III. Methodology

The following section talks briefly about the strategies employed to implement the floor cleaning robot. Two main problems were to be tackled for the implementation, the problems being.

- Coverage Path Planning Problem
- Obstacle Avoidance

The floor cleaning robot should effectively avoid obstacles in the room and cover majority of the room at the same time. While the *full coverage path planning* problem has not been solved successfully yet, there are theoretical frameworks and solutions that pose to be good implementations for the coverage problem. The section will first discuss the Coverage Path Planning problems implementation is length to understand why the choices of random walk were taken and how the solution can be a good fit to solve the problem of coverage when it comes to floor cleaning robots in domestic settings.

A. Brownian Motion for Coverage Problem

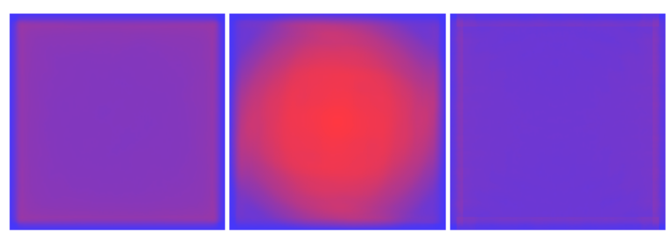
Brownian motion refers to the random movement displayed by small particles that are suspended in fluids. *Brownian motion* is defined as chaos and is one of the simplest models of randomness. The motivation from the random movement of particles in fluids is taken for the movement of robots for the coverage problem or in swarm robotics which is a field in multi-robotics where a large number of robots are connected in a centralized manner. While *Brownian motion* has been used extensively in swarm robotics. Using the theory of Brownian motion, [6] [7] implement swarm robotics and for efficient area coverage by following *Brownian motion* to explore.

Keeping the strategy that researchers have previously employed in the past for swarm robotics, *Brownian motion* and its random movement theory could be an efficient strategy to implement and solve the coverage problem when it comes to movement of floor cleaning robots in domestic settings. The randomness of the robot movement could lead to better results depending on the type of random movement that is being implemented. This is where deciding on what type of random walk for the robot becomes an extremely important discussion.

B. Spiral Random Walk

Spiral random walk or the spiral algorithm combines the use of making spirals and random turns to move the robot [8]. To begin with, the robot checks if there is enough space to start making a spiral. This is decided with the use of on board sensors. If there is enough space for a spiral, then continuous turns with increasing radius from the current position are performed [9].

Traditionally in spiral algorithm, if the robot senses that it is close to an obstacle, it moves away from the obstacle by moving in a straight path away from the obstacle and once the robot senses that the distance from the obstacle is far enough, it initiates the spirals once again. As can be seen from the strategy, that this will not be extremely successful when it comes to extremely cluttered rooms. Discussing spiral walk, the research [9] discussed how spiral walk takes the most updates to achieve the maximum coverage, this meant that the algorithm revisited the same surface the most times, but in context of floor cleaning robots that is not a bad thing since a cleaner surface is desired..



(a) Random algorithm (b) Spiral algorithm (c) Snaking algorithm

Fig. 1. Figures showing the heat maps for the three tested algorithms after 2000 runs in the square room. A strong red colour indicates frequent visits, while a strong blue colour indicates few visits. [9]

Figure 1 & 2 show the extent of revisits in a single area the spiral algorithm takes place. In terms of the heat map, it is seen that the spiral algorithm revisits the same area way more than the random and snaking algorithm. This revisiting is also shown in the result in figure 3 where we can see that while the spiral walk is slower in visiting the area, it ensures that the cleaning is robust by revisiting the same area.

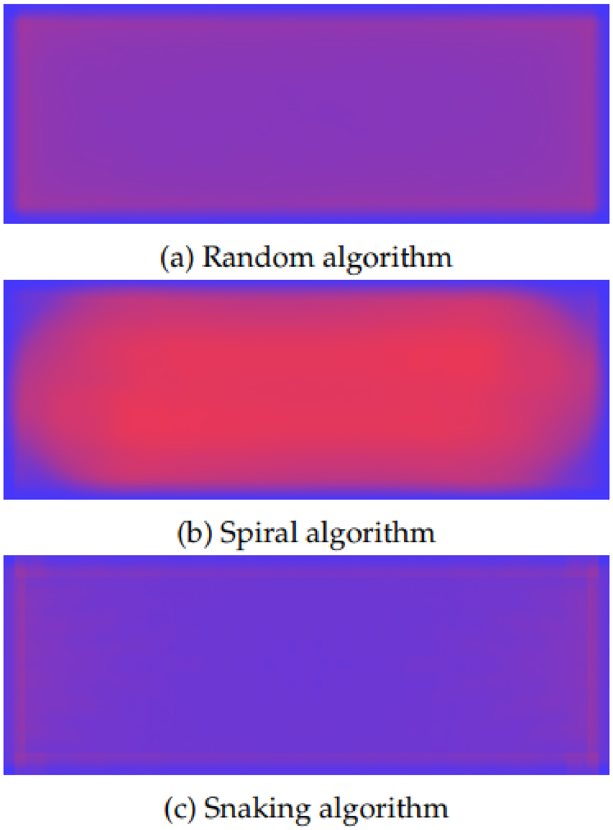


Fig. 2. Figures showing the heat maps for the three tested algorithms after 2000 runs in the elongated room. A red colour indicates frequent visits, while a blue colour indicates few visits. [9]

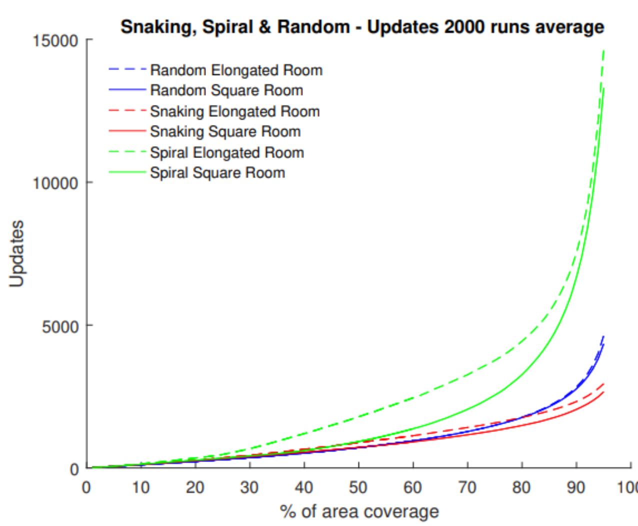


Fig. 3. Graph showing average number of updates required to achieve each percentage of coverage. 2000 runs of the three algorithms in both rooms. [9]

Additionally, [1] discussed the performance of different random walk algorithms as well, and it came to the same conclusion as the [9]. Following is the performance graph of different random walk algorithms on a 12x12 room.

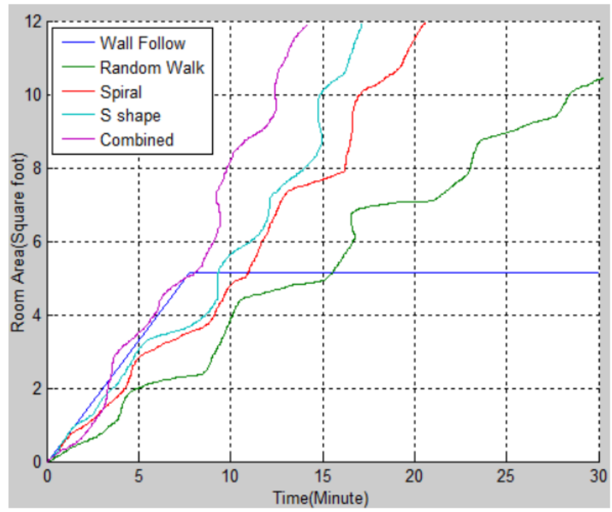


Fig. 4. Performance graph of various algorithms in 12×12 foot room arena. [1]

C. Vector Field Histogram for Obstacle Avoidance

Obstacle avoidance is a key component of any successful mobile robot. Without perfectly implementing obstacle avoidance the robot can be prone to many problems during the simulation such as getting stuck and acting "drunk". As a floor cleaning robot it is essential that the robot does not get stuck by hitting obstacles and that is why robust obstacle avoidance is needed. This is where the implementation of Vector Field Histogram (VFH) is necessary for our mobile robot.

VFH was first introduced in 1991 as a new real-time obstacle avoidance method for mobile robots. The VFH method uses a two-dimensional Cartesian histogram grid as a world model [10]. However, the algorithm implemented for our robot is the VFH+ algorithm for obstacle avoidance. VFH+ algorithms implementation from mathworks was used as reference to implement the obstacle avoidance algorithm. VFH+ works in the following manner. VFH+ is explained as followed in [11].

1) The first step in the VFH+ is to obtain the updated histogram grid and use only the cells in the active window for next computation. Cells on the edge of the active window has to be odd.
2) While the basic VFH algorithm uses one histogram, the VFH+ algorithm creates three histograms. The primay polar hisogram is created using the equation below, where k is defined as the number of sectors, [i,j] are coordinates f the active cell.

$$H_k^p = \sum_{i,j \in C^*} m_{i,j} \cdot h'_{i,j} \tag{1}$$

3) The movement of the robot is affected by every active cell, the movement is calculated using.

$$m_{i,j} = (c_{i,})^2(a - bd_{i,j}) \quad (2)$$

4) The second histogram H^b is called binary polar histogram that is created using two thresholds, T_{high} and T_{low} that determines whether the sector is free or blocked.
5) The third histogram H^m is called masked polar histogram. The masked histogram stops the sectors that cannot be reached without obstacle collision. After the creation of this third histogram we can see the opening and blocked spaces.

VFH+ algorithms works as a pair of eyes for the robot by providing a clear view of reference for openings and blocked spaces in front of the robot. The robustness of the VFH+ algorithm in its implementation of obstacle avoidance makes it a perfect choice for our implementation of a floor cleaning robot in domestic settings.

D. Environment

Matlab was linked to Robot Operating System (ROS) in the Gazebo Office environment. Gazebo Office was chosen because of its simplicity yet still having the necessary obstacles needed to test our robot robustly.

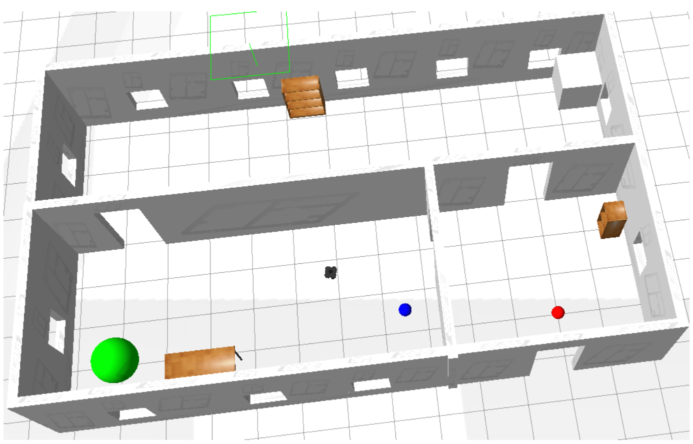


Fig. 5. Gazebo Office Environment

While Gazebo House and Gazebo Warehouse were good options, Gazebo House was presenting the problem of having too cluttered a structure. Whereas, Gazebo warehouse had moving obstacles as humans in reference.

IV. Results and Discussion

The following section will introduce the results of our implementation and explain why it works the way it works. In addition, there are a couple of easy ways that could have been changed or implemented to make the implementation more robust.

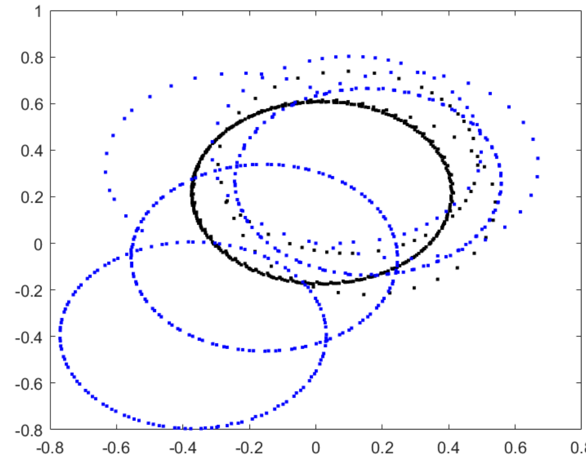


Fig. 6. The black dots are tracing spiral paths for thirty updates (blue dots are odometry)

While there is very little difference between the number of updates in figure 6 & 7, the difference in their plots highlights the randomness of the implementation, proving that if the implementation is ran to number of updates that are high enough, the robot will eventually cover the entire area of the room.

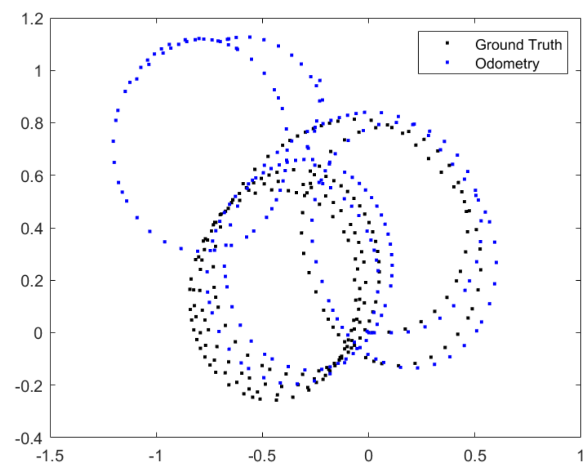


Fig. 7. The black dots are tracing spiral paths for sixty updates (blue dots are odometry)

Figures 6 & 7 are the initial plots that were obtained after the robot was simulated for thirty and sixty updates respectively. The robot had some disturbances due to the odometry running simultaneously but overall the robots performed the spiral walk algorithm to a high degree of accuracy.

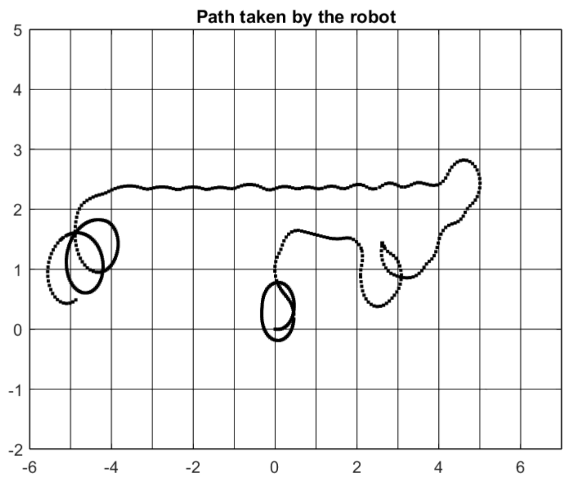


Fig. 8. Path coverage for 600 updates in gazebo office

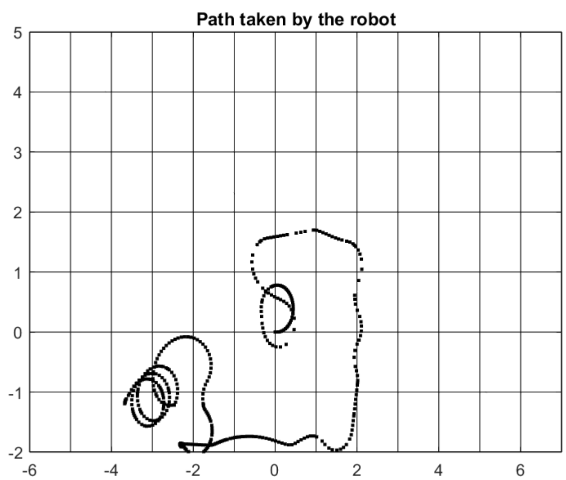


Fig. 9. Path coverage for 600 updates in gazebo office taking a different route

Figures 8 and 9 depict two plots generated from simulating the robot in Gazebo Office within ROS. In Figure 8, the robot initially engages in spiral movements until the VFH+ algorithm detects a wall, triggering obstacle avoidance. Consequently, throughout Figure 8, the robot follows the wall. Toward the end, it resumes spiral movements upon finding sufficient space. Similarly, the movement in Figure 9 begins with spiral motions. However, upon wall detection, the robot navigates along the wall's length before returning to spiral movements in open space.

The obstacle avoidance and spiral algorithm for random walk work exactly how it was intended to work in our implementation. The implementation could have also changed the amount of uncertainty theorem in the robot movement. The implementation could have been ran on different values of uncertainty to find the optimal value, however, changing the uncertainty value without any prior logic led to unsatisfactory results as evident in the plot attached below.

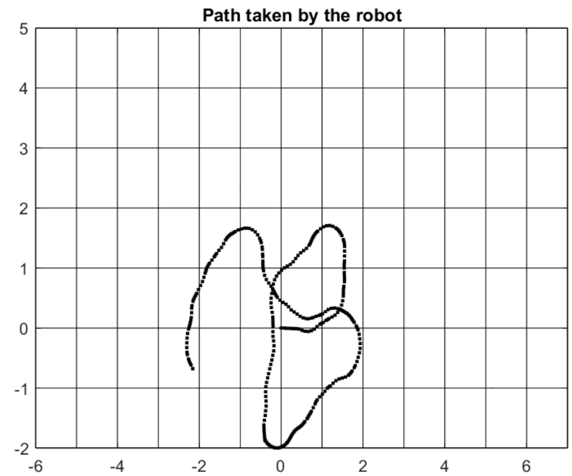


Fig. 10. Path taken by the robot on a different uncertainty value

Different values of uncertainty lead to the robot not working as intended by moving completely randomly rather than moving in a spiral sequence.

V. Conclusion and Future Work

Robotics and automation have been facilitating humanity in creative ways for quite some time. Robots could be programmed to help humans in super complex tasks such as surgeries and operating heavy machinery, but at the same time, the robots could be really helpful in fulfilling dirty and repetitive tasks such as floor cleaning. This research paper introduces an implementation of floor cleaning robots that could lead to the floor cleaning robot covering the entire area of the room, leading to great cleaning results to spiral algorithms repetitiveness of traversing an area, and avoiding obstacles found in its way. Future work for this project can vary in different manners. We can vary the turning radius of our robot to optimize it's performance, similarly, we can run the robot for a higher number of updates such as twenty thousand updates similar to the implementation in the research paper [9] to get more extensive results which can help us tune the parameters accordingly.

References

[1] Hasan, K. M., & Reza, K. J. (2014, May). Path planning algorithm development for autonomous vacuum cleaner robots. In 2014 International Conference on Informatics, Electronics & Vision (ICIEV) (pp. 1-6). IEEE.

[2] Ravindran, R. (2014). Comparison of Simulation Models for a Cleaner Robot. International Journal of Computer Applications, 96(13).

[3] Suryani, V., Agustriana, K., Rakhmatsyah, A., & Pahlevi, R. R. (2023). Room cleaning robot movement using A* algorithm and imperfect maze. Jurnal Infotel, 15(1), 75-81.

[4] Liu, H., & Zhang, Y. (2022). ASL-DWA: An Improved A-Star Algorithm for Indoor Cleaning Robots. IEEE Access, 10, 99498-99515.

[5] Yakoubi, M. A., & Laskri, M. T. (2016). The path planning of cleaner robot for coverage region using genetic algorithms. Journal of innovation in digital ecosystems, 3(1), 37-43.

[6] Martinez, F., Jacinto, E., & Acero, D. (2012, December). Brownian motion as exploration strategy for autonomous swarm robots. In 2012 IEEE international conference on robotics and biomimetics (ROBIO) (pp. 2375-2380). IEEE.

[7] Deshpande, A., Kumar, M., & Ramakrishnan, S. (2017, October). Robot swarm for efficient area coverage inspired by ant foraging: the case of adaptive switching between brownian motion and lévy flight. In Dynamic Systems and Control Conference (Vol. 58288, p. V002T14A009). American Society of Mechanical Engineers.

[8] Skrzypczyk, K., & Pieronczyk, A. (2010). Surface covering algorithms for semiautonomous vacuum cleaner. ACMOS, 10, 294-298.

[9] Sörme, J., & Edwards, T. (2018). A comparison of path planning algorithms for robotic vacuum cleaners.

[10] Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. IEEE transactions on robotics and automation, 7(3), 278-288.

[11] Babinec, A., Vitko, A., Duchoň, F., & Dekan, M. (2013). Navigation of robot using vfh+ algorithm. Journal of Mechanics Engineering and Automation, 3, 303-310.