

Enhancing Performance of Path Planning PRM Algorithm for Automated Boat Using PID Controller

Abdulkader M. Saeed, Khalida S. Rijab

Electrical Engineering Department, University of Technology, Baghdad, Iraq.

ARTICLE INFO

*Received: 21 Sep 2024,
Revised: 3 Oct 2024,
Accepted: 7 Oct 2024,
Online: 5 Nov 2024*

Keywords:
(Path Planning, Boat, Obstacles,
Prm Algorithm, Pid Controller)

ABSTRACT

In recent years, automated boats have gained traction in both civil and military applications, particularly for order delivery, search and rescue, and surveillance. A key challenge in autonomous boat operation is route planning, which involves navigating from a starting point to a destination while avoiding obstacles, especially in urban flood scenarios. Effective route planning requires real-time adaptability to dynamic obstacles, with factors like average route length, power consumption, and trip duration being crucial for optimization. Traditional algorithms like the probabilistic roadmap (PRM) are often used for global search but face challenges in aquatic environments due to disturbances like waves and wind. To address these issues, this research introduces a modification to existing algorithms by integrating a proportional integral derivative (PID) controller. This new system adjusts the boat's path to minimize travel distance and time. Evaluations conducted in various static environments indicate that the PRM algorithm, when combined with the PID controller, significantly outperforms the standard PRM in terms of efficiency, resulting in shorter paths and quicker arrival times at the target destination.

1. Introduction

The use of "Unmanned Surface Vehicles" (USVs) has become an essential feature nowadays, with its various types, shapes, and functions, which includes many civil and military applications, such as their use in surveillance, tracking, object detection, material delivery, marine surveillance, and geography. The survey monitored and hit targets with extreme accuracy due to their small size and low cost [1].

The autonomy of the USV, the required built-in components, guidance, endurance, and

functionality all depend on path planning, making it a crucial task. Extra batteries and power banks cannot be added because USVs can only carry a certain amount of cargo. Thus, path planning becomes the key problem to be addressed in order to resolve a time-limited issue and enable USVs to carry out the necessary activities [2].

The research in USV and its path planning algorithms maintain a focus on the missions of mapping or covering specific areas of interest. The majority of the research focused on the land-based

Corresponding author:

E-mail addresses: abdalkadermsq@gmail.com (Abdulkader), khalida.S.Rijab@uotechnology.edu.iq (Khalida)

doi: [10.5281/jgsr.2024.14039138](https://doi.org/10.5281/jgsr.2024.14039138)

2523-9376/© 2024 Global Scientific Journals - MZM Resources. All rights reserved.



This work is licensed under a Creative Commons Attribution Share Alike 4.0 International License.
<https://creativecommons.org/licenses/by-sa/4.0/legalcode>

vehicles as well as their tips and tricks for completing the given mission. In most of the research, only terrestrial-based vehicles were examined, and their operational strategies were described for mission accomplishment. Afterwards, the target was widened to cover the functioning of the USVs operation as well. However, environment safety rules and road code regulations that are viable for land vehicles usually do not necessarily imply that they are also universal for the rules of water transport regulation. When handling USVs, it is crucial to consider more apparent and substantial supplementary elements. For example, a watercraft is subject to limitations on the amount of cargo it can carry, as well as specific physical attributes and weight specifications. It is also constrained by restrictions on manoeuvrability and various other factors that can hinder its overall performance.

Given its significance, it was imperative to implement substantial and enduring enhancements to enhance its object detection capabilities, path determination, and obstacle avoidance while preserving the path's integrity to the greatest extent possible. Additionally, real-time implementation was pursued to achieve the desired objective [3].

2 Related Work

Now, there is a huge debate about how to go around the blockades of trajectory planning for robots with the range of path planning techniques and inspiration from nature as well. The analysis presents a general survey of the recent studies that have tackled this issue.

The research [4], focuses on using UAV aerial imaging for ground map generation and path planning in flooded urban environments. A* algorithm provides a shorter path but worse computation time compared to GA and PRM algorithms. Experimental results show the effectiveness of the proposed algorithms in finding near-optimal paths for rescue boats in disaster scenarios. Increasing the number of waypoints in GA algorithm leads to larger paths and longer computation times. Overall, the study highlights the importance of efficient path planning

algorithms for enhancing rescue operations in flooded areas.

In [5], the paper focused on the employment of A* and PRM algorithms by mobile robots to define the shortest path and ascertain the overall time that they will take during the execution of the task. While A* is an algorithm that may be widely used and optimized for finding routes, it takes a long time to generate routes on large maps. On the contrary, PRM renders a probabilistic search, which works out a solution in a short time. This feature can be very helpful if there is a large area to be searched. The paper deals with the development of a hybrid algorithm that consists of A* and PRM and enables recognizing a path with a reduced time of processing. Efficiency and time elapsed were two factors held in focus while comparing the hybrid algorithm with the conventional A* algorithm in terms of path length and routes generated. The output revealed that the proposed combination algorithm produced paths in shorter run times compared to the classical A* algorithm without neglecting the path query. Thus, the hybrid algorithm can be the most resourceful and effective technology for robots working in large areas.

In [6], proposed enhanced version of the PRM algorithm for route planning of smart vehicles with mobility personalities in complicated environments. This approach uses a cosine distance metric and a pseudo-random sampling method with the primary spatial axis as the reference axis. It minimizes the generation of sampling points, excludes reduced distances, adopts the distance threshold between the road points, and uses two-way incremental collision detection to maximize road construction efficiency. The most significant waypoints that form part of the plan are specific areas of the given Bezzel curve that become control points, and the routes are developed with vehicles' driving situations in mind. The upgraded version of the PRM was used within MATLAB and ROS as a testing environment to guarantee the expected reliability. The comparison of the PRM method with and without modifications shows a superiority of PRM with modifications in comparison with the basic PRM method from the perspective of both the length of the path and the calculation time.

3. Basic Principles work

In this part, we will described route planning and obstacles avoidance system:

3.1 Path Planning and Obstacles Avoidance

Path planning and obstacle avoidance are critical components of autonomous navigation systems, enabling robots to navigate efficiently and safely in complex environments. Path planning involves computing an optimal path from a starting point to a goal while avoiding obstacles and adhering to constraints. Algorithms such as Dijkstra's, A*, and Rapidly-exploring Random Trees (RRT) are commonly used to generate paths based on environment representations like grids or graphs. Obstacle avoidance complements path planning by dynamically adjusting robot trajectories in real-time to circumvent unforeseen obstacles detected through sensors such as lidar or cameras. Techniques like potential fields and reactive methods are employed to ensure smooth and collision-free navigation.

Integration of path planning and obstacle avoidance forms a continuous loop where global path plans are refined locally based on real-time sensor data, adapting to dynamic changes in the environment. Challenges include navigating in cluttered spaces, managing real-time computation constraints, and handling uncertainties in sensor readings. These concepts find application in autonomous vehicles, drones, and robotic systems deployed in diverse operational scenarios, underscoring their significance in advancing autonomous navigation capabilities.

The boat, which obtains information on obstacles in the region, determines the distance between an obstacle and a boat, and estimates the direction of the moving obstacle [7]. If the obstacle's position is fixed, then it is static. Or it's dynamic. The direction of the barrier is evaluated by an assisting sensor, and the boat chooses the next path based on environmental information. When an impediment approaches near the boat route, the boat will manoeuvre to avoid colliding with it. Furthermore, if the distance between the boat and the barrier increases continually, it indicates that the dynamic moves away from the boat. Otherwise, the obstruction may be shifted toward

the boat. The path planning model for the proposed system is depicted in Fig. 1.

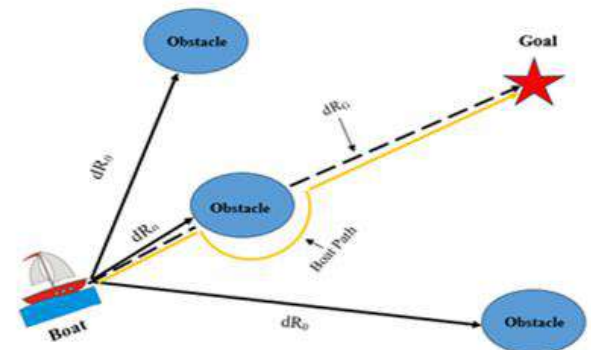


Figure 1. route planning and obstacle avoidance mechanism.

Where dR_0 is the distance between the boat and the obstacle, and dR_g is the distance between the boat and the objective.

Using the distance formula, parameters dR_0 and dR_g are utilized to characterize the distance between a boat and an obstacle, as well as the distance between the boat and the target location, respectively.

$$dR_0 = \sqrt{(X_s - X_{obs})^2 + (Y_s - Y_{obs})^2} \quad (1)$$

$$dR_g = \sqrt{(X_s - X_g)^2 + (Y_s - Y_g)^2} \quad (2)$$

where X_g and Y_g are the environment's goal coordinates, X_s and Y_s represent the current boat coordinates, and X_{obs} and Y_{obs} explain the environment's obstacle coordinates. Robots should be designed and constructed first, followed by a navigation and guidance system tailored to their applications. These robots can perform navigation and guidance in a variety of environments, including at home and in agricultural fields. Simply said, using these robots in any situation requires navigation and supervision. Owing to the significant proportion of robotic researches in the field of navigation and guidance, it can be divided into four main categories [8]:

- Positioning,
- Path planning (Routing),
- Motion planning (Control)
- Mapping

3.2 Probabilistic Roadmap (PRM) Algorithm

It is a commonly-used motion planning technique in robotics and computation graphics to discover possible paths for the automated moving of robots or virtual phenomenon distributed within complex environments. It offers a system for the systematic studying of possible links between various layouts of the confined area [9]. The PRM algorithm operates in two main stages: planning and performance of a roadmap as well as the process of satisfying the query. In this state, the sample paths are randomly generated into the sample space within the work space. The construction is complex because for each design there is a new possibility of the robot or the character. The algorithm then generates a set of possible placements for the samples, checks if they collide with any object in the environment, and prioritizes them based on their feasibility. At the same time, this will trigger the collision detection techniques like the bounding volume hierarchy, or the spatial partition [9]. When the samples are validated, the algorithm substitutes the graph with its roadmap onto a path called roadmap. Here, each sample configuration becomes a node and edges are established between neighbouring nodes. Flee tactic would work for such configurations because nodes are linked without crossing other nodes. The interaction process ideas try different techniques to connect the samples together, for example, using linear interpolation between the scans and local optimization of the alignment [10]. The next step after building up the roadmap for this PRM algorithm is the query resolution segment. This phase is where the problem instances, setting specific start and goal configurations are defined, and the algorithm aims at finding a solution which is a path between the start and the goal configurations. It is able to accomplish this task by generating possible routes from the graph using some graph search algorithms like Dijkstra's algorithm or A* algorithm. The graph search

algorithm focuses on the links and adjacencies of the nodes, and computes the associated cost or distance[10].

The robustness of the PRM algorithm is related to its probabilistic property resulting in a good search capability over a large possible space. Thanks to the ability to generate random samples, it can save from the dimensionality problem that usually affects high-dimensional planning tasks. Along with it, the pattern along the paths allows reuse, which means that the built graph can be used as a query solver multiple time efficiently [5]. PRM has found numerous applications in real-world scenarios, such as robot motion planning in complex environments, path planning for autonomous vehicles, and character animation in video games. Its ability to handle complex and dynamic environments makes it a valuable tool for a wide range of robotics and computer graphics applications. Fig.2, describes the flowchart of the PRM algorithm [5].

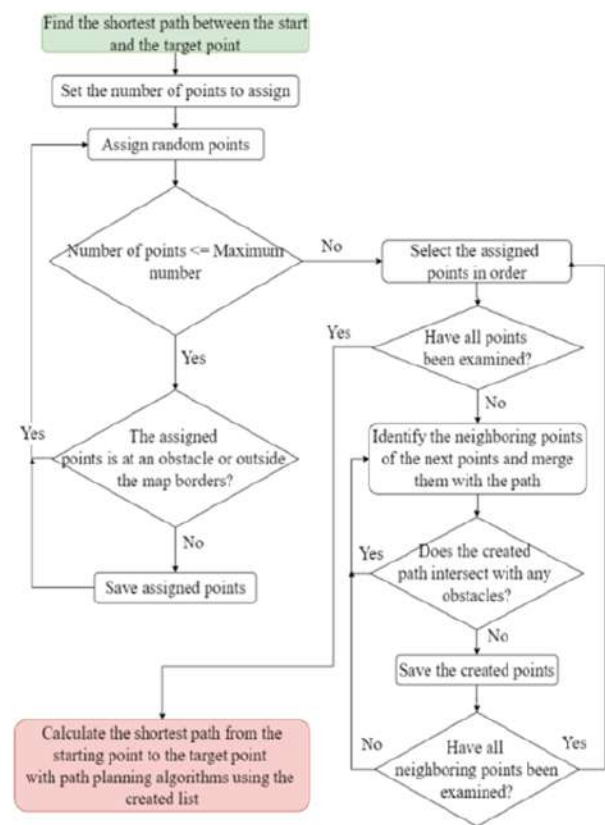


Figure2. Flowchart of the PRM algorithm [5].

4. Hardware and Software Item Required for an Automated Boat

In this section, we will cover in detail the key compounds necessary to create an autonomous boat, such as:

4.1 3D Printer

A 3D printer creates three-dimensional objects by layering materials, typically plastic or resin. It's widely used in manufacturing, prototyping, and education, allowing for rapid design and customization [11]. Fig.3 shows 3d printed to make a boat.

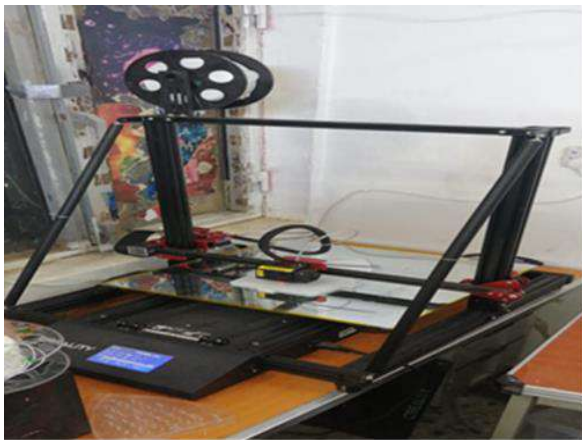


Figure 3. 3D printer make boat.

The boat was printed using pure technical specs and measurements using solidwoks, as illustrated in Fig. 4

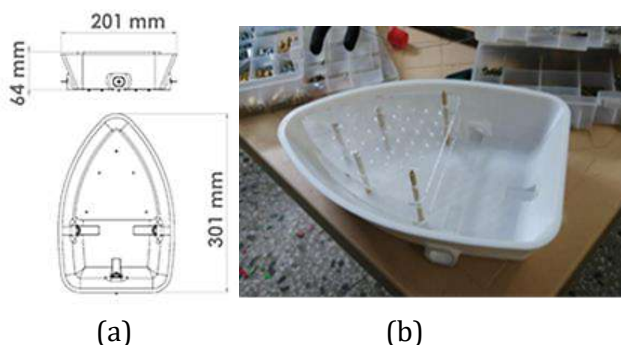


Figure 4. (a) show the measures for a boat and (b) show a boat after it was made.

4.2 Raspberry pi 4

The Raspberry Pi 4 is a compact, powerful single-board computer featuring improved processing speed, multiple USB ports, dual-display support, and enhanced RAM, ideal for projects, learning, and IoT applications, fig.5 show raspberry pi 4 used to control on the boat .



Figure 5. show raspberry pi 4.

4.3 D.C Motors

DC motors are electric motors powered by direct current, converting electrical energy into mechanical energy. They are widely used in robotics, appliances, and automotive applications due to their simplicity and efficiency fig.6 show type of a d.c motor used in boat .



Figure 6. D.C motor used to move a boat.

4.4 Motor Driver ln298

The LN298 is a dual H-bridge motor driver IC, enabling control of DC motors and stepper motors. It allows bidirectional operation and is commonly used in robotics and automation projects fig.7 show type of driver used.



Figure 7. show L298 driver for d.c motor

4.5 Ultrasonic Sensor

An ultrasonic sensor measures distance by emitting sound waves and calculating the time it takes for the echo to return. It's commonly used in robotics, obstacle detection, and level sensing applications[11]. Fig.8 show ultrasonic sensor type used in a boat in order to detect an obstacle .



Figure 8. ultrasonic sensor

4.6 Battery

Power supply it necessary needed in order to operate a system, a lipe batteries are lithium ion was used, this batteries know for high energy density and long cycle life as shown in fig.9.



Figure 9. lipe battery type

5. Software for Applied Algorithm

In this section, the proposed algorithm was used in the proposed environment and the Python language was used to write a program and upload it to Raspberry Pi 4. At first PRM algorithms was used without the PID controller. Finally, the PRM algorithm used with the PID controller.

5.1 Probabilistic Roadmap Algorithm (PRM) without PID Controller

In this part, the steps of the PRM algorithm are explained in details as shown below:

Step 1: Define the Environment:

- Set up the physical environment where the boat will navigate. This includes identifying the boundaries, obstacles (like rocks, other boats, or buoys), and the boat's starting and goal positions.

Step 2: Map Representation:

- Create a map representation of the environment. You can use a grid-based approach or continuous representation, depending on the complexity of the environment.

Step 3: Install Necessary Software on Raspberry Pi 4:

- Set up the Raspberry Pi 4 with an operating system (e.g., Raspbian) and install the required software libraries for robotics and motion planning.

Step 4: Implement Motion Control:

- Set up the necessary motor control and sensors on the boat to allow it to move in the environment. This may involve using motors, servos, or thrusters, as well as sensors like GPS, compass, and obstacle detection sensors.

Step 5: Generate Random Configurations:

- Randomly generate valid configurations (positions and orientations) for the boat in the environment. Ensure that these configurations are collision-free and within the defined boundaries.

Step 6: Build the PRM:

- Create a graph representation where each node represents a valid configuration of the boat, and edges connect nodes that are within a certain distance and do not intersect with obstacles.

Step 7: Connect Nodes:

- The graph node is connected by first checking whether the neighboring nodes have feasible paths. Apply the technology of path-line-of-sight checking or exaggerated detection of collisions to result in the consideration of safe paths.

Step 8: Add Start and Goal Nodes:

- Use the initial boat configuration node to represent state nodes of a graph and connect it to nearby nodes. Repeat this step for the goal configuration.

Step 9: Find the Path:

- Use a path planning algorithm (e.g., Dijkstra's algorithm, and A* search) to find the shortest path between the start and the goal nodes in the PRM graph.

Step 10: Execute the Path:

- Translate the planned path into control commands for the boat's motors or thrusters. Ensure that the boat follows the planned path while considering any real-time obstacles or environmental changes.

Step 11: Implement Feedback Control:

- Incorporate feedback control mechanisms to account for deviations from the planned path due to environmental disturbances or model inaccuracies.

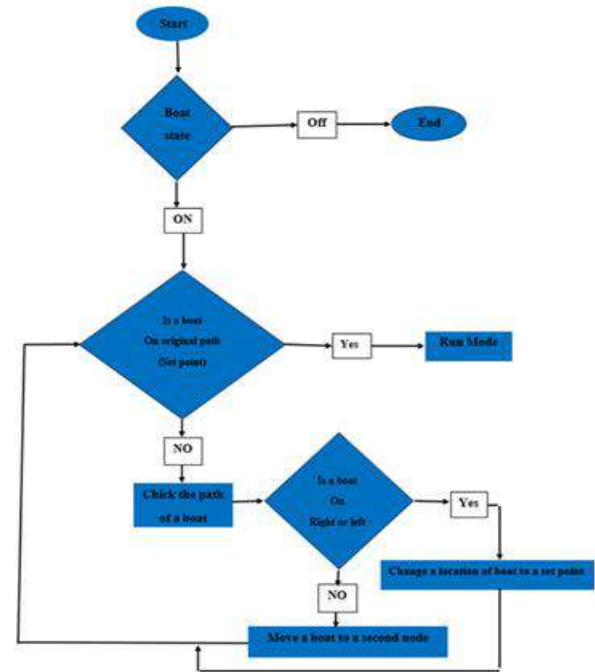


Figure 10. The flowchart of the PRM algorithm.

5.2 PRM Algorithm with PID Control

The Probabilistic Roadmap (PRM) algorithm, combined with a PID controller, provides an efficient and accurate approach to robotic path planning and motion control[12]. PRM constructs a roadmap of possible paths by randomly sampling the configuration space and connecting these samples to create a network of feasible paths. Once a path is planned, the PID (Proportional-Integral-Derivative) controller ensures precise path-following by adjusting the robot's motion based on the error between the desired and actual positions. The PID control law is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{e(t)}{d(t)} \quad (3)$$

where $e(t)$ is the error at time t , K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. This integration of PRM and PID enhances navigation accuracy, stability, and overall system performance [11].

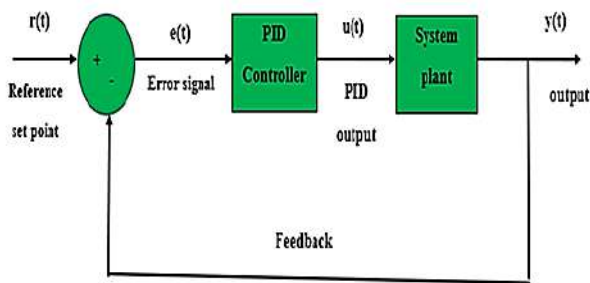


Figure 10. PID Contrller system.

The flowchart of PRM algorithm with PID controller as described in fig.11.

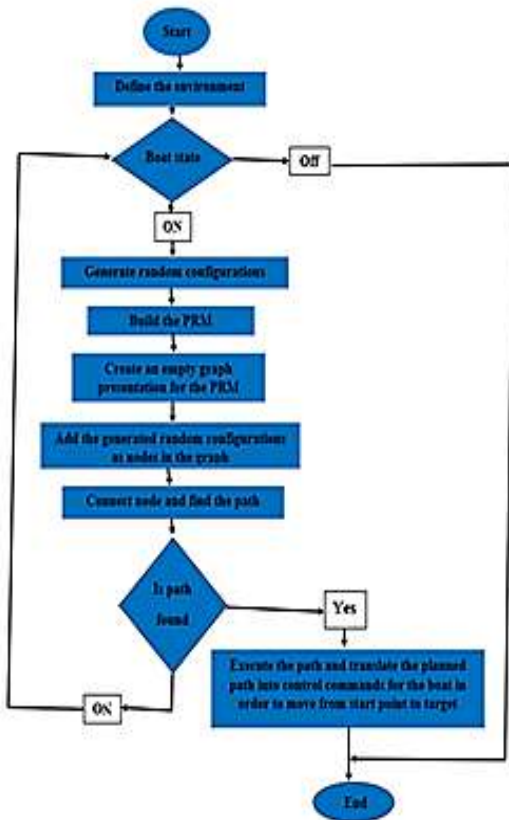


Figure 11. flowchart section for PRM algorithm with PID controller.

6. Performance Metrics

In this study, the proposed method is tested five times. The evaluation is based on the following terms:

- **Path Length:** This parameter describes the route length evaluated over the twelve runs; it is measured by centimeter (cm).

$$path\ planning = n \times t \times C \quad (4)$$

Where:

n : is the number of rotations of the DC motor in one second, which equals 1.5 r/s.

t : is the time taken by boat from the start point to the target (sec).

C : is the circumference of the fan of DC motor = $2 \times \pi \times r$ (4)

r : is the radius of the fan of DC motor, which equals 0.47 cm.

- **Average Path Length (Ava. PL):** This is a measure that calculates the average number of steps taken to get from one node to another in a network. This parameter is the mean distance traveled during the five iterations, calculated as follows:

$$Ava \cdot PL = \frac{\sum_{i=1}^m L_i}{m} \quad (5)$$

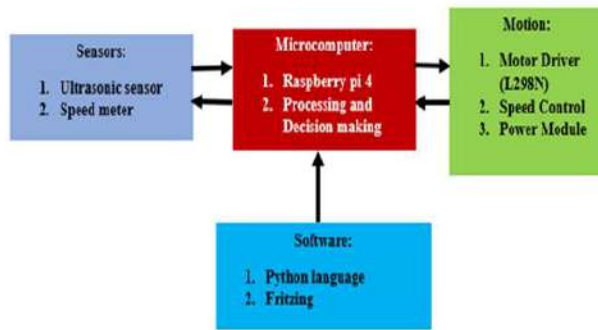
where L_i is the best path length found at the i th run and m is the number of runs; it is measured by centimeters (cm).

- **Standard deviation (SD)** is a statistic that measures the variance between the twelve runs based on their optimum path length in centimeters (cm).

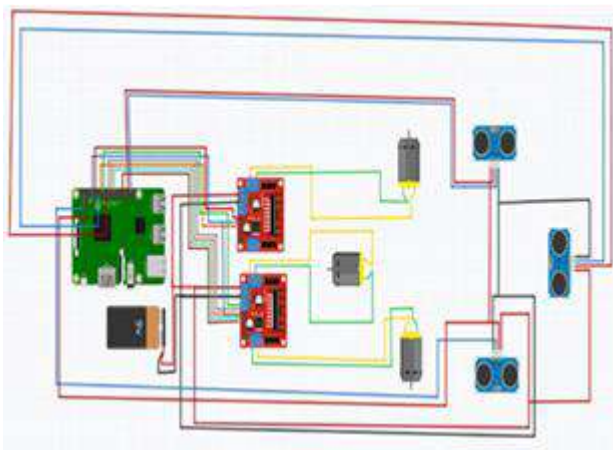
$$SD = |Ava. PL - PL| \quad (6)$$

7. Evolution scenario

The evaluation scenario and analytical experiments of the suggested A PRM algorithm with PID controller are tested utilizing the proposed system, as shown in Figure 12 (a) (b) system interface.



(a)



(b)

Figure 12. show (a) block diagram for proposed system and (b) implementation system by using fritzing.

After that, the application was uploaded to the Raspberry Pi 4. It was written in Python 3.10. The evaluation situation is represented using the following specifications:

- A static environment with two obstacles. Each barrier has a varied size and form.
- There have been five cases implemented. Figure 14 shows that each example represents a beginning point and a target point. Ten runs are conducted for each case.
- The PRM algorithm and a modified PRM method with a PID controller are applied to the suggested environment depicted in Fig.13.



Figure 13. proposed environment.

8. Results

The planning area is (240 cm x 110 cm). The beginning position is $(X1, Y1)$, whereas the goal location is $(X2, Y2)$. Fig.14 depicts the dimensions of the setting used in this work the system was run on five cases, each case having a starting point and a goal point. In the first case (start point (45,5), goal point (95,215)), we can see that the path length using the traditional PRM algorithm is 198.2 cm, while the path length using the improved algorithm PRM with PID controller is 165.9 cm, indicating an improvement in the performance of PRM with PID controller. In contrast, the findings reveal that for the identical example 1, the average path length using the PRM algorithm is 16.516 cm, but the average path length using the PRM with PID controller is 13.825 cm, indicating a considerable difference. Improvement. PRM with a PID controller is quite near to the shortest path, as opposed to the standard PRM method, which was far from it. Table 1 and 2 show the comparison PRM algorithm without PID controller and PRM algorithm with PID controller, and Fig.15 demonstrates a smaller standard deviation (SD) in the improved algorithm PRM with PID controller. Figure 18 also compares the path length and duration for PRM without PID controller with

PRM algorithm with PID controller in the suggested setting.

Table 1. PRM algorithm without PID controller for proposed environment.

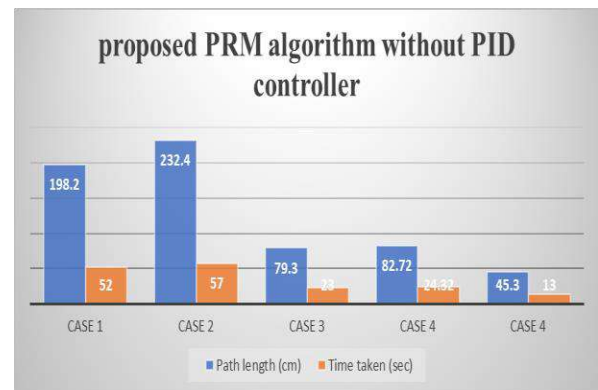
Cases	Start point	Goal point	Path length (cm)	Time taken (sec)	Average Length (cm)	Standard Deviation (S.D)(cm)
1	(45,5)	(95,215)	198.2	52	16.516	181.684
2	(95,15)	(7,180)	232.4	57	19.367	213.033
3	(40,100)	(85,10)	79.3	23	6.6	72.7
4	(32.5,22)	(81,119)	82.72	24.32	6.89	75.83
5	(60,205)	(52,95)	45.3	13	3.75	41.55

Table 2. PRM algorithm with PID controller for proposed environment.

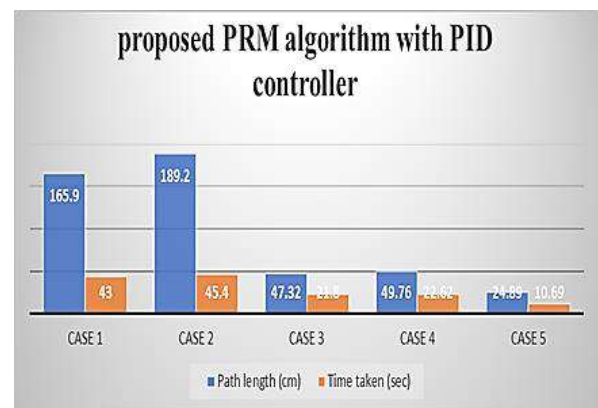
Cases	Start point	Goal point	Path length (cm)	Time taken (sec)	Average Length (cm)	Standard Deviation (S.D)(cm)
1	(45,5)	(95,215)	165.9	43	13.825	152.075
2	(95,15)	(7,180)	189.2	45.4	15.76	173.44
3	(40,100)	(85,10)	47.32	21.8	3.94	43.38
4	(32.5,22)	(81,119)	49.76	22.62	4.146	45.614
5	(60,205)	(52,95)	24.89	10.69	2.07	22.82



Figure 14. environment dimensions.



(a)



(b)

Figure 15. A comparison between path length and time for (a) PRM algorithm without PID controller and (b) PRM algorithm with PID controller.

The results demonstrate the superiority of the new algorithm of PRM with the PID controller

compared to the standard PRM algorithm without the PID controller. The path length was reduced by 23% and the average path length was reduced by 19%, resulting in decreased path and arrival time. Fig.16 shows the standard deviation (SD) of the two suggested techniques.

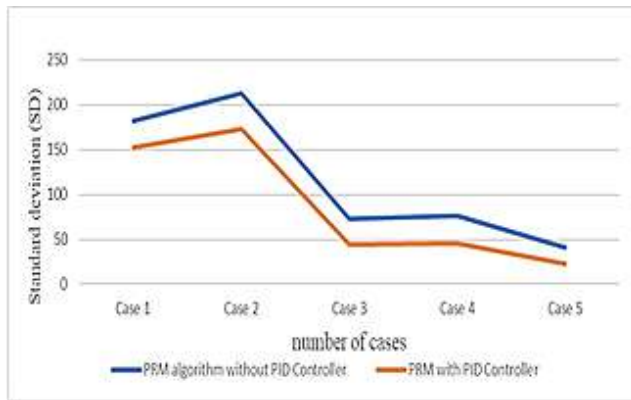


Figure. 16 Standard deviation (SD) between the PRM algorithm without the PID controller and the PRM with the PID controller.

9. Conclusion

This paper present in details, improving a system for path planning algorithms through the use a hybrid system with the PRM algorithm with the PID controller. The new proposed system applied for autonomous boats, and from the collected results from the proposed environments achieved were verified. The conclusions from this work can be summarized by the following points:

1. The merging of the PRM algorithms with the PID controller help to increase the effectiveness and capability of path planning and obstacle avoidance that are used in independent systems. PRM is good at managing dynamic situations and uncertainties. The prediction will be based on a scenario that has the possibility of change; thus, a plan will be considered that accommodates changing conditions.
2. The PID controller comes into the picture and acts as a catalyst that gives the final input to maximize the performance of the system. A PID controller is a tool that is able to take charge of changes in the path and obstacle avoidance utilizing a unique method.
3. The new hybrid system renders the system more reliable in practical solutions. For instance, a

robotic system, like a vehicle or a drone, handles dynamic environments with items or moving obstacles. The combined global planning ability of the local adaptability of PRM as well as the PID controller serve to make the control system more productive since the ultimate goal is reaching its destination, avoiding any hindrances it might experience along the way.

And last it can say, the method that combines the PRM algorithm with the PID controller makes a system stronger and more stable in different environments.

And last it can say, the method that combines the PRM algorithm with the PID controller makes a system stronger and more stable in different environments.

10. References

- [1]. M. Shu et al., "Application of UAV Multisensor Data and Ensemble Approach for High-Throughput Estimation of Maize Phenotyping Traits," *Plant Phenomics*, vol. 2022, 2022, doi: 10.34133/2022/9802585.
- [2]. A. Montazeri, A. Can, and I. H. Imran, "Unmanned aerial systems: Autonomy, cognition, and control," in *Unmanned Aerial Systems: Theoretical Foundation and Applications: A Volume in Advances in Nonlinear Dynamics and Chaos (ANDC)*, Elsevier, 2021, pp. 47–80. doi: 10.1016/B978-0-12-820276-0.00010-8.
- [3]. J. Kim, S. Kim, C. Ju, and H. Il Son, "Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 105100–105115, 2019. doi: 10.1109/ACCESS.2019.2932119.
- [4]. "T. L. Harman et al., 2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR): the University of Houston-Clear Lake, 2700 Bay Area Blvd, Houston, Texas, 77058, 19-21 September 2019. ".
- [5]. B. Uzun, H. Gozde, and M. C. Taplamacioglu, "IJTPE Journal HYBRID OPERATION OF A * AND PRM ALGORITHMS FOR EFFICIENT PATH PLANNING IN AUTONOMOUS MOBILE ROBOTS," *International Journal on "Technical and Physical Problems of Engineering" (IJTPE) Issue*, vol. 50, pp. 183–189, 2022, [Online]. Available: www.ijtpe.com
- [6]. Q. Li, Y. Xu, S. Bu, and J. Yang, "Smart Vehicle Path Planning Based on Modified PRM Algorithm,"

- Sensors, vol. 22, no. 17, Sep. 2022, doi: 10.3390/s22176581.
- [7]. M. L. Cummings, J. J. Marquez, and N. Roy, "Human-Automated Path Planning Optimization and Decision Support."
- [8]. Sotnezov, R. M. (2009). Genetic algorithms for problems of logical data analysis in discrete optimization and image recognition. Pattern Recognition and Image Analysis, 19, 469-477
- [9]. <https://ln.run/uRmJo>
- [10]. Pettersson, P. O., & Doherty, P. (2006). Probabilistic roadmap based path planning for an autonomous unmanned helicopter. Journal of Intelligent & Fuzzy Systems, 17(4), 395-405.
- [11]. A. M. Saeed and K. S. Rijab, "PID Controller Enhanced A* Algorithm for Efficient Water Boat," Journal Europeen des Systemes Automatises, vol. 56, no. 6, pp. 1083-1093, Dec. 2023, doi: 10.18280/jesa.560618.
- [12]. Fülöp, J. (2005, November). Introduction to decision making methods. In *BDEI-3 workshop, Washington* (pp. 1-15).
- [13]. K. M. Passino and Stephen. Yurkovich, Fuzzy control. Addison-Wesley, 1998.
- [14]. Abdulridah, M. N., Abdulridah, H. N., Mohammed, A. S. (2022). Voice Controlled System by Using Bluetooth and Wi-Fi Technique. *Journal of Global Scientific Research in Electrical and Electronic Engineering*. 7(9): 2659-2667.