Adaptive path-planning for AUVs in dynamic underwater environments using sonar data.

B, B., HASAN, M.J., KANNAN, S. and PRABHU, R.

2024

Adaptive Path-Planning for AUVs in Dynamic Underwater Environments Using Sonar Data

Bryan B.[a,b], Md Junayed Hasan[b, c], Somasundar Kannan[b], and Radhakrishna Prabhu[b, *]

[a]ENSEIRB-MATMECA, Bordeaux Institute of Technology, Talence 33400, France
[b]School of Computing, Engineering, and Technology, Robert Gordon University, Aberdeen AB10 7AQ, Scotland, UK
[c]National Subsea Centre, Aberdeen AB21 0BH, Scotland, UK
[*]Corresponding Author

ABSTRACT

This paper presents an innovative approach to path-planning for Autonomous Underwater Vehicles (AUVs) in complex underwater environments, leveraging single-beam sonar data. Recognizing the limitations of traditional sonar systems in providing detailed environmental data, we introduce a method to effectively utilize Ping360 sonar scans for obstacle detection and avoidance. Our research addresses the challenges posed by dynamic underwater currents and obstacle unpredictability, incorporating environmental factors such as water temperature, depth, and salinity to adapt the sonar's range detection capabilities. We propose a novel algorithm that extends beyond the capabilities of the A* algorithm, considering the underwater currents' impact on AUV navigation. Our method demonstrates significant improvements in navigational efficiency and safety, offering a robust solution for AUVs operating in uncertain and changing underwater conditions. The paper outlines our experimental setup, algorithmic innovations, and the results of comprehensive simulations conducted in a controlled tank environment, showcasing the potential of our approach in enhancing AUV operational capabilities for defense and security applications.

Keywords: Autonomous Underwater Vehicles (AUV), Sonar Data Processing, Path-Planning Algorithms, Underwater Obstacle Avoidance

1. INTRODUCTION

Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) are integral to underwater exploration and operations, where they frequently rely on sonar technology to detect objects of interest or to avoid collisions in environments where visibility is severely compromised. A significant challenge in this domain is the high cost associated with multibeam sonar systems, which are capable of producing high-resolution 3D images of the underwater environment but are often beyond the financial reach of many projects.

The primary aim of this paper is to explore how a more cost-effective single-beam sonar can be utilized to achieve effective obstacle avoidance during underwater navigation. We focus on the Ping360 sonar, a product from Blue Robotics, which offers a distinctive capability to conduct 360° scans despite its lower cost. This characteristic makes it a promising tool for underwater navigation, particularly in scenarios where budget constraints preclude the use of more expensive multibeam systems.

In this study, we delve into the application of the Ping360 sonar for detecting obstacles in the surrounding environment, which is critical for the implementation of path-planning algorithms. Initially, we will discuss the challenges inherent in underwater navigation and review existing solutions to these challenges. Subsequently,

Further author information: (Send correspondence to R.P.)
B.B.: Email: bryanb3018.bb@gmail.com,
M.J.H.: Email: j.hasan@rgu.ac.uk,
S.K.: E-mail: s.kannan1@rgu.ac.uk,
R.P.: E-mail: r.prabhu@rgu.ac.uk

we will apply these solutions to our specific context, demonstrating the practical use of the Ping360 sonar in real-world scenarios.

Path-planning algorithms are designed to find the optimal path from a starting point to a goal while avoiding obstacles and minimizing associated costs. The concept of "cost" in this context can refer to various factors such as time, distance, or energy consumption, depending on the specific environment or mission requirements. These algorithms are crucial not only for robotic navigation but also for broader applications such as autonomous driving systems,[1] where efficiency and safety are paramount.

The A* algorithm is one of the most widely recognized and utilized path-planning algorithms, renowned for its efficiency. This efficiency is largely due to the algorithm's use of heuristics, which allows it to prioritize paths that are likely to be close to the optimal solution—often conceptualized as the "Bird's Eye View" path—enabling quicker decision-making.

In addition to A*, other algorithms, such as Rapidly Exploring Random Tree (RRT), are designed to find feasible paths that, while not necessarily optimal, are computationally less expensive to calculate. RRT, for instance, is particularly effective in scenarios involving high-resolution maps, though it typically produces less optimal results compared to A*.[2]

The rise of Artificial Intelligence (AI) has led to the development of more sophisticated path-planning algorithms, particularly those utilizing supervised Deep Learning[3] or Reinforcement Learning.[4] In the former, models are trained using large datasets to predict optimal paths, while in the latter, agents learn through a reward system, improving their navigation strategies over time. These AI-driven approaches have found applications in diverse fields, including space navigation[5] and the movement control of industrial robots.[6]

Sonar remains one of the most critical tools for underwater exploration and operation, particularly for tasks such as obstacle detection and target identification. High-end sonar systems not only detect obstacles but can also be used for object recognition, providing detailed images that can be analyzed either by human operators or by automated systems using Deep Learning models for classification[7] or detection tasks.[8] However, one of the major challenges in this field is the creation and availability of high-quality datasets, which are often expensive to produce or are kept confidential due to the sensitive nature of the data.

To utilize sonar effectively, a thorough understanding of the sonar equation is essential. This equation provides a mathematical model for understanding how the power of the received signal is influenced by factors such as the source level, target strength, and environmental conditions. For those seeking a deeper understanding of these principles, "Principles of Underwater Sound" by R.J. Urick[9] remains a seminal reference in the field.

The structure of this paper is as follows: Section 2 provides a detailed discussion of the relevant technical challenges and outlines the mitigation strategies. Section 3 presents the experimental results obtained through our continuous research, development and practical experiments, and Section 4 offers concluding remarks and discusses potential future work.

2. TECHNICAL CHALLENGES AND SOLUTION STRATEGIES

2.1 Sound Propagation Underwater

Underwater, the propagation of sound differs significantly from that in the air. A key distinction is the variability in the speed of sound c (measured in m/s), which cannot be considered constant underwater. The speed of sound varies based on factors such as water temperature T (in °C), depth D (in meters), and salinity S (in practical salinity units, psu). The relationship governing the speed of sound underwater is expressed by equation (1).

$$c = 1410 + 4.21 \times T - 0.037 \times T^2 + 1.1 \times S + 0.018 \times D \tag{1}$$

In our study, since we are working in freshwater environments, the salinity S will consistently be 0 psu. This equation highlights the necessity for the AUV to accurately measure the surrounding temperature and its depth position to determine the sound speed precisely.

Understanding the speed of sound is crucial because it directly impacts the range of our sonar system. For instance, the Ping360 sonar in Python does not allow the user to directly set the maximum scan range. Instead,

this range is indirectly determined by setting the speed of sound, the sample period S_p (in seconds), the sample distance S_d (in meters), and the sample number S_n. The relevant relationships are given by equations (2) and (3).

$$S_d = \frac{c \times S_p}{2} \tag{2}$$

$$\text{Max Range} = S_d \times S_n \tag{3}$$

These equations enable us to configure the sonar for optimal performance by adjusting the sound speed based on environmental conditions, ensuring accurate and reliable obstacle detection during AUV operations.

2.2 Path-Planning Algorithm

Path-planning is a critical component of AUV navigation, particularly in complex and dynamic underwater environments. The most commonly used algorithm for this purpose is the A* algorithm, known for its efficiency in finding the optimal path by using heuristics to prioritize paths close to the most direct route. However, the standard A* algorithm, as implemented in MATLAB's `plannerAStarGrid` function,[10] does not account for the influence of underwater currents, a significant limitation in real-world applications.

To address this, we developed an alternative algorithm, often referred to as the grid algorithm in robotics. This method begins by assigning a high value to the goal position's pixel on the grid map. From this pixel, values are propagated to adjacent pixels (right, left, top, and bottom), decrementing the value by one at each step, provided the adjacent pixel is not an obstacle. This process continues until the entire map is evaluated. To find the optimal path, the algorithm then traces back from the start position, moving to the adjacent pixel with the highest value, continuing this process until the goal is reached.

Figure 1 illustrates the working principle of this algorithm.

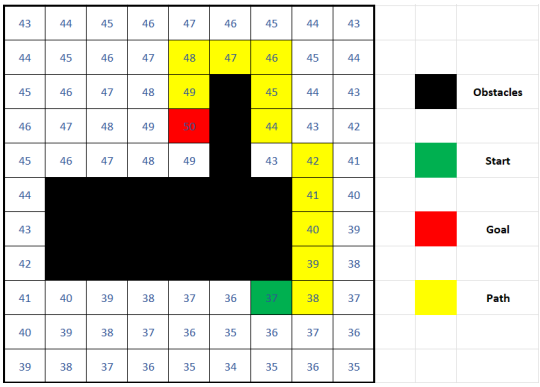


Figure 1: Illustration of the grid algorithm for path planning

By calculating the cost in terms of time required for each movement between two cells, we can determine the estimated time remaining to reach the goal, allowing for more informed navigation decisions.

2.3 Influence of Water Currents

In underwater environments, non-uniform currents present additional challenges for AUV navigation. To ensure the selection of an optimal path, it is crucial to account for the impact of these currents on the AUV's movement.

We begin by establishing a key assumption: if a region is subject to a current with an intensity equal to or greater than the AUV's maximum speed, that region becomes inaccessible. In such areas, it would be impossible for the AUV to maintain its position, making stabilization unfeasible.

Next, for each possible direction the AUV can take, we must calculate the influence of the current on the AUV's speed. This involves a two-step process:

1. Compensation for Current Influence: For each direction (x and y coordinates), the AUV's speed is adjusted to counteract the current's effect. For example, if the AUV needs to move eastward, it may have to angle slightly northward or southward to compensate for a current that would otherwise push it off course, resulting in a reduction of speed in the intended direction.

2. Calculation of the Combined Speed: After adjusting for the current, the current's speed components (V_{c_x} and V_{c_y}) are added to the AUV's adjusted speed components to determine the resulting speed vector. In the case of diagonal movement, the AUV's trajectory is calculated to ensure consistent speed in both the x and y directions.

For example, consider the initial speed vector of the AUV during a diagonal movement, $\overrightarrow{V_{AUV}} = [V_{init};\ V_{init}]$, and the speed vector of the current, $\overrightarrow{V_c} = [V_{c_x};\ V_{c_y}]$. The resulting speed vector of the AUV, considering the current's influence, is $\overrightarrow{V_{Tot}} = [V_{c_x} + V_{init};\ V_{c_y} + V_{init}]$.

To maintain the desired trajectory along the diagonal, the AUV's speed must be distributed appropriately across both axes, while keeping the initial speed magnitude constant. This requires calculating a new speed vector, $\overrightarrow{V_{AUV_{NEW}}} = [V_X - V_{c_x};\ V_Y - V_{c_y}]$, where the magnitude $\|\overrightarrow{V_{AUV_{NEW}}}\|$ matches the original speed $\|\overrightarrow{V_{AUV}}\|$.

The speed calculation is given by equation (4).

$$\|\overrightarrow{V_{AUV}}\| = \sqrt{(V_X - V_{c_x})^2 + (V_Y - V_{c_y})^2} \tag{4}$$

Using this approach, we can determine the new speed components V_X and V_Y, and subsequently calculate the combined speed intensity of the AUV and the current.

The AUV can ascertain the current intensities around it by referencing its position and consulting a pre-mapped database of water streams, enabling it to make informed decisions about its path.

2.4 Problem Specifications

While it is increasingly feasible to instruct a robot to navigate from a starting point to a goal, selecting an optimal path becomes complex in environments where the map is incomplete or obstacles are unknown. This is particularly challenging underwater, where only about 5% of the seabed has been mapped.[11] This lack of comprehensive mapping means that not all potential obstacles are accounted for, necessitating constant vigilance and adaptability during AUV operations.

It is impractical to map the entire underwater environment in a single scan to find the optimal path due to two primary reasons: the limited range of sonar scans and the possibility of undetected obstacles behind the detected ones.

To address this, our approach involves scanning the area directly in front of the sonar when the AUV enters an unmapped region, setting a series of short-term goals (subgoals) as it progresses. For each scan, a path-planning algorithm identifies a subgoal within the maximum range and within a specified angular tolerance. If no valid subgoal is found due to obstacles, the angular tolerance is adjusted, and the search is repeated.

Once a subgoal is identified, the AUV moves toward it, performs another scan, and repeats the process. The initial path can be informed by a satellite map, providing a preliminary route. For example, in Fig. 2, the red path represents the AUV's potential route toward the final goal (red star), while the AUV scans the yellow area (white star) within a 20° angular tolerance (between the blue lines).

If obstacles prevent finding a subgoal, the angular tolerance increases until a viable path is identified. The process ensures that the AUV continues to progress toward its final destination while adapting to previously unknown obstacles.

Given the Ping360 sonar's maximum range of 50 meters,[12] each scan covers a semi-circular area with a radius of 50 meters. The process is depicted in the flow diagram shown in Fig. 3.

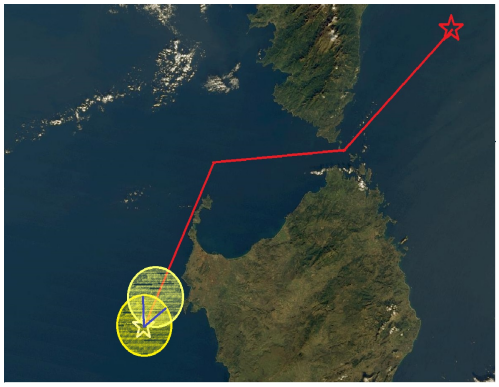


Figure 2: Illustration of AUV navigation and obstacle scanning

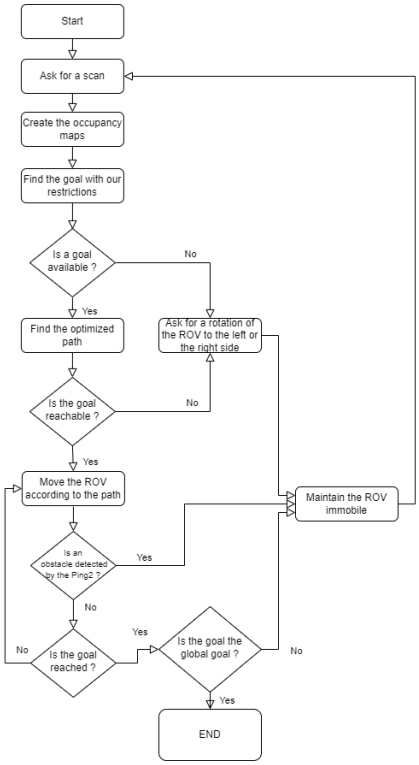


Figure 3: Flow diagram of the proposed path-planning system

2.5 Experiment Conditions

The following conditions apply to our experimental setup:

1. The Ping360 sonar has a minimum detection range of 0.5 meters due to its own sound pulse, a limitation that may vary with different sonar models.
2. Given our test environment—a water tank measuring 96 x 115 x 69 cm—we set a maximum sonar range of 2 meters to avoid reflections from the tank walls.
3. The AUV's movement is restricted to eight possible directions (similar to a king's movement in chess): vertical, horizontal, and diagonal.

4. The AUV's depth is assumed to remain constant between the start and goal positions, simplifying the calculations by focusing on a 2D plane.

5. In our experiments, since a real AUV is not used, the subgoal is defined as the farthest reachable point within the scan range.

6. Data acquisition and storage are handled in Python, utilizing the Ping360 object and appropriate libraries for CSV file storage—a format that is easily parsed for further analysis.

7. Data processing and algorithm implementation are carried out in the MATLAB environment.

8. The final implementation is intended to run on a Raspberry Pi 4.

The desired system configuration is depicted schematically in Fig. 4.

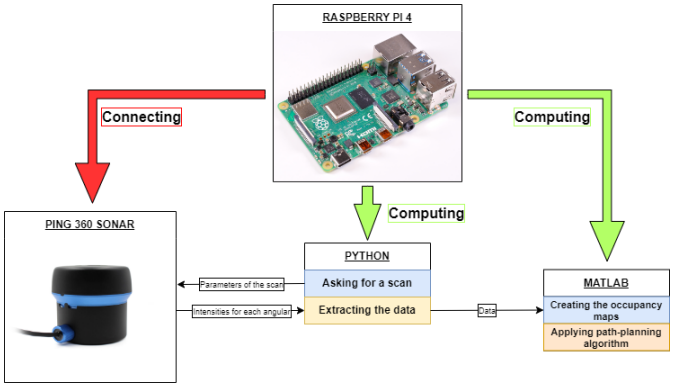


Figure 4: Schematic configuration of the experimental setup

3. RESULT ANALYSIS

3.1 Occupancy Map

By utilizing sonar scans and storing the data in a CSV file, we can generate an occupancy map of the area. For each angle, we capture the corresponding intensity values at various ranges, enabling us to construct a matrix that represents the polar map.

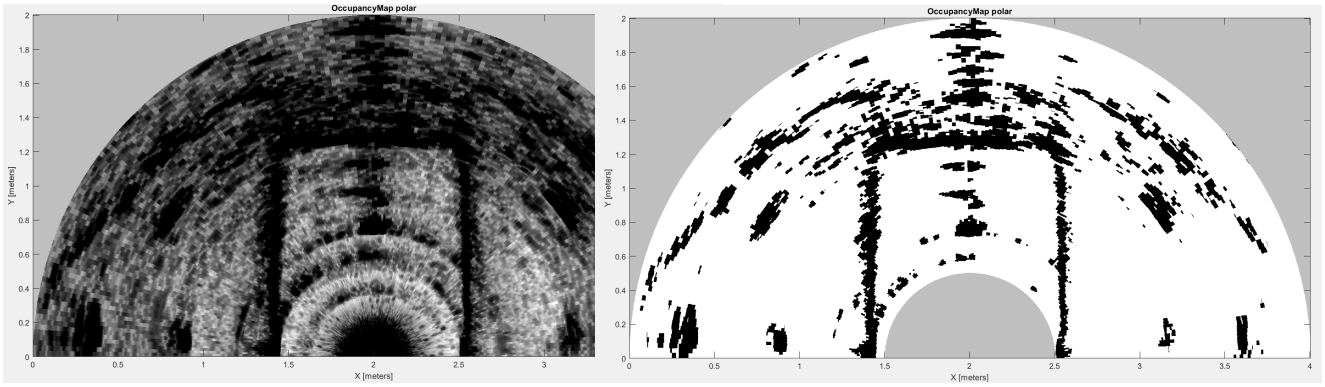
However, a challenge arises due to the angular resolution of 1 gradian, which is the minimum settable resolution. This resolution is too coarse for creating a usable occupancy map, as it leaves gaps between each gradian. These gaps may be interpreted as free space by path-planning algorithms, leading to inaccuracies in obstacle detection.

To mitigate this issue, we adopt the following assumption: any object detected at a specific angle is also present in the space between that angle and the previous one. To implement this, we duplicate the data from one gradian across the intermediate spaces up to the previous gradian.

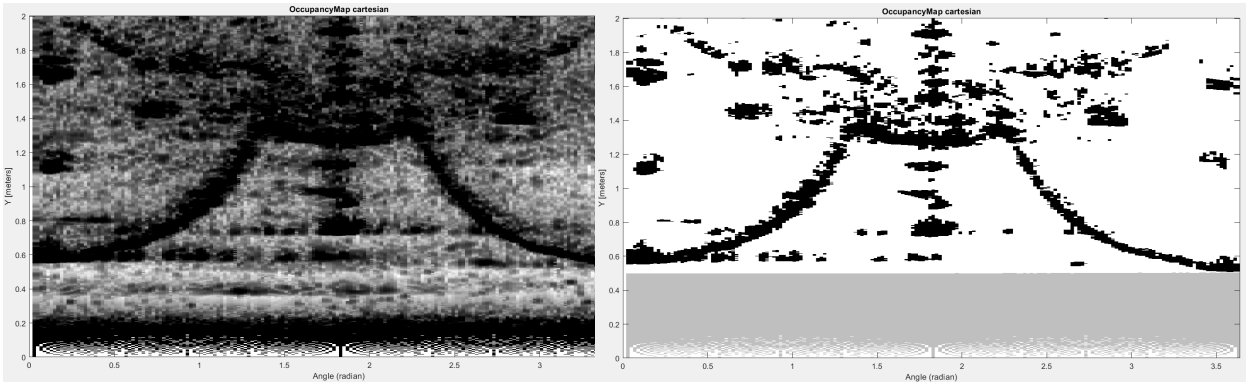
We generate two types of occupancy maps: the polar map, which depicts the area around the sonar from a top-down perspective, and the Cartesian map, which plots the angle scanned on the x-axis and the range where the intensity was detected on the y-axis.

For effective path-planning, the occupancy maps must be binary, where each pixel is either 0 (representing free space) or 1 (representing an obstacle). To achieve this, our map generation functions accept a threshold value between 0 and 255, which determines the intensity level at which a pixel is classified as an obstacle.

Figure 5 shows the original and modified occupancy maps, with a threshold set at 255.



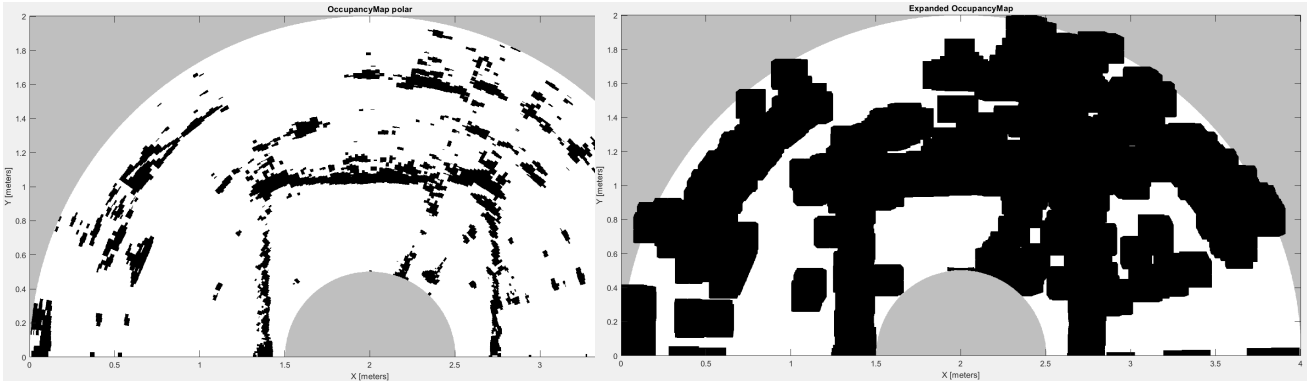
(a) Polar Map (b) Modified Polar Map



(c) Cartesian Map (d) Modified Cartesian Map

Figure 5: Original and modified occupancy maps

Furthermore, it is crucial to consider the size of the AUV to determine whether it can navigate between obstacles. This is done by expanding the pixels that represent obstacles by half the size of the AUV. This expansion ensures that the AUV's size is accounted for, even though the robot is represented by a single pixel. Figure 6 illustrates an expanded occupancy map.



(a) Polar Map without Expansion (b) Expanded Polar Map

Figure 6: Example of an expanded occupancy map

3.2 Subgoal Identification

Our objective is to identify a goal position within a specified area on the polar map, as defined by the operator. In our experiments, the goal position is constrained within two angular limits and two range boundaries.

In Figure 7, we show an example where the ROV needs to find a goal within a specific sector: the left angular tolerance is marked in red, the right angular tolerance in orange, the farthest range in yellow, and the nearest range in blue. The optimal goal, ideally located at the farthest possible point within this sector, is indicated by the center of the green circle.

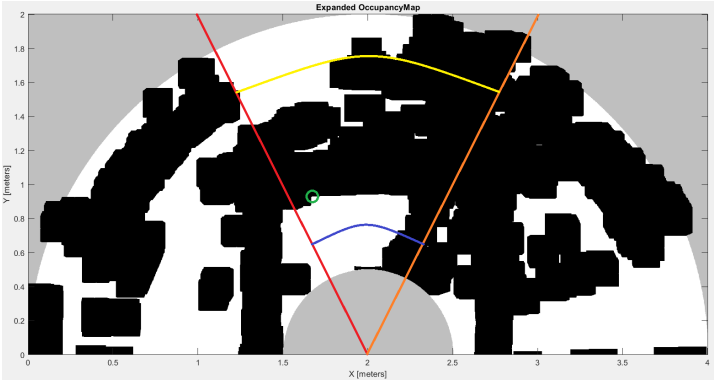


Figure 7: Illustration of the constraints for subgoal identification

We explore two approaches to finding the subgoal:

- A* Algorithm: We use MATLAB's A* algorithm to attempt to reach the farthest allowable range. If the goal is not reachable, the algorithm returns an error, allowing us to identify non-reachable areas.
- Grid Search Method: Starting from the initial position, we identify all reachable pixels and then select the farthest one as the subgoal.

Both methods can be applied to either the polar or Cartesian map. To expedite the search process, the Cartesian map is simplified with abscissas varying directly by 1 degree. Once the subgoal is identified, we calculate its corresponding position on the polar map.

For the A* approach, we first attempt to reach the farthest range within the allowed area. Given that using A* in non-reachable areas can be time-consuming, we assume that if a free space is non-reachable, adjacent free spaces to the right and bottom are also non-reachable. In the grid method, we analyze each point along the same abscissa as the start position, incrementing the ordinate position until the farthest point is identified.

Figure 8 compares the temporal performance of both methods when applied to the previous map, with a maximum range of 1 meter and zero-degree angular tolerance on each side.

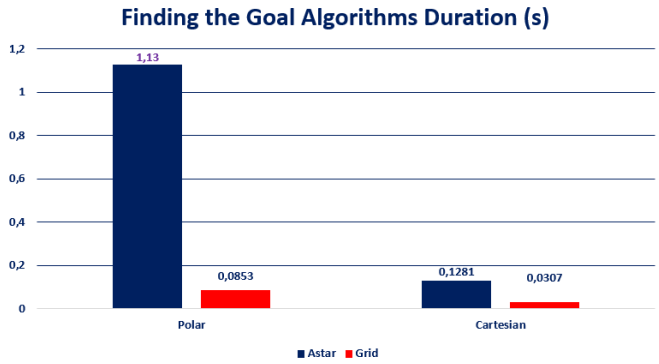


Figure 8: Time performance for subgoal identification at 1 meter

The following observations can be made:

- The Cartesian map yields better performance in both methods due to its lower pixel count compared to the polar map.
- The grid method outperforms the A* algorithm in efficiency.

However, the A* algorithm's efficiency is contingent on the map and the specific constraints. For example, if the operator's chosen farthest goal is directly reachable, the A* method performs faster. In our case, where the farthest goal was within the water tank, A* was relatively efficient. However, when extending the maximum range to 2 meters (beyond the tank's limits), the A* algorithm's time consumption increases significantly, as shown in Figure 9.

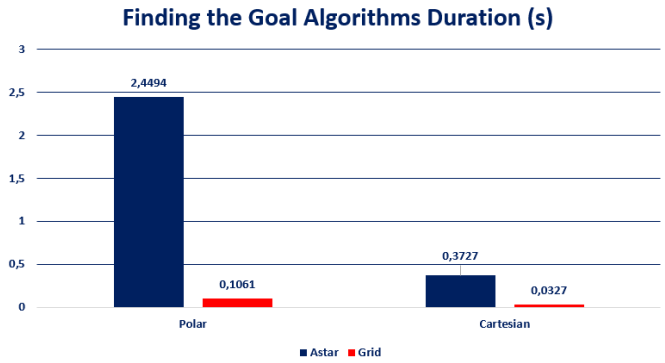


Figure 9: Time performance for subgoal identification at 2 meters

This demonstrates that while the grid method is consistently time-efficient, the A* method may vary greatly depending on the scenario. Despite this, the A* method has the advantage of providing the path between the start and subgoal, which can be beneficial for path planning.

3.3 Path-Planning Algorithm

We now apply the path-planning algorithm discussed in Section 2.2. For this experiment, we directly incorporate currents as vectors into the MATLAB environment. We introduce both impassable strong currents and navigable currents that the AUV can utilize. The goal is to observe whether the generated path avoids the overly strong currents while leveraging beneficial currents.

Figure 10 displays the generated path between the start and subgoal positions, with different current regions shown, including the impassable strong currents marked in red.

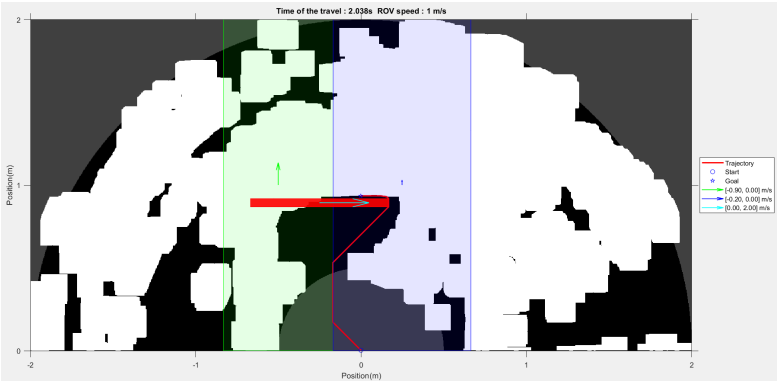


Figure 10: Generated path considering current effects

As shown, the ROV deviates to the right to take advantage of a favorable current while avoiding the stronger currents.

Next, we compare the time consumption of this method with the A* algorithm discussed previously. The results are presented in Figure 11.

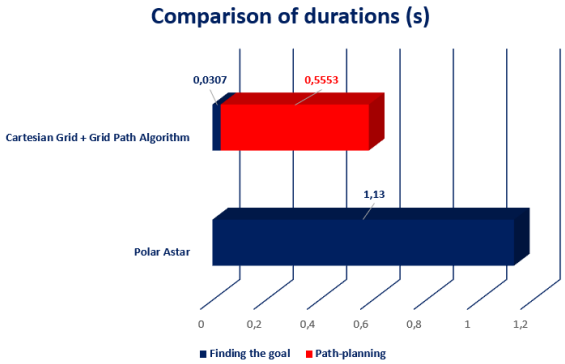


Figure 11: Comparison of path-planning algorithms

The comparison reveals that the grid path search is significantly more efficient than the A* approach. Additionally, our method successfully integrates the effects of underwater currents, enhancing its practical utility.

3.4 Overall Results

Finally, we summarize the duration of each process for two different sample numbers, as shown in Figure 12.

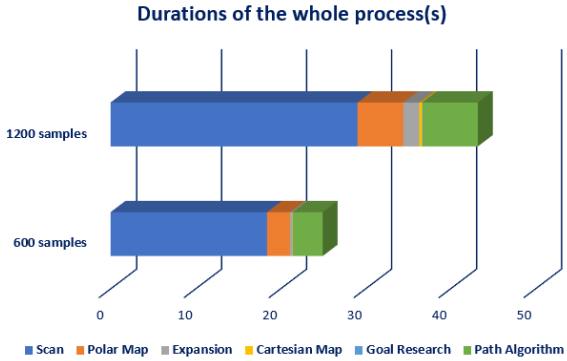


Figure 12: Summary of the steps in the path-planning process

The most time-consuming step is the sonar scan, whose duration increases with the number of samples. Each MATLAB process also takes longer as the sample number increases.

Overall, the entire process took less than one minute, which is satisfactory for real-time navigation scenarios.

The primary goal of path-planning algorithms is to find a path between a start and a goal, avoiding obstacles while minimizing costs, which could include time, distance, or energy consumption. The A* algorithm is well-known for its efficiency due to its heuristic approach, which effectively tests paths close to the optimal one, often referred to as the "Bird's Eye View."

Other algorithms, like Rapidly Exploring Random Tree (RRT), aim to find paths that are not necessarily optimal but are less costly in terms of computation. RRT tends to yield worse results than A* except in high-resolution maps.[2]

With the advent of AI, new algorithms have been developed, utilizing Deep Learning or Transfer Learning. Sonar is one of the most widely used tools underwater. To use it effectively, it is necessary to understand the sonar equation, which relates the power of the received signal to the source, the target, and the environment. A fundamental reference for understanding sonar principles is "Principles of Underwater Sound" by R.J. Urick.[9]

Sonar is primarily used for detecting obstacles or targets, but high-quality sonar can also be employed for object recognition, providing detailed images that enable classification by users or through advanced AI models utilizing deep learning ,[13–15] applicable to various industrial use cases.

4. CONCLUSION AND FUTURE WORK

This paper has introduced a novel path-planning approach for Autonomous Underwater Vehicles (AUVs) operating in dynamic and complex underwater environments, utilizing single-beam sonar data from the Ping360 sonar. We addressed the limitations of traditional sonar systems by developing a method that effectively leverages sonar scans for accurate obstacle detection and avoidance. By incorporating environmental factors such as water temperature, depth, and salinity into the sonar's range detection capabilities, and by extending the algorithm beyond the traditional A* algorithm to account for underwater currents, we have demonstrated significant improvements in the navigational efficiency and safety of AUVs.

The results of our comprehensive simulations, conducted in a controlled tank environment, validate the effectiveness of our approach. These simulations have shown that our method can significantly enhance AUV operational capabilities, particularly in scenarios that require reliable navigation in uncertain and changing underwater conditions, such as in defense and security applications.

Looking forward, several areas of future work have been identified to build upon the foundation laid by this research. First, real-world testing in diverse underwater environments will be crucial to validate the robustness and reliability of the proposed method. Such field trials will help to identify any potential challenges that may arise when transitioning from controlled simulations to more unpredictable real-world conditions. Additionally, integrating advanced sensors, such as multibeam sonars, Doppler velocity logs (DVL), or optical cameras, could further enhance the accuracy of obstacle detection and environmental mapping.

Another promising direction for future research involves the incorporation of adaptive learning algorithms, such as reinforcement learning or online machine learning. These algorithms could enable the AUV to continually improve its path-planning capabilities based on real-time experiences and environmental feedback, leading to more autonomous and efficient navigation. Moreover, given the energy constraints inherent to AUV operations, optimizing the path-planning algorithm to minimize energy consumption while maintaining accuracy is a critical area for further exploration.

Finally, the concept of deploying multiple AUVs in a collaborative network presents an exciting opportunity to explore. Such an approach could allow for more efficient and comprehensive mapping and navigation of large underwater areas, offering significant benefits for complex missions.

In conclusion, the path-planning method proposed in this paper represents a significant advancement in AUV navigation, particularly in challenging underwater environments. The continued development of this approach, along with the exploration of new technologies and methodologies, will be essential in pushing the boundaries of what AUVs can achieve in the future.

Declaration on the Use of Generative AI Tools

In the preparation of this manuscript, various generative AI tools, including ChatGPT and QuillBot, were utilized to assist in the modification and enhancement of the writing. These tools were employed to refine the language, improve clarity, and ensure coherence throughout the document. However, it is important to emphasize that the core research, including the development of methods, execution of experiments, analysis of results, and formulation of conclusions, was solely conducted by the authors.The use of AI tools was restricted to linguistic enhancements and did not influence the scientific content or integrity of the research presented in this paper. Following the use of these tools, the authors meticulously proofread the manuscript to verify the accuracy of the content and to ensure that the final text accurately reflects the original research and the authors' intentions.

We declare that the intellectual contributions, ideas, and findings presented in this paper are the result of our own work and research, and that AI tools were used solely as a supplementary resource for improving the manuscript's readability.

REFERENCES

[1] M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," *Robotics and Autonomous Systems* **174**, p. 104630, 2024.

[2] J. Braun, T. Brito, J. Lima, P. G. d. Costa, P. Costa, and A. Y. Nakano, "A comparison of a* and rrt* algorithms with dynamic and real time constraint scenarios for mobile robots," in *9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2019*, pp. 398–405, Scitepress, 2019.

[3] S. Chehelgami, E. Ashtari, M. A. Basiri, M. T. Masouleh, and A. Kalhor, "Safe deep learning-based global path planning using a fast collision-free path generator," *Robotics and Autonomous Systems* **163**, p. 104384, 2023.

[4] A. I. Panov, K. S. Yakovlev, and R. Suvorov, "Grid path planning with deep reinforcement learning: Preliminary results," *Procedia computer science* **123**, pp. 347–353, 2018.

[5] T. Blum, W. Jones, and K. Yoshida, "Deep learned path planning via randomized reward-linked-goals and potential space applications," *arXiv preprint arXiv:1909.06034* , 2019.

[6] M. G. Tamizi, H. Honari, A. Nozdryn-Plotnicki, and H. Najjaran, "End-to-end deep learning-based framework for path planning and collision checking: bin-picking application," *Robotica* **42**(4), pp. 1094–1112, 2024.

[7] M. Valdenegro-Toro, "Object recognition in forward-looking sonar images with convolutional neural networks," in *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–6, IEEE, 2016.

[8] G. Neves, M. Ruiz, J. Fontinele, and L. Oliveira, "Rotated object detection with forward-looking sonar in underwater applications," *Expert Systems with Applications* **140**, p. 112870, 2020.

[9] R. J. Urick, "Principles of underwater sound-2," 1975.

[10] Mathworks, "plannerAStarGrid." Mathworks, Introduced in R2020b https://ch.mathworks.com/help/nav/ref/plannerastargrid.html?searchHighlight=plannerAstargrid&s_tid=srchtitle_support_results_1_plannerAstargrid. (Accessed: July 2023).

[11] M. Fava, "How much of the Ocean has been explored?." Ocean Literacy Portal, 09 May 2022 https://oceanliteracy.unesco.org/ocean-exploration/. (Accessed: June 2023).

[12] B. Robotics, "Ping360 Scanning Imaging Sonar." Blue Robotics https://bluerobotics.com/store/sonars/imaging-sonars/ping360-sonar-r1-rp/. (Accessed: May 2023).

[13] M. J. Hasan, E. Elyan, Y. Yan, J. Ren, and M. M. K. Sarker, "Segmentation framework for heat loss identification in thermal images: Empowering scottish retrofitting and thermographic survey companies," in *International Conference on Brain Inspired Cognitive Systems*, pp. 220–228, Springer Nature Singapore, 2023.

[14] R. Muthukrishnan, S. Kannan, R. Prabhu, Y. Zhao, P. Bhowmick, and M. J. Hasan, "Tracking and estimation of surgical instrument position and angle in surgical robot using vision system," in *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*, pp. 1–6, IEEE, 2023.

[15] M. Sohaib, M. J. Hasan, M. A. Shah, and Z. Zheng, "A robust self-supervised approach for fine-grained crack detection in concrete structures," *Scientific Reports* **14**(1), p. 12646, 2024.