

PATH PLANNING OF UNDERWATER ROBOTS IN UNKNOWN ENVIRONMENT

Rishabh Gupta

Literature Review Paper

ME 656: Autonomous Navigation for Mobile Robots, Spring 2022

Stevens Institute of Technology

ABSTRACT

These days multiple types of research are going on on path planning of underwater robots. The term path planning refers to discovering a path that allows a robot to move from one point to another while avoiding collisions with obstacle. This paper presents three different frameworks and real-world results of path planning of underwater vehicles in an unknown environment. They are Real-time path planning of underwater robots, Online path planning of underwater vehicles, and Path planning of underwater gliders are discussed.

INTRODUCTION

The most common types of underwater vehicles are Remotely Operated Vehicles, Autonomous Underwater Vehicles (AUV), and underwater gliders. The main objective behind developing underwater vehicles is for oceanography and surveillance include autonomous mine detection, structural inspection, inspecting ship hulls and security. One of the most debated research subjects in the underwater vehicle field is how to enhance the vehicle's autonomy. The term "autonomy" refers to a vehicle's ability to do operations without the need for human interaction. Path planning has been a hot topic among many autonomous systems in recent decades. To navigate autonomous underwater robots in an unknown environment, different path planning algorithms are required.

In this paper, three different studies of path planning algorithm of underwater robots (UR) in unknown environment are discussed. The first is online path planning of AUV uses combination of anytime path planning algorithm & lazy collision evaluation to expand the RRT* (rapidly random tree) approach for planning online collision-free paths [1]. The second, real time path planning of underwater robots where a* algorithm is applied to plan optimal path [2]. The third shows path planning of underwater glider using combined A* search and nearest search (ND) algorithm [3].

Online path planning of AUV'S

The day-by-day development in AUV's technology introduced new research and development area in applications like Imaging and examining various structures such as in-water ship hulls, complex structures on the seabed, or enclosed natural structures. In these situations, it is very difficult to operate AUVs in an unknown potentially dense environment where they are exposed to collisions. To overcome this issue a path planner with online capabilities is introduced to solve global position inaccuracy while navigating through close proximity to nearby obstacles.

In this study, the authors have introduced an online path planning technique for AUVs using an RRT* sampling-based algorithm with the combination of lazy collision evaluation and anytime path planning algorithms used to extend RRT* for online path planning while moving in an unknown environment. To accomplish this, they introduced a framework that includes an online mapping module as well as a mission controller that organizes task execution. The framework presented consists of three primary modules:

- a) The *mapping* module builds an occupancy map of the environment using data from various perception sensors like laser rangefinders, echo sounders, etc.
- b) a *planning* module that generates a collision-free path from start to goal online with evaluating parameters like computing time, minimum distance to the target, and workspace limits.
- c) a *mission handler* module that works as a high-level coordinator of the planner and the AUV controllers.

An (RRT*) is a space-filling tree that is randomly built to effectively explore nonconvex, high-dimensional environments to deliver the shortest possible path to the goal. To solve online path planning tasks, they have extended concepts of RRT* with the combination of two approaches [1].

Any time approach for replanning online

RRT* is also an example of an anytime algorithm, where authors have improved and formalized such extension by adding other rapidly-exploring-random-tree (RRT) variants. This version consists of two procedures, *build* and *extend* where it generates a tree rooted at the start configuration and controls its execution based on other variables such as the build procedure's time (t_{build}) for each callback and the computation time (t_{comp}) for extending the tree [1].

Lazy collision evaluation for replanning online

In the proposed sampling-based algorithm RRT we present, most of the configurations use to expand and explore the c-space are in undiscovered regions of the workspace. By using octomap we can verify in advance if the configuration is in an explored area or not, which helps in avoiding unnecessary collision routine callbacks. If the configuration sampled from expansion is out of an unknown or explored area, the planner assumes it as valid. As vehicles move and explore the area, the parts of the tree are verified and discovered if found under collision. This approach is called the Lazy Collision Evaluation.

Real-World Results

To evaluate the presented framework, they used the SPARUS-II AUV, a torpedo-shaped vehicle, and performed an experiment at Sant Feliu de Gu'ixols in Catalonia (Spain). In

the experiment, the SPARUS-II AUV had to navigate through the concrete blocks without any prior knowledge of their placement [1].

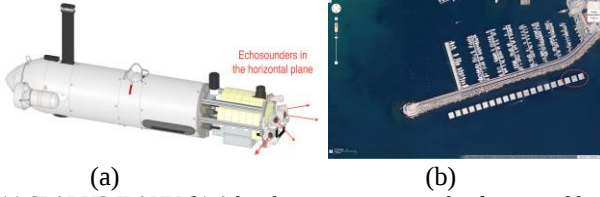


Fig.1 (a) SPARUS-II AUV (b) A breakwater structure made of concrete blocks at Sant Feliu de Gu'xols in Catalonia, Spain.

The start and goal configurations were on opposite sides of the breakwater structure. At the start of path planning, it is observed that octomap was covered by a real image of concrete blocks. This is most commonly produced when an echosounder sends a single noisy signal over an undiscovered area on the map that is marked as occupied. However, this problem was resolved using a probabilistic framework [1].

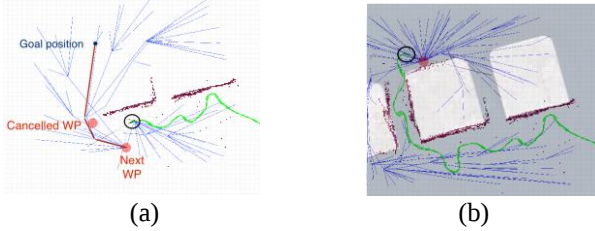


Fig.2 (a) Near a series of occupied sites, a waypoint has been created. Later, the point is invalidated, and the replanning procedure generates a new waypoint. (b) Resulting map shows the concrete block and last waypoint.

The quality of the first solution was inefficient in terms of distance because of t_{comp} increases and the quality of the first result decreases after replanning due to the greater computational power of the vehicle. It was discovered that when a non-zero pitch motion is combined with an echosounder that functions as a single beam sensor, obstacles in the vehicle's direction of motion take longer to identify, which results in delaying replanning operations. Along with this, it is seen that repetitive replanning due to resulting pathways that did not consider the vehicle's motion constraints, particularly in turning maneuvers [1].

To overcome observed issues, they have simulated start-to-goal queries and used the presented framework to solve them using an RRT considering vehicle differential constraints. Fig.3 shows one of the simulated paths in UWSim. These simulations produce pathways that are seen to be collision-free and feasible, which minimize callbacks in the replanning stage and unnecessary movements [1].

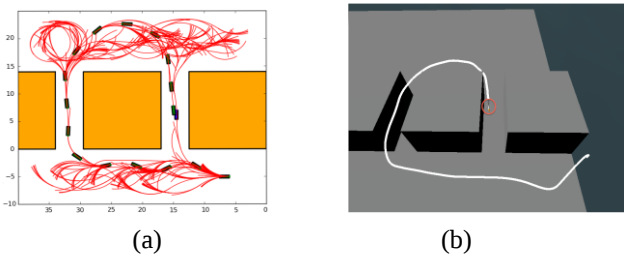


Fig.3 (a) the RRT expansion with differential constraints. (b) simulation over UWSim of SPARUS-II

This proposed framework for planning collision-free paths online for AUV uses a modified form of RRT*, which allows replanning while moving from an unknown environment. This framework has an online mapping module and mission handler to execute the task from outside. The replanning capacity of the presented approach and probabilistic mapping framework helped in overcoming the noisy measurement. The capacity to incrementally generate maps while planning paths has been demonstrated in AUV mapping results. Preliminary results on path planning with differential constraints were also presented [1].

Real time path planning of Underwater vehicles

In this study, we look at a scenario in which an underwater vehicle is traveling through unknown terrain. For obstacle detection, the vehicle has a 2D front-facing image sonar positioned in front. As a result, from the standpoint of path planning, assuming that there are no obstacles in this unknown area may be a logical choice. As a result, 2D space can be separated into two types of regions: known and unknown. The obstacle is represented as a type of polygon based on the detected contour, and its shape can vary depending on the contour shape. Taking this assumption into consideration visibility graph is generated which is called a rubber band visibility graph (RBVG). A visibility graph is a graph of intervisible locations in the Euclidean plane, usually for a series of points and obstacles. To make an RBVG graph to explore an unknown area author has considered some assumptions they are a) Its two endpoints are set at q_{init} and q_{goal} , respectively, and the length satisfies $l_{rb} < ||q_{goal} - q_{init}||$ addition, the band can be stretched indefinitely, b) It cannot pass through the unknown region, c) It will never intersect itself. As this graph differs from the real one, we call this virtual RBVG. Using this graph, we use the A* algorithm to find a kind of optimal path, with the traveling time acting as the heuristic cost function. In today's AUV technology, energy usage is generally considered one of the top priorities. The shortest path has the least amount of energy consumption if there is no external disturbance. When the sea current and flow are significant, however, this is not the case. The least energy usage reflects the least travel time in the case of torpedo-type vehicles, which normally move at constant forward speeds in most of their operations. The travel time is also used as a criterion for selecting the best option. It's interesting to think about some of the unique situations that can arise during the path-finding process [2].

A) *Multiple Same Cost Paths*: There may be more than one path with the same heuristic cost function given a VRBVG. Various points of view can be examined in this case. From the standpoint of collision risk, it might make sense to avoid the road that has more barriers. The more pieces in $C_{unknown}$ will also be at higher danger. We can choose one at random or simply pre-determine the vehicle to always use the right-sided path, or inversely if two tracks have the same length of links adjoining barriers and the same length of pieces placed in $C_{unknown}$, as shown in Fig.4 [2].

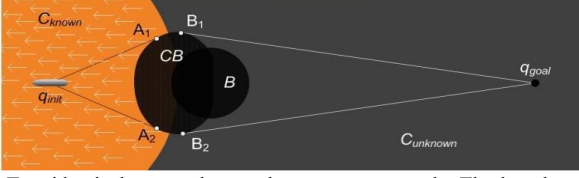


Fig.4 Two identical cost paths are shown as an example. The lengths of two pathways, $q_{init}A_1B_1q_{goal}$ and $q_{init}A_2B_2q_{goal}$, are the same, and the sea current flows in the direction of the vehicle. In this example, it is self-evident that these two roads take the same amount of time to travel.

B) *Wall following*: If the cost of the path $q_{init}, t_kA_1B_1q_{goal}$ is greater than the cost of $q_{init}, t_kA_kB_kq_{goal}$, the vehicle will follow the wall B in the right direction, as shown in Fig.5. If the wall stretches indefinitely to the right, the vehicle will follow it indefinitely. This implies that, even if there is a path, the algorithm presented may fail to find it in some cases. It is important to note that if the sea current flows from right to left or q_{goal} approaches B_1 more closely, then there will be $t_k > 0$ such that the cost of the path $q_{init}, t_kA_1B_1q_{goal}$ becomes less than that of $q_{init}, t_kA_kB_kq_{goal}$, and the vehicle will be able to discover the way approaching q_{goal} [2].

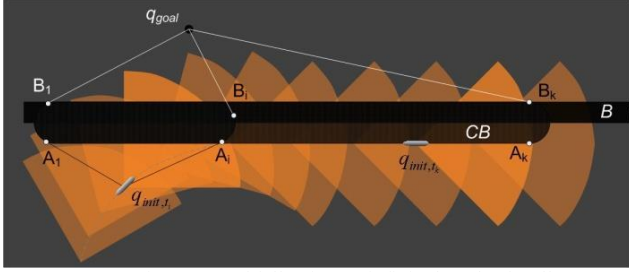


Fig.5 Case of following an infinite length of wall.

Simulation Studies

To demonstrate the efficiency of the suggested path planning, certain numerical simulation studies are carried out. In the simulation, we use the REMUS AUV's 6DOF nonlinear dynamics. We assume that this torpedo-type vehicle travels at a steady forward speed. The path planning algorithm can give the reference heading of the explored path at any point, which is denoted as Ψ_d . The vehicle's reference heading Ψ_r can be considered as Ψ_d without the sea current.

Figures 6 and 7 show the simulation findings. Figures 6(a) and 6(b) are nearly identical examples with the exception of a little change in the placement of the second barrier. As previously stated, the VRBVG described in this paper is often built using just partially available environmental data. As a result, the searched path is only slightly suboptimal, as illustrated in Figure 3.

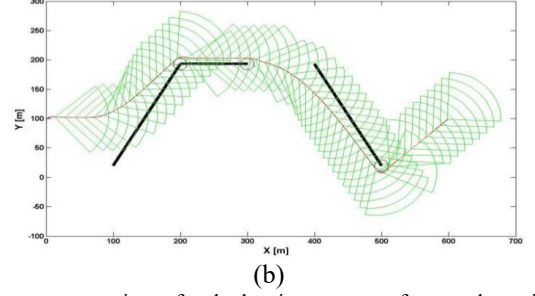
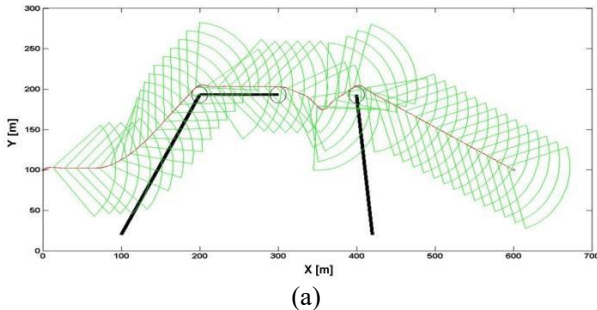


Fig.6 shows a comparison of path planning outcomes for two almost identical scenarios with only minor differences in obstacle placement.

Fig.7 is a zoomed-in portion of the graph in Figure 3, showing both the matching reference line. As previously stated, the impacts of sea current and flow prevent Ψ_d from directly converting to Ψ_r , which is derived from the path planning concept. Both Ψ_d and the accompanying reference heading Ψ_r can be seen here. As previously stated, Ψ_r cannot directly replace the Ψ_d , which is obtained from the path planning scheme, because of sea current and flow. Instead, a change should be made to overcome the sea current/flow [2].

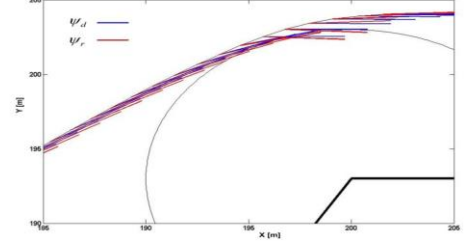


Fig.7 The reference heading Ψ_d is determined using the path planning scheme, while the real one Ψ_r is changed to account for the impacts of the sea current and flow.

The main goal behind path planning is to establish a reference path for the vehicle to follow to manage its movements. As a result, the vehicle's dynamics and control problem should also be considered during the path planning step. In this study, the authors have proposed real-time path planning for underwater vehicles in an unknown environment with zero sea current and flow. A visibility graph (RBVG) is made to generate a roadmap. To create the roadmap, a type of visibility graph known as RBVG (rubber band visibility graph) is used. This graph is based on the premise that there are no obstacles outside the sonar range of the vehicle. We call this graph virtual RBVG since it may differ from the real one. Where they have used an A* method to find a suboptimal path using this graph. The heuristic cost function is selected with the goal of reducing energy use [2].

Path planning of Underwater Glider

A glider is a form of underwater vehicle that functions by cyclically changing its buoyancy. The combined impact of internal mass displacements and the vehicle wings and tail orientation produces vertical impulsion that is translated into an effective but low surge speed, resulting in a series of up/downslope or climb/dive transects. They're ideal for long-range operations because of their unrivaled autonomy, yet their slow surge speed makes them vulnerable to ocean currents. Path

planning is a major problem for this sort of vehicle because it can shorten the time it takes to reach a particular destination or conserve energy. It is difficult to discover an ideal solution in such a dynamic environment, and any such solution would demand a lot of processing power. In this study, researchers describe a low-cost path planning technique for this type of underwater vehicle that permits static or dynamic obstacle avoidance, which is typically required in coastal situations with land areas, high tides, shipping routes, etc [3].

In this study, the author introduced a novel path planning technique for underwater gliders in hazardous coastal areas that incorporates an initialization module to avoid obstacles and is based on A*-based search and Nearest Diagram (ND) algorithms. The focus of the researchers was to reduce the time and energy consumption of gliders. Along with this, they have compared the results of the proposed approach with previous algorithms. Here are following algorithm:

Direct to Goal algorithm: The next bearing is determined as the direction to the destination location at each surfacing. It does not take into consideration ocean current forecasts. This isn't a path planning method, but it does mimic glider behavior. To compare new developments, we mostly used this algorithm as a benchmark. For managing ocean currents, they have used A* method. The main disadvantage of this method is that it does not produce continuous time periods as gliders do. Because ocean currents are non-static, optimality is no longer expected [3].

CTS-A* Method: They created the CTS-A* algorithm a version of A*, to overcome previous constraints. A set of bearing is considered at each surfacing point, and the glider trajectory is integrated for each one over a constant-time stint. Although they are maintained in a search grid, the surfacing places remain continuous. This method has two drawbacks: the bearing space is discretized, and as the number of bearings grows, the computing cost grows exponentially [3].

Optimization-based algorithm: Finally, they solved the problem using optimization approaches. They employed the last surfacing distance to the goal waypoint as an objective function, and the bearing at each submerged stint as variables, which were iteratively optimized to determine the cheapest approach. The optimization procedure's cost function is calculated using a stint simulator that recreates the glider trajectory using the commanded bearing, nominal glider speed, and 2D ocean currents. This proposed approach gives acceptable results but as shown in this work they are looking for short-term-costal navigation. Due to the complexity of the environment, the optimization can easily be trapped on the wrong paths [3].

They built a new path planner called Optimization with Intelligent Initialization to solve the constraints of obstacle avoidance that we discovered in prior versions of our system. This algorithm includes a bootstrap module that is based on the CTS-A* search and ND algorithms. In the initialization phase,

candidate trajectories are divided into two or more stages. These possible trajectories combine all the stints into one stage using a fixed bearing. The nodes serve as junctions between stages. The algorithm allows these spots to be moved around [3].

To verify the proposed algorithm, they have compared their results with those produced by other algorithms used in the planning of trajectories for gliders. The first collection of examples corresponds to coastal trajectories, whereas the second solely contains offshore trajectories are performed in canary Island from ESEOO project model. Here, I have included two scenarios of experiment results that portray trajectory simulated near coastal& offshore of all proposed algorithms. The author observed that proposed algorithm will help in the reduction of computational and economical costs with the ability to cover the distance in less time compared with others [3].

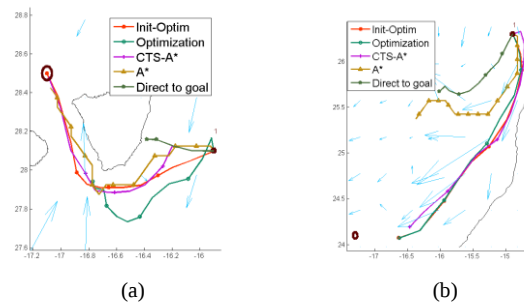


Fig.8 Trajectories simulated near (a) coastal areas & (b) offshore area

CONCLUSION

In sum, the proposed three approaches performed satisfactorily in their field of study. Although, according to my point of view online path planning of AUVs is a more effective way of path planning in today's scenario. The RBVG visualization graph and proposed approach further help optimize path planning in a vacuum. The third approach shows that if the gliders are provided an artificial source of propulsion, they could possibly out speed the first two approaches.

REFERENCES

- [1] J. D. Hernández, E. Vidal, G. Vallicrosa, E. Galceran and M. Carreras, "Online path planning for autonomous underwater vehicles in unknown environments," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1152-1157, doi: 10.1109/ICRA.2015.7139336.
- [2] J. Li, H. Kang, G. Park and J. Suh, "Real Time Path Planning of Underwater Robots in Unknown Environment," 2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), 2017, pp. 312-318, doi: 10.1109/ICCAIRO.2017.64.
- [3] Isern, Josep & Hernández-Sosa, Daniel & Fernández Perdomo, Enrique & Cabrera, Jorge & Domínguez Brito, Antonio & Víctor, Prieto-Marañón. (2012). Obstacle Avoidance in Underwater Glider Path Planning. Journal of Physical Agents. 6. 11-20. 10.14198/JoPha.2