Simulation for Path Planning of Autonomous Underwater Vehicle Using Flower Pollination Algorithm,Genetic Algorithm and Q-Learning

Utkarsh Gautam
JSS Academy of Technical Education
Noida, India
ugautam_91@yahoo.com

R.Malmathanraj
Department of ECE, NIT
NIT, Trichy
Trichy, India
rmathan@nitt.edu

Chhavi Srivastav
JSS Academy of Technical Education
Noida, India
chhavi.nancy@gmail.com

***Abstract*—the motivation behind this paper is to address the necessity for exploration in near bottom ocean environment employing Autonomous Underwater Vehicles. This paper presents a simulation for an optimized path planning for an autonomous underwater vehicle in benthic ocean zones. The statistical data pertaining to the near-bottom ocean currents has been sourced from the Bedford Institute of Oceanography, Canada. A cost function is developed which incorporates the interaction of the underwater vehicle with the ocean currents. This cost function takes the source and destination coordinates as the inputs and outputs the time taken by the vehicle to travel between them. This paper aims to minimize this cost function to obtain a path having the least travel time for the vehicle. Various biologically inspired algorithms such as Flower Pollination Algorithm and Genetic Algorithm have been used to optimize this cost function. The optimization of the cost function has also been performed using Q-Learning technique and the results have been compared with the biologically inspired algorithms. The results depict that Q-Learning Algorithm is better in computational complexity and ease of simulating the environment. Thus, an efficient Path planning technique, which has been tested for the cost function of an autonomous underwater vehicle is proposed through this paper.**

Keywords— Simulation, Path planning, Near Bottom Ocean Currents, Benthic Ocean Zones, Q-Learning, Flower Pollination Algorithm,Genetic Algorithm.

I. INTRODUCTION

The exploration and analysis of the bottom of an ocean, referred to as the benthic zone, is crucial for various scientific research activities. Study of the benthic zone plays a pivotal role in processes such as procurement of the location of minerals, oils, gases and other resources, and prediction of Geo-hazards such as Landslides which may lead to Tsunami turbidity currents [8]. Recent advancements in technology have facilitated the increased use of Autonomous Underwater Vehicles (AUV) for efficient exploration of the vastness of the water bodies. The AUV's are competent for a variety of missions investigating the base of water bodies. To help facilitate the analysis of benthic zones, this paper presents a simulation for the path planning of an AUV. The simulation has been done on MATLAB platform. The statistics for the velocity vectors of the ocean currents at near bottom ocean zones has been sourced from the Bedford Institute of Oceanography, Canada [9]. Using this data, a cost function is devised, which represents the AUV's interaction with the benthic environment. Section II of this paper describes the details of the cost function. This cost function is optimized to obtain the shortest path between the source node and destination node. This optimization is established using multiple biologically inspired algorithms such as Flower Pollination Algorithm (FPA), and Genetic Algorithm (GA). Simulation is also performed using Q-Learning technique and the results derived from both the biologically inspired algorithms and Q-Learning technique are compared. The results depict that the Q-Learning technique simulates the environment more precisely, and is more computationally economical than biologically inspired algorithms. Since the introduction of AUV's, lot of research has been done to increase their efficiency in path planning and obstacle avoidance capabilities. M.Eichhorn et al published a series of works [2-5], using graph based methodologies, on the path planning and obstacle avoidance of the AUV SLOCUM Glider. In India, a team of technocrats from AUV-IITB have developed 'Matsya' [10]. M.P. Aghababa used a plethora of ground breaking algorithms for facilitating path planning and obstacle avoidance in an underwater vehicle [6]. K.F. Man et al published a detailed work on Genetic Algorithm: Concepts and Application [1]. Xin-She Yang et al published extensive works on Flower Pollination Algorithm [7]. Such exemplary works have inspired us to strive harder, and taking the cue, we have proposed a method for path planning of an AUV utilizing Q-Learning technique and have compared it with other algorithms.

978-1-4799-7171-8/15/$31.00 ©2015 IEEE

II. COST FUNCTION

A cost function is essential to depict the interaction of the AUV with the near bottom ocean environment, specifically with the near bottom ocean currents. In this paper a cost function has been developed, which takes into account the principle of locomotion of the AUV and the spatially varying velocity of ocean currents at benthic levels. This cost function takes the coordinates of the source and destination as the input and produces the output in the form of time taken to travel from the source to the destination. We seek to minimize this time using various optimization techniques. This Time taken for the AUV to travel from source point to goal point is denoted by 'T', and is derived by the kinematics equation:

$$S=UT+\frac{1}{2}aT^2 \tag{1}$$

Where 'S' denotes the displacement, 'U' is the initial velocity, 'a' is the acceleration and 'T' is the time. Since AUV's generally possess low cruising speed (0.2-0.4 ms^{-1}), we can take acceleration to be zero, thus:-

$$T=\frac{S}{U} \tag{2}$$

Here 'T' is the time taken from source point to goal point, S denotes the distance between the source coordinate and goal coordinate and U is the average velocity with which the AUV travels. The following sub sections illustrate the development of the cost function and the parameters essential for it.

A. Ocean Current Determination

The data for the benthic zone current velocity has been sourced from Bedford Institute of Oceanography. These data constitute of the magnitude and direction of the ocean current velocity at various Latitudes, Longitudes and Depths near the bottom surface of the ocean. They are processed to obtain the values required for a realistic simulation. Following processes are involved in obtaining of the Ocean Current values:

1) Obtaining the data related to the near bottom ocean currents from Bedford Institute of Oceanography

2) Segregating the data to obtain distinct databases of ocean current velocity values for different latitudes, longitudes and depth.

3) Identifying the region of interest.

4) Interpolating the distinct databases of latitude, longitude and depth varying ocean current values using one dimensional interpolation functions within the domain of the region of interest.

5) The ocean current value at a particular coordinate is derived by averaging the interpolated value of the ocean current at the required latitude, longitude and depth.

Since the data obtained has ocean current values at random coordinates, it is essential to interpolate within the range of region of interest to get the ocean current values at regular intervals of the coordinates, thereby improving the accuracy of the simulation.

B. Determination of Average Velocity 'U'

An AUV's velocity at a particular coordinate, denoted by 'V_{ef}' depends on its cruising speed, denoted by 'V_c', the magnitude and direction of the ocean current velocity, denoted by '$V_{current}$' and the direction of the path from the previous point to the present point, denoted by V_{path} (for the source node the previous node is taken as the origin).The interaction of the ocean currents with AUV's can be approximately simulated as an intersection between a line and a circle or a sphere (Ref. 3). The point of intersection gives the V_{ef} as explained by the following equations:

$$\text{Line:} \quad x\,(V_{ef}) = V_{ef}V_{path} \tag{3}$$

$$\text{Circle/Sphere:} \quad V_c^{\,2} = \|x- V_{current}\|^2 \tag{4}$$

$$D = (V_{path}^{\,T}.\, V_{current})^{2} + V_c^{\,2} - V_{current}^{\,T}.\, V_{current} \tag{5}$$

The Discriminant given by equation 5 defines whether a particular trajectory should be completely avoided or not, if the 'D' becomes negative, then Vef has no real value, thus it would be Not a Number, NaN.

$$V_{ef} = V_{path}^{\,T} + \sqrt{D}, \text{ for } D > 0 \tag{6}$$

$$V_{ef} = \text{NaN}, \qquad \text{otherwise.} \tag{7}$$

The average velocity 'U' is calculated by taking the average of the effective velocities at source and destination points, Vef1 and Vef2 respectively.

$$U=\frac{(Vef1+Vef2)}{2} \tag{8}$$

C. Calculation of Distance

The distance formula is applied to compute the distance between the source point and destination point.

$$S=\sqrt{((x2-x1)2+(y2-y1)2(z2-z1)2)} \tag{9}$$

Here S denotes the distance between the source node and destination node, x1, y1, z1 represent the coordinates of source point and x2, y2, and z2 represent the coordinates of destination point.

D. Dive Profile

To either ascend or descend in the depth of a water body, the AUV considered in this paper changes its buoyancy. This describes its locomotion principle. Due to this principle, the AUV possesses a characteristic sinusoidal dive profile. The simulation of dive profile has been approximated, and a saw tooth shaped profile has been obtained as a result. For the sake of simulation we have sampled this dive profile and the discrete coordinates, so derived, are given as input to the optimization algorithms to estimate the shortest trajectory traversing all these nodes. Thus, we ensure that the AUV's characteristic dive profile is included in the path planning. The results depicted in the section IV of the paper also illustrate that the optimized paths derived from the algorithms are in accordance with the dive profile of the AUV. The cost function is first optimized within itself, and only then is it optimized using the several algorithms. Due to its dive profile, the AUV can split its trajectory from a source point to a destination point into multiple saw tooth shaped segments. A function within the cost function estimates the time taken in the case when the number of segments is unity to when the number of segments is maximum. The minimum time taken is selected as the final output of the cost function. The division of a single trajectory into x number of segments can be both a merit and a demerit for the AUV depending on whether it avoids highly turbulent ocean currents or not, the pre-optimization ensures that all the possible cases are taken into consideration, thus facilitating a realistic simulation and efficient path planning.

III. OPTIMIZATION ALGORITHMS

After the development of the cost function, several optimization algorithms are applied to minimize the cost function, which provides us with the minimum time taken from source node to destination node. To achieve this, random coordinates are sampled from the dive profile and fed as inputs to the optimization algorithms. The results are observed to follow the dive profile of the AUV. These results are then compared. The inference from these comparisons is illustrated in section IV.

A. Genetic Algorithm

Genetic Algorithm is classified under Evolutionary Algorithms, which are inspired by the methods of natural evolution. This algorithm utilizes several techniques such as inheritance, mutation, selection and crossover repeatedly till only the fittest individual, which is analogous to the most suitable solution to the cost function remains. For the implementation of the Genetic Algorithm , a cost function has to be defined that can compute the fitness of a solution. The potential solutions have to be represented in a suitable format (binary, real, gray coded etc.) The Genetic Algorithm views these prospective solutions of the cost function as the genes of a chromosome and keeps altering them randomly to get the fittest gene. This is the optimized solution to the cost function as depicted in Figure 1.

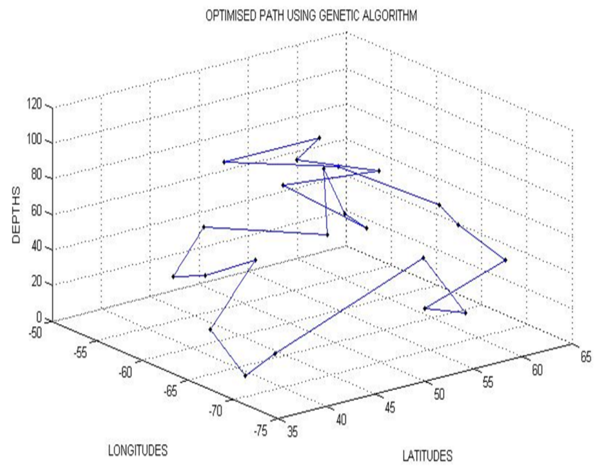

Figure-1: Optimised path using Genetic Algorithm

The pool of prospective solutions is termed as a Population. This population constitutes of randomly generated solutions for the cost function. Using Roulette Wheel method, the crossover operator chooses 'parents' from this population and randomly recombines them to produce 'offspring'. These are also valid solutions of the cost function. The 'offspring' solutions survive only if they yield better results, otherwise they are discarded. The mutation operator randomly selects a candidate from the population and makes random alterations to it. This mutated solution is then checked with the help of the cost function, and survives only if it yields optimized results. Thus, the Genetic Algorithm imitates the principles of natural evolution, and selects the most optimum result.

B. Flower Pollination Algorithm

A nature-inspired algorithm, Flower Pollination Algorithm (FPA) is based on the characteristics of flowering plants. Flower Pollination occurs through the transfer of pollens via agents, termed as pollinators. Broadly classifying the pollination process as local pollination and global pollination, in biotic and cross-pollination processes of global pollination, the pollen-carrying pollinator obey L´evy flights as per the following equation:

$$A^{t+1} = A^t + \beta\, F(x)\, (A^t - S_b) \tag{10}$$

Where A^t is the solution vector A at iteration t, and S_b is the current best solution among all solutions for the current iteration, β is the scaling factor to control the step size and F(x) corresponds to the strength of the pollination.

Local abiotic and self-pollination processes are modelled according to the equation:

$$A^{t+1} = A^{t} + \alpha\,(P_1 - P_2) \qquad (11)$$

Where P_1, and P_2 are pollens from different flowers of the same plant species.

This algorithm works on the principle of the survival of the fittest. For the implementation of FPA, a cost function is formulated, which computes the fitness of a solution. A set of random coordinates are fed to cost function, analogous to the pollens of a flower. The cost function keeps on modifying these coordinates over multiple iterations to achieve a minimum. This is the optimized solution for the cost function.

Using L´evy flights technique, the set of coordinates are randomly altered to achieve the solutions. These are then compared with the previous minimum, and are fit only if they are lesser than the previous minimum solution. Thus, Flower Pollination Algorithm chooses the shortest path from the source node to the destination node and yields the most optimum solution.

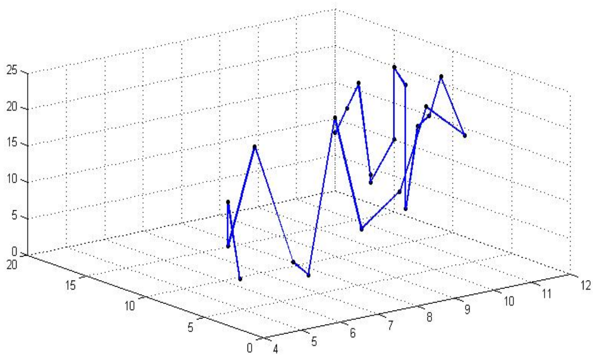

Figure-3: Optimized Path using FPA

C. Q-Learning

A form of Reinforcement Learning, Q-Learning belongs to the super class of Machine Learning. A computer system is said to learn from a data set denoted by ‘D’ to perform the task denoted by ‘T’, if after learning, the system’s performance on T improves, as measured by a performance measurement index, denoted by ‘M’. Trial and error method is utilized by Reinforcement Learning to learn. Random actions are considered to make a system achieve its goal and then, a feedback is obtained in the form of rewards or penalties that determine whether the action taken was correct or incorrect. Therefore, after a number of iterations the algorithm comes to know which steps are beneficial and which are not with respect to the goal to be achieved. In this paper, Tabular Q – Learning Technique is utilized to optimize the path between the source node and goal node.

To implement the Q-Learning, the cost function has been converted to a matrix form, termed as the Reward Matrix and denoted by ‘R’. The Q matrix was initialized by all zeroes. The states are denoted by ‘s’ and actions by ‘a’. Now for a multiple iterations, the Q matrix is altered using the following equation:

$$Q\,(s, a) = R\,(s, a) + \gamma.\ \mathrm{Max}\,(Q\,(\text{next } s, \text{all } a)) \qquad (12)$$

The initial state constitutes of the coordinates of the source node, whereas the next state is chosen randomly. Then all the possible paths from the next state to the goal state are checked and awarded rewards accordingly. After a few iterations, the coordinates of the optimized path possess the highest Q values and hence are chosen as the optimised path.

Figure-3 depicts the value of Q over multiple iterations.

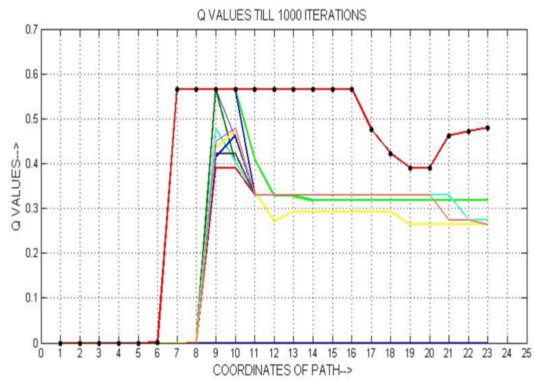

Figure-3: Q values till 1000 iterations

IV. RESULTS AND DISCUSSIONS

Random points sampled from the dive profile are given to the GA. These points include the starting node and goal node. The GA is programmed in MATLAB in such a way that it finds the shortest distance with reference to the cost function of the AUV, from the starting node to goal node, whilst traversing all the given random points and also finding its way back to the starting node. The results obtained show that not only does the GA find the shortest path, the path thus obtained also replicates the dive profile of the AUV as evident from the figure-1. Time taken by the GA to perform is 12.765700 CPU seconds. Figure -1 clearly shows the optimized path generated by the GA. Similarly, random points sampled from the dive profile are given as input to the FPA algorithm. The FPA also gives a similar result with shortest path being evaluated incorporating all the random points given as input. The resulting path also was as expected from the dive profile of the AUV (evident from figure-2). Time taken by the FPA to function was 9.286780 CPU seconds. Figure-2 showcases the optimized path generated by the FPA algorithm.Figure-3 shows the maximum Q value of path obtained after standard one thousand iterations in red color and the other Q value of the tabular column shown in different colors. The coordinates corresponding to the maximum Q values (in the Q matrix) depict the shortest path with reference to the cost function as optimized by the Q-Learning technique. The time taken by the Q-Learning is 7.466598 CPU seconds. It is also seen from the figure-3 that the Q values have saturated for only the optimized path after a standard number of iterations. The γ is used with a constant value of 0.8.

IV. CONCLUSION AND FUTURE SCOPE

It can be concluded from the paper that the Q-Learning approach was better in path planning of the AUV with respect to computational complexity and ease of environment simulation. The paper, therefore presents an effective method of path planning using tabular Q-Learning technique. Concept of function approximation can be used instead of tabular Q-Learning so as to improve computational complexity.

References

[1] K. F. Man, K. S. Tang, and S. Kwong, "Genetic Algorithms: Concepts and Applications," IEEE Transactions on Industrial Electronics, vol. 43, pp. 519-531, October 1996.

[2] Eichhorn, M., "A new concept for an obstacle avoidance system for the AUV "SLOCUM glider" operation under ice," OCEANS 2009 - EUROPE, vol. 1, no. 8, pp.11-14, May 2009

[3] Eichhorn, M, "Optimal Path Planning for AUVs in Time-Varying Ocean Flows," in Proc. 16th Symposium on Unmanned Untethered Submersible Technology (UUST09), Durham NH, USA, August 23-26 2009, in press.

[4] Eichhorn, M., Williams, C.D., Bachmayer, R., de Young, B., "A mission planning system for the AUV "SLOCUM Glider" for the Newfoundland and labrador shelf," OCEANS 2010 IEEE - Sydney, vol. 1, no. 9, pp. 24-27 May 2010.

[5] Eichhorn, M., "Solutions for practice-oriented requirements for optimal path planning for the AUV "SLOCUM Glider"," OCEANS 2010, vol. 1, no. 10, pp. 20-23 Sept. 2010.

[6] Aghababa, M.P., "3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles", Applied Ocean Research, vol. 38, pp. 48-62, October 2012.

[7] Yang, X.S., Karamanoglua, Am., He, X., "Multi-objective Flower Algorithm for Optimization", Procedia Computer Science, vol. 18, pp. 861-868, May 2014.

[8]http://noc.ac.uk/research-at-sea/exploration-at-sea/ocean-floor [Accessed on 20th June 2014]

[9]http://www.bio.gc.ca/science/datadonnees/archive/current_statistics/bottom currents-eng.php [Accessed on 20th June 2014]

[10] www.auv-iitb.org