Energy Optimisation through Path Selection for Underwater Wireless Sensor Networks

Kenechi G. Omeke, Michael S. Mollel, Lei Zhang, Qammer H. Abbasi and Muhammad Ali Imran
James Watt School of Engineering, University of Glasgow, United Kingdom

***Abstract*—This paper explores energy-efficient ways of retrieving data from underwater sensor fields using autonomous underwater vehicles (AUVs). Since AUVs are battery-powered and therefore energy-constrained, their energy consumption is a critical consideration in designing underwater wireless sensor networks. The energy consumed by an AUV depends on the hydrodynamic design, speed, on-board payload and its trajectory. In this paper, we optimise the trajectory taken by the AUV deployed from a floating ship to collect data from every cluster head in an underwater sensor network and return to the ship to offload the data. The trajectory optimisation algorithm models the trajectory selection as a stochastic shortest path problem and uses reinforcement learning to select the minimum cost path, taking into account that banked turns consume more energy than straight movement. We also investigate the impact of AUV speed on its energy consumption. The results show that our algorithm improves AUV energy consumption by up to 50% compared with the Nearest Neighbour algorithm for sparse deployments.**

***Index Terms*—underwater communication, autonomous underwater vehicles, AUV path planning, acoustic communication, underwater wireless sensor networks, AUV trajectory optimisation, q-learning**

I. Introduction

Autonomous underwater vehicles (AUVs) are used in different marine applications for gathering data. It has been established that using AUVs to retrieve data from underwater wireless sensor networks (UWSNs) increases the network lifetime [1]. However, these vehicles have limited power supplies, as they use only the power stored in the on-board batteries. Careful balance must be made between time sensitivity and energy efficiency in underwater sensing applications that employ AUVs for data collection. The time sensitivity of the collected data determines its delay tolerance, which in turn determines how long an AUV trajectory can be, how many network locations it can visit and how many AUVs are needed for a given application. Energy efficiency determines the network lifetime and also has important implications for the delay tolerance. These considerations must be made carefully to improve the performance of UWSNs.

There are three major options [2] to consider in balancing the above considerations: 1) Optimise the path or trajectory that the AUV takes to reach the underwater sensor network location. This depends heavily on the network size. 2) Use clustering or network hierarchy to group the network so that the AUV visits only designated sensor nodes or selected locations to collect data. 3) Use of multiple AUVs to traverse different network paths. Most commercially available propeller-driven AUVs such as the REMUS class of vehicles have endurance of less than 24 hours; hence, energy consumption is a critical consideration. UWSNs become disconnected when there is no path to the sink; hence, improving AUV energy consumption is vital in keeping the network functional and improving the AUV range.

In this paper, we propose an AUV-based protocol for UWSN data collection. The AUV visits clusters in the network to retrieve sensed data and surface to offload it to the sink or data user. The energy consumed by the AUV in this endeavour depends on the trajectory it follows to reach the underwater sensor nodes. The proposed protocol aims to maximise the network lifetime by optimising the AUV energy consumption. To ensure that the overall network energy is improved, we implement a network grouped into clusters. The AUV is deployed from a floating platform and visits only the cluster heads and the network has some level of delay tolerance. We employ reinforcement learning to select the optimal route that minimises the energy consumed by the AUV per mission.

Since the AUV must visit all cluster heads and return to the floating platform, we model the problem as a variant of the travelling salesperson problem (TSP), where an underwater robot must select a path that minimises its travel time or energy expenditure [3]. There are different ways to address this problem, including using genetic algorithm [4], ant colony optimisation [5], [6], using reinforcement learning [7], using nearest neighbour optimisation [8], [9], etc. However, unlike classical shortest path problems, we treat this as a stochastic shortest path problem [10] since the AUV must consider the energy cost of a path in addition to the path length. This is because a shorter path may be more expensive than a longer path if the AUV needs to make many turns for the shorter path. Reinforcement Learning (RL) has been proven suitable for path finding problems with given constraints such as energy consumption or delay [11]. We model the AUV energy consumption to account for straight flights and flights at angled inclinations which consume more energy. The simulation results show that the proposed algorithm outperforms the nearest neighbour algorithm both as the number of clusters increase and at realistic AUV speeds.

The remainder of this paper is structured as follows: we present the system model in Section II and describe the proposed algorithm in Section III. Simulation results are presented in Section IV, followed by discussion of the important results. Section V serves as the conclusion and also presents some

This work was funded by the Petroleum Development Trust Fund of the Federal Repulic of Nigeria.

future research directions.

II. SYSTEM MODEL

We consider the problem of AUV navigation [12] for collecting data from a wireless sensor network deployed to monitor underwater pipeline and other oil and gas production facilities.

An AUV submerged in water experiences a force which can be resolved into a horizontal drag force and a perpendicular lift force [13].

The underwater sensor network comprises N sensor nodes, grouped into clusters with K cluster heads, as shown in figure 1. The sensors are distributed in a 2D grid bounded by (x, y). The AUV operates in three dimensions (x, y, z), where z represents the depth of the network location compared to the water surface. The power consumed by the AUV is a function of the forces acting on it. AUV designers must strive to achieve an effective drag to lift ratio (called the load factor) to minimise energy losses.

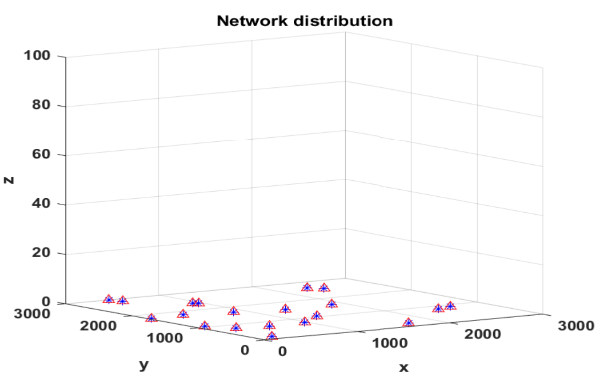

Fig. 1. Network deployment. Only the distribution of cluster heads in 2D plane is shown. The AUV trajectory starts from the node closest to the origin.

The trajectory of an AUV from an initial position, $p = (x_p, y_p)$ to a final position, $q = (x_q, y_q)$ underwater can be described by three motion primitives: straight ahead movement, right turn at a given turning radius (or turning angle) and right turn at a given turning radius (or turning angle) [14]. The AUV thus follows a path, referred to as a Dubins path, consisting only of arc and line segments. The straight ahead movement consumes the least energy to execute. The power consumed in executing turn motions depend on the turning radius, which is in turn affected by the load factor of the AUV.

The electrical power, P consumed by an AUV can be expressed [15] as

$$P = P_{\text{prop}} + H = Dv + H \tag{1}$$

where P_{prop} is the propulsion power given by the product of the drag force, D and the velocity of the AUV, v

The drag force depends on the vehicle design, speed and nature of the surrounding fluid and is given by

$$D = \frac{1}{2}\frac{C_{\text{D}} A \rho v^2}{\eta}. \tag{2}$$

Thus, the propulsion power can then be expressed as

$$P_{\text{prop}} = \frac{1}{2}\frac{C_{\text{D}} A \rho v^3}{\eta}, \tag{3}$$

H is the so-called hotel load, which is a combination of all the power consumed by the AUV subsystems apart from the propulsion system. As shown in equation (3), the power consumed by the propulsion depends on the drag coefficient, C_{D}, area of the AUV, A, density of water, ρ and η, which expresses the efficiency of the propulsion system, defined as the relationship between the mechanical power required for dive, P_m and the electrical power supplied to the motor, P_e.

$$\eta = \frac{P_m}{P_e} \tag{4}$$

The energy consumed by the AUV to travel from cluster head at point P to another cluster head at point Q is the product of the propulsion power and the taken time, t_{pq}. This can be expressed as

$$E = P_{\text{prop}} t_{pq} = P_{\text{prop}} \frac{d_{pq}}{v}, \tag{5}$$

where t_{pq} is can be obtained from the distance between the points, d_{pq} and the velocity of the AUV

The straight line distance between two cluster heads is given by

$$d_{p,i} = \sqrt{\left(x_{p,q} - \hat{x}_{p,q}\right)^2 + \left(y_{p,q} - \hat{y}_{p,q}\right)^2} \tag{6}$$

where $x_{p,q}, y_{p,q}$ represent the x and y coordinates of the first point and $\hat{x}_{p,q}, \hat{y}_{p,q}$ represent the x and y coordinates of the second point.

Banked turns create a centripetal acceleration component, a_c for the AUV, given [16] by

$$a_c = mv^2/R \tag{7}$$

where m is the mass of the AUV and R is the turn radius, which can be expressed in terms of the turn angle, phi. The cetripetal force acting on the AUV must then be added to the drag force in calculating the energy consumption, E^t for banked turns.

$$E_{AUV} = \sum_{i=1}^{N} \left(E^s + E^t\right) \tag{8}$$

where E^s, E^t represent the energy consumed for straight travel and banked turns, respectively, while N is the number of cluster heads in the network.

Equation (3) also shows that the faster the AUV travels through water, the more rapidly its power will be expended. The speed at which the AUV uses the lowest power can be derived from the above equation, but lower speeds affect data timeliness.

Our goal is formulated as: given an AUV position, $p = (x_p, y_p)$ and a final position, $q = (x_q, y_q)$, find the path from p to q that costs the minimum AUV energy. To simplify the analysis, we consider only the motion of the AUV at constant depth. Thus, the cluster head nodes visited by the AUV are all at the same water depth in the network.

III. Proposed Algorithm

We hereby present the RL algorithm for finding the stochastic shortest path that minimises AUV energy consumption. This problem is known to be NP-complete, which is best solved using heuristics. A number of options are available for solving such problems [17] such as minimum spanning trees, incremental insertion methods, K-optimal tours, etc.

To avoid pathological distance functions and keep the problem tractable, we assume that the distance function between a pair of vertices are symmetric. That is, given a pair of vertices x, y, the distance between them $d(x, y) = d(y, x)$.

We select a type of RL called Q-Learning to address the path selection problem defined here. In our implementation, the RL algorithm checks different paths and selects the path that consumes the lowest energy. Unlike other fields of machine learning that rely on data to learn, learning occurs in RL through a series of interactions between an *agent* (e.g. AUV) and an *environment* as shown in figure 2; a *reward* is given for each interaction . In these interactions, the environment reveals itself to the agent as a series of *states*, S_t. The agent takes *actions*, A_t in the environment and obtains a reward, R_t according to how the action performs against a *reward function*. A *policy* is used to decide which action to choose. The agent aims to maximise its reward for each action.

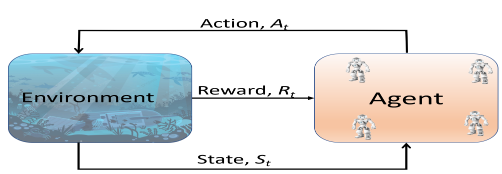

Fig. 2. Reinforcement learning scenario

In our implementation, the environment represents the underwater sensor network comprising sensor nodes and cluster heads whereas the AUV represents the agent that aims to minimise its energy consumption. We define the reward as the inverse of the total energy, E_{AUV}, required by the AUV to travel a complete path. That is, the energy that the AUV requires to travel from the starting cluster head to all the cluster heads in the network and back to the starting node. The state represents which cluster head (coordinates) the AUV is at at a given point in time. A reward is given for each action taken by the AUV, that is, for each move to a new cluster head. Different RL methods exist; however, we adopted Q learning, an empirical, model-free RL technique that achieves early convergence [18]. We take the energy consumption into account in the reward function design by considering the energy expended by the AUV in moving between two points, given by the angle the AUV needs to turn to reach a point.

The network distribution accounts for the underwater signaling technique employed; acoustic waves travel further than radio frequency and light signals in water, and thus, allows more sparse deployments. In this paper, no two cluster heads are allowed within radio range of each other. To reduce the computational demand, we conducted the training offline since the location of the cluster heads are known and do not change over the course of deployment. The solution was implemented in MATLAB (MATLAB R2020a, Mathworks Inc, USA).

TABLE I
Simulation parameters

Parameter	Value	Parameter	Value
m	37 kg	ρ	1027 kg/m^3
v	1.54 m/s	g	9.81 m/s^2
H	10 W	η	0.5
C_D	0.2	A	0.082

IV. Results and Discussion

We evaluated the performance of the proposed algorithm against the nearest neighbour algorithm, which is a common algorithm used in AUV path planning. The network was deployed in an area of 3000 m x 3000 m, with a restriction placed on the minimum separation between two cluster heads (10 m). Monte Carlo simulation was performed for 1000 runs and the results averaged out. The simulation parameters are shown in Table I.

First we evaluated the AUV energy consumption at constant speed as the network size increases. Figure 3 shows the energy consumption of our algorithm versus the nearest neighbour algorithm as the number of clusters grows and for an AUV velocity of v = 1.54 m/s. This is the cruising speed of the majority of the REMUS class of AUVs. The energy consumption increases as the number of clusters to be visited grows because the energy consumed increases the longer the AUV stays in operation. In addition, the number of curved turns that the AUV makes to collect data increases as the number of clusters increase. It is observed that the performance margin between our algorithm and the nearest neighbour algorithm narrows as the number of clusters increase. This is because the network area is fixed and the nearest neighbour algorithm is particularly designed for dense networks, as noted already.

Figure 3 shows that the proposed algorithm outperforms the nearest neighbour algorithm as the number of cluster heads to be visited grows. The performance margin increases significantly for more sparsely distributed networks but narrows for dense deployments. In practice, cluster distribution depends on the target application e.g., ocean sampling, facilities monitoring, etc. More dense deployments have high data redundancy.

In figure 4, we show the performance of our algorithm versus the nearest neighbour algorithm as the speed of the AUV changes. It is seen that at low speeds, the proposed algorithm outperforms the nearest neighbour algorithm and vice versa as speed increases beyond 4.3 m/s. In reality, the best-performing class of commercially available AUVs have a speed of less than 3 m/s. The speed of the AUV affects the timeliness of

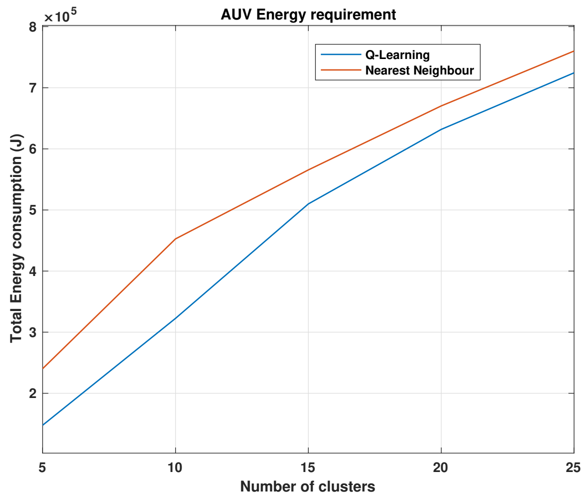

Fig. 3. Energy consumed by the AUV for a selected trajectory as the number of cluster heads increases. We compare our algorithm with the nearest neighbour algorithm.

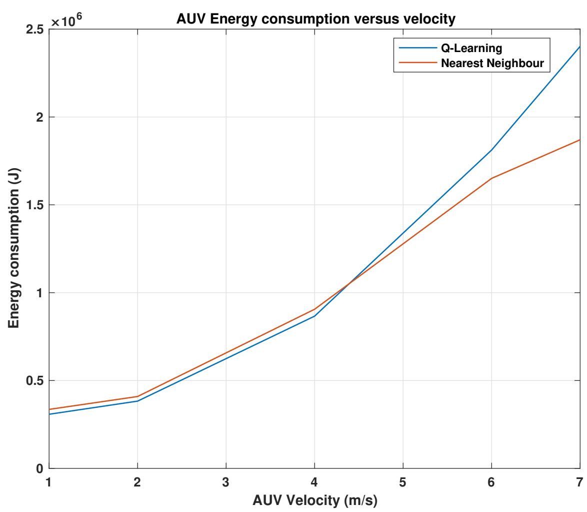

Fig. 4. AUV energy consumption as speed increases.

receiving the sensed data and becomes a major consideration in real-time monitoring and in military applications where large delays might have catastrophic consequences. However, hydrodynamic drag is roughly proportional to the square of the AUV speed [13], which impacts the range and endurance of the AUV. Thus, the AUV must travel slower to cover a longer range.

V. Conclusion and Future Work

In this paper, we presented a Q-learning algorithm for selecting the most energy efficient trajectory for an AUV to conserve energy in an UWSN. The AUV visits every cluster head in network and returns to the origin. Our algorithm models the trajectory selection as a stochastic shortest path problem and considers the cost associated with each path. We presented results that showed that the proposed algorithm outperforms the Nearest Neighbour algorithm. Our future work will consider dynamic networks whereby the cluster locations change over time and the efficiency of data collection by evaluating the fraction of data collected for the time the AUV spends within each cluster.

References

[1] J. Yan, X. Yang, X. Luo, and C. Chen, “Energy-efficient data collection over auv-assisted underwater acoustic sensor network,” *IEEE Systems Journal*, vol. 12, no. 4, pp. 3519–3530, 2018.

[2] G. Han, S. Shen, H. Wang, J. Jiang, and M. Guizani, “Prediction-based delay optimization data collection algorithm for underwater acoustic sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6926–6936, 2019.

[3] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. S. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, “Underwater data collection using robotic sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 899–911, 2012.

[4] Y. A. Sambo, P. V. Klaine, J. P. B. Nadas, and M. A. Imran, “Energy minimization uav trajectory design for delay-tolerant emergency communication,” in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.

[5] M. Dorigo and T. Stützle, “Ant colony optimization: overview and recent advances,” in *Handbook of metaheuristics*. Springer, 2019, pp. 311–351.

[6] J. Wang, J. Cao, R. S. Sherratt, and J. H. Park, “An improved ant colony optimization-based approach with mobile sink for wireless sensor networks,” *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6633–6645, 2018.

[7] U. Challita, W. Saad, and C. Bettstetter, “Deep reinforcement learning for interference-aware path planning of cellular-connected uavs,” in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[8] J. J. Bentley, “Fast algorithms for geometric traveling salesman problems,” *ORSA Journal on computing*, vol. 4, no. 4, pp. 387–411, 1992.

[9] T. S. Alemayehu and J.-H. Kim, “Efficient nearest neighbor heuristic tsp algorithms for reducing data acquisition latency of uav relay wsn,” *Wireless Personal Communications*, vol. 95, no. 3, pp. 3271–3285, 2017.

[10] W. Xia, C. Di, H. Guo, and S. Li, “Reinforcement learning based stochastic shortest path finding in wireless sensor networks,” *IEEE Access*, vol. 7, pp. 157 807–157 817, 2019.

[11] Z. Cao, H. Guo, W. Song, K. Gao, Z. Chen, L. Zhang, and X. Zhang, “Using reinforcement learning to minimize the probability of delay occurrence in transportation,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2424–2436, 2020.

[12] A. Wolek and C. A. Woolsey, “Model-based path planning,” in *Sensing and Control for Autonomous Vehicles*. Springer, 2017, pp. 183–206.

[13] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi, “Seaglider: a long-range autonomous underwater vehicle for oceanographic research,” *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 424–436, 2001.

[14] F. Scibilia, U. Jørgensen, and R. Skjetne, “Auv guidance system for dynamic trajectory generation,” *IFAC Proceedings Volumes*, vol. 45, no. 5, pp. 198 – 203, 2012, 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016306024

[15] J. Bellingham, “Platforms: Autonomous underwater vehicles,” in *Encyclopedia of Ocean Sciences (Second Edition)*, second edition ed., J. H. Steele, Ed. Oxford: Academic Press, 2009, pp. 473 – 484. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978012374473900730X

[16] A. Wolek, “Optimal paths in gliding flight,” Ph.D. dissertation, Virginia Tech, 2015.

[17] S. S. Skiena, *The Algorithm Design Manual*. London: Springer, 2008.

[18] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.