Energy-Constrained Informative Routing for AUVs

Nikolaos Tsiogkas, Valerio De Carolis and David M. Lane
Ocean Systems Laboratory
Heriot Watt University, EH14 4AS Edinburgh, Scotland, UK
Email: nt95, vd63, d.m.lane@hw.ac.uk

***Abstract*—Autonomous underwater vehicles have proven their roles as useful tools for scientific exploration and data gathering. The current state of the art in commercial vehicles involves a statical mission planning phase which does not take into account any energy or time limits. This work proposes the use of a mixed integer quadratic programming solution that maximises the utility of data gathering missions under time or energy constraints. This method is compared against other constrained and unconstrained mixed integer linear programming methods. Results confirm that the method is producing better quality paths in the expense of higher computation time.**

I. Introduction

Advances in the technology of underwater vehicles in the past decade have allowed them to become stable platforms for scientific exploration of the underwater environment. Autonomous underwater vehicles (AUVs) allow scientists to perform data gathering in a faster, cost effective and safer way as no human divers are involved in the process. The current state of mission planning for underwater vehicles involves missions that are usually statically planned on the surface by operators. This is the case for present commercial vehicles, where missions don't take into account the dynamic underwater environment and external disturbances. These can limit the performance of the AUV and thus the scientific outcome. For example, in high current conditions the vehicle may consume more energy and time while executing tasks to complete its mission [1]. There may be cases where a whole mission cannot be completed within the expected time or energy requirements.

This paper focuses on improving the vehicle performance in these cases and introduces energy awareness in the task scheduling stage. The survey requires the visit of various *inspection points* (IPs) inside a constrained area for data acquisition. Moreover, it is assumed the mission is to be performed using a maximum amount of energy. The AUV starts from a user defined point and reaches a specific extraction point to be recovered at the end of the mission. Its goal is to gather as much information as possible without exceeding the given mission constraints.

For this reason the use of a mixed integer quadratic programming optimisation algorithm is proposed called *Correlated Orienteering Problem* (COP) and was first presented in [2]. It maximises the number of inspection points visited by the vehicle, taking into account the correlation of the information in each IP, while respecting the overall energy usage. In the formulation presented in this work extra constraints have been introduced for defining a specific starting and ending point

Fig. 1. Nessie AUV during an inspection mission at The Underwater Centre, Fort William, UK. In environment like this the presence of strong tidal currents affect sensibly the energy usage during long-term autonomous missions.

for the vehicle. To formulate this problem an energy-aware navigation model is used. This estimates the required energy for locomotion tasks among inspection points in presence of external disturbances. Methods for energy estimation for autonomous vehicles are discussed in [3], [4].

To measure the benefits of the proposed method it is compared against two other methods. The first is a mixed integer linear programming flavour of the *Orienteering Problem* (OP), presented in [5]. The second is a variant of a simple *Open Vehicle Routing Problem* (OVRP) instance described in [6]. That formulation also allows for user defined starting and ending points but ignores any extra constraints. A general benefit of MILP and MIQP solutions is that they can be easily extended in multi-AUV scenarios as shown in [7], [8].

The rest of the paper is organised as follows. In section II previous work regarding the solution of energy constrained routing problems is presented. Section III presents the methods used in this paper. In section IV the experimental setup is presented. Section V analyses the results. Finally, in section VI the paper concludes presenting future directions.

II. Previous Work

In the literature many methods exist considering autonomous sensing under constraints using various flavours of the orienteering problem. In [9] a submodular OP instance is used to maximise information gathering. Solution is provided

978-1-4673-9724-7/16/$31.00 ©2016 IEEE

through an approximation algorithm. In [10] a submodular OP is utilised for maximising the additional information gain in a sensor network. Solution for the OP is provided through a recursive greedy algorithm for its speed and quality of solutions. Finding that the submodular OP is slow for real-time use in large graphs, [11] proposes the use of a linear approximation for area coverage using a micro aerial vehicle. In [12] a sampling method is presented that maximises the visits to uncertain areas. The problem is solved by solving a linearised version of a non-linear function as an orienteering problem using genetic algorithms. In [13] a multi-robot sensing system is presented for agriculture applications. It uses an OP instance to maximise visits to potentially misclassified points. The solution is given through a 4-approximation algorithm, that guarantees that it will visit at least a quarter of the points visited by the optimal solution.

A method for energy constrained path planing for AUVs is presented in [14]. In this work the dynamic routing of AUVs taking into consideration the energy of the vehicles is studied. Solution to the problem is provided by Markov decision processes. The work of [15] tries to solve the problem of energy aware planning of AUVs using genetic algorithms. Finally, [16] approaches the same problem by using search strategies.

III. Methods

To achieve the maximised mission performance the use of a mixed integer quadratic programming (MIQP) formulation is proposed. This formulation maximises the inspection points the vehicle visits, considering their correlation, while respecting the energy requirements provided by operators as input during the *planning process*. This formulation is presented in III-A. Moreover in III-B a mixed integer linear programming formulation (MILP) is presented. This formulation only respects energy constraints provided as input. Finally, in III-C a simple OVRP formulation is also introduced. This is used to compare the use of the energy-constrained approaches with a well-known scheduling method.

A. Correlated Orienteering Problem

The correlated orienteering problem tries to maximise the objective function described in (1).

$$\sum_{i \in V}(r_i x_i + \sum_{v_j \in N_i} r_j w_{ij} x_i (x_i - x_j)) \tag{1}$$

The objective function is the sum over the rewards of all the visited vertices. The set V includes all the vertices. The reward for visiting vertex i is described by variable r_i, while x_i is a binary variable denoting that a vertex i is visited in a solution. As mentioned in section I it is assumed that visiting a vertex gives information regarding vertices around it. These vertices form the neighbourhood of vertex i, represented by N_i. The utility of this information is added to the utility obtained by visiting vertex i. It is calculated as the sum of the rewards for visiting each vertex j in the neighbourhood, multiplied by a weight w_{ij} depending of vertices i and j and the quadratic term $x_i(x_i - x_j)$. This quadratic term ensures that extra reward, produced by correlation, is only added for vertices that are not visited in the current solution.

In the version of the COP presented in this work the robot must start and finish in user defined points. This is enforced by constraints (2) and (3).

$$\sum_{i \in V \setminus \{s\}} x_{is} = \sum_{i \in V \setminus \{f\}} x_{fi} = 0 \tag{2}$$

$$\sum_{i \in V \setminus \{s\}} x_{si} = \sum_{i \in V \setminus \{f\}} x_{if} = x_s = x_f = 1 \tag{3}$$

Here the binary variable x_{si} denotes that vertex i is visited right after the starting vertex s and x_{if} shows that the finishing vertex is after vertex i. In general x_{ij} is a binary variable denoting that a path exists from vertex i to vertex j. For all the other vertices constraints (4), (5) and (6) are applied.

$$\sum_{i \in V \setminus \{sf\}} x_{ik} = x_i \leq 1 \qquad \forall k \in V \setminus s, f \tag{4}$$

$$\sum_{i \in V \setminus \{sf\}} x_{ki} = x_i \leq 1 \qquad \forall k \in V \setminus s, f \tag{5}$$

$$\sum_{i \in V \setminus f} x_{ik} = \sum_{i \in V \setminus s} x_{ki} \qquad \forall k \in V \setminus s, f \tag{6}$$

Constraints (4) and (5) allow vertices to be left out from a solution and enforce that a vertex is visited at most once. Constraint (6) ensures that if a vertex is visited it must continue the path to another vertex, unless it is the user defined finish vertex. Constraint (7) bounds the vehicle to use a maximum amount of energy.

$$\sum_{i \in V} x_i c_i + \sum_{j \in V} c_{ij} x_{ij} \leq C_{max} \tag{7}$$

In constraint (7), $x_i c_i$ represents a fixed cost of doing the required operations in vertex i. The cost of travelling from vertex i to the next vertex is added to that, as denoted by $c_{ij} x_{ij}$. It should be noted that the fixed cost of operations should not be added for the user defined start and finish vertices.

$$u_i - uj + 1 \leq (|V| - 1)(1 - x_{ij}) \qquad \forall i, j \in V, i \neq j \tag{8}$$

$$0 \leq u_i \leq |V| \qquad \forall i \in V \tag{9}$$

Finally, constraints (8) and (9) allow only a single path to be generated, preventing smaller disjoint tours.

B. Orienteering Problem

The aim of the OP formulation is to maximise the objective function shown in (10).

$$\sum_{i \in V} \sum_{j \in V} S_i x_{ij} \tag{10}$$

It describes the total amount of inspection points the vehicle will visit, by maximizing the profit acquired in that path. Variable S_i is the profit of visiting inspection point i. The binary variable x_{ij} is denoting the existence of a path between the two points in a proposed solution. V is the set of inspection points, s being the starting point and f being the finishing point.

$$\sum_{i\in V\setminus\{s\}} x_{is} = \sum_{i\in V\setminus\{f\}} x_{fi} = 0 \tag{11}$$

$$\sum_{i\in V\setminus\{s\}} x_{si} = \sum_{i\in V\setminus\{f\}} x_{if} = 1 \tag{12}$$

Constraint (11) ensures that there is no entry path to the starting point and no exit path from the finishing point, making them respectively the first and last point the vehicle visits. While constraint (12) enforces that there formulated path will include the staring and finishing points.

$$\sum_{i\in V\setminus\{sf\}} x_{ik} \leq 1 \qquad \forall k \in V \setminus s, f \tag{13}$$

$$\sum_{i\in V\setminus\{sf\}} x_{ki} \leq 1 \qquad \forall k \in V \setminus s, f \tag{14}$$

$$\sum_{i\in V\setminus f} x_{ik} = \sum_{i\in V\setminus s} x_{ki} \qquad \forall k \in V \setminus s, f \tag{15}$$

Constraints (13), (14) and (15) allow inspection points to be omitted from the solution.

$$\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij} \leq C_{max} \tag{16}$$

$$u_i - uj + 1 \leq (|V| - 1)(1 - x_{ij}) \qquad \forall i, j \in V \tag{17}$$

$$0 \leq u_i \leq |V| \qquad \forall i \in V \tag{18}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{19}$$

Constraint (16) bounds the maximum solution cost to a user specified value C_{max}. Constraint (17) is a sub-tour elimination constraint that forces all the inspection points to be connected in a single path. Finally constraints (18) and (19) bound the values of the problem variables.

C. Open Vehicle Routing Problem

For comparison reasons a simple OVRP with selected starting and exiting points is defined. This solver tries to minimise the cost of travel through all inspection points as defined in the following objective function:

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \tag{20}$$

The (20) describes the total cost of traversing all the inspection points. c_{ij} is the cost of travelling from point i to point j. x_{ij} is a binary variable denoting the existence of a path between the two points in a proposed solution. A is a set of arcs that connect the points. The rest of the formulation is written as:

$$s.t. \sum_{j=1}^{n} x_{js} = 0 \tag{21}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, ..., n \text{ and } j \neq s \tag{22}$$

$$\sum_{j=1}^{n} x_{fj} = 0 \tag{23}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, ..., n \text{ and } i \neq f \tag{24}$$

$$u_i - u_j + n \cdot x_{i,j} \leq n - 1 \qquad \forall (i, j) \in A \text{ and } i \neq j \tag{25}$$

Constraint (21) ensures that there is no entry path to the start point. Constraint (22) enforces a single entry point to each of the other points. Constraint (23), likewise, ensures that there is no exit path from the extraction point, making it the final point of the route. Constraint (24), allows only one exit path from all the other points. Finally, constraint (25), eliminates any sub-tours.

IV. Experimental setup

Performance of the COP method is compared to a simple OP and a reference OVRP implementation using simulations. These are based on the software developed by the Ocean Systems Laboratory. This simulates the dynamics of a Nessie VII AUV [17], the energy consumption and the behaviour of its thrusters. It interfaces with the vehicle software stack using the ROS framework [18]. The use of simulations demonstrates the effects that the above methods have on the *planning* and *execution* process. To solve the MIQP and MILP formulations the Gurobi 6.5 commercial solver [19] is used.

The simulated mission consists of multiple *inspection points*, arranged in a grid, which the vehicle needs to visit. In addition, the vehicle is given specific starting and finishing points. Moreover, the vehicle is given a limited amount of energy to complete its tasks. If a safety threshold is exceeded the vehicle is forced to navigate towards a safe extraction point. The overall vehicle's goal is to maximise the gained utility before reaching the energy threshold.

Evaluation of the methods is done by comparing the accumulated utility and the planning time for each method. Experiments were run on a 2.6GHz Intel Core i5-3320M processor utilising 16GB of RAM.

V. Results

To evaluate the performance of the proposed method several tests were run. It was decided that the cost of movement would be directly proportional to the distance that had to be travelled. Additionally, the robot would have a fixed sensing cost of one unit for each inspection point it visited. For simulating the correlation of information between points, and thus produce a weight for equation (1), the exponential decay function was used as seen in (26) where d represents the distance between the vehicle and the neighbouring vertex. This choice was made in order to simulate the drop in measurement quality as the distance grows. For the purpose of this paper the value of λ was set to 2.

$$N(d) = e^{-\lambda * d} \tag{26}$$

The first point of evaluation is the utility each method manages to collect. For that three different sizes of grid and four different energy budgets were chosen. The energy budget was calculated by finding the optimal path to traverse all the vertices and adding all the required sensing costs. It was then reduced accordingly to form the required budgets. The results of the various tests can be seen in table I.

TABLE I
UTILITY GAINED FOR DIFFERENT GRID SIZES AND BUDGETS

Grid Size	Algorithm	Budget 100%	75%	50%	25%
5x5	OVRP	25.00	18.88	12.39	5.28
	OP	24.88	20.87	14.75	8.01
	OP 5%	24.88	20.87	14.75	8.01
	COP	24.96	22.37	16.47	8.43
	COP 5%	24.61	22.23	16.47	8.43
6x6	OVRP	36.00	27.22	18.75	7.59
	OP	35.97	29.53	22.44	12.52
	OP 5%	35.97	29.53	22.44	12.52
	COP	35.97	32.75	24.37	12.87
	COP 5%	35.92	32.20	24.29	12.85
7x7	OVRP	49.00	34.89	24.73	11.32
	OP	48.65	41.35	31.81	16.87
	OP 5%	48.65	41.35	30.22	16.87
	COP	48.99	45.3	34.03	18.08
	COP 5%	48.65	44.84	34.03	18.03

As expected the COP is performing better than the two other methods when there is some energy constraint. Given that it takes into account the sensing correlation it tries to maximise the utility obtained. An example path for the COP for a 6x6 grid can be seen in figure 2. The OP falls on average 7% behind the COP solution. It still tries to maximise the points visited under energy constraints but totally ignores the correlation and sensor ranges. An example path for the same conditions as the COP can be seen in figure 3. The OVRP performs bad as it doesn't consider at all the energy constrains. The COP performs on average 27% better than the OVRP. An instance of the OVRP solution can be seen in figure 4. It should be noted that in cases where there is enough energy all methods perform equally good. Moreover, for the OP and the COP, two sets of experiments where performed. One for the optimal solution and one for a solution that is 5% away from the optimal. This is done to reduce the runtime of each algorithm and measure the effect it has on the solution quality.

In a path quality analysis, it can be seen that the COP tries to spread the survey in the whole area. This happens because it tries to maximise the total sensed area, and thus utility. On the other hand the OP solution gathers almost the same amount of utility but from only one side of the area to be sensed. This

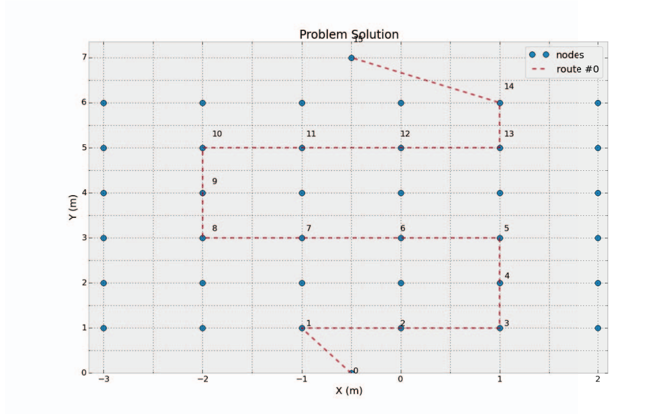

Fig. 2. Correlated orienteering problem solution for a 6x6 grind and energy budget of 30 units. The utility gathered was 20.7.

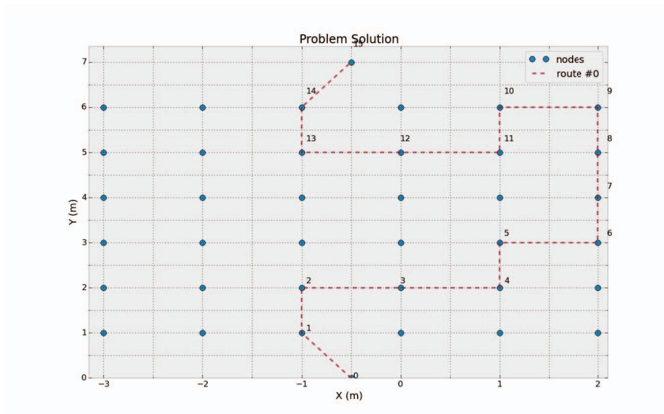

Fig. 3. Orienteering problem solution for a 6x6 grind and energy budget of 30 units. The utility gathered was 19.09.

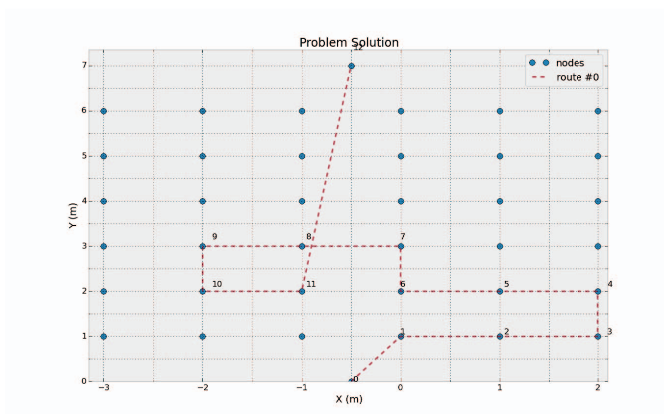

Fig. 4. Open vehicle routing problem solution for a 6x6 grind and energy budget of 30 units. The utility gathered was 14.28.

is considered to have lower quality in sensing missions as it leaves a big part of the area without gathering any information. Finally the OVRP instance finishes the energy budget visiting less than half of the area and gathering the least utility.

The second evaluation point is the path planning time. This is a vital part for an autonomous system as it is required to perform its missions online and as fast as possible. For this reason a maximum of 600 seconds was allowed for planning. The results can be seen in table II.

TABLE II
COMPUTATION TIME(S) FOR DIFFERENT GRID SIZES AND BUDGETS

Grid Size	Algorithm	Budget 100%	75%	50%	25%
5x5	OVRP	0.21	0.21	0.21	0.21
	OP	2.04	2.27	0.81	0.30
	OP 5%	1.30	2.12	0.87	0.31
	COP	2.04	16.31	58.89	1.14
	COP 5%	0.61	8.63	19.01	1.16
6x6	OVRP	4.17	4.17	4.17	4.17
	OP	60.27	7.58	2.88	0.69
	OP 5%	6.14	6.43	2.86	0.73
	COP	45.90	516.71	MAX	40.90
	COP 5%	1.51	17.68	231.41	14.71
7x7	OVRP	9.74	9.74	9.74	9.74
	OP	60.49	13.59	10.96	4.07
	OP 5%	6.38	13.23	7.18	4.54
	COP	MAX	MAX	MAX	MAX
	COP 5%	4.76	4.58	MAX	174.39

It can be seen that as the grid size increases the planning time increases also. The OVRP is in general the fastest method. The OP comes second in computation speed being close to the OVRP. The COP is the most expensive computationally. Especially in larger cases it reaches the maximum time without finding an optimal solution. In these cases the 5% optimality gap shows its use. One can trade off some of the solution quality with a huge improvement in the computation speed. Another interesting aspect is the effect the available energy budget has on the computation time of the COP. As it can be seen the computation time increases as the budget decreases to the regions of around 50% and then decreases again.

VI. CONCLUSION

In this work the problem of task scheduling for AUVs is studied. The use of a MIQP formulation is proposed which takes into account the correlations of the information in various inspection points while having constraints in its time or energy usage. This allows an AUV to maximise the mission performance in cases where end users introduce energy or time constraints. This method is then compared with simpler MILP scheduling methods using simulations. Comparison among the methods reveals the benefits of the MIQP method in solution quality. Regarding the time complexity the MIQP method is the most expensive and it is impractical to use in large cases. As a future direction the use of meta-heuristics for the solution of the MIQP is considered as an option to allow the use in larger problem instances in reasonable amounts of time.

REFERENCES

[1] J. Bellingham and J. Willcox, "Optimizing auv oceanographic surveys," in *Autonomous Underwater Vehicle Technology, 1996. AUV '96., Proceedings of the 1996 Symposium on*, Jun 1996, pp. 391–398.

[2] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 342–349.

[3] J. Willcox, J. Bellingham, Y. Zhang, and A. Baggeroer, "Performance metrics for oceanographic surveys with autonomous underwater vehicles," *Oceanic Engineering, IEEE Journal of*, vol. 26, no. 4, pp. 711–725, Oct 2001.

[4] V. D. Carolis, D. Lane, and K. Brown, "Low-cost energy measurement and estimation for autonomous underwater vehicle," in *Proceedings of IEEE-MTS Oceans'14, Taipei, Taiwan*, 2014.

[5] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[6] F. Li, B. Golden, and E. Wasil, "The open vehicle routing problem: Algorithms, large-scale test problems, and computational results," *Computers & Operations Research*, vol. 34, no. 10, pp. 2918–2930, 2007.

[7] I. Kara and T. Bektas, "Integer linear programming formulations of multiple salesman problems and its variations," *European Journal of Operational Research*, vol. 174, no. 3, pp. 1449–1458, 2006.

[8] N. Tsiogkas, G. Frost, N. Monni, and D. Lane, "Facilitating multi-auv collaboration for marine archaeology," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–4.

[9] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, pp. 707–755, 2009.

[10] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Robotics and automation (icra), 2010 IEEE international conference on*. IEEE, 2010, pp. 4791–4796.

[11] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1071–1078.

[12] S. Frolov, B. Garau, and J. Bellingham, "Can we do better than the grid survey: Optimal synoptic surveys in presence of variable uncertainty and decorrelation scales," *Journal of Geophysical Research: Oceans*, vol. 119, no. 8, pp. 5071–5090, 2014.

[13] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5321–5326.

[14] P. A. Forero, S. K. Lapic, C. Wakayama, and M. Zorzi, "Rollout algorithms for data storage-and energy-aware data retrieval using autonomous underwater vehicles," in *Proceedings of the International Conference on Underwater Networks & Systems*. ACM, 2014, p. 22.

[15] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *Oceanic Engineering, IEEE Journal of*, vol. 29, no. 2, pp. 418–429, April 2004.

[16] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an a* approach," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 194–198.

[17] N. Valeyrie, F. Maurelli, P Patron, J. Cartwright, B. Davis, Y. Petillot, "Nessie v turbo: a new hover and power slide capable torpedo shaped auv for survey, inspection and intervention," in *AUVSI North America 2010 Conference*, 2010.

[18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[19] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com