

Motion Planning for Autonomous Underwater Vehicle Supervision

Graeme Best^{1,2} and Stuart Anstee¹

¹Littoral Unmanned Systems Group, Defence Science and Technology Organisation
Sydney, Australia

stuart.anstee@dsto.defence.gov.au

²Australian Centre for Field Robotics, University of Sydney
Sydney, Australia
g.best@acfr.usyd.edu.au

Abstract

Autonomous Underwater Vehicles (AUV) play an important role in surveying undersea environments. It is often essential for AUV operators to be able to supervise the progress of their vehicles during missions, to enable informed decisions in real-time. In this paper we consider supervision of an AUV executing a preplanned mission from a mobile surface vessel equipped with an acoustic communications system. We consider the surface vessel to be supervising effectively if it is stationary and located within a fixed maximum range from the AUV. When the supervising vessel is not stationary, we assume it is transiting to the next supervision point. We seek trajectory plans that maximise the proportion of time the supervising vessel is able to supervise the AUV effectively. In this paper we propose two solutions to this trajectory planning problem; a greedy planner, which instructs the supervising vessel to move when the AUV moves beyond its effective range but restricts the vessel to points along the path of the AUV; and a global optimiser based on a genetic algorithm that relaxes this limitation. Sea-trial and simulation results demonstrate the effectiveness of our proposed solutions.

1 Introduction

Autonomous Underwater Vehicles (AUV's) are tools for surveying and understanding undersea environments. They are used in a wide variety of applications such as surveying pipeline routes, monitoring marine life and locating mines on the sea floor. At present, most AUV's have *navigational autonomy*, that is, the capability to autonomously perform preplanned missions defined by sequences of waypoints with associated speed and depth instructions [Bovio *et al.*, 2006; Fang and Anstee, 2010]. While AUV's navigate autonomously, it is currently prudent to supervise their progress when possible, so that

information they transmit can be used for intelligent and timely decision making [German *et al.*, 2012; Hagen *et al.*, 2008]. Responses vary from adjusting priorities to performing time-critical tasks, such as prevention of a collision between a surface vessel and an AUV. A supervising platform may also augment AUV systems; e.g., by supplying positional information [Fallon *et al.*, 2010; Bahr *et al.*, 2009; Walls and Eustice, 2014].

The research direction in this paper originated from the experiences of AUV operators responsible for supervising AUV's operating in shallow waters. Motivating factors have included singular events: AUV's have aborted missions, run aground or become trapped underwater for various reasons; also, near-collisions between surface vessels and AUV's have occurred. The value of information transmitted by AUV's has also increased as they have gained capabilities for on-board data reduction, yielding environmental information and the locations and characteristics of objects of interest that have immediate value to operators.

Almost all AUV operators supervise their AUV's as standard procedure. Multiple options exist when an AUV is on the surface, but for the bulk of its mission it is underwater and only accessible via an acoustic communications system (ACOMMS), which is likely to be effective over short ranges – typically hundreds to a few thousands of metres. Operators may supervise AUV's working in compact geographical areas with fixed 'gateway' buoys incorporating ACOMMS and surface or satellite communications systems, but more typically they supervise AUV missions from powered surface vessels.

This paper proposes the automation of a path-planning and scheduling task – controlling the position of the supervising vessel as a function of time – that is currently the responsibility of a human operator. The motivation is the observation that AUV operators are often poorly suited to the task of maintaining a vessel within a range from the AUV that will allow them to respond effectively to a singular event or to important information generated by the AUV.

We assume that a single motor-driven surface vessel is the supervising platform for a single AUV. The vessel is capable of transiting much faster than the AUV. It may be manned or unmanned. It is equipped with an ACOMMS that becomes fully effective when its acoustic transducer is lowered a few metres below the keel of the vessel and the motor is switched off or idling. The ACOMMS is unavailable when the vessel is transiting at speed. We model the vessel as having two states: stationary, when its position is fixed and the ACOMMS is functional, and moving, when the ACOMMS is not functional and the vessel is either preparing to move, moving from one supervising waypoint to another at fixed speed, or preparing to become stationary.

We assume that the trajectory of the AUV is known in advance. The objective of the planning system is to minimise risk, expressed as the period that the AUV is unsupervised. The output of the planning system is a time-scheduled sequence of supervising waypoints. Upon arrival at each supervising waypoint, the supervising platform enters the stationary state until the schedule instructs it to transit to the next supervising waypoint. This differs from a traditional mission plan, in which the schedule is implicit in the speed instructions attached to each leg.

A variety of work in the literature discusses algorithms for marine vehicles tracking moving targets [Encarnaçao and Pascoal, 2001; Bibuli *et al.*, 2009; 2012; Lapierre *et al.*, 2004], and similarly for aerial vehicles [Cetin and Zagli, 2012; Grancharova *et al.*, 2012] and ground vehicles [Ghabcheloo *et al.*, 2007; Michael *et al.*, 2009; Xu *et al.*, 2013]. Most authors employ sliding-horizon closed-loop non-linear control theory and related techniques, which aim to minimise the distance between the vehicle and a (possibly virtual) target. A related problem is the path planning of mobile recharging stations for persistent tasks, which can be formulated as a generalised travelling salesman problem [Mathew *et al.*, 2013], but requires predefined position/time regions for recharging. We have similar aims in this paper to the above mentioned work, however there are three key differences that characterise our problem specification.

1. **Predetermined mission plan** - The AUV mission is known in advance and can be accurately predicted. This allows us to search for solutions that are optimal over the duration of the mission.
2. **Inexact goal position** - The goal is to maximise the amount of time the supervising vessel is within an effective supervising range rather than aiming to be as close as possible to a goal point at all times.
3. **Alternating behaviour** - The model requires alternation between stationary and fixed-speed states rather than continuously varied speed.



Figure 1: The Remus 100 Autonomous Underwater Vehicle.

These differences motivate us to solve our trajectory planning problem for the surface supervising vessel using optimisation techniques rather than control theory. In this paper we propose two approaches. The first, greedy algorithm selects waypoints along the planned mission path. The planner instructs the supervising vessel to enter the transiting state as soon as the AUV moves beyond supervising range and selects the next waypoint such that the AUV comes within supervising range when the supervising vehicle arrives at the waypoint. This approach is effective, but restrictive. The second approach removes the restriction that the waypoints be on the planned path and adds time as an optimisation variable. We use a genetic algorithm (GA) based iterative search algorithm to select points from the resulting space and solve the resulting global optimisation problem. We conclude this paper by demonstrating the effectiveness of our trajectory planner through simulations and sea trials in which a manned vessel supervised an AUV, the REMUS 100 shown in Figure 1, during its missions.

The remainder of the paper is organised as follows. Section 2 specifies the problem definition for the trajectory planner. Section 3 presents our proposed solutions and Section 4 discusses our implementations. Section 5 describes results from a sea-trial and simulations. Section 6 summarises our work and proposes future research directions.

2 Problem Specification

The system includes two vehicles: an AUV and a supervising surface vessel, which may or may not be manned. The AUV follows a predefined mission plan. The supervising vessel aims to stay within effective range of the AUV so that the progress of the mission can be monitored.

2.1 AUV Mission

The AUV is instructed to survey an undersea environment by following a predetermined mission plan. A mission plan consists of a sequence of waypoints (denoted $X = x_1, x_2, \dots, x_N$, where each x_i represents a geographical position) that the AUV must visit in order, maintaining a specified speed over ground v_i along the path to x_i . The AUV can follow the mission plan autonomously

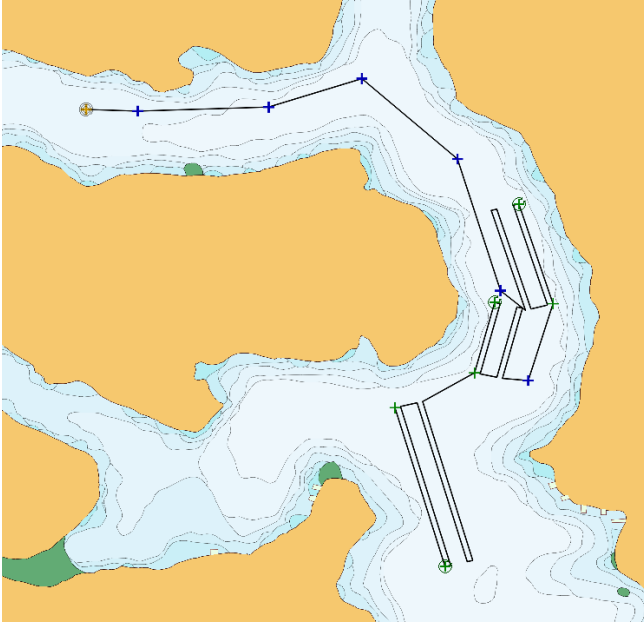


Figure 2: An example AUV mission plan in Middle Harbour, Sydney.

from start to end without requiring any external inputs. During the mission, the AUV will follow an approximately linear trajectory between the waypoints. Note that other possible mission parameters, particularly depth, are not considered for our purposes. An example mission plan, which will be referred to throughout the paper, is shown in Figure 2.

Let the expected duration of the mission be T . In this work, we express the planned path of the AUV as a sequence of locations $X^* = x_1^*, x_2^*, \dots, x_n^*$ where $n = 1 + T/\Delta t$, constructed from the mission plan X by piecewise linear interpolation at regular time intervals $\Delta t = 10$ s. Let the associated times be $T^* = t_1^*, t_2^*, \dots, t_n^*$, where $t_1^* = 0$ and $t_n^* = T$.

2.2 Supervising Vessel Trajectory Plan

The goal of the supervising vessel is to effectively supervise the AUV. To achieve this, it must be able to communicate with the AUV, which requires it to be stationary, and it must be within an effective response range r of the AUV. The effective response range may be the maximum range of the communications system, or a smaller range that allows the supervising vessel to respond to a hazardous situation or provide positional information. The key constraint is that the supervising vessel cannot communicate with the AUV while it is moving under power. Therefore the supervising vessel's planned motion must be defined as an ordered sequence of waypoints (denoted $Y = y_1, y_2, \dots, y_M$) at which it will remain stationary, and associated arrival and departure times (denoted $T^a = t_1^a, t_2^a, \dots, t_M^a$ and $T^d = t_1^d, t_2^d, \dots, t_M^d$,

respectively) that satisfy the constraints:

$$t_i^a < t_i^d < t_{i+1}^a, \forall i \in \{1, 2, \dots, M-1\} \quad (1)$$

The supervising vessel is in a stationary state at waypoint y_i during the time interval $[t_i^a, t_i^d]$ and it moves from waypoint y_i to y_{i+1} during the time interval $[t_i^d, t_{i+1}^a]$. We assume that the supervising vessel moves at a fixed speed over ground v_s between the waypoints. The travel time between two waypoints includes an additional constant time penalty T_{pen} to allow time for the supervising vessel to prepare for departure and setup after arrival at each waypoint. In this work, we assume $T_{\text{pen}} = 30$ s based on experience. From these restrictions we have:

$$t_{i+1}^a = t_i^d + \frac{\text{DIST}(y_i, y_{i+1})}{v_s} + T_{\text{pen}}, \quad (2)$$

where $\text{DIST}(y_i, y_{i+1})$ is the Euclidean distance between successive waypoints.

We assume that the AUV is launched and recovered by the supervising vessel, yielding the constraints:

$$\begin{cases} y_1 = x_1 \\ t_1^a = 0 \\ y_M = x_n \\ t_M^a \leq T. \end{cases} \quad (3)$$

For convenience, a single trajectory plan solution which encompasses the waypoints Y and the associated arrival and departure times T^a, T^d is denoted $S = (Y, T^a, T^d)$.

2.3 Fitness Function

We introduce a fitness function $\text{FITNESS}(X^*, S, r)$ for evaluating the effectiveness of a trajectory plan S given an interpolated AUV mission X^* and an effective response range r . The fitness is calculated as the fraction of the mission duration T during which the supervising vessel is stationary and within effective response range of the AUV. For the purposes of this study we choose $r = 200$ m. The fitness function is evaluated as shown in Algorithm 1.

2.4 Cylinder Plot Representation

The problem specification as described above can be visualised graphically as shown in Figure 3. The blue line shows the path of the AUV (X^*) with a local geographical scale on the horizontal axes and time as the vertical axis. The red cylinders show the effective response range (r) around the supervising vessel waypoints (Y). The bottom of each cylinder represents the arrival time (t_i^a) and the top represents the departure time (t_i^d). The green stars are points along the AUV path during which

Algorithm 1 Fitness Function

```

1: function FITNESS( $X^*, S = (Y, T^a, T^d), r$ )
2:    $count \leftarrow 0$  ▷ num. in range points
3:    $state \leftarrow \text{STATIONARY}$ 
4:    $y \leftarrow y_1$  ▷ supervisor position
5:    $t_{event} \leftarrow t_1^d$ 
6:   for  $t \leftarrow [0 : \Delta t : T]$  do ▷ iterate AUV plan
7:      $x \leftarrow X^*(t)$  ▷ AUV position at time  $t$ 
8:     if  $state = \text{STATIONARY}$  then
9:       if  $\text{DIST}(x, y) < r$  then ▷ in range
10:         $count = count + 1$ 
11:       if  $t > t_{event}$  then
12:         $state \leftarrow \text{MOVING}$ 
13:         $t_{event} \leftarrow T^a.next$ 
14:     else if  $state = \text{MOVING}$  then
15:       if  $t > t_{event}$  then ▷ arrived at  $y$ 
16:         $state \leftarrow \text{STATIONARY}$ 
17:         $t_{event} \leftarrow T^d.next$ 
18:         $y \leftarrow Y.next$ 
19:   return  $\frac{count}{n}$ 
    
```

the AUV is not effectively supervised. The cost function ($\text{FITNESS}(X^*, S, r)$) is 1 minus the ratio of the number of unsupervised locations to the total mission time, which is to be maximised by the proposed algorithms.

Now we have defined a way of encoding a trajectory plan $S = (Y, T^a, T^d)$ for the surface supervisor vehicle and a way of evaluating the effectiveness of a solution given an interpolated AUV mission X^* and an effective response range r . The following section, and the ultimate goal of this paper, describes our algorithms to determine a trajectory plan S that maximises $\text{FITNESS}(X^*, S, r)$.

3 Methodology

We propose two algorithms as solutions to the trajectory planning problem: a greedy planner and a globally optimising planner. The greedy planner is deterministic and chooses supervising waypoints along the path of the AUV. It is based on the heuristic that the supervising vessel must depart from a waypoint when the AUV moves outside its effective response range and move to the next location along the path at which the AUV will have just come into range when the supervising vessel arrives. The globally optimising planner is implemented using a genetic algorithm. It removes the restriction that supervising waypoints must lie along the path of the AUV and is therefore able to generate superior trajectory plans.

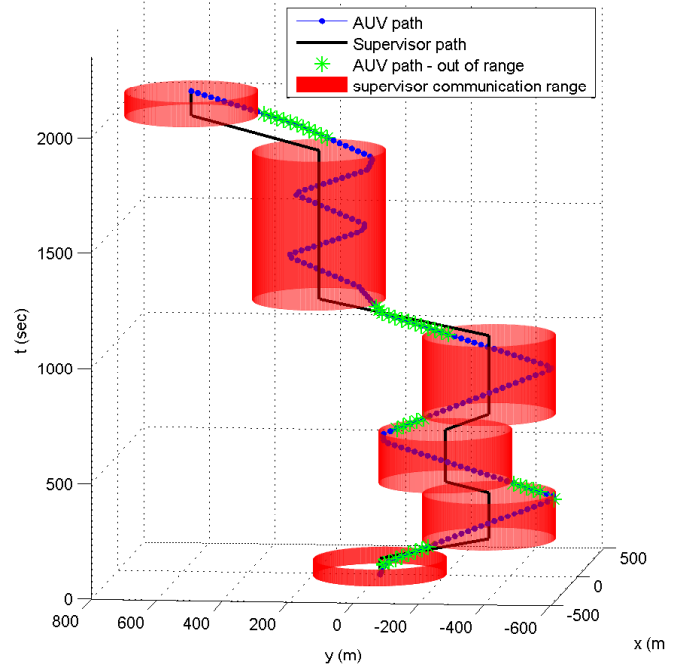


Figure 3: An example AUV mission plan (the southern part of Figure 2) and a solution for the surface vehicle plan. Note that the vertical axis shows time, with the mission starting at the bottom and ending at the top of the plot. The red cylinders represent the effective range of the surface vehicle while stationary during the mission.

3.1 Greedy Trajectory Planner

The greedy path planner is fast, efficient and simple to implement. The algorithm is an iterative process that alternates between two states: calculating the time to depart the current waypoint, and calculating the best location for the next waypoint. The algorithm is fast because we constrain the waypoints for the supervising vessel to lie on the path of the AUV (i.e. $y_i \in X^*, \forall i$). Despite this constraint, our tests have shown that the algorithm produces acceptable results.

Algorithm 2 shows pseudocode for the greedy planner. It follows a similar structure to $\text{FITNESS}(X^*, S, r)$, as it iterates through the times T^* associated with the AUV locations X^* while alternating between *stationary* and *moving* states. The supervising vessel starts moving when the AUV goes out of range ($\text{DIST}(x_{\text{current}}, y_{\text{current}}) > r$).

The next waypoint location y_{i+1} is chosen such that the AUV is just coming back into range at the time the surface vessel arrives there (t_{i+1}^a calculated in (2)). It is found by iteratively trialling each successive AUV location x_j^* as the next supervising waypoint y_i^* , following the NEXT Y function in Algorithm 2. The algorithm continues iterating until it finds a waypoint that is out of

Algorithm 2 Greedy Trajectory Planner

```

1: function GREEDY( $X^*, r$ )
2:    $state \leftarrow \text{STATIONARY}$ 
3:    $y_1 \leftarrow x_1^*$  ▷ Initialise  $Y$ 
4:    $y \leftarrow y_1$  ▷ current supervisor position
5:    $t_1^a \leftarrow 0$ 
6:   for  $t \leftarrow [0 : \Delta t : T]$  do ▷ iterate AUV plan
7:      $x \leftarrow X^*(t)$  ▷ AUV position at time  $t$ 
8:     if  $state = \text{STATIONARY}$  then
9:       if  $\text{DIST}(x, y) > r$  then ▷ out of range
10:         $state \leftarrow \text{MOVING}$ 
11:         $T^d.append(t)$ 
12:         $[y, t^a] \leftarrow \text{NEXTY}(y, t)$  ▷ next waypoint
13:         $Y.append(y)$ 
14:         $T^a.append(t^a)$ 
15:       else if  $state = \text{MOVING}$  then
16:         if  $t > t^a$  then ▷ arrived at  $y$ 
17:            $state \leftarrow \text{STATIONARY}$ 
18:       return  $S = (Y, T^a, T^d)$ 
19: function NEXTY( $y_{current}, t_{current}$ )
20:   for  $t \leftarrow [t_{current} : \Delta t : T]$  do ▷ iterate AUV plan
21:      $y \leftarrow X^*(t)$  ▷ let  $y$  be a future  $X^*$ 
22:      $d \leftarrow \text{DIST}(y_{current}, y)$  ▷ distance to  $y$ 
23:      $t^a \leftarrow \text{Eqn. (2) (using } d)$  ▷ arrival time at  $y$ 
24:     for  $i \leftarrow [t^a : \Delta t : t]$  do ▷ iterate AUV plan
25:       if  $\text{DIST}(X^*(i), y) > r$  then
26:         ▷ Out of range during  $[t^a, t]$ 
27:         return  $[y_{prev}, t_{prev}^a]$  ▷ next waypoint
28:        $t_{prev}^a \leftarrow t^a$ 
29:        $y_{prev} \leftarrow y$ 
30:   ▷ reached end of mission
31:    $t^a \leftarrow T$ 
32:    $y \leftarrow X^*(T)$ 
33:   return  $[y, t^a]$  ▷ next waypoint
    
```

range any time from t^a when the supervisor arrives until t when the AUV arrives. Once this is confirmed, the algorithm selects the waypoint from the previous iteration as the new y , since the AUV will just be coming back in to range at this point. It is necessary to check between t^a to t (lines 24 to 27) rather than just at t^a , since it is possible that t^a decreases between successive iterations of the outer loop. This could potentially result in the AUV going out of range again sometime between t^a and t , despite having been in-range during this time interval in previous iterations of the outer loop.

3.2 Global Trajectory Planner

Our second trajectory planner removes the limitation that supervising waypoints lie along the planned path of the AUV. This limitation becomes important when the AUV's path is curved. Consider the extreme case when

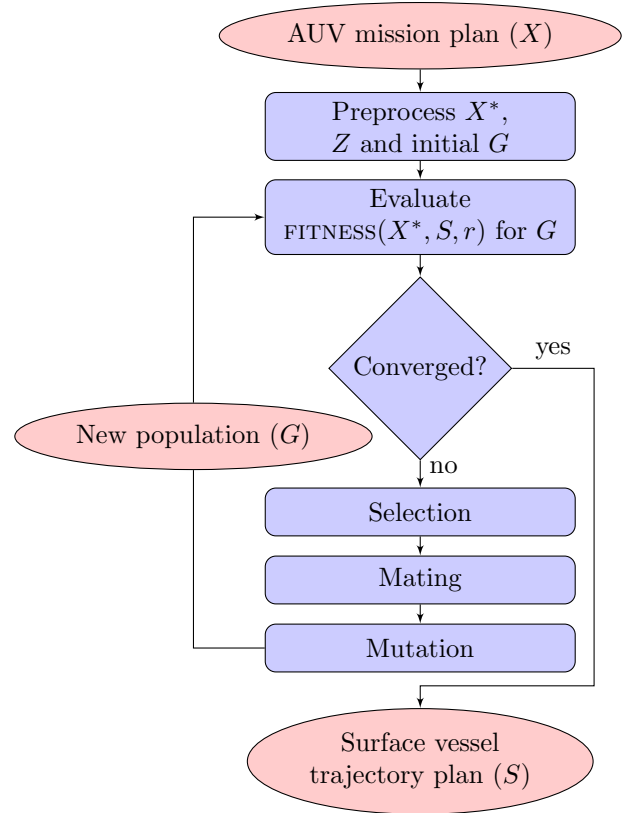


Figure 4: Overview of the proposed Genetic Algorithm trajectory planner.

the AUV follows a circular path with $\frac{r}{2} < \text{radius} < r$. In this case, the optimal solution would require the supervising vessel to remain stationary in the centre of the circle, which is not on the path of the AUV, for most of the mission. The difficulty with allowing the supervising vessel to move off the path of the AUV is that it greatly increases the size of the search space, making an exhaustive search infeasible.

For this reason, we formulate our solution using a genetic algorithm (GA) optimisation. GAs are a widely used optimisation technique that is inspired by biological evolutionary processes whereby *fitter* variations of a species are more likely to stay in a population and pass on favourable characteristics to following generations.

Our proposed algorithm follows the standard GA structure which iterates through a selection, mating and mutation cycle as depicted in Figure 4. The effectiveness of a GA implementation for a specific problem is strongly dependent on the design of the individual parts of the algorithm. The novelty of our approach is our method of encoding the candidate solutions to reduce the search dimensionality while encompassing the search space defined by the problem specification.

Candidate Solution Encoding

The relationship between the candidate solutions and their encoding as chromosomes is the foundation of the algorithm; the encoding must be simple and specific enough for population manipulation and complex enough to encompass the search space efficiently.

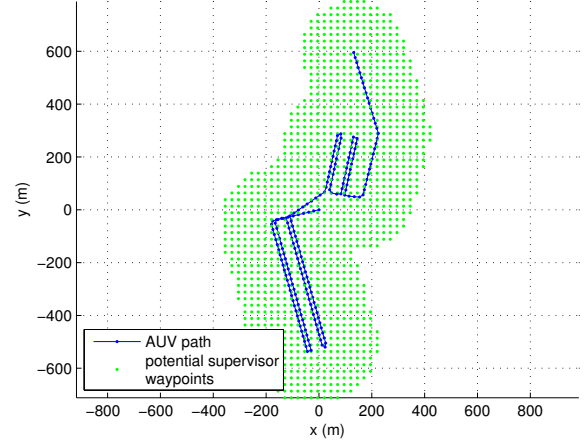
In this work, the search space from which supervising waypoints will be selected is limited to a set of discrete geographical locations P that are no further distant from the predicted path of the AUV than the effective range; that is, the set of points lying on a rectangular grid that are within range of at least one point x_i^* in the interpolated path of the AUV X^* , as shown in Figure 5a. In this work, we use a square grid spacing of 25 m, because this approximates the accuracy with which the supervising vessel can be controlled. A smaller spacing may result in better solutions but will also decrease the convergence rate of the search.

The path X^* of the AUV may approach within effective range of a point $p_k \in P$ multiple times during a mission. We consider each contiguous section of AUV locations $x_i^*, x_{i+1}^*, x_{i+2}^* \dots$ falling within range of p_k as a separate event at time τ_l which we assign as the mean of the times $t_i^*, t_{i+1}^*, t_{i+2}^*, \dots$ associated with the AUV's locations, resulting in a three-dimensional entity $z_l = (p_k, \tau_l)$. Let the search space be $Z = z_1, z_2, z_3, \dots$, as depicted in Figure 5b. Denote two additional special cases necessary to fulfil the constraints (3): the point $z_{\text{start}} = (x_1, 0)$ at which the AUV is launched and the point $z_{\text{end}} = (x_N, T)$ at which the AUV must be recovered.

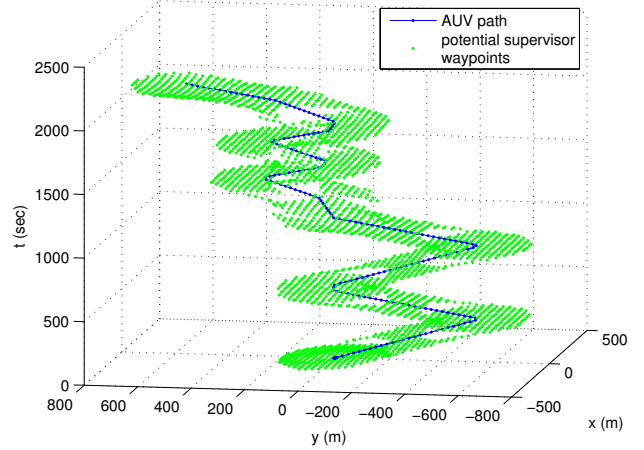
We construct candidate solutions, which are trajectory plans for the supervising vessel, from subsets of Z . Let an individual chromosome be $\hat{g} = g_1, g_2, \dots, g_M$, where $g_1 = z_{\text{start}}$ and $g_M = z_{\text{end}}$ in all cases, but the number of intermediate elements is arbitrary. Let $g_2, g_3, \dots, g_{M-1} = z_a, z_b, z_c, \dots \in Z$ where the elements, say $z_\eta = (p_\eta, \tau_\eta)$ are arranged in ascending order of τ_η . We interpret the p_η as supervising waypoints and require that the supervising vessel be located at p_η at time τ_η . We now require arrival and departure times t_η^a and t_η^d satisfying

$$t_\eta^a \leq \tau_\eta \leq t_\eta^d \quad (4)$$

and the constraint (2) relating successive departure and arrival times t_η^d and $t_{\eta+1}^a$. The travel time $\Delta t_{\eta, \eta+1}$ between p_η and $p_{\eta+1}$ is known from (2), because the vessel speed v_s and movement penalty T_{pen} are fixed. Therefore, as depicted in Figure 6, we vary the departure time t_η^d between τ_η and $\tau_{\eta+1} - \Delta t_{\eta, \eta+1}$ and select the departure time that minimises the number of times the AUV will fall at an unsupervised location between them. It is possible that $\tau_{\eta+1} - \tau_\eta < \Delta t_{\eta, \eta+1}$, in which case the candidate solution is infeasible and must be discarded. If every leg of the candidate solution is feasible, then



(a) Discrete possible waypoint locations P around the planned path of the AUV X^* .



(b) Each possible waypoint in P is allocated to a time τ corresponding to the middle of the interval

Figure 5: Discretising the waypoint search space around the AUV mission. This is using the same AUV plan as Figure 3.

the sequence of arrival times, waypoints and departure times constitutes a valid trajectory and we add it to the population.

Initial Population

The algorithm manages a fixed population G of candidate solutions \hat{g} . We chose this size empirically as 80, as this gave a good balance between convergence rate and solution quality. The initial population is made up of a combination of 10 seeded solutions and 70 randomly generated solutions. The seeded solutions are formed by running the greedy planner 10 times with r varying between 50 % and 200 % of its initial value. These solutions must be adjusted slightly such that each path

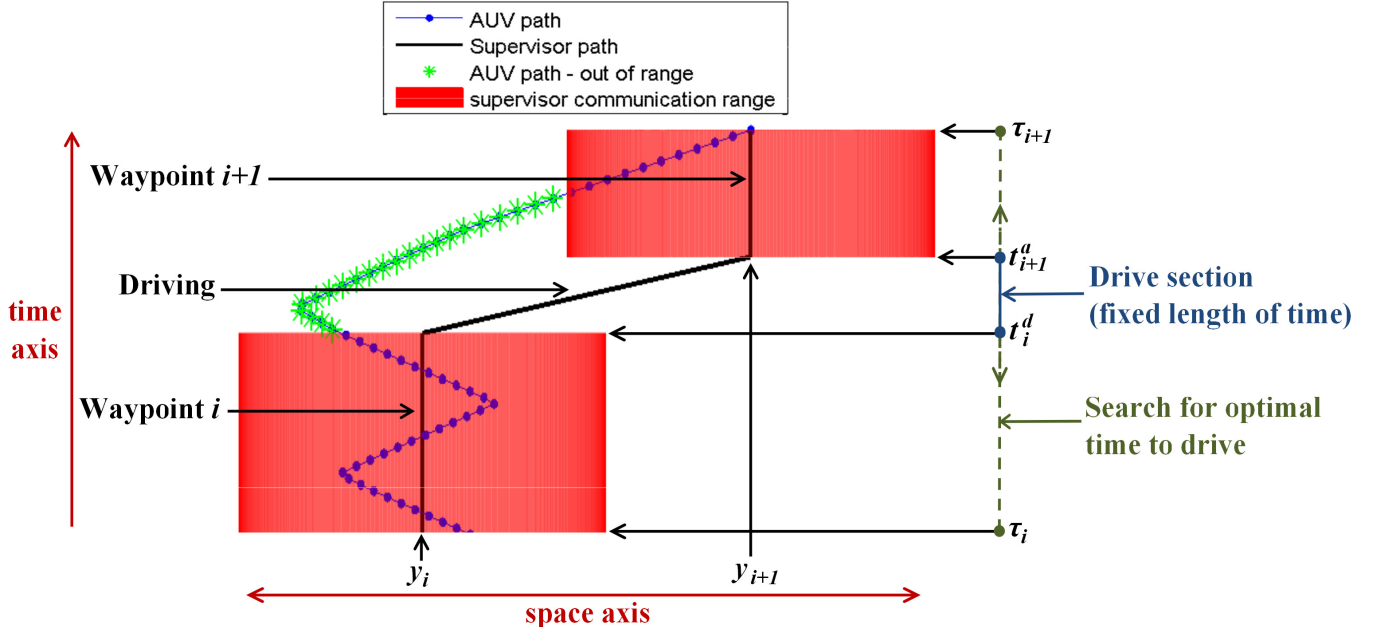


Figure 6: Sliding-window search for the best time to drive between two consecutive waypoints. The search finds t_i^d and t_{i+1}^a which minimises the number of locations at which the AUV is unsupervised (green stars) between τ_i and τ_{i+1} .

waypoint is mapped to the nearest waypoint in the set Z . The seeded solutions may be omitted, but including them typically increases the convergence rate. The remainder of the initial population is formed by choosing random sets of waypoints $\hat{g} = z_a, z_b, z_c, \dots$ from Z . Any invalid \hat{g} are discarded and recomputed. The number of waypoints in each \hat{g} is randomly sampled from a uniform distribution with a range obtained from the seeded solutions.

Selection

At the start of each iteration, the fitness of each \hat{g} in G is evaluated by decoding it to a trajectory S following the steps described in Section 3.2, then evaluating $\text{FITNESS}(X, S, r)$. We follow the widely used *elitist selection* policy, retaining the 20 \hat{g} with greatest fitness from the previous G (G_{prev}) into the new G for the next iteration. The remaining 60 \hat{g} in G are formed by mating and mutation. This is achieved by first sorting G_{prev} in ascending order of their fitness values $\hat{g}_1, \hat{g}_2, \hat{g}_3, \dots$, and giving each \hat{g}_i a relative selection weighting of $w_i = i+40$. Pairs of parents \hat{g}_a, \hat{g}_b are selected from G_{prev} , such that \hat{g}_j is w_j/w_k times more likely to be selected than \hat{g}_k .

Every pair selected from the non-elite group has 50% chance of undergoing mating or mutation. Both operations result in two children, which will replace their parents in the population. The children of the mating step subsequently have a 50% chance of being included in the group selected for mutation. A mating or mutation step

is repeated if a child is infeasible.

Mating

Mating is performed by first randomly selecting a time $0 < t_c < T$. The two children are then formed by combining the two parents as follows:

$$\hat{g}_A = \{z_i \in \hat{g}_a : \tau_i < t_c\} \cup \{z_i \in \hat{g}_b : \tau_i > t_c\} \quad (5)$$

$$\hat{g}_B = \{z_i \in \hat{g}_b : \tau_i < t_c\} \cup \{z_i \in \hat{g}_a : \tau_i > t_c\} \quad (6)$$

Mutation

All children \hat{g}_i that were not formed through mating go through the mutation process. The children that were formed through mating have a 50% chance of mutation. The elitist 20 do not get mutated. For every child \hat{g}_i that is chosen to be mutated, one of the following four equally-likely actions is performed:

- A **Substitute globally** - Select one of the waypoints $z \in \hat{g}_i$ and replace it with a randomly selected waypoint $z' \in Z$.
- B **Substitute locally** - Select one of the waypoints $z \in \hat{g}_i$ and replace it with a randomly selected waypoint $z' \in Z$ which is a short distance (less than 100m) away from z . This action was included in addition to A to encourage small adjustments of highly-performing solutions.
- C **Remove** - Select one of the waypoints $z \in \hat{g}_i$ and remove it from the set. The length M of \hat{g}_i is decreased by 1.

D Insert - Add a randomly selected new waypoint $z' \in Z$ to \hat{g}_i . The length M of \hat{g}_i is increased by 1.

Actions C and D in combination with the selection process allows the algorithm to naturally find the most suitable number of waypoints M .

Stopping Criteria

There are various possible choices for stopping criteria of a GA. A simple solution we follow is to stop if the fitness of the best \hat{g} in G has not improved over the previous 100 iterations.

3.3 Optimality

The greedy planner is not optimal by design. The degree to which the solutions found by the global planner are suboptimal, within the constraints of the discretised search space, is unclear. In our approach we introduce a new constraint (4), to reduce the search dimensions and provide a convenient way of encoding the candidate solutions. Introducing this constraint affects the search space in two ways; (i) the supervising vessel must be at p_η at time τ_η , which therefore defines the bounds of the sliding window search; and, (ii) enforces a strict ordering of z_a, z_b, z_c, \dots , and therefore waypoint reordering need not be considered by the search. We argue that in most cases this constraint does not affect the optimality of the search, despite reducing the search space. However there may be some cases where it may be suboptimal; e.g., when the sequences of AUV positions that are associated with different τ_i overlap in time and the trajectory plan associated with g , which ignores such overlaps, may not be optimal for the sequence. We intend to explore other approaches to investigate this further.

4 Implementation

Our trajectory planning algorithms exist in two forms: we employ offline Matlab implementations of both algorithms for experimentation and testing through simulations. We employ an online Python implementation of the greedy planner integrated into a map-centric graphical AUV monitoring tool to supervise AUV missions at sea. The online tool receives and interprets acoustic communications from the AUV, which are input to the greedy planner. The planner progressively updates its predictions to account for the progress of the AUV through its mission. It gives audible warnings to inform the operators when they should move their vessel, and visible instructions through the map and a command window showing the next supervision point.

The online implementation represents a minor modification to the algorithms presented in previous sections, because the actual trajectory of the AUV differs from the prediction. We account for deviations in the online implementation by adjusting the clock when a significant

Table 1: Percentage of time where the supervisor vehicle is within the effective range of the AUV. The number of stationary waypoints M is listed in parenthesis. The middle two columns are simulation results, while the last column is from a sea trial while using the greedy planner.

	Greedy	Global	Sea trial
Middle Harbour	72.6 (10)	78.3 (8)	-
Jervis Bay	73.3 (10)	78.5 (7)	71.2 (10)
<i>circular</i>	65.0 (10)	95.4 (3)	-
<i>linear</i>	47.4 (7)	51.4 (8)	-

deviation between the predicted and actual AUV path is measured. For example, if the AUV arrives at a point 10 seconds later than predicted, then the clock driving the supervision process is delayed by the same amount, such that all remaining supervising vehicle waypoints are effectively delayed by 10 seconds. With our equipment we found the deviations to be small and this method was appropriate. If larger deviations were experienced, then it may be necessary to re-execute the full planning algorithm, using the vessel and AUV locations as starting positions and removing the completed part of the AUV mission.

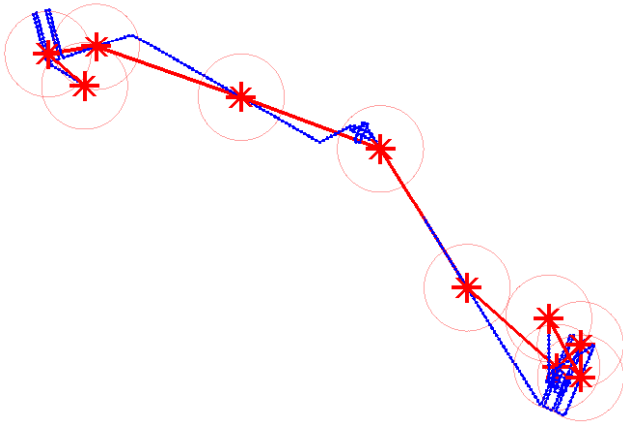
At present there is no online implementation of the global planner, because it runs much slower than the greedy algorithm. For example, the present implementation of the global planner required approximately 200 iterations and between 10 and 30s to converge to a solution for a mission plan that the greedy planner solved in approximately 10ms on the same computer. The global planner is naturally variable because it is a randomized algorithm, and its running time is strongly dependent on parameters such as population size and grid spacing.

5 Evaluation

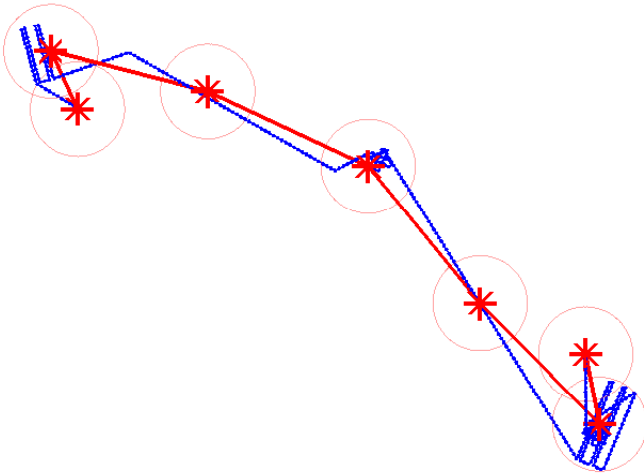
We evaluate our proposed algorithms using by several simulated and one real mission. In all cases we used an AUV speed $v_i = 2\text{ m/s } \forall i$ and a vessel speed $v_s = 5\text{ m/s}$. Table 1 compares the evaluated fitness function expressed as a percentage and the required number of stops for each mission and algorithm.

We conducted our initial sea trials in Middle Harbour, Sydney, using the mission plan depicted in Figure 2. The purpose of the trials was to evaluate an early version of the GUI rather than the algorithm, so we did not collect comparable statistics. Offline simulations based on the mission plan showed the global planner to be slightly superior to the greedy planner.

We conducted our second sea trials in Jervis Bay, New South Wales. The supervising vessel followed the directions given by the online implementation of the greedy



(a) Greedy planner solution.



(b) Global optimisation planner solution.

Figure 7: Solutions for the Jarvis Bay mission. AUV path in blue and supervisor path in red with effective range circles at waypoints. Approximately 70 min duration and 8 km travel distance.

planner. As shown in Figure 7, the 70 min mission consisted of long transits separating three small sections of dense coverage. The greedy planner solution selected 10 supervising waypoints. The operators found that the supervisor received 71.2 % of the messages sent by the AUV over the course of the mission. This is in good agreement with the simulator’s prediction of 73.3 % effective supervision and suggests the validity of the approach. Later simulations showed that the global planner required only 7 supervising waypoints and was marginally superior in terms of supervision effectiveness. An online, continually updating tool based on a global planner might therefore be preferred by the vehicle operators, if it were implemented.

The cases listed in Table 1 include two semi-realistic missions and two extreme cases: a circular mission and a linear mission. The extreme cases highlight the differences between the algorithms. The *circular* mission required the AUV to follow the circumference of a circle with radius 180 m for 5 complete loops. The global planner optimised the supervision trajectory to a single location at the centre of the circle, within range of the AUV’s entire path. The supervising vessel moved directly from the starting position to the centre of the circle and from there to the end position, remaining within range of the AUV for almost the entire mission. In contrast, the greedy planner was restricted to locations along the AUV’s path, so the AUV continually moved out of range, forcing the supervising vessel to move. The greedy planner selected 10 supervising waypoints and achieved significantly lower supervising efficiency due to the larger travel time.

The *linear* mission comprised a single 5000 m long line segment. In contrast to the *circular* mission, the global planner performed only marginally better than the greedy planner, selecting locations along the path of the AUV. The global planner performed better than the greedy planner because it was able to take better advantage of the speed of the vessel relative to the speed of the AUV. Both algorithms performed worse on the linear mission than the others because the mission plan minimised the opportunities to remain stationary.

6 Conclusions

In this paper we have presented two trajectory planners for a surface vessel that supervises an AUV during its mission. The supervising vessel is only effective as a supervisor while it is stationary and the AUV is within a fixed minimum range. Our first trajectory planner is greedy and deterministic, and therefore has a much faster computation time. Our second planner, which is based on optimisation of a global cost function using a genetic algorithm, removes the limitation that the supervisor’s waypoints must lie along the AUV’s mission plan. This improves the planner’s performance, but at the cost of a slower computation time. Simulations and sea trial results show that the greedy algorithm is arguably sufficient for the application we considered; however, the global approach is more general and more suited to future development.

In particular, it would be worthwhile to perform planning with an uncertain AUV trajectory, to allow for larger deviations from the initial mission and the possible consequences caused by mission failures. Additionally, we would like to extend this work to allow for supervising multiple AUV’s simultaneously. These problems would require modifications to the presented algorithms, such as formulating new cost functions, which

would be more applicable to the global planner. Additionally, in many situations it is desirable that the supervising be performed by an unmanned surface vehicle; which presents new practical challenges such as how to have the planner effectively cooperate with a low-level autonomous controller, relay valuable information back to humans, and autonomous deployment and retrieval.

Acknowledgements

This work was performed while Graeme Best held a Summer Vacation Scholarship at DSTO in 2013-2014. The authors wish to acknowledge the assistance of Unmanned Systems and Autonomy Group members during this period. We also thank Robert Fitch from ACFR for insightful comments.

References

- [Bahr *et al.*, 2009] A. Bahr, J. J. Leonard, and M. F. Fallon. Cooperative localization for autonomous underwater vehicles. *International Journal of Robotics Research*, 28(6):714–728, 2009.
- [Bibuli *et al.*, 2009] M. Bibuli, G. Bruzzone, M. Caccia, and L. Lapierre. Path-following algorithms and experiments for an unmanned surface vehicle. *Journal of Field Robotics*, 26(8):669–688, 2009.
- [Bibuli *et al.*, 2012] M. Bibuli, M. Caccia, L. Lapierre, and G. Bruzzone. Guidance of unmanned surface vehicles: Experiments in vehicle following. *Robotics Automation Magazine, IEEE*, 19(3):92–102, 2012.
- [Bovio *et al.*, 2006] E. Bovio, D. Cecchi, and F. Baralli. Autonomous underwater vehicles for scientific and naval operations. *Annual Reviews in Control*, 30(2):117–130, 2006.
- [Cetin and Zagli, 2012] O. Cetin and I. Zagli. Continuous airborne communication relay approach using unmanned aerial vehicles. *Journal of Intelligent and Robotic Systems*, 65(1-4):549–562, 2012.
- [Encarnação and Pascoal, 2001] P. Encarnação and A. Pascoal. Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft. In *Proc. IEEE Decision and Control*, volume 1, pages 964–969, 2001.
- [Fallon *et al.*, 2010] M. F. Fallon, G. Papadopoulos, J. J. Leonard, and N. M. Patrikalakis. Cooperative auv navigation using a single maneuvering surface craft. *International Journal of Robotics Research*, 2010.
- [Fang and Anstee, 2010] C. Fang and S. Anstee. Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle. In *IEEE OCEANS*, pages 1–8, 2010.
- [German *et al.*, 2012] C. R. German, M. V. Jakuba, J. C. Kinsey, J. Partan, S. Suman, A. Belani, and D. R. Yoerger. A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of autonomous surface vehicles and autonomous underwater vehicles. In *Autonomous Underwater Vehicles (AUV), IEEE/OES*, pages 1–7, 2012.
- [Ghabcheloo *et al.*, 2007] R. Ghabcheloo, A. Pascoal, C. Silvestre, and I. Kaminer. Non-linear co-ordinated path following control of multiple wheeled robots with bidirectional communication constraints. *Int. Journal of Adaptive Control and Signal Processing*, 21(2-3):133–157, 2007.
- [Grancharova *et al.*, 2012] A. Grancharova, E. I. Grøtli, and T. A. Johansen. Distributed mpc-based path planning for uavs under radio communication path loss constraints. In *Proc. IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control*, pages 254–259, 2012.
- [Hagen *et al.*, 2008] P. E. Hagen, N. Størkersen, B.-E. Marthinsen, G. Sten, and K. Vestgård. Rapid environmental assessment with autonomous underwater vehicles examples from hugin operations. *Journal of Marine Systems*, 69(1):137–145, 2008.
- [Lapierre *et al.*, 2004] L. Lapierre, D. Soetanto, and A. Pascoal. Coordinated motion control of marine robots. In *Proc. IFAC Conference on Manoeuvring and Control of Marine Craft*, 2004.
- [Mathew *et al.*, 2013] N. Mathew, S. L. Smith, and S. L. Waslander. A graph-based approach to multi-robot rendezvous for recharging in persistent tasks. In *Proc. IEEE ICRA*, pages 3497–3502, 2013.
- [Michael *et al.*, 2009] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas. Maintaining connectivity in mobile robot networks. In *Experimental Robotics*, pages 117–126. Springer, 2009.
- [Walls and Eustice, 2014] J. M. Walls and R. M. Eustice. An origin state method for communication constrained cooperative localization with robustness to packet loss. *International Journal of Robotics Research*, 2014.
- [Xu *et al.*, 2013] Z. Xu, R. Fitch, J. Underwood, and S. Sukkarieh. Decentralized coordinated tracking with mixed discrete-continuous decisions. *Journal of Field Robotics*, 30(5):717–740, 2013.
- [Fang and Anstee, 2010] C. Fang and S. Anstee. Coverage path planning for harbour seabed surveys using an