# Sonar Path Planning Using Reinforcement Learning

Joshua J. Wakefield, Adam Neal, Stewart Haslinger, Jason F. Ralph
Department of Electrical Engineering & Electronics, University of Liverpool, UK
Email: {sgjwakef, sganeal, sgh, jfralph}@liverpool.ac.uk

***Abstract*—Passive towed array sonar systems play an essential role in submarine situational awareness. However, the detection and localisation of sound-emitting objects is a more challenging task compared to their active counterparts due to a lack of immediate range information. By making manoeuvres and changing the bearings at multiple positions, a passive sonar can localise and track the source of the sound. Reinforcement learning is the process of learning an optimal strategy to guide an agent's actions towards optimising its cumulative reward for a given task. This work evaluates an agent's ability to control a passive towed array sonar system for optimal source localisation and tracking in the underwater environment, using collision avoidance as a practical example application.**

***Index Terms*—Tracking, localisation, reinforcement learning, particle filters, passive sonar**

## I. Introduction

Passive towed array sonar systems play a critical role in enhancing submarine situational awareness in the complex underwater environment. Unlike active sonar, these systems rely on detecting the acoustic signatures of other objects without emitting detectable signals, thus maintaining a stealthy posture while surveilling the local area. This passive approach presents unique challenges, notably the accurate detection and localisation of sound-emitting objects due to the absence of range information. Historically, operators would manually interpret sonar data to determine the location of a sound source, a process now increasingly augmented by sophisticated mathematical and computational methods.

The challenge in bearings-only target motion analysis (TMA) lies in accurately determining a target's motion from directional measurements alone, without range information. This problem requires satisfying several key observability criteria including manoeuvring the observer to introduce bearing variations and ensuring diverse geometric configurations [1]. Addressing these challenges motivates the development of advanced techniques to enhance target tracking accuracy and reliability in complex, real-world environments.

A significant challenge in maritime surveillance involves the use of towed linear arrays. A common issue with these arrays is directional ambiguity, where the direction of the signal received by the sensor array cannot be distinguished between port side and starboard side. However, this ambiguity can be resolved through Sequential Monte Carlo (SMC) methods, such as particle filters due to their ability to handle multimodal distributions [2]. Resolving such ambiguities is essential for meeting the observability criteria.

This work lays the groundwork for scenarios where a submarine needs to surface in areas congested with surface vessels. Surface ships tend to be noisy, emitting characteristic narrowband tones and broadband noise. Approaching the source of the sound improves the signal-to-noise ratio (SNR), yielding clearer signal reception and more precise localisation. Closer proximity facilitates greater changes in bearing angle, further enhancing localisation efficiency, but raises the risk of collisions. Balancing between accurately localising a sound source and minimising the risk of a collision with surface vessels is a key consideration in operational strategies, and is fundamental to the safety of both the submarine and the other vessel.

Reinforcement learning (RL) [3] has the potential to aid this sensor management task by learning an optimal strategy that proposes a manoeuvre at each time step for maximum reward. RL can systematically learn and execute strategies that find a balance between the objectives and constraints, aiming to maximise the overall effectiveness of the sonar system. This method enables adaptive decision-making based on the dynamic conditions of the underwater environment, potentially enhancing both the safety and efficacy of submarine operations.

Our research builds on the sensor path planning work by Hoffmann et al. [4], addressing the challenges of passive sonar situational awareness in oceanic environments. These challenges include acoustic scattering, absorption, and dynamic obstacles, which complicate sonar operation and navigation. Our contributions are the application of RL algorithms and SMC methods to improve support safety and surveillance in submarine situational awareness, adapting these advanced techniques to the specific needs of underwater environments.

In this paper, we first review related work in Section II. We then discuss the particle filter tracking algorithm utilised in this work in Section III. Section IV describes our reinforcement learning approach, including the construction of the environment and the design of the reward function. In Section V, we provide a full description of the scenario. Section VI presents the results of our study, followed by a summary and conclusions in Section VII.

## II. Related Work

In recent years, the application of efficient tracking algorithms to sonar systems has increased. An example of this is shown in [5] where an end-to-end signal processing chain and particle filtering based track-before-detect (TBD) algorithm is proposed for multi-target tracking using a passive sonar array.

A comprehensive overview of various sensor management tasks is given in [6], where the effects of operational con-

straints on task objectives are discussed. The work conducted in [7] attempts to find optimal trajectories in a multi-sensor setup with bearing-only measurements. Here, the optimisation is based on minimising the state estimation error. In the context of collision and conflict avoidance, [8] implement an efficient subset simulation method to estimate the probability of conflict for an air-to-air scenario. In this work we deal with a slightly different problem, which is to use reinforcement learning to propose routes that are optimised over several potentially conflicting factors, including the need to localise another vehicle, whilst maintaining a safe distance, and maintaining the efficacy of the sensor.

The applications of RL have grown considerably, and its use in autonomous navigation and collision avoidance systems presents many conceivable benefits. Of particular interest is the capability for solutions to be discovered that may have previously been overlooked. In [4], RL is utilised to localise a target using bearing-only radar measurements with a maximum likelihood estimator, demonstrating the potential for an RL agent to navigate and optimise sensor paths. In addition, an evaluation is provided comparing a myopic policy, which only considers immediate rewards, to a policy learned by the RL model. It is shown that the learned policy outperforms that of the myopic policy. In the context of underwater navigation, [9] propose a sophisticated adaptive path planning and collision avoidance solution using RL for autonomous underwater vehicles (AUV). A multi-objective optimisation approach to sonobuoy placement is proposed in [10] where the aim is to minimise sensor placement time and localisation uncertainty.

## III. Target Tracking & Localisation

We approach the target tracking problem from a Bayesian perspective where we recursively estimate the belief of the system as new measurements become available.

### A. Dynamic State-Space Model

At each time step $k \in \mathbb{N}$, the target is assumed to move according to a state evolution model defined by

$$\mathbf{x}_k^t = \mathbf{f}_k(\mathbf{x}_{k-1}^t, \mathbf{q}_{k-1}), \tag{1}$$

where $\mathbf{f}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_q} \to \mathbb{R}_{n_x}$ is a possibly nonlinear function, $\mathbf{x}_k^t$ is the state of the target, and $\mathbf{q}_{k-1}$ is an independent and identically distributed (i.i.d.) process noise sequence representing the uncertainty in the evolution.

Generally, it can be assumed that the target exhibits a certain degree of motion consistency, i.e., they typically maintain a steady velocity. This is commonly represented by modelling the acceleration as white noise. A more realistic representation of the target dynamics is the Integrated Ornstein-Uhlenbeck (IOU) model [11] that introduces a damping, or drag, factor $\zeta$ leading to an exponential decay in velocity over time. The IOU model accounts for a decrease in velocity which is more realistic in scenarios where the target is moving through a medium that exerts resistance such as water.

We represent the state of the target with a four-dimensional vector that includes both position and velocity components in two-dimensional space:

$$\mathbf{x}_k^t = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T, \tag{2}$$

where $x$ and $y$ are the horizontal and vertical Cartesian coordinates, respectively, and $\dot{x}$ and $\dot{y}$ are the corresponding velocities.

We then define the evolution of the system using a kinematic equation of motion:

$$\mathbf{x}_k^t = F_{IOU}\mathbf{x}_{k-1}^t + \mathbf{q}_{k-1}, \quad \mathbf{q}_{k-1} \sim \mathcal{N}(0_{4\times1}, Q_{IOU}), \tag{3}$$

where

$$F_{IOU} = \begin{bmatrix} 1 & 0 & F_1 & 0 \\ 0 & 1 & 0 & F_2 \\ 0 & 0 & F_2 & 0 \\ 0 & 0 & 0 & F_2 \end{bmatrix}, \tag{4}$$

$$Q_{IOU} = \begin{bmatrix} Q_1 & 0 & Q_2 & 0 \\ 0 & Q_1 & 0 & Q_2 \\ Q_2 & 0 & Q_3 & 0 \\ 0 & Q_2 & 0 & Q_3 \end{bmatrix} \sigma^2, \tag{5}$$

and

$$\begin{aligned} F_1 &= \zeta^{-1}(1 - e^{-\zeta\Delta t}), \\ F_2 &= e^{-\zeta\Delta t}, \end{aligned} \tag{6}$$

$$\begin{aligned} Q_1 &= \zeta^{-2}[\Delta t - 2\zeta^{-1}(1 - e^{-\zeta\Delta t}) + \frac{1}{2}\zeta^{-1}(1 - e^{-2\zeta\Delta t})], \\ Q_2 &= \zeta^{-2}[(1 - e^{-\zeta\Delta t}) - \frac{1}{2}(1 - e^{-2\zeta\Delta t})], \\ Q_3 &= \frac{1}{2}\zeta^{-1}(1 - e^{-2\zeta\Delta t}). \end{aligned} \tag{7}$$

### B. Measurement Model

We define a function that describes the transformation between the sensor measurement and the state-space, in addition to capturing the uncertainties inherent in sensor measurements:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k^t, \mathbf{n}_k), \tag{8}$$

where $\mathbf{h}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \to \mathbb{R}^{n_z}$ is a possibly nonlinear mapping function, and $\mathbf{n}_k$ is an i.i.d. measurement noise sequence representing the uncertainty in the measurement.

In the case of passive sonar, measurements are generated in terms of azimuth only which requires a transformation to relate it to the Cartesian coordinate system. More suitably, the estimated state is converted to a bearing. Thus, the measurement model is defined as:

$$\mathbf{h}_k(\mathbf{x}_k^t, \mathbf{x}_k^s, n_k) = \theta_k = \arctan 2(y_k^t - y_k^s, x_k^t - x_k^s) + n_k, \tag{9}$$

where $\mathbf{x}_k^s$ is the state of the sensor array's origin sensor at time $k$, and $x_k^s$ and $y_k^s$ are the horizontal and vertical Cartesian coordinates of the sensor array

### C. Particle Filter

The optimal Bayesian solution to the target tracking problem cannot be determined analytically [12]. Under certain constraints, one can find an optimal solution, e.g., in the case of a linear system and Gaussian noise, the Kalman filter [13] can be utilised. For passive towed array sonar the measurements are non-Gaussian. The optimal solution must then be approximated by methods such as the extended Kalman filter, unscented Kalman filter, and particle filter. In this paper, we only consider the particle filter.

The particle filter is an SMC method designed to approximate the posterior probability density function (pdf) using a set of weighted samples (or particles) [2]. Each particle represents a candidate solution to the state of the system at a given time, and the set of all particles approximates the probability distribution of the system's state.

The weighting associated with each particle represents the particle's likelihood of being the true state of the system. Initially, particles are distributed according to prior knowledge, or uniformly across the state-space if prior knowledge is limited.

The Bayesian framework for a particle filter can be summarised as three steps: predict, update, and resample.

*1) Predict*

The evolution of the particles as they are propagated forward in time is defined by the dynamics of the system. Thus, (3) can be written as:

$$\mathbf{x}_k^{(i)} = F_{IOU}\mathbf{x}_{k-1}^{(i)} + \mathbf{q}_{k-1}^{(i)}, \quad \text{for } i = 1, \ldots, N_s. \tag{10}$$

*2) Update*

Once a new measurement becomes available, the weights are updated based on the likelihood of the new measurement given the particle's state:

$$w_k^{(i)} = e^{\frac{d(\mathbf{h}(\mathbf{x}_k^{(i)}), \mathbf{z}_k)^2}{2\sigma^2}} \cdot w_{k-1}^{(i)}, \quad \text{for } i = 1, \ldots, N_s, \tag{11}$$

where $w_k^{(i)}$ is the weight of the $i^{th}$ particle, and $d(\mathbf{x}_k^{(i)}, \mathbf{z}_k)$ represents the function that calculates the difference between the predicted state of the particle $\mathbf{x}_k^{(i)}$ and the new measurement $\mathbf{z}_k$.

Given the measurements are bearing only, $d(\cdot)$ computes the angular difference between the measured bearing and the bearing from the sensor array to the particle. To maintain a probability distribution, the weights are then normalised.

*3) Resample*

After several iterations of weight updates, a small subset of particles accumulate higher weights compared to the rest. This leads to a situation where a large proportion of particles have negligible weights and thus contribute little to the estimation of the posterior distribution. We use systematic resampling [14] to minimise the degeneracy of the particles and add larger noise with probability $p = 0.1$ to avoid sample impoverishment.

### D. State Estimation

For a unimodal distribution of particles, the estimated state of the target can be computed as the weighted average of all particles:

$$\bar{\mathbf{x}}_k = \frac{1}{N_s}\sum_{i=1}^{N_s} \mathbf{x}_k^{(i)} w_k^{(i)}. \tag{12}$$

However, for a multimodal distribution of particles where the distribution represents multiple hypotheses about the state of the system, computing the state estimation requires more sophisticated approaches than calculating the weighted average of all particles. This is because the weighted average may not necessarily represent any actual hypothesis accurately, especially in cases of ambiguous or conflicting information that leads to distinctly separated clusters of particles.

To resolve this problem, we utilise the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [15] clustering algorithm to identify the distinct groups of particles that represent different hypotheses. HDBSCAN is well-suited to this task due to its ability to find clusters of varying densities, its flexibility in handling noise, and its robustness to hyperparameter selection.

After clustering, the weighted average of each cluster is then calculated for each cluster independently. Initially, we considered only the likelihood of each cluster as the metric for selecting the estimated state but found this to be unstable. By factoring in the uncertainty of the clusters and computing the harmonic mean between the uncertainty and likelihood we were able to improve the stability of the state estimation selection process.

The uncertainty in the estimated state of particles is quantified by the determinant of their covariance matrix, denoted:

$$\hat{\sigma}_{\bar{\mathbf{x}}_k} = |\Sigma_k|, \tag{13}$$

where $\Sigma_k$ represents the covariance matrix of the particle distribution.

## IV. Reinforcement Learning

This section outlines our application of RL to enable a vessel to autonomously navigate through a series of discrete actions, aiming for trajectories with optimal source localisation and collision avoidance.

RL is a branch of machine learning (ML), characterised by an agent's capacity to learn optimal decision-making through interactions with its environment, aiming to maximise a user-defined cumulative expected reward. This learning paradigm is defined by three distinct aspects:

- **Agent**: The decision-making entity.
- **Environment**: The context or space within which the agent operates.
- **Policy**: The strategy adopted by the agent, denoted as a function $\pi : S \rightarrow A$, mapping states to actions.

RL can be formally described through Markov Decision Processes (MDPs) which provide the foundation for modelling decision-making in scenarios where outcomes are stochastic. An MDP is represented by the tuple $(S, A, P, R, \gamma)$, where:

- $\mathcal{S}$ is the set of all possible states,
- $\mathcal{A}$ denotes the action space,
- $\mathcal{R}$ is the reward function, with $\mathcal{R}(s, a)$ specifying the reward received when taking action $a$ in state $s$,
- $\mathcal{P}$ represents the transition probabilities, where $\mathcal{P}(s'|s, a)$ defines the probability of transitioning to state $s'$ from state $s$ upon taking action $a$,
- $\gamma$, the discount factor, a weighting/importance applied to future rewards.

Our choice of RL algorithm is Proximal Policy Optimisation (PPO) — a state-of-the-art policy gradient method that optimises policy performance with respect to the cumulative reward using gradient descent. The primary motivation for using PPO is its sample complexity and the use of a clipping function which improves performance over other objective functions [16].

### A. Environment Setup

*1) Observation Space*

The observation space $\mathcal{O}$ captures essential spatial and orientation information of both the agent and target. The specific inputs can be seen in Table I. By providing the agent with these inputs at each time step, the algorithm can process the current state of the environment and its relationship to the target. This processed information then informs the selection of the next action to take. All inputs are normalised in the range [0, 1] except for the agent's heading, $\varphi_k$, which is normalised in the range [-1, 1].

TABLE I
OBSERVATION SPACE

| Input | Description |
|---|---|
| $x_k$ | Horizontal position of the agent |
| $y_k$ | Vertical position of the agent |
| $\varphi_k$ | Heading of the agent |
| $\bar{x}_k$ | Estimated horizontal position of the target |
| $\bar{y}_k$ | Estimated vertical position of the target |
| $\hat{\sigma}_{\bar{\mathbf{x}}_k}$ | Uncertainty |
| $d_k^s$ | Distance between the agent and estimated source position |
| $d_k^l$ | Distance between the agent and surveillance area boundaries |

*2) Action Space*

The action space, denoted as $\mathcal{A}$, consists of three discrete actions that enable the agent to navigate its environment by adjusting its heading. These actions modify the agent's heading at a fixed angular rate of $\Delta\varphi = 0.1^\circ$ per second and correspond to a counter-clockwise change in heading, no change in heading, and a clockwise change in heading, respectively. They are mathematically defined as:

$$\mathcal{A} = \{-\Delta\varphi,\ 0,\ \Delta\varphi\}. \tag{14}$$

At each time step $k$, the agent selects an action $a$ from the set $\mathcal{A}$ of valid actions.

*3) Transition Dynamics*

The dynamics of the agent's movement are governed by its chosen action. Specifically, the agent's heading at time step $k$, denoted $\varphi_k$, is updated based on its previous heading and the applied action:

$$\varphi_k = \varphi_{k-1} + \Delta\varphi_k, \tag{15}$$

This updated heading then informs the agent's position at the next time step:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{v}_k \Delta t, \tag{16}$$

where the velocity vector, $\mathbf{v}_k$, is represented as:

$$\mathbf{v}_k = \begin{bmatrix} v_x \cos\varphi_k \\ v_y \sin\varphi_k \\ 0 \end{bmatrix}. \tag{17}$$

Here, $v_x$ and $v_y$ are the components of the agent's velocity in the horizontal and vertical directions, respectively. The velocity is assumed to remain constant.

*4) Reward Structure*

The reward function $\mathcal{R}(s, a)$ is defined with respect to the state $s$ and action $a$ taken by the agent. We provide the agent with the incentive to stay within the surveillance area and to minimise the uncertainty of the particle filter estimation. Additionally, an episode ends if the agent either exits the surveillance area or gets too close to the source. The reward structure is as follows:

$$\mathcal{R}(s, a) = \begin{cases} -E_{max} & \text{if } d_k \leq D_{min}, \\ -E_{max} & \text{if } \mathbf{x}_k \notin L, \\ +1 & \text{if } \hat{\sigma}_{\bar{\mathbf{x}}_k} \leq 0.2, \end{cases} \tag{18}$$

where $E_{\text{max}}$ is the maximum length of the episode, $\hat{\sigma}_{\bar{\mathbf{x}}_k}$ is the normalised uncertainty, $d_k$ is the Euclidean distance between the agent and estimated position of the source, and $L$ is the dimensions of the surveillance area.

The reward structure encourages two distinct behaviours. Firstly, the agent not only learns to reduce uncertainty below a threshold value of 0.2 but to also maintain an uncertainty below this threshold. Secondly, a large penalty is given if the agent exits the boundaries of the surveillance area or moves within a minimum distance, $D_{min}$, of the estimated source position. This encourages the agent to find optimal positions from which to operate, reinforcing our initial motivation of collision avoidance and reliable measurements from higher SNR values.

### B. Invalid Action Masking

Invalid action masking is a common technique within RL that is shown to improve convergence time to an optimal policy [17]. Action masking allows us to encourage certain behaviours of the agent that may be more sensible in a given scenario, whilst scaling better than the alternative approach of penalising the agent.

We employ action masking primarily to discourage frequent heading changes, mirroring the cautious maneuvering typical

of submarine operations, and to ensure the sensor array is properly aligned for accurate bearing measurements. If the sensor array has a degree of curvature or inconsistency in its alignment then the directional power will be misaligned. Due to this consideration, we do not incorporate the measurement into the particle filter's update procedure unless the array is approximately straight.

To approximate the time taken for the sensor array to straighten we store the heading $\varphi_k$ when the agent starts to turn, and the difference in heading $\Delta\varphi_k$ is calculated once the agent begins to straighten out. We incorporate the speed of the submarine and calculate the time to straighten as:

$$t_s = \left\lceil \frac{1}{v_k} \Delta\varphi_k \right\rceil. \tag{19}$$

The result is rounded up to incorporate any sway that may be present at the end of the straightening process.

We define the action mask as a Boolean vector $[a_{\text{left}}, a_{\text{forward}}, a_{\text{right}}]$, where each element can be either True or False, representing which actions are valid:

$$M_i^{\text{inv}}(s) = \begin{cases} True, & \text{if action } a_i \text{ is valid in state } s, \\ False, & \text{otherwise.} \end{cases}$$

### C. Implementation

We use the `MaskablePPO` implementation from the *Stable Baselines 3 Contrib* (SB3) package [18]. SB3 is a common framework that provides many RL algorithms and can directly interact with standard RL environments. Our environment is built in *Gym* from *OpenAI* [19], which allows for standardised communication between environments and RL algorithms.

To select the optimal hyperparameters, we use the *Optuna* framework [20], which utilises Bayesian optimisation (in the form of a Tree-structured Parzen Estimator [21]) along with pruning techniques to tune the hyperparameters of our agent. 100 trials were conducted for tuning a range of different parameters such as: the learning rate, discount factor, and step size per update; however, due to the robustness of PPO, we found no tangible improvement in performance over the default hyperparameter values.

## V. Scenario

We consider the scenario of tracking a moving surface vessel in the ocean environment. The scenario consists of a single submarine towing a passive sonar array moving at a constant speed of 5 m/s, 200m below the sea-surface. The surveillance area is 50×50 km in size. We do not consider additional clutter or sound emitting sources other than the ship stated in each scenario.

For each run, we randomise the starting position of both the agent and source such that they start in different quadrants of the surveillance area. In addition, there are three pre-calculated trajectories followed by the source for a total of 36 unique scenarios. These trajectories include a sinusoid, an ellipse, and a straight line. This randomisation ensures that our evaluation is comprehensive, accounting for a wide range of operational scenarios.

The source emits a complex sinusoidal signal with three harmonics reflecting the gearing of rotating mechanical components of the ship, which presents a unique narrowband signature. This signal is corrupted with broadband source noise, representing the sporadic and varied sounds produced onboard, such as human activity, mechanical noise or other vibrations. We model the source noise as pink noise where the intensity falls off at higher frequencies [22]. We also acknowledge the ambient noise in the ocean as a factor and model this as white Gaussian noise that is uniformly distributed in space. The signal parameters can be found in Table II which represent a modern container ship's radiated noise as stated in [23], where the spectral characteristics of various modern ships are compared. The signal for each harmonic can be expressed as follows:

$$S_j(t) = A_j \exp(2\pi i f_j(t - \delta_i - \tau) + i\phi_j), \tag{20}$$

where $A_j$ is the amplitude of the $j$th harmonic, $f_j$ is the frequency, $\phi_j$ is the initial phase, $\delta_i$ is the sensor delay for the $i$th sensor, and $\tau$ is the propagation time of the signal.

Thus, the combined signal is defined as:

$$S(t) = \sum_{j=1}^{N_h} S_j(t) + \mathbf{n}_p + \mathbf{n}_w, \tag{21}$$

where $N_h$ is the number of harmonics, $\mathbf{n}_p$ is a pink noise sequence and $\mathbf{n}_w$ is a white noise sequence, representing the broadband source noise and ambient noise, respectively.

TABLE II
Signal Parameters

| Parameter | Values | Description |
|---|---|---|
| $SL$ (dB) | [176, 172, 174] | Source levels of each harmonic |
| $f$ (Hz) | [45, 70, 800] | Frequencies of each harmonic |
| $\phi$ (rad) | [-112, -151, 86] | Phases of each harmonic |
| $SL_{\mathbf{n}_p}$ (dB) | 90 | Broadband source noise level |
| $SL_{\mathbf{n}_w}$ (dB) | 80 | Broadband ambient noise level |

For reasons of computational efficiency over a large number of scenarios, the results in this paper were generated using straight line propagation to generate signal attenuation characteristics. However, we also tested it on a smaller set of example scenarios using an industry standard acoustic propagation model (Bellhop [24], [25]) and the results were consistent. In these tests, the bathymetry was uniform and the sound speed in the water column was calculated using the Munk sound speed profile [26], with a maximum depth of 5 kilometres.

Using straight line propagation we are able to efficiently compute transmission loss and signal propagation time — key parameters that we integrate into our signal model. The propagation time is calculated as:

$$\tau = \frac{d}{c}, \tag{22}$$

where $d$ is the Euclidean distance between the source and receiver, and $c$ is the speed of sound obtained from the Munk sound speed profile at the depth of the submarine. The transmission loss, incorporating attenuation due to absorption, is calculated under the assumption of geometric spreading:

$$TL = 20\log_{10}(d) + \alpha \cdot \frac{d}{1000}, \tag{23}$$

where $\alpha$ is the attenuation factor.

The signal received by the sensor array undergoes beamforming, a process that spatially filters the acoustic signals to enhance the SNR from the desired direction. This technique involves combining the signals received at multiple sensors in the array in such a way that signals from a specific direction are constructively added [27]. A delay-and-sum beamformer is used in this example. We then compute the directional power across the beamformed output to identify areas of interest. The direction exhibiting the maximum power is then extracted and considered as the measurement, representing the relative bearing of the sound source.

## VI. Results

We trained 5 seeds for 1 million time steps each. Figure 1 shows the mean reward of each agent during training. The reward increases rapidly during the initial 100 thousand time steps, then plateaus as the agent becomes more confident in its environment. By exploring various actions, the agent identifies and exploits those yielding higher rewards, leading to early gains such as those that easily localise the source. Continued improvement then occurs as the agent also learns to avoid collisions with the surveillance boundaries and the estimated source position. While this behavior is effective, it has the downside of potentially leading to overfitting to early training scenarios, resulting in high initial rewards but poor generalisation to new situations later. Fluctuations can be explained by the agent's exploration and the variance in the particle filter. As the agent becomes more confident, it shifts from exploration to exploitation, indicated by the convergence during the final time steps.

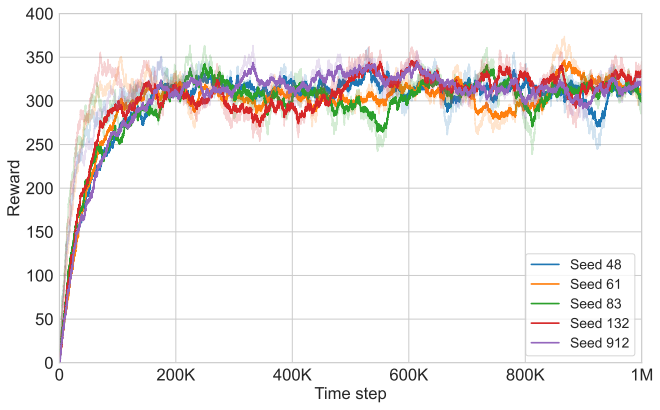


Fig. 1. Mean reward of each agent over 1 million time steps. The raw data is shown with shaded lines, while the smoothed data, obtained using exponential weighted moving average, is depicted with bold lines.

Explained Variance (EV), shown in Figure 2 is a measure of how much of the original variance can be explained by the model, allowing us to evaluate the performance of the policy evaluations. EV ranges from -1 to 1, where a value of 1 indicates perfect prediction in explaining the variability in the target variable, 0 indicates the model's predictions perform no better than the mean of the observed values resulting in performance appearing random, and negative values indicating worse than random. It is computed as:

$$EV(Y, \hat{Y}) = 1 - \frac{Var(Y - \hat{Y})}{Var(Y)}, \tag{24}$$

where $Y$ is the actual reward and $\hat{Y}$ is the predicted reward. Here, $Var$ represents the variance. Thus $Var(Y - \hat{Y})$ is the variance of the prediction errors and $Var(Y)$ is the variance of the actual reward.

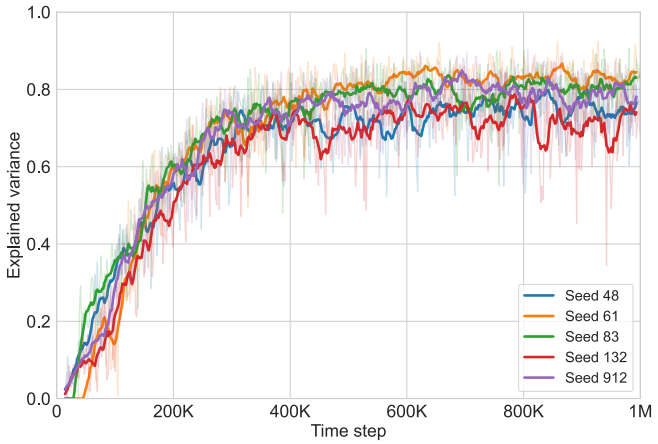


Fig. 2. Explained variance for each of the 5 trained seeds over 1 million time steps, shown with raw data (shaded) and smoothed data (bold).

We observe that the seeds converge at approximately 0.8, suggesting that a significant proportion of the variance can be explained by the model.

As a visual example of the agent's actions during an episode, we will consider a scenario where the agent starts in the upper-right quadrant of the surveillance area, while the source will follow a straight line trajectory. We can first see at time step 1 in Figure 3, after a single measurement, how the directional ambiguity results in two clusters formed around the hypotheses. As the agent changes its course, the directional ambiguity is resolved by the particle filter and the number of possible hypotheses reduces to one. This can be seen at time step 50. The distribution of particles is elongated due to bearing-only measurements. In order to better localise the source, the agent moves closer to the source, increasing the changes in bearing angle (often referred to as bearing rate). This behaviour is demonstrated at time step 100. The cluster of particles becomes smaller, representing the improvement in uncertainty and indicating a more precise estimation of the source's location. We can see that the agent's motion exhibits zigzag-like behaviour to improve the estimation of the range.

This motion is visually similar to that of Ekelund ranging, which is a manual TMA technique to calculate the range of a target by estimating bearing rates and speed of advance [28]. Over the remaining time steps during the episode, the source remains localised.

The evolution of the scenario can be quantified by monitoring the estimated bearing and range and calculating the radial error, as illustrated in Figure 4.

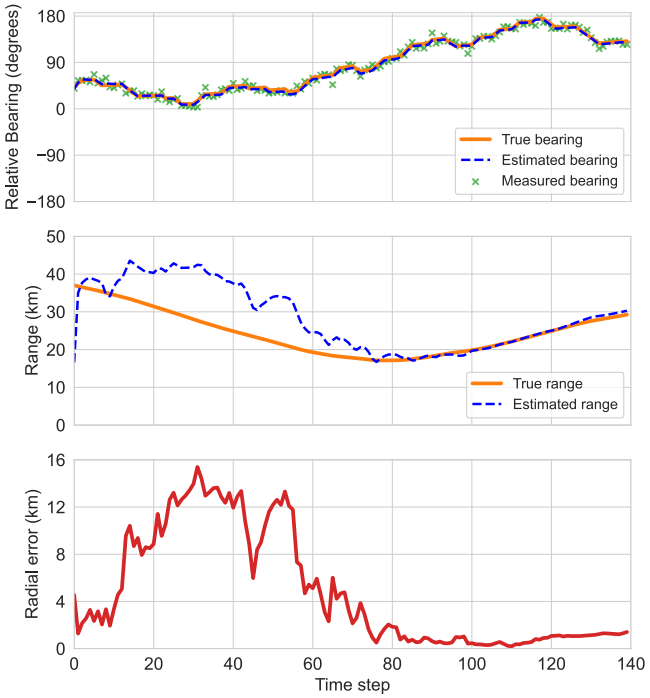


Fig. 4. Top: Bearings relative to the agent's heading (true, estimated, measured). Middle: Estimated vs. true range between agent and source. Bottom: Radial error between estimated and true positions.

Bearings are shown relative to the agent's heading. In the depicted scenario, the source remains on the port side of the agent, resulting in consistently positive bearings. Due to minimal noise in the received signal from the source, the measurements are accurate, as are the estimated bearings.

Given that we are dealing with bearings-only measurements, the difference between the true range and the estimated range remains large until the particle filter eventually converges at approximately time step 80. The convergence occurs rapidly as the agent approaches the source, driven by the increasing bearing rate.

The estimation error is quantified by the radial error, reflecting the difference between the true and estimated positions of the source. Since performance hinges on accurate range estimation, the error pattern mirrors the range estimation error. Initially, the error increases as the agent faces the source directly, reducing port-starboard discrimination and increasing uncertainty. When the agent is broadside to the source, the particle filter converges, reducing the error. The slight increase in error in the final time steps is due to the agent moving away from the source.

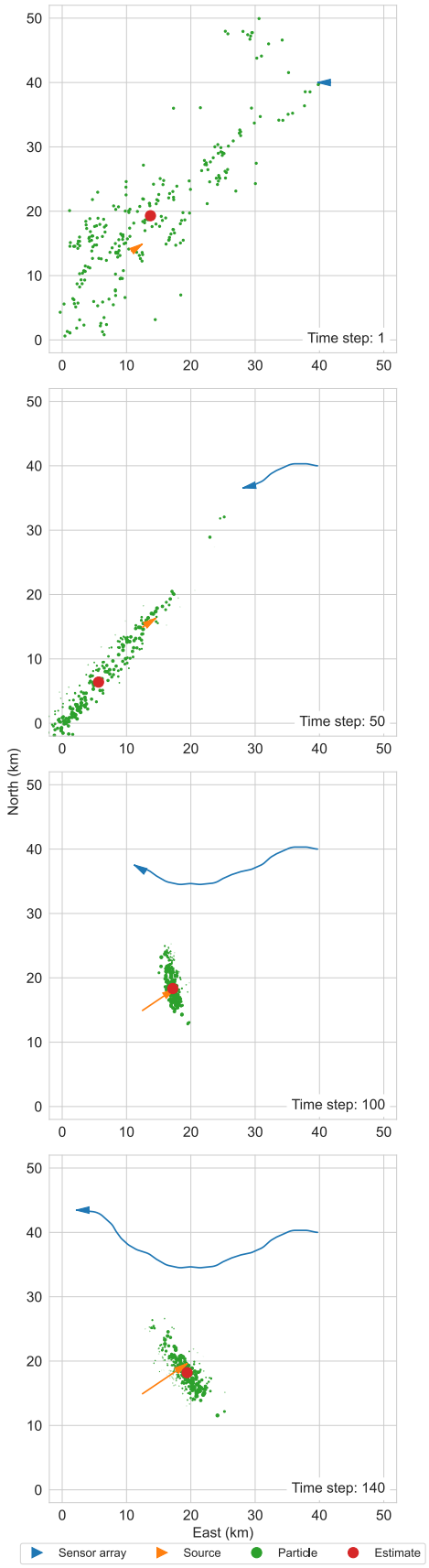


Fig. 3. Evolution of the scenario over time, illustrating the sensor array, source, source estimate and particle distributions at four distinct time steps.

## VII. Conclusion

In this paper, we have deployed a state-of-the-art reinforcement learning algorithm in a simplified underwater environment to control a passive towed array sonar system, with the objective of tracking a moving source whilst maintaining desirable operating procedures. Our results show convergence to an optimal policy given a sparse reward structure, utilising invalid action masking to improve convergence time and promote practical behaviour. We have considered challenges such as directional ambiguity inherent in the sonar system and have shown that reinforcement learning, in collaboration with an efficient particle filtering method, can learn an effective and sensible policy. We have also accounted for transmission loss in underwater acoustic propagation and implemented a signal processing chain to more accurately reflect sonar operations.

In this work, we explored proximal policy optimisation and observed promising results. Future research will extend to Bayesian reinforcement learning, which incorporates uncertainty into the learning process and decision-making [29]. This approach enhances adaptability and robustness in complex, dynamic environments, enabling better navigation and reasoning within complex and dynamic environments. Additionally, improving the reward structure to provide more frequent feedback could help the agent to discover strategies that better generalise to new scenarios.

Our aim is to introduce greater complexity into the simulation by incorporating clutter, multiple targets, and a three-dimensional state-space. Evaluating our methods in environments with clutter will help us assess their robustness under more realistic and challenging conditions. The inclusion of multiple targets will enable us to test and refine our algorithms for scenarios involving target discrimination and association. Expanding to a three-dimensional state-space will enhance the realism of our simulations, allowing us to explore underwater applications where both the source and receiver vary with depth. Recent developments in three-dimensional bearings-only target motion analysis (TMA), such as those in [30], incorporate both azimuth and elevation to improve tracking performance. These advancements will provide a more comprehensive evaluation of our approach and pave the way for its application in a wider range of operational scenarios.

## VIII. Acknowledgments

This work was supported by the EPSRC Centre for Doctoral Training in Distributed Algorithms through a Research Studentship under Grant EP/S023445/1.

## References

[1] S. C. Nardone and V. J. Aidala, "Observability criteria for bearings-only target motion analysis," *IEEE Trans. on Aerospace and Electronic Systems*, pp. 162–166, 1981.

[2] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," *Sequential Monte Carlo methods in practice*, pp. 3–14, 2001.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2 ed., 2018.

[4] F. Hoffmann, A. Charlish, M. Ritchie, and H. Griffiths, "Sensor path planning using reinforcement learning," in *2020 IEEE 23rd Int. Conf. on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2020.

[5] W. Yi, L. Fu, Á. F. García-Fernández, L. Xu, and L. Kong, "Particle filtering based track-before-detect method for passive array sonar systems," *Signal Processing*, vol. 165, pp. 303–314, 2019.

[6] A. O. Hero and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, pp. 3064–3075, 2011.

[7] M. L. Hernandez, "Optimal sensor trajectories in bearings-only tracking," in *Proc. of the Seventh Int. Conf. on Information Fusion*, vol. 2, pp. 893–900, Citeseer, 2004.

[8] C. Mishra, S. Maskell, S.-K. Au, and J. F. Ralph, "Efficient estimation of probability of conflict between air traffic using subset simulation," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 55, pp. 2719–2742, 2019.

[9] B. Hadi, A. Khosravi, and P. Sarhadi, "Deep reinforcement learning for adaptive path planning and control of an autonomous underwater vehicle," *Applied Ocean Research*, vol. 129, p. 103326, 2022.

[10] C. M. Taylor, S. Maskell, and J. F. Ralph, "Using hybrid multiobjective machine learning to optimise sonobuoy placement patterns," *IET Radar, Sonar & Navigation*, vol. 17, pp. 374–387, 2023.

[11] K. Bell, T. Corwin, L. Stone, and R. Streit, *Bayesian Multiple Target Tracking, Second Edition*. 2013.

[12] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, pp. 174–188, 2002.

[13] G. Welch, G. Bishop, *et al.*, "An introduction to the Kalman filter," 1995.

[14] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[15] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, (Berlin, Heidelberg), pp. 160–172, Springer Berlin Heidelberg, 2013.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[17] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *The Int. FLAIRS Conf. Proc.*, vol. 35, 2022.

[18] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, pp. 1–8, 2021.

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," 2016.

[20] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *The 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

[21] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[22] P. T. Arveson and D. J. Vendittis, "Radiated noise characteristics of a modern cargo ship," *The Journal of the Acoustical Society of America*, vol. 107, pp. 118–129, 2000.

[23] M. F. McKenna, D. Ross, S. M. Wiggins, and J. A. Hildebrand, "Underwater radiated noise from modern commercial ships," *The Journal of the Acoustical Society of America*, vol. 131, pp. 92–103, 2012.

[24] M. B. Porter, "The bellhop manual and user's guide: Preliminary draft," *Heat, Light, and Sound Research, Inc., CA, USA, Tech. Rep*, vol. 260, 2011.

[25] M. B. Porter, "Beam tracing for two-and three-dimensional problems in ocean acoustics," *The Journal of the Acoustical Society of America*, vol. 146, pp. 2016–2029, 2019.

[26] W. H. Munk, "Sound channel in an exponentially stratified ocean, with application to SOFAR," *The Journal of the Acoustical Society of America*, vol. 55, pp. 220–226, 1974.

[27] M. A. Ainslie, *Principles of Sonar Performance Modelling*. Springer, 2010.

[28] D. H. Wagner, W. C. Mylander, and T. J. Sanders, *Naval Operations Analysis*. Annapolis, MD, USA: Naval Institute Press, 1999.

[29] R. Dearden, N. Friedman, and S. Russell, "Bayesian q-learning," *Aaai/iaai*, vol. 1998, pp. 761–768, 1998.

[30] L. Badriasl and K. Dogancay, "Three-dimensional target motion analysis using azimuth/elevation angles," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, pp. 3178–3194, 2014.