Optimizing Source Localization via Reinforcement Learning in Multi-Agent Underwater Networks

J. Moos Middelkoop[1], Federico Celi[2], Alessandro Faggiani[2], Hilde I. Hummel[1], Sandjai Bhulai[3], Alessandra Tesei[2], Robert Been[2], Gabriele Ferri[2]

[1]*Centrum Wiskunde & Informatica, Amsterdam, The Netherlands*

[2]*NATO's Centre for Maritime Research and Experimentation, La Spezia, Italy*

[3]*Vrije Universiteit, Amsterdam, The Netherlands*

***Abstract*—In this paper, we propose a novel approach to multi-agent underwater source localization with passive sonar. Our framework optimizes the trajectories of two autonomous underwater vehicles, each towing an antenna, to maximize the probability of detection of the source. We implement a shared-parameter Multi-Agent Reinforcement Learning (MARL) strategy with non-synchronous actions to address the challenges posed by non-stationary multi-agent environments. We train a neural network using proximal policy optimization (PPO) to act in a simplified simulation environment, and evaluate it in a realistic simulation engine, demonstrating robustness to communication losses of up to 60%. Our preliminary results indicate that the RL-based trajectory optimization trained in a simplified simulation engine can achieve comparable performance to traditional approaches in a realistic simulation engine. At the same time, this transfer may be due to simplifying assumptions made in the experimental setting presented here. We provide recommendations for future work to further evaluate this approach.**

I. Introduction

The localization of an underwater sound-emitting source is a challenging problem with several real-world applications, including tracking marine wildlife [1]–[3] and search & rescue operations [4], [5]. Recent advances in maritime robotics have made the use of autonomous underwater assets for underwater source localization (SL) a reality. By networking these assets, it becomes possible to enable cooperative localization strategies, distributed sensing, and real-time information sharing. This is likely to enhance the accuracy and robustness of SL.

While localization is not a domain-specific problem *per se*, the underwater environment introduces challenges that are usually not found in land, surface, or air operations. Crucially, underwater communication is severely limited compared to over-water communication in both bandwidth and reliability. Information can be wirelessly exchanged underwater only via sound through the acoustic channel. This is severely unreliable though, as sound propagation in water is affected by many factors, including water temperature, pressure, and salinity. At the same time, due to the nature of the sensors, SL requires the coordination in space and time of multiple sensing assets to locate the source successfully. For these reasons, developing autonomous networks of underwater assets for SLn that can coordinate through very low bandwidth channels and which are robust to missed communications is a problem of ongoing interest.

This work has been supported by the NATO Allied Command Transformation under the Autonomous Anti-Submarine Warfare research programme.

Passive sonar technology is widely used for underwater sensing [6]. It relies on the detection of sound waves emitted by external sources rather than actively transmitted signals. This makes it particularly useful for stealth applications, as it does not introduce additional noise into the environment. Linear hydrophone arrays, a common implementation of passive sonar, consist of multiple hydrophones arranged in a line to measure the direction of arrival (DOA) of incoming acoustic signals [7]. These arrays enable precise bearing estimation by exploiting phase differences between signals received at different elements. However, they often face challenges such as port/starboard ambiguity, meaning they cannot inherently distinguish whether the sound is coming from the left or right relative to the array. Furthermore, direct range measurement is difficult with passive sonar as this relies solely on received sound waves without actively sending out signals to determine the distance to the sound source.

Recently, networks of autonomous underwater vehicles (AUVs) have been used to improve SL by leveraging principles from optimal and distributed control theory [8]–[10]. By employing cooperative control strategies and decentralized decision-making, multiple AUVs can dynamically position themselves to resolve port/starboard & range ambiguity and enhance localization accuracy by triangulating the source's position. Additionally, this coordination allows the vehicles to optimize their trajectories in real-time and maximize the probability of detection (Pd) of the source, all while maintaining robust performance under environmental uncertainties and communication constraints.

Traditional approaches to underwater SL typically rely on model-based methods grounded in signal processing and estimation theory. Techniques such as Beamforming and Time Difference of Arrival use measurements from multiple hydrophones to estimate the position of a sound-emitting source. Optimization-based methods, such as those leveraging the Fisher Information Matrix (FIM), have also been employed to design optimal sensor trajectories that maximize localization accuracy [11]. These approaches are effective but require precise knowledge of the acoustic environment and sensor characteristics, which may not always be available in real-world scenarios. Furthermore, they can struggle to adapt to

dynamic and uncertain conditions, particularly in multi-agent settings where communication constraints and environmental variability play a significant role.

Lately, reinforcement learning (RL) has gained significant attention thanks to its success in solving complex decision-making and control problems. These include environments with high-dimensional state spaces and limited prior knowledge. Advances in deep learning, coupled with improved training algorithms and computational resources, have enabled RL to achieve state-of-the-art performance in multiple domains, such as robotics and autonomous navigation. In the context of underwater SL, RL offers a promising framework for both low-level control [12] and optimizing the trajectories of AUVs in a data-driven manner [13]–[15]. Unlike traditional model-based approaches, which often rely on strong assumptions about the environment and sensor characteristics, RL can learn adaptive policies directly by interacting with the environment. This makes it particularly well-suited for handling uncertainties and dynamic constraints inherent in multi-agent underwater networks. One of the main challenges of RL is its reliance on large amounts of high-quality training data, which are particularly difficult to obtain for underwater operations. Therefore, developing realistic simulations and efficient data collection strategies is crucial to ensure that RL-based approaches generalize well to real-world deployments.

In this work, we investigate the use of Multi-Agent Reinforcement Learning (MARL) for the problem of underwater SL using multiple AUVs equipped with passive sonar. We leverage simulation-based training to develop adaptive and data-driven AUV coordination strategies. The use of RL has been explored and successfully implemented for many closely related problems, including single-AUV SL with passive sonar sensing [16], [17], and multi-AUV SL with active sonar sensing [18]. However, to the best of our knowledge, an RL implementation for the setting of multi-AUV SL with bearing-only measurements from passive sonar sensing has not been attempted before.

Our primary objectives are to (i) formulate the multi-asset localization problem in a way that is amenable to RL-based solutions, (ii) design MARL policies that optimize AUV trajectories to acoustic source location estimation accuracy while mitigating the limitations of passive sonar, (iii) evaluate the performance of these policies against traditional model-based approaches, and (iv) evaluate robustness to a setting with disrupted communication.

In particular, building upon previous work [9], [10], we synthesize the state of the surrounding environment to create a World Model by means of probabilistic Occupancy Grids (OG). A detailed description of OGs and their use in underwater localization is provided in Section II-B. An estimated target position is extracted from the OG. This, together with the AUV positions is used to compute a set of quantities describing the state of the system which serves as the input layer to the neural network. In return, the network outputs a heading decision to command each AUV. By training RL agents in a simulated environment, we aim to demonstrate that MARL can effectively coordinate multiple AUVs to successfully perform SL, even under conditions of uncertainty and intermittent communication. The network performance is then evaluated in a high-fidelity simulation environment. The results from our simulations provide insights into the potential benefits and limitations of RL-based approaches for autonomous underwater sensing and multi-agent coordination.

II. Background

This section introduces the key preliminary concepts essential to the approach presented in Section III. Due to space constraints, we provide a concise overview and refer the reader to the original works cited throughout for further details.

A. Multi-agent underwater sensing networks

For this work, we focus on a simplified underwater sensing network made up of two AUVs traveling at a constant speed of 1 m/s. Each AUV tows a linear array in a passive sonar configuration and we assume that the location of the linear array, i.e., the sensor, corresponds to that of the AUV. Passive sonars quietly listen to sound waves propagating in water at different frequencies and can distinguish the direction of arrival of a sound. Crucially, however, these sensors cannot easily estimate the distance to the sound source. We further assume that both linear arrays have port/starboard ambiguity and, therefore, cannot distinguish whether the sound is coming from the left or the right relative to their longitudinal axes.

Working in a networked configuration, the AUVs are required to exchange information with the goal of optimizing their behavior for SL. One of the key challenges of underwater robotics is addressing the limitations of communication capabilities described in Section I. In this setting, the AUVs communicate by means of an acoustic modem. At best, this enables the transmission of one message with a payload of 60 bytes every second. On top of this inherently small data throughput, packets are lost at high rates. For this reason, we will pay particular attention to the robustness of the developed approach to missed communications among AUVs in this work. For more details regarding the typical hardware used in our experimental campaigns, we refer to [19], [20].

B. State estimation via Occupancy Grids

The estimate of the system's state relies on a perception approach based on Occupancy Grids [21], [22]. An OG is a discretized representation of the environment into a collection of cells. Each cell can either be occupied (e.g., when the cell is associated to the position in space of a sound emitting source) or empty, see Fig. 1 (a)-(b). An estimate of the occupancy state of each cell can be built from the sensors' readings. Let $C \in \mathbb{R}$ be the number of cells in the OG and $m \in \mathbb{R}^C$ be the collection of all binary cell states. Further, let $z \in \mathbb{R}^{p \times q}$ denote the collection of all measurements, where $p, q \in \mathbb{R}$ are the dimension of the sensor's readings and the cardinality of

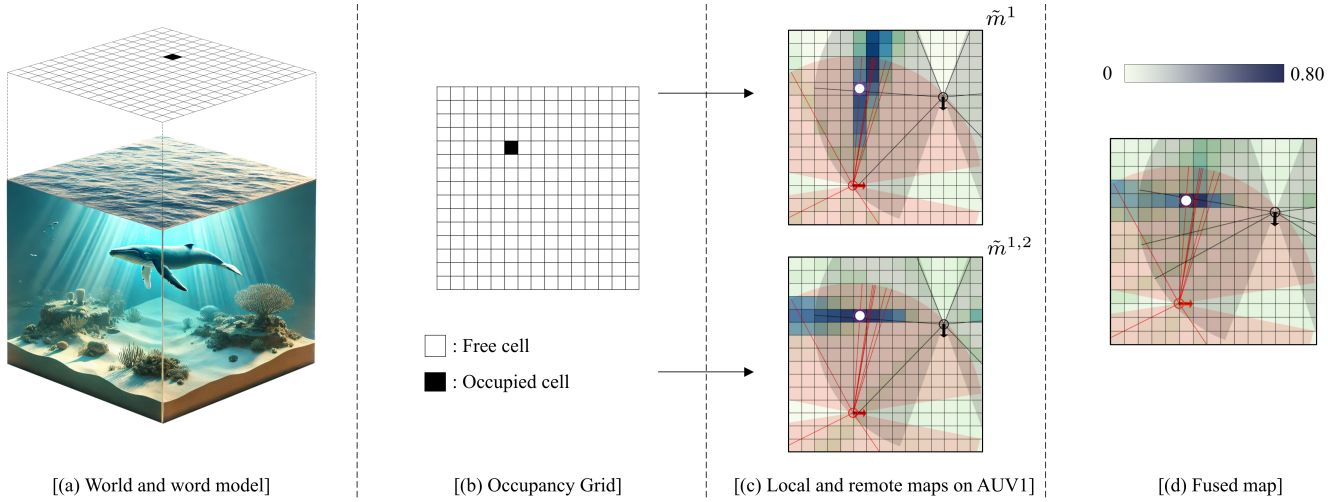


Fig. 1. This figure shows the OG mapping framework for multi-agent source location described in Section II-B. In this setting, the underwater sensing network is composed of two AUVs, each carrying a linear array in a passive sonar configuration. Panel (a) shows the environment of interest and its discretization into cells. Panel (b) shows the OG associated with the environment: white (black) cells depict an area of the environment that is free from (occupied by) a sound source. Panel (c) shows the local map $\tilde{m}^1$ and remote map $\tilde{m}^{1,2}$ stored onboard AUV 1. The positions and orientation of AUV 1 (AUV 2) are depicted with a red (black) arrow. Darker cells correspond to areas of the environment with a higher occupancy probability. The white dot shows the position of the sound source. Given the sensors' bearing-only nature, the source's position cannot be estimated precisely from a single sensor reading (notice how higher probabilities span a ray from the sensor through the correct source location but are rather uniformly distributed in space). Panel (d) shows the result of fusing maps $\tilde{m}^1$ and $\tilde{m}^{1,2}$ in $\tilde{m}^1_F$ and the consequent improvement of the localization performance.

the readings, respectively.[1] Then, the OG can be estimated via the posterior $\hat{m} = \text{prob}\,(m \,|\, z)$.

Since the number of possible map combinations is 2^C, it is usually prohibitive to compute the full posterior $\hat{m}$. Consequently, OG mapping methods typically compute the marginal posteriors for the probability of occupancy of each cell, independently, as $\tilde{m}_i = \text{prob}\,(m_i \,|\, z)$, where m_i is an element of m, with $i \in \{1, 2, \ldots, C\}$. Further approximations are usually required to allow the computation of all $\tilde{m}_i$ in real-time, as discussed in [22]. Ultimately, as a result of all simplification steps involved in the process, it is possible to obtain an estimation of the state m as $\tilde{m} = [\tilde{m}_1\, \tilde{m}_2\, \ldots\, \tilde{m}_C]^\top$, which is suitable for real-time processing, see Fig. 1 (c).

When a new sensor measurement is available, the estimate $\tilde{m}$ should be updated accordingly. In [8], we proposed a perception layer based on the Independence of Posterior update rule [23]. Independence of Posterior works better than traditional Bayesian-based update rules in scenarios characterized by a low prior probability, i.e., the default probability associated with an unexplored cell. Notice that this approach does not require the occupancy state of each cell to be time-invariant, as is often assumed in similar approaches. A scheme of the OG-based cooperative framework is shown in Fig. 1. A mathematical model of each sensor is developed (for example, by characterizing the probability of detection at different bearing/ranges and the false alarm rate) and is used to create an estimation of the state in $\tilde{m}$. Each AUV i leverages its onboard sensor to generate a *local map* $\tilde{m}^i$. Since AUVs are allowed to share raw data, each AUV i also stores an estimate of the state using exogenous information from AUV j by creating a map $m^{i,j}$, i.e., the *remote map* of j on i. Clearly, $\tilde{m}^{i,j}$ ideally tends to $\tilde{m}_j$ as communication frequency increases. Finally, each AUV i fuses its local map and the available remote maps into a unified view of the state, which we refer to as *fused map* $\tilde{m}^i_F$ and that can be used to estimate the source location, see Fig. 1 (d).

The quality of SL can be quantified using the Laplacian of the fused map $\mathcal{L}(\tilde{m}^i_F)$, obtained by computing the divergence of the gradient of the scalar field $\tilde{m}^i_F$. Cells with high occupancy probability that are surrounded by cells with low occupancy probability are associated with negative and small values of $\mathcal{L}(\tilde{m}^i_F)$. These are, in practice, the cells that we consider the most likely for source location [9].

C. Optimal geometry in bearing-only localization

As discussed in Section II-B, the information originating from multiple bearing-only sensors can be fused together to improve SL. One might ask, then, if there exists an optimal sensors-source geometry (i.e., an optimal position of the sensors with respect to the source's location) that improves SL overall. An intuitive answer to this question can be inferred through Fig. 1 (c), where the angle described by the sensors and the source is roughly $|\phi| = \pi/2$. This sensors-source geometry minimizes the number of cells with high occupancy probability that are overlapping between the two sensor readings (see Fig. 1 (d) for the resulting fused map). Conversely, if $|\phi| = \{0, \pi\}$, the two sensors become colinear with the target, and their readings are not able to resolve the ambiguity of the source's position, as most cells with high probability remain overlapping.

[1] In passive sonar applications, each sensor reading (also called a *contact*) is made up of the bearing angle and the position of the sensor in the plane and therefore $p = 3$.

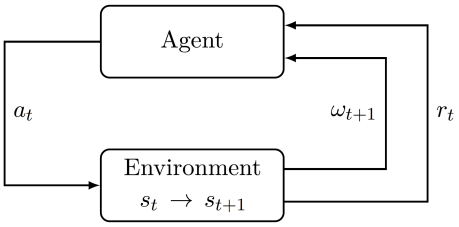


Fig. 2. The standard reinforcement learning loop. At time t, An agent makes an action a_t based on an observation ω_t. This causes a state transition in the environment from s_t to s_{t+1}, with a corresponding reward r_t. Based on the new state s_{t+1}, the agent receives a new observation ω_{t+1} and uses r_t to evaluate a_t. Adapted from [26].

This intuitive conclusion was discussed and formally shown in [24], where the authors derive that the Fisher Information Matrix (FIM) for N bearing-only sensor measurements as

$$\mathcal{I}(p, \mathbf{s}) = \sum_{i=1}^{N} \frac{1}{d_i^2 \sigma_i^2} \begin{bmatrix} \cos^2 \phi_i(p) & -\frac{\sin 2\phi_i(p)}{2} \\ -\frac{\sin 2\phi_i(p)}{2} & \sin^2 \phi_i(p) \end{bmatrix}, \quad (1)$$

where p is the source location, σ_i is the measurement error standard deviation for sensor i, d_i is the distance from sensor i to the source, and ϕ_i is the bearing angle between sensor i and the source, with $\mathbf{s} = \{d_1, \ldots, d_N, \phi_1, \ldots, \phi_N\}$. Since the variance of an unbiased estimate $\hat{p}$ of p can be related to the FIM though the Cramér-Row bound as

$$E[(\hat{p} - p)(\hat{p} - p)^\top] \geq \mathcal{I}^{-1}(p, \mathbf{s}), \quad (2)$$

one can indeed conclude that $|\phi| = \pi/2$ is the optimal two-sensors angle for SL [24, Proposition 1].[2] This fundamental result is also at the core of the method presented in this paper, as discussed in Section III.

D. Reinforcement Learning

Reinforcement learning is a machine learning paradigm where agents learn to make sequential decisions by interacting with an environment. The goal is for an agent to find an optimal sequence of actions a based on observations ω of the current state s of the environment. Unlike supervised learning, which requires labeled data, RL agents learn through trial and error, receiving feedback in the form of rewards r on their actions (larger positive rewards are associated with actions that take the system closer to the desired state). We distinguish between the abstract agent (the learning algorithm) and agent *instances*, which are individual trained instances of the agent with a unique set of parameters. The actions should be chosen so to maximize the expected value of the (discounted) accumulated reward [25]. The RL approach is particularly well-suited for complex control problems where optimal solutions are difficult to derive analytically. A high-level schema of the standard RL loop is presented in Fig. 2. One iteration of this observation–action–reward loop is called a *step* or *timestep*.

[2]In particular, the estimation error is inversely proportional to $\det(\mathcal{I}(p, \mathbf{s}))$.

The RL framework is typically formalized as a Markov Decision Process, defined by states, actions, transition probabilities, and rewards. In our underwater SL context, states represent the positions and internal model of the surroundings, captured in the internal fused map of the AUVs. Actions correspond to heading changes, and rewards reflect how effectively the vehicles position themselves to maximize detection probability and information gain. An RL task is often episodic, where each episode represents complete task sequences from initialization to termination. Starting from an initial state, a sequence of states, actions, and rewards that occur when an agent interacts with the environment is called a *rollout* or *trajectory*.

The core objective in RL is to optimize the agent's decision-making policy. A policy π maps states to actions, and the goal is to find a policy that maximizes the expected cumulative reward. Value functions estimate the expected return from a given state, either following a specific policy (state-value function $V(s)$) or taking a specific action and then following the policy (action-value function $Q(s, a)$).

Modern RL approaches often employ neural networks to approximate these functions, allowing them to handle high-dimensional state spaces. Policy gradient methods directly optimize the policy parameters θ by estimating the gradient of the expected return with respect to these parameters, while "actor-critic" methods combine policy optimization with value function approximation to reduce variance in the gradient estimates. An in-depth discussion of these techniques and more can be found in [25].

E. Simulation engines for data generation and validation

Training a neural network via RL is a data-intensive process, and training an agent by interacting with the true environment is usually prohibitively expensive or dangerous in robotic applications. For this reason, simulated environments are often used [26, Chapter 11.2]. A simulated environment should accurately model the dynamics of the problem, while being efficient at fast data generation. Collecting sufficient high-quality data is especially challenging in the underwater domain due to the limitations of underwater communication and the inherently high costs of operating sea vehicles.

For this work, we use two distinct simulation engines. The first, which we refer to as Σ_1, is used for training. It is developed in Python as an OpenAI Gymnasium object [27], and it is designed to be as quick as possible at the cost of approximations over several aspects of the problem dynamics. The second simulation engine, Σ_2, is used to validate the approach we develop in Section III. This simulation engine approximates the problem's dynamics more closely, including a more accurate model for the vehicle dynamics, the sensor's performance, and the communication channel. Consequently, Σ_2's additional functionalities make it too resource-intensive to be used efficiently for training purposes.

A key distinction between simulation engines Σ_1 and Σ_2 is the source position estimation process. Σ_1 assumes perfect

knowledge, while Σ_2 incorporates realistic bearing measurement generation via the OG pipeline described in Section II-B. This creates an estimation of the source position which may contain inaccuracies, especially during early episode stages. We hypothesize that training with perfect information enhances learning efficiency without impeding policy transfer between environments. As agents establish reasonable target estimates through OG mapping during deployment, their learned behavior optimizes sensor-source geometry, progressively improving position estimation accuracy.

III. Multi-Agent Reinforcement Learning for Source Localization

A. The Multi-Agent Reinforcement Learning framework

In RL, an agent is an entity that interacts with the environment to learn a behavior that maximizes a given reward function. To do this, an agent implicitly learns its environment dynamics – how the environment changes as a result of the actions that the agent takes. In a situation where multiple agents cooperate to complete a task, one agent's actions affect the overall state of the environment and, therefore, the behavior of the other agents. To perform well, all agents need to account for the behavior of their collaborators: the *joint behavior*. In this scenario, we consider that the agents are concurrently learning to adapt their behavior independently from each other. Then, the joint behavior, and thus the environment dynamics, are nonstationary from the point of view of one agent [28].

A possible solution to coordinate the behavior of multiple agents in MARL is a centralized training, distributed execution (CDTE) scheme, often implemented with shared parameters. When each agent learns to perform a cooperative task independently, they effectively chase a moving target since the other agent's policy is also shifting. However, with a shared policy network, any update from one agent is immediately reflected in both agents' behavior. This does not necessarily eliminate the nonstationarity of the environment dynamics but has been shown empirically to be a simple and effective method in MARL [29].

In the setting presented here, the two AUVs share a full set of policy network parameters θ, which effectively creates one agent instance that controls both AUVs. The agent instance learns how to cooperate with a collaborator of which it knows the position and heading direction. Both AUVs interchange contacts to form a fused probabilistic OG map estimate of the environment. In turns, either agent takes an action a_t based on an observation ω_t and collect a reward r_t, accordingly. Tuples (ω_t, a_t, r_t) are added to the collective trajectory to update θ when the episode terminates. Fig. 3 shows how the RL loop is adapted to a multi-agent scenario with fully shared parameters. Once training is finished, the agents are deployed with identical policy networks.

One drawback of all agents sharing the same parameter network is the inability to create heterogeneous agents that could perform separate tasks or behaviors. This limits the potential benefits of having a system with multiple nodes.

B. Proximal Policy Optimization (PPO)

The Proximal Policy Optimization (PPO) algorithm is used to train the agents [30]. PPO is an actor-critic policy gradient algorithm, where a differentiable parametrized policy π_θ is optimized via gradient ascent over an estimate of the policy gradient to approximate the optimal policy π^*.

This policy gradient estimator is obtained by differentiating an objective function with respect to θ. A standard objective function at a certain timestep t is

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t[\log \pi_\theta(a_t|s_t)\hat{A}_t], \tag{3}$$

where, $\hat{A}_t$ is an estimator of the *advantage function*. At any time t, the advantage function describes how much better or worse the return of a certain trajectory is than the expected value of that return, given the value functions at time t. Therefore, $L^{PG}(\theta)$ evaluates a stochastic policy π_θ by weighing the achieved performance during a trajectory by how likely the performed actions were under the current policy.

Performing these rollouts can be expensive, especially when using heavyweight simulations. Therefore, doing multiple gradient ascent updates using the same trajectory is appealing. However, in practice, this often leads to destructively large policy updates [30]. Small changes in the policy parameters could cause unwanted large changes in the resulting policy. In Deep RL it is a common problem to suddenly see the performance drop off a figurative 'cliff' during training and not recover.

PPO is the latest iteration in a series of approaches to address this issue [31]. In particular, it introduces a clipped component to the objective function as

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t], \tag{4}$$

where $r_t(\theta)$ is defined as $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta\text{old}}(a_t|s_t)}$, with $\pi_{\theta\text{old}}$ being the policy computed at the previous step. For the rationale behind these design choices, we refer to [31].

We make use of the PPO implementation by Stable Baselines 3 (SB3) [32]. We use the standard 'MlpPolicy' actor-critic policy network. Both actor and critic are represented by a multi-layer perceptron (MLP) with two hidden, fully connected layers, each with 64 units. The full default set of algorithm parameters from SB3 was used, and this was sufficient to show promising learning capacity on this task.

C. Observation space

The observation space, which serves as input to the policy neural network, comprises seven quantities representing Euclidean distances (expressed in meters) and angles (expressed in radians) between the sensors and the source:

- actor: L2 distance to the source (d_a)
- actor: bearing to the source (ϕ_a)
- collaborator: L2 distance to the source (d_c)
- collaborator: bearing to the source (ϕ_c)
- L2 distance between the AUVs ($d(a, c)$)
- inter-sensor angle measured at the source position ($\phi_{a,c}$)

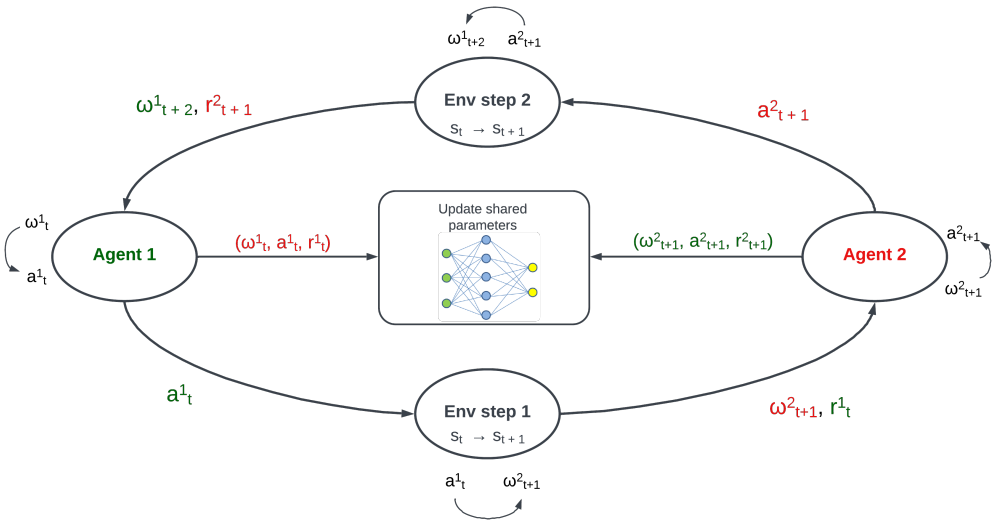


Fig. 3. This figure depict the asynchronous MARL loop with parameter sharing, as described in Section III-A. Agent 1 receives an observation ω_t^1 of its state at a certain time t, and selects an action a_t^1 according to the current policy π_θ. The environment takes a step forward in time based on a_t^1. Agent 2, then, receives observation ω_{t+1}^2 and the reward r_t^1 corresponding to a_t^1. The tuple $(\omega_t^1, a_t^1, r_t^1)$ is added to the trajectory which is then used to perform policy updates on the shared network. Finally, Agent 2 selects a_{t+1}^2 based on ω_{t+1}^2 and initiates the next environment step.

- last action of the collaborator (ψ_c)

At a given timestep, which AUV is selected as the 'actor' and which is selected as the 'collaborator' depends on which AUV made the last action. The latter collects the subsequent observation and is thus responsible for updating their shared parameters.

The observation elements are normalized in $[-1, 1]$ before being fed to the policy network as input.

D. Action space

The action space is one-dimensional: a continuous heading change in $[-\frac{\pi}{4}, \frac{\pi}{4}]$. The output of the neural network is a scalar in $[-1, 1]$, which is then mapped to the action space range.

E. Reward function

The reward function is based on theoretical sensor management principles. It is a combination of the sensor's (Pd) function, which is unique to each sensor type and can be changed based on the particular sensor used, and $\det(\mathcal{I}(p, \mathbf{s}))$, the determinant of the FIM, which is maximized when any configuration of a set of sensors $\mathbf{s}$ and a single acoustic source p is optimal in an information-theoretic sense, see Section II-C.

The distance factor in the FIM is fixed to a predetermined value so that changes in the value of the determinant of this matrix will only be due to changes in the angular configuration. This modification is necessary to prevent AUVs from attempting to crash into the source, which would occur due to the singularity of $\det(\mathcal{I}(p, \mathbf{s}))$ when the range is 0. A theoretical optimal trajectory would end up at the precise source position, but, in reality, one would want to keep some distance from the source.

The FIM component is normalized for use in the reward function. Throughout each episode, we keep track of the highest value of $\det(\mathcal{I}(p, \mathbf{s}))$ observed up to the current time step t. At each time step, we divide the current $\det(\mathcal{I}(p_t, \mathbf{s}_t))$ by its maximum value to obtain the normalized metric

$$\widehat{\det}\, \mathcal{I}(p_t, \mathbf{s}_t) = \frac{\det \mathcal{I}(p_t, \mathbf{s}_t)}{\max\limits_{t' \leq t} (\det \mathcal{I}(p_t, \mathbf{s}_t))}. \tag{5}$$

Sensors are rarely perfect, and for each sensor, a probability density function $Pd(p, s)$ can be defined, which describes the probability that a source p will be detected, given its position relative to the sensor s. This function can be used to evaluate the optimality of a sensor-target configuration, and in our case, it will be used in the reward function. In the multi-agent setting, the Pd of multiple sensors are averaged:

$$\overline{Pd} = \frac{\sum_n^N Pd(p, s_n)}{N}. \tag{6}$$

Finally, an additional penalty when the source is outside of the detection range for the given sensor gives us the reward

$$r_t(p, \mathbf{s}) = \begin{cases} \overline{Pd} + \widehat{\det}\, \mathcal{I}(p, \mathbf{s}), & \text{if } d(p, s_n) < r_{\max}\ \forall n, \\ \overline{Pd} + \widehat{\det}\, \mathcal{I}(p, \mathbf{s}) - 1, & \text{otherwise.} \end{cases} \tag{7}$$

IV. Experiments Design

In this section, we describe the process of training a neural network for multi-agent SL with the approach proposed in Section III. An agent instance is trained for 3,000,000 steps in simulation engine Σ_1, while its performance is validated

through the simulation engine Σ_2, see Section II-E for a discussion on the differences between the two simulations.

Two AUVs are free to move in an operating box of $6,000 \times 6,000$ meters. Their starting locations are randomly initialized within a smaller box of $4,000 \times 4,000$ meters, centered within the operating box. A stationary acoustic noise source is positioned at $x = 50$, $y = 50$. In one timestep, each AUV travels for 40 meters, after which one AUV makes a new heading decision. Which AUV makes the decision alternates between timesteps. This entails that each AUV adjusts its heading every 80 meters (since speed is fixed at $1\,m/s$). Episodes are terminated with an additional penalty of -50 when at least one of the AUVs either leaves the operating box or gets within 100 meters of the source. Episodes are truncated without an additional penalty after 200 timesteps when no other termination criteria are met.

The resulting policy is evaluated in the simulation engine Σ_2. In this simulation engine, the AUV that makes the observation consults its internal fused OG map $\tilde{m}_F$ to estimate the source position. The source is estimated to be at the center of the cell with the minimum value of $\mathcal{L}(\tilde{m}_F)$, see Section II-B

A. Comparison with Baseline

Our optimized policy is evaluated against an analytical baseline solution proposed in [10]. This solution was designed for cooperative SL via passive sonar for two cooperating AUVs with limited communication. Similarly to our approach, the AUVs take their actions in alternating fashion and communicate their latest decision.

A priori, a set of eight pairs of starting points is chosen randomly on a circle with a radius of $1,500$ meters from the source. Both solutions will be evaluated with this identical set of starting points, for accurate comparison. For each initial condition, a 90 steps rollout is performed with the MARL policy and for a corresponding amount of time (1 hour) with the baseline algorithm. The average performance of both methods is compared in terms of absolute error of source position estimation.

B. Uncertain Communication

Throughout the training, a perfect arrival of all communication is assumed. This second experiment explores the resilience of the policy to missed communications. To give the agents an fair chance of dealing with decreased communication, we add a simple heuristic to create a sensible observation. The agents keep track of the last known position and heading of their collaborator. Whenever an agent notices that a message has not arrived, it predicts the position of the collaborator one timestep into the future. This prediction is based on the last known position and heading, in a Model Predictive Control fashion [33]. The observation of that agent is then built based on this prediction and the last known action of the collaborator. Once a new message arrives, the predicted position is overridden with the new information.

For this experiment, we evaluate the reward in place of the absolute error of source position estimation, which would normally be the more relevant metric for actual performance. We make this choice because constrained communication naturally leads to fewer contacts arriving to update the OG maps, and thus, the source position estimation is more erroneous. The agents' ability to overcome the challenge of constrained communication requires accurate evaluation. Using a measure that is directly influenced by communication success rate would compromise this assessment.

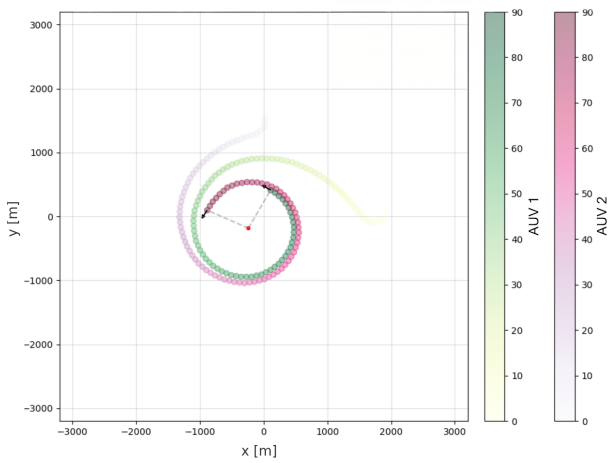


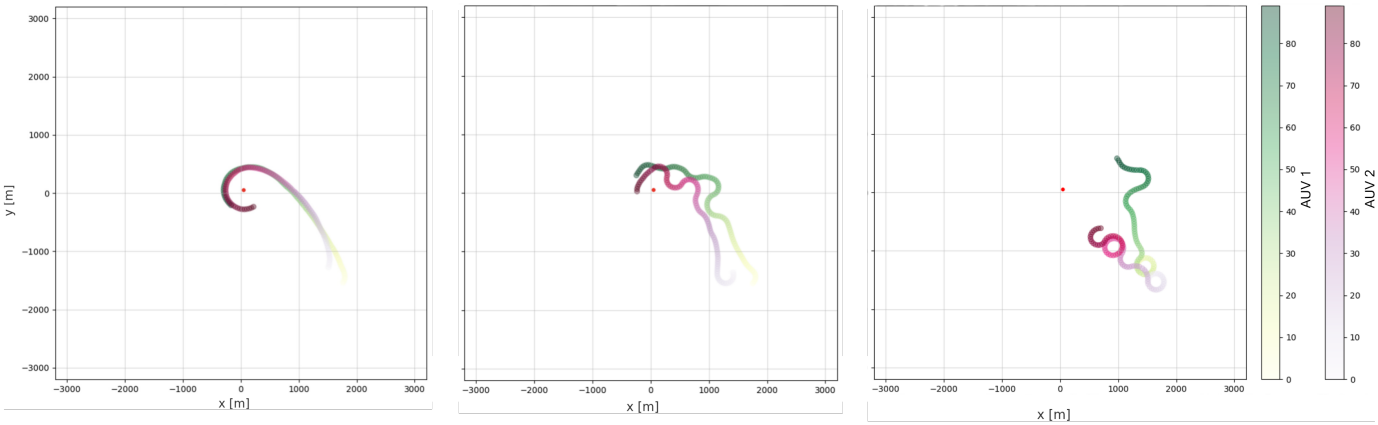
Fig. 4. A rollouts of 90 timesteps of the trained MARL solution, evaluated in the lightweight training simulation engine

V. Results Discussion

After $3,000,000$ steps, the training converges with the total reward per episode consistently ranging between [250, 280]. When evaluating the model, in both simulations Σ_1 and Σ_2, the paths show that the algorithm managed to learn a sensible policy. The AUVs find the source, approach it, and circle around it. The AUVs show ability to cooperate. They consistently circle in the same direction, and keeping a constant radius to the target. Furthermore, they manage a sensible phase difference in the circling behavior between each other, near the theoretical optimal $|\phi_{a,c}|$ of $\pi/2$, as discussed in II-C. If needed, one or both AUVs show that they can perform a maneuver to ensure this phase difference. Fig. V shows an example of this behaviour in Σ_1, and Fig. 5a shows the same results for Σ_2. This is a promising sign for potential sim-to-real transferability of the solution presented here.

A. Comparison with Baseline

Results show performance that is comparable with the baseline in terms of average error per episode. The mean absolute estimation error, averaged per run, was approximately 87.402 meters for the MARL solution, and approximately 81.898 meters for the analytical solution. However, even though the average performance of the MARL solution is slightly better, it does appear to be less reliable. The standard deviation around this mean was approximately 53.617 compared to approximately 38.240 of the analytical solution.



[a) 100%] [b) 40%] [c) 20%]

Fig. 5. This figure shows the robustness of the approach proposed in Section III to degradation in communication performance. Panel (a) shows the ideal condition, with perfect communication between the two AUVs. Panel (b) shows the same experiment of panel (a), this time carried out with a 40% probability for each message sent by an AUV to reach the the other AUV. As one can observe, the overall performance decreases, but the source location is nonetheless identified and successfully circled. Finally, panel (c) shows the same experiment as panels (a), carried out with a 20% probability for each message sent by an AUV to reach the the other AUV. This is an extreme condition, where the AUVs ultimately fail to successfully cooperate to circle the source location.

B. Uncertain Communication

When the reliability of successful communication is reduced, the paths of the AUVs become unstable, but the strategy of cooperating to approach and circle the source remains. Examples of the paths can be seen in Fig. 5, which displays examples of paths of two AUVs starting SL behavior about $1,5000m$ southeast of the source's location. As the percentage of successful communication decreases, the paths become less stable. However down to a communication arrival success rate of 40%, the AUVs remain able to locate the source and circle it. It is only with a rate of 20% of messages arriving, that the agents fail to showcase the key aspects of a good solution. The AUVs do not show coordinated behavior, nor manage to reliably circle the source.

Furthermore, performance in terms of reward remains acceptable. Fig. 6 shows the result in terms of mean reward over an episode of 90 steps. A slight decrease in performance can be observed as the percentage of dropped messages increases, with a drop in reward in the experimental setting where only 20% of messages arrive successfully. In line with this, the results of the one-way ANOVA test showed that some of the differences between groups were significant ($F = 11.4896$, $p < 0.001$). A Tuskey's HSD post-hoc test revealed that there were significant differences ($p < 0.05$) between the pairs: 20%-70% and 20%-100%.

To make the result of the linear regression more intuitive, these can be expressed as a function of the probability of packet loss, which we cluster over the four test groups $\left[0\%,\ 30\%,\ 60\%,\ 80\%\right]$. A linear regression through the mean rewards, corresponding to these communication loss probabilities per evaluation rollout was performed. It found a slight negative correlation between the level of communication success rate and mean episode reward. The fitted line has a slope of -0.003 and an intercept of approximately 1.061, with an R^2 value of 0.288.

The theoretical maximum value for the average reward over an episode is 1.8. This value would be achieved if, at each step, the maximum reward is achieved. This is, however, just a theoretical maximum. It would only be achieved if the agents started in a perfect geometric configuration, and if every following action succeeded in preserving the FIM theoretically perfect configuration.

In this experiment, the starting locations were set in such a way that quickly obtaining a high reward was difficult. In each experiment run, the agents started close to each other. Consequently, they first had to increase the distance between each other before circling the target, in order to increase the Fisher matrix component of the reward function. This is a possible explanation for why, even in the scenario with a communication success of 100%, the mean reward per episode was at most 75% of the theoretical maximum.

VI. Conclusions

A. Summary of findings

The results demonstrate that the MARL approach can successfully coordinate multiple underwater assets to perform a SL task with performance comparable with an analytical solution. Agents learn adequate behavior for an underwater SL task with passive sonar measurements, purely on a reward signal based on the optimal sensor configuration. Their paths are similar in shape to theoretical optimal sensor paths for localization of a stationary target with bearing-only measurements [11]. A difference with the theoretical optimal paths is that the AUVs don't reach the exact source location, but find a trajectory in which they are able to circle the source indefinitely. This behavior is achieved by setting the range factor to a constant in the FIM component of the reward, and the added penalty that is given when the AUVs come too

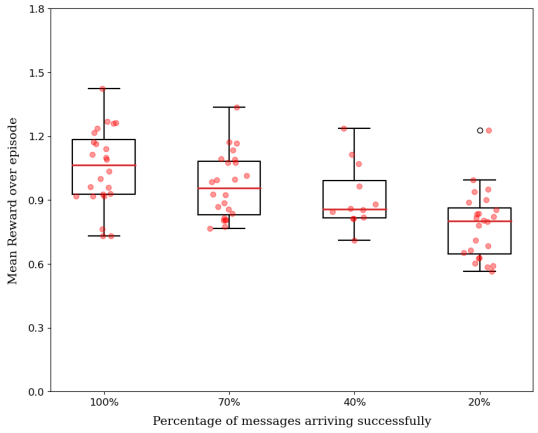


Fig. 6. Box plots for the average reward per episode, over 24 different episodes of 90 steps per condition.

close to the source, as discussed in section III-E. In a real life setting, this is important since maintaining a safe distance from the source preserves optimal sensing capabilities and avoids collisions with the acoustic source. Moreover, by adding a Model Predictive Control heuristic, the learned policies show robustness to communication losses of up to 60%. This is a critical consideration for practical underwater deployments where acoustic communication is inherently unreliable.

B. Limitations of the proposed approach

Our simulation approach presented some challenges. The lightweight simulation engine (Σ_1) used for training, while efficient, required significant simplifications that potentially limit real-world applicability. Conversely, the heavyweight simulation (Σ_2) offered more realism but severely restricted our ability to perform extensive experiments due to computational demands. This constraint is evident in the baseline comparison, where only eight evaluation runs could be completed due to time and efficiency limitations of the heavyweight simulation.

A fundamental limitation of our approach is that it only allows for homogeneous agents. By using full parameter sharing, both AUVs execute identical behaviors modulated only by their respective inputs. This restricts the potential benefits of having a multi-agent system where agents could specialize in complementary roles or behaviors. The system cannot develop heterogeneous strategies that might be better at optimizing information gain across the network.

Despite these limitations, our approach demonstrates promising initial results for applying RL to underwater SL tasks and establishes a foundation for future improvements in multi-agent underwater acoustic networks.

C. Extensions and future work

Building on this proof-of-concept, several promising directions emerge for future research.

In particular, we plan to extend this framework to more complex and challenging scenarios, where we envision RL approaches to outperform traditional methods. In fact, adding multiple dynamic acoustic sources as well as coordinating larger AUV fleets would push the boundaries of what is computationally tractable for real-time analytical approaches. Conversely, RL concentrates the computational burden to the training phase, and is otherwise efficient during inference. These scenarios represent the real-world complexity of underwater surveillance and would showcase the adaptability of RL in handling dynamic environments.

Beyond these environmental complexities, an important theoretical and practical investigation should involve comparing centralized against decentralized training approaches for this task. While in this work we focused on parameter sharing for simplicity, future work could explore decentralized training. This could validate or challenge existing claims in the MARL literature about effectiveness of coordination in communication-constrained environments. Additionally, given the constraints of underwater communication, developing MARL systems specifically trained to handle severely degraded communication (success rates of $10 - 20\%$) would be valuable for real-world deployments. Our results suggest resilience down to 40% success rates, but specialized training could potentially push this boundary further. Furthermore, creating heterogeneous agents represents another promising direction. This could be achieved by adding agent indicator nodes to the input layer as suggested by [29], allowing agents to develop specialized behaviors while still benefiting from shared learning experiences.

Finally, exploring more sophisticated neural network architectures that directly process OG maps could improve performance. Using a convolutional neural network (CNN) feature extractor might help agents distinguish between different environmental characteristics and adapt their behaviors accordingly, potentially enabling them to switch seamlessly between detection and localization tasks.

REFERENCES

[1] A. Maguer, R. Dymond, A. Grati, R. Stoner, P. Guerrini, L. Troiano, and A. Alvarez, "Ocean gliders payloads for persistent maritime surveillance and monitoring," in *2013 OCEANS-San Diego*. IEEE, 2013, pp. 1–8.

[2] Y. Lin, J. Hsiung, R. Piersall, C. White, C. G. Lowe, and C. M. Clark, "A multi-autonomous underwater vehicle system for autonomous tracking of marine life," *Journal of Field Robotics*, vol. 34, no. 4, pp. 757–774, 2017.

[3] E. T. Küsel, T. Munoz, M. Siderius, D. K. Mellinger, and S. Heimlich, "Marine mammal tracks from two-hydrophone acoustic recordings made with a glider," *Ocean Science*, vol. 13, no. 2, pp. 273–288, 2017.

[4] J. Li, G. Zhang, C. Jiang, and W. Zhang, "A survey of maritime unmanned search system: Theory, applications and future directions," *Ocean Engineering*, vol. 285, p. 115359, 2023.

[5] L. Wang, D. Zhu, W. Pang, and Y. Zhang, "A survey of underwater search for multi-target using multi-auv: Task allocation, path planning, and formation control," *Ocean Engineering*, vol. 278, p. 114393, 2023.

[6] D. I. Havelock, S. Kuwano, and M. Vorländer, *Handbook of signal processing in acoustics*. Springer, 2008, vol. 1.

[7] C. H. Sherman and J. L. Butler, *Transducers and arrays for underwater sound*. Springer, 2007, vol. 4.

[8] G. Ferri, M. V. Jakuba, A. Mondini, V. Mattoli, B. Mazzolai, D. R. Yoerger, and P. Dario, "Mapping multiple gas/odor sources in an uncontrolled indoor environment using a bayesian occupancy grid mapping based method," *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 988–1000, 2011.

[9] G. Ferri, A. Tesei, P. Stinco, and K. D. LePage, "A Bayesian Occupancy Grid Mapping Method for the Control of Passive Sonar Robotics Surveillance Networks," in *OCEANS 2019 - Marseille*, Jun. 2019, pp. 1–9.

[10] G. Ferri, P. Stinco, A. Tesei, and K. LePage, "An AUV Cooperative Target Localisation Strategy with Bearing-Only Measurements Based on Bayesian Occupancy Grid Mapping," in *Global Oceans 2020: Singapore – U.S. Gulf Coast*, Oct. 2020, iSSN: 0197-7385.

[11] S. Hammel, P. Liu, E. Hilliard, and K. Gong, "Optimal observer motion for localization with bearing measurements," *Computers & Mathematics with Applications*, vol. 18, no. 1-3, pp. 171–180, 1989.

[12] Y. Hsu, H. Wu, K. You, and S. Song, "A selected review on reinforcement learning based control for autonomous underwater vehicles," Nov. 2019, arXiv:1911.11991 [cs].

[13] B. Yoo and J. Kim, "Path optimization for marine vehicles in ocean currents using reinforcement learning," *Journal of Marine Science and Technology*, vol. 21, no. 2, pp. 334–343, Jun. 2016.

[14] L. Zheng, M. Liu, S. Zhang, and J. Lan, "A Novel Sensor Scheduling Algorithm Based on Deep Reinforcement Learning for Bearing-Only Target Tracking in UWSNs," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 4, pp. 1077–1079, Apr. 2023.

[15] Z. Chu, F. Wang, T. Lei, and C. Luo, "Path Planning Based on Deep Reinforcement Learning for Autonomous Underwater Vehicles Under Ocean Current Disturbance," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 108–120, Jan. 2023.

[16] J. Shi, J. Fang, Q. Zhang, Q. Wu, B. Zhang, and F. Gao, "Dynamic Target Tracking of Autonomous Underwater Vehicle Based on Deep Reinforcement Learning," *Journal of Marine Science and Engineering*, vol. 10, no. 10, Oct. 2022.

[17] F. Hoffmann, A. Charlish, M. Ritchie, and H. Griffiths, "Sensor Path Planning Using Reinforcement Learning," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, Jul. 2020.

[18] Z. Yang, J. Du, Z. Xia, C. Jiang, A. Benslimane, and Y. Ren, "Secure and Cooperative Target Tracking via AUV Swarm: A Reinforcement Learning Approach," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2021.

[19] G. Ferri, R. Grasso, E. Camossi, A. Faggiani, K. Bereta, M. Vodas, D. Kladis, D. Zissis, and K. D. LePage, "Developing a robotic hybrid network for coastal surveillance: the infore experience," in *OCEANS 2021: San Diego–Porto*. IEEE, 2021, pp. 1–10.

[20] G. Ferri, R. Grasso, A. Faggiani, F. de Rosa, E. Camossi, A. Grati, P. Stinco, A. Tesei, R. Been, K. D. LePage *et al.*, "A hybrid robotic network for maritime situational awareness: Results from the infore22 sea trial," in *OCEANS 2022, Hampton Roads*. IEEE, 2022, pp. 1–10.

[21] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[22] S. Thrun, W. Burgard, D. Fox *et al.*, *Probabilistic robotics, vol. 1*. MIT press Cambridge, 2005.

[23] M. V. Jakuba, "Stochastic mapping for chemical plume source localization with application to autonomous hydrothermal vent discovery," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[24] A. N. Bishop, B. Fidan, B. D. Anderson, K. Dogancay, and P. N. Pathirana, "Optimality Analysis of Sensor-Target Geometries in Passive Localization: Part 1 - Bearing-Only Localization," in *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. Melbourne, Australia: IEEE, 2007, pp. 7–12.

[25] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction*, second edition ed., ser. Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press, 2020.

[26] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforcement Learning*, 2018.

[27] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Mar. 2023.

[28] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," Mar. 2019, arXiv:1707.09183 [cs].

[29] J. K. Terry, N. Grammel, S. Son, B. Black, and A. Agrawal, "Revisiting Parameter Sharing in Multi-Agent Deep Reinforcement Learning," Oct. 2023, arXiv:2005.13625 [cs, stat].

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017, arXiv:1707.06347 [cs].

[31] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust Region Policy Optimization," Apr. 2017, arXiv:1502.05477 [cs].

[32] 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, no. 268, pp. 1–8, 2021.

[33] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer, 2007.