sensors

MDPI

Article

Reinforcement Learning-Based Multi-AUV Adaptive Trajectory Planning for Under-Ice Field Estimation

Chaofeng Wang [1], Li Wei [1], Zhaohui Wang [1,*], Min Song [2] and Nina Mahmoudian [3]

[1] Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931, USA; cwang8@mtu.edu (C.W.); liwei@mtu.edu (L.W.)
[2] Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA; msong6@stevens.edu
[3] Department of Mechanical Engineering-Engineering Mechanics, Michigan Technological University, Houghton, MI 49931, USA; ninam@mtu.edu
* Correspondence: zhaohuiw@mtu.edu; Tel.: +1-906-487-2054

Received: 1 October 2018; Accepted: 4 November 2018; Published: 9 November 2018

Abstract: This work studies online learning-based trajectory planning for multiple autonomous underwater vehicles (AUVs) to estimate a water parameter field of interest in the under-ice environment. A centralized system is considered, where several fixed access points on the ice layer are introduced as gateways for communications between the AUVs and a remote data fusion center. We model the water parameter field of interest as a Gaussian process with unknown hyper-parameters. The AUV trajectories for sampling are determined on an epoch-by-epoch basis. At the end of each epoch, the access points relay the observed field samples from all the AUVs to the fusion center, which computes the posterior distribution of the field based on the Gaussian process regression and estimates the field hyper-parameters. The optimal trajectories of all the AUVs in the next epoch are determined to maximize a long-term reward that is defined based on the field uncertainty reduction and the AUV mobility cost, subject to the kinematics constraint, the communication constraint and the sensing area constraint. We formulate the adaptive trajectory planning problem as a Markov decision process (MDP). A reinforcement learning-based online learning algorithm is designed to determine the optimal AUV trajectories in a constrained continuous space. Simulation results show that the proposed learning-based trajectory planning algorithm has performance similar to a benchmark method that assumes perfect knowledge of the field hyper-parameters.

Keywords: underwater communication networks; under-ice exploration; field estimation; AUVs; adaptive trajectory planning; reinforcement learning

1. Introduction

Autonomous underwater vehicles (AUVs) are attractive platforms for remote underwater exploration and monitoring, e.g., seafloor mapping [1,2] and under-ice hydrographic observations [3]. The AUV trajectories can be determined prior to the deployment or adjusted online based on recent observations. Given the high deployment cost of AUVs, adaptive trajectory planning is desirable for the collection of the "best" data over scalar or vector fields that vary in a range of spatial and temporal scales [4–7]. This work studies the online adaptive trajectory planning of multiple AUVs in the under-ice environment for the estimation of a scalar water parameter field of interest.

Adaptive trajectory planning has been under extensive investigation in terrestrial robotic networks. Both myopic solutions and non-myopic solutions have been proposed in different contexts (e.g., mapping, physical phenomenon monitoring and field maxima tracking). In myopic solutions, the trajectories (or sampling positions) in the next time step are determined to optimize some predefined performance metric, such as the reduction of the field estimation error [8,9] and the conditional information entropy [10]. Both a single robot [9] and multiple robots [8,10] have been considered in existing solutions. In non-myopic solutions, the trajectories are determined to optimize a long-term performance. Specifically, the trajectory planning is often formulated as a partially observable MDP, in which the state is formed by the robot status and the collected field knowledge, and the action corresponds to the waypoints to travel or parameterized continuous trajectories [11–14]. The partially observable Markov decision process is typically solved via Bayesian optimization methods and Monte Carlo tree search [11,13,14]. Due to the high computational complexity of Bayesian optimization and Monte Carlo tree search, existing solutions mainly consider a single robot. For multiple robots, the objective function can be designed to have a certain structure (e.g., local submodular) to make the problem computationally tractable [12]. In both myopic and non-myopic solutions, the field is typically modeled as a Gaussian process, and the field covariance function is assumed known a priori, which can be estimated based on historical measurements.

Relative to terrestrial robotic networks, studies on adaptive trajectory planning of underwater AUVs have been very limited. Existing solutions have been developed for feature tracking (e.g., tracking of thermal fronts) at relatively small spatial scales using gradient climbing strategies and for coverage of a field of interest at large spatial scales. This work focuses on the latter applications, such as mapping of the temperature, salinity, flow or biological variables in a two- or three-dimensional water region. To achieve synoptic coverage of the field of interest, coordination among the AUVs via wireless communications is critical to keep the AUVs appropriately distributed in space according to the field spatial variability [4]. Thus far, satellite links are the most common solution for information exchange between an AUV and the control center when the AUV surfaces every few hours (e.g., 2 h). Such a large communication latency prevents timely uploading of data from AUVs and timely update of AUV trajectories, especially in the presence of strong currents.

For the estimation of an unknown water field, due to the communication constraint, existing works have mainly focused on offline AUV trajectory planning. Specifically, the unknown field is typically represented as an uncertainty field described by a Gaussian process with known covariance, and the AUV trajectories are determined to maximize the uncertainty reduction (i.e., the collected information) subject to the constraints on, e.g., the primary motion, anti-curling, vicinity, communications and obstacle avoidance. The AUV trajectory is represented by a series of waypoints on a discrete grid in the region of interest [5,15,16] or parameterized by a restricted number of parameters [4]. For a single AUV, the branch and bound algorithm [5], the recursive greedy algorithm [15] and the sampling-based redundant roadmap method [16] have been applied to find the near-optimal solution. In the presence of multiple AUVs, the AUV trajectories can be jointly computed prior to the deployment by the control center [5].

Although with a human in the loop, the trajectory of an AUV can be adjusted during its deployment via satellite links once it surfaces [4], research on the machinery and online adaptation of AUV trajectories to maximize the collected information has been limited. Two existing myopic solutions to adaptive AUV trajectory planning are described in the following. In References [17,18], to prevent the existence of a single point of failure and to achieve the scalability to the number of AUVs, a decentralized strategy for multi-AUV sampling and patrolling is developed. The region of interest is dynamically partitioned into multiple Voronoi cells according to the AUV locations. Based on the samples collected by all or neighboring AUVs in the past, each AUV computes the next visiting point within its own Voronoi cell in a myopic way to maximize the amount of information to be collected. The above operation requires information exchange among the AUVs. Field experimental results were presented with two AUVs exchanging information via acoustic communications [19] and three surface vehicles exchanging information via radio frequency links [17]. In the above works, the unknown field is modeled as a Gaussian process with known covariance function. Such a priori field knowledge, however, could be difficult to obtain in practice, particularly in the presence of large field dynamics.

This work studies the non-myopic adaptive trajectory planning of multiple AUVs in the under-ice environment for the estimation of a water parameter field of interest. In particular, we consider a centralized system as illustrated in Figure 1, where the fixed access points on the ice layer serve as gateways for communications between the AUVs and a remote data fusion center. The AUV trajectories are determined by the fusion center on a time epoch-by-epoch basis based on the samples collected in the past epochs. Different from the open-water AUV network in which the AUVs can access satellite links wherever they surface, the under-ice AUVs can only communicate with the fusion center when they are within the communication range of an access point.

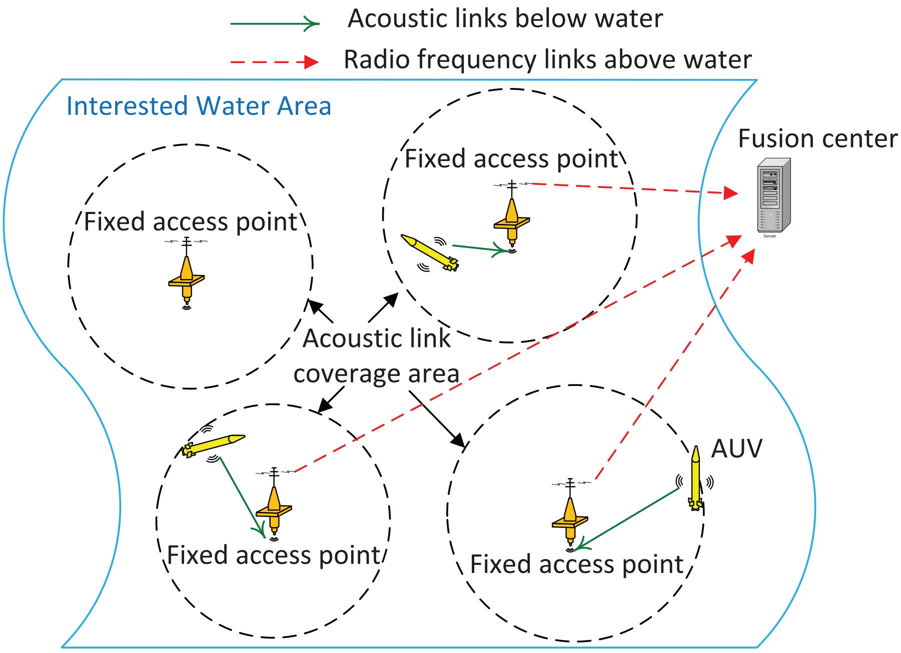


Figure 1. An illustration of the system layout with three autonomous underwater vehicles (AUVs) and four access points.

In this work, the water parameter field of interest is modeled as a Gaussian process with an unknown covariance function specified by hyper-parameters [20]. At the end of each epoch, the field samples collected by the AUVs are relayed via the access points to the fusion center where the field hyper-parameters are estimated via the maximum likelihood method [20], and the posterior field distribution and the field uncertainty are computed via the Gaussian process regression [21]. The AUV trajectories in the next epoch will then be determined by the fusion center based on the current system state including the current positions of all the AUVs and the field knowledge, with the aim of maximizing a long-term system reward that is defined based on the field uncertainty reduction and the AUV mobility cost. The AUV trajectories are expected to satisfy several practical constraints, including the kinematics constraint, the constraint on communications and the constraint of being within the area of interest.

The adaptive trajectory planning problem is formulated as a MDP [22] with a constrained continuous action space. A reinforcement learning-based method is designed for online learning of the optimal action, i.e., the trajectories of all the AUVs, which satisfies the constraints. The knowledge for determining the optimal trajectories in each epoch is first obtained by transferring the historical knowledge of determining the trajectories in the previous epoch and then is further adjusted based on the newly-collected reward. The proposed reinforcement learning-based trajectory planning algorithm is validated using simulated two-dimensional (2D) fields. The simulation results show that the proposed algorithm achieves performance similar to a benchmark method that assumes perfect knowledge of the field hyper-parameters.

The main contributions of this work are summarized in the following.

- The developed algorithm is non-myopic and for multiple AUVs, while existing works consider either non-myopic planning for a single vehicle [11–14] or myopic planning for multiple vehicles [8,10,17,18]. To tackle the high computational cost of non-myopic multi-vehicle planning, instead of using Monte Carlo tree search, we employ a learning algorithm which can be implemented via parallel computation. To further speed up the convergence of the planning algorithm, the decision-making strategy is adjusted on the fly by transferring the knowledge learned in previous epochs.
- By introducing a deep deterministic policy gradient (DDPG) method, the developed algorithm allows adaptive trajectory planning in a continuous action space, while many existing works consider either a discrete action space or a finite number of pre-determined trajectory patterns [10,12–14].
- This work performs the online learning of the field hyper-parameters according to the maximum likelihood principle, while many existing works assume those parameters known a priori [5,8,15,17].

Remark 1. *The proposed solution for adaptive trajectory planning can be directly implemented on AUVs that are equipped with an autonomy package, appropriate sensors and an acoustic communication unit. Specifically, with the support of wireless communications between AUVs and a remote fusion center, the field samples collected by AUVs can be sent periodically to the fusion center for centralized processing and planning for the AUV trajectories in the near future. To ensure the communication reliability, both forward error control via error correction coding and backward error control via automatic repeat request can be applied [23]. Different from mission adaptive systems where the AUV uses an on-board decision-making system for online mission planning [24], the decision-making in this work occurs at the fusion center, which can have access to sufficient computational and storage resources. The AUVs are mainly responsible for navigation according to the determined trajectories by the fusion center, taking field samples around waypoints and sending the field samples to the fusion center via gateways (access points).*

Remark 2. *The proposed solution has several advantages for field missions. First, the proposed solution does not require prior knowledge about the field spatial variability, benefiting from the online estimation of the field spatial correlation parameters (the hyper-parameters). Secondly, the trajectory obtained in the proposed solution is not constrained to a set of predetermined discrete points, rather it can be an arbitrary path in the continuous water region of interest. Thirdly, the proposed trajectory planning algorithm is non-myopic, benefiting from the relatively large computational power of the remote fusion center.*

The rest of this paper is organized as follows. The system model is presented in Section 2. The online adaptive trajectory planning is formulated into an optimization problem in Section 3. A reinforcement learning-based algorithm is developed to solve the optimization problem in Section 4. Evaluation of the proposed algorithm is included in Section 5. Conclusions are drawn in Section 6.

Notation 1. *Bold upper case letters and lower case letters are used to denote matrices and column vectors, respectively.* $\mathbf{A}^{\mathrm{T}}$ *denotes the transpose of matrix* $\mathbf{A}$. $[\mathbf{a}]_m$ *denotes the m-th element of vector* $\mathbf{a}$. *For* $\mathbf{a} = [a_1, a_2, \cdots, a_n]^{\mathrm{T}}$, $\|\mathbf{a}\|_2 := \left(a_1^2 + a_2^2 + \cdots + a_n^2\right)^{1/2}$ *denotes the Euclidean norm (also known as the two-norm) of* $\mathbf{a}$. $[\mathbf{A}]_{i,j}$ *denotes the* (i,j)*-th element of matrix* $\mathbf{A}$. $\mathbf{A}^{-1}$ *denotes the inverse matrix of* $\mathbf{A}$. $\det(\mathbf{A})$ *denotes the determinant of matrix* $\mathbf{A}$. $\mathbf{I}_N$ *denotes an identity matrix of size* $N \times N$. $|\mathcal{A}|$ *denotes the cardinality of set* $\mathcal{A}$.

2. System Model

2.1. System Description

The system under consideration consists of multiple AUVs, several fixed access points and a remote fusion center. Denote the AUV index set as $\mathcal{M} = \{1, 2, ..., M\}$. The AUVs are equipped with sensors and acoustic communication units. They take field measurements at discrete sampling locations as they navigate along their trajectories. A total number of N_{AP} access points are placed at fixed locations, serving as gateways for communications between AUVs and the fusion center. The acoustic links are used for underwater information exchange between AUVs and access points, and the radio frequency links are used for in-air information exchange between access points and the fusion center. With the access points, the data collected by AUVs can be sent to the fusion center for centralized processing. An illustration of the system layout with three AUVs and four access points is shown in Figure 1.

We consider a water parameter field within an area of interest $\mathcal{X}_{\text{area}} \subset \mathbb{R}^{\text{Dim}}$, with Dim $= 2$ or 3 denoting the dimension of the area. The water parameter field is represented by $z(\mathbf{x})$, with $\mathbf{x} \in \mathcal{X}_{\text{area}}$. Within the area $\mathcal{X}_{\text{area}}$, a discrete set of target points $\mathcal{X}$ is selected based on application requirements. The overall system mission is to minimize the field uncertainty (namely, the estimation inaccuracy) over those target points.

The system operates on an epoch-by-epoch basis. An illustration of the system operation within each epoch is shown in Figure 2. The trajectories for all the AUVs in the ℓ-th epoch are determined at the end of the $(\ell-1)$-th epoch based on the collected field samples. For each AUV, the planned trajectory consists of K waypoints. It takes one time slot for the AUV to travel from one waypoint to the next. Each AUV collects field measurements around the waypoints. After the AUV reaches the last waypoint in the current epoch, it transmits the observed data and the corresponding sampling locations to the nearest access point via underwater acoustic links. The access points then relay all the information to the fusion center via radio frequency links. The fusion center estimates the field values at target points in $\mathcal{X}$ based on all the observations and updates the estimation of the field spatial correlation parameters. It then determines the trajectories for all the AUVs in the next epoch and transmits via access points the planned trajectories to all the AUVs. At the end of the ℓ-th epoch, all the AUVs receive their planned trajectories in the next epoch.

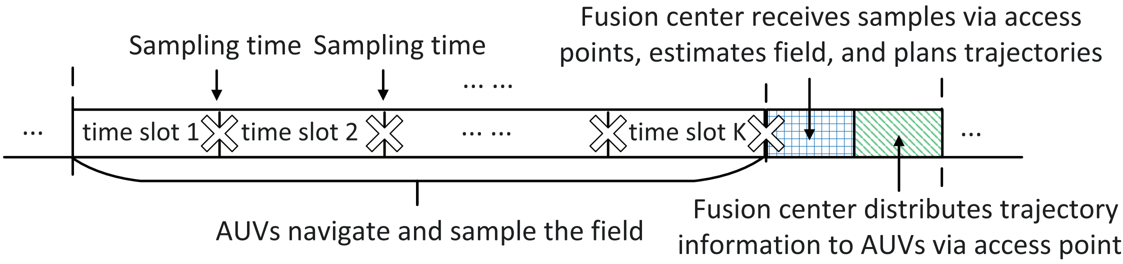

Sampling time
Sampling time
Fusion center receives samples via access points, estimates field, and plans trajectories
··· ···
···
time slot 1
time slot 2
··· ···
time slot K
···
AUVs navigate and sample the field
Fusion center distributes trajectory information to AUVs via access points

Figure 2. System operation within an epoch. The AUV samples at the end of each time slot. There are K time slots within an epoch for AUV navigation and sampling.

2.2. *Autonomous Underwater Vehicles Trajectory Modeling*

For the i-th AUV and the ℓ-th epoch, the planned trajectory that consists of K waypoints is represented as $\check{\mathcal{X}}_i(\ell) := \{\check{\mathbf{x}}_{i,\ell}(1), \check{\mathbf{x}}_{i,\ell}(2), \cdots, \check{\mathbf{x}}_{i,\ell}(K)\}$. The last waypoint in the ℓ-th epoch is the AUV starting position in the $(\ell+1)$-th epoch, namely $\check{\mathbf{x}}_{i,\ell}(K) = \check{\mathbf{x}}_{i,\ell+1}(0)$.

Based on the system description, the AUV trajectories need to satisfy some practical constraints. We consider three constraints in the following.

- Kinematics constraint: Due to the limited travel speed of an AUV, the distance between any two consecutive waypoints for each AUV is constrained as:

$$\|\check{\mathbf{x}}_{i,\ell}(k) - \check{\mathbf{x}}_{i,\ell}(k+1)\|_2 \leq \kappa_{\text{up}}, \qquad 0 \leq k \leq K-1, \; \forall i \in \mathcal{M} \tag{1}$$

where κ_{up} is the maximal distance that an AUV can travel within one time slot.

- Communication constraint: For each epoch, since the AUV needs to send its field samples to an access point when it arrives at the last waypoint, the AUV should be within the communication range of at least one of the N_{AP} access points, namely,

$$\|\check{\mathbf{x}}_{i,\ell}(K) - \mathbf{x}_{\mathrm{AP}}^{(j)}\|_2 \leq \kappa_{\mathrm{comm}}, \qquad \exists j \in \mathcal{I}_{\mathrm{AP}}, \, \forall i \in \mathcal{M} \tag{2}$$

  where $\mathcal{I}_{\mathrm{AP}} := \{1, 2, \cdots, N_{\mathrm{AP}}\}$ is the access point index set, $\mathbf{x}_{\mathrm{AP}}^{(j)}$ is the location of the j-th access point and κ_{comm} is the communication range that ensures reliable transmission between an access point and an AUV.

- Sensing area constraint: All the AUVs should stay within the area of interest, namely,

$$\check{\mathbf{x}}_{i,\ell}(k) \in \mathcal{X}_{\mathrm{area}}, \qquad 1 \leq k \leq K, \, \ell \geq 0, \, \forall i \in \mathcal{M}. \tag{3}$$

Due to the water current and obstacles, the AUVs may not arrive at each planned waypoint exactly. In field missions, the "arrival" is defined based on the allowable location discrepancy specified in the mission file. For the i-th AUV, we model the k-th sampling location within the ℓ-th epoch as:

$$\mathbf{x}_{i,\ell}(k) = \check{\mathbf{x}}_{i,\ell}(k) + \mathbf{e}_{i,\ell}(k), \tag{4}$$

where $\mathbf{e}_{i,\ell}(k) \in \mathbb{R}^{\mathrm{Dim}}$ is a noise vector, which describes the location inaccuracy, and each of its elements is assumed following an independent and identical uniform distribution $\mathcal{U}(-\epsilon, \epsilon)$ with $\epsilon \ll \kappa_{\mathrm{comm}}$ and $\epsilon \ll \kappa_{\mathrm{up}}$ [25].

2.3. Unknown Field Modeling

We model the unknown field $z(\mathbf{x})$ as a Gaussian process with zero mean,

$$z(\mathbf{x}) \sim \mathcal{GP}(0, \mathcal{K}(\mathbf{x}, \mathbf{x}')), \qquad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}_{\mathrm{area}} \tag{5}$$

where $\mathcal{K}(\mathbf{x}, \mathbf{x}')$ is the covariance function of the field values at locations $\mathbf{x}$ and $\mathbf{x}'$.

There are various types of covariance functions that can be employed to describe the field spatial correlation [20]. In this work, we consider the squared exponential covariance function,

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\big(-(\mathbf{x} - \mathbf{x}')^{\mathrm{T}} \mathbf{\Lambda}^{-2} (\mathbf{x} - \mathbf{x}')\big), \tag{6}$$

where $\mathbf{\Lambda} = \mathrm{diag}([d_1, \cdots, d_{\mathrm{Dim}}])$ with $\mathrm{Dim} = 2$ or 3 being the dimension of the water area and d_i being the distance scale that determines the field spatial correlation at two locations and σ_f^2 is the signal variance. The matrix $\mathbf{\Lambda}$ and σ_f^2 are referred to as field hyper-parameters. In this work, we consider the lack of prior knowledge about the field hyper-parameters $\boldsymbol{\theta}_{\mathrm{hyper}} := \{\sigma_f^2, \mathbf{\Lambda}\}$ and develop a method to estimate the hyper-parameters based on the sequentially-collected field samples by AUVs.

For the i-th AUV, the field observation at the k-th sampling location in the ℓ-th epoch is described as:

$$y_{i,\ell}(k) = z(\mathbf{x}_{i,\ell}(k)) + w_{i,\ell}(k), \tag{7}$$

where $w_{i,\ell}(k)$ is the observation noise and assumed following a Gaussian distribution $\mathcal{N}(0, \sigma_w^2)$.

3. Problem Formulation for Adaptive Trajectory Planning

In this section, we develop a mathematical model for the field estimation. The trajectory planning for multiple AUVs is then formulated as a constrained optimization problem, to maximize the field estimation accuracy while minimizing the AUV mobility cost.

3.1. Gaussian Process Regression for Field Estimation

Based on the field observations, we use the Gaussian process regression to estimate the field values at target locations and estimate the field hyper-parameters $\boldsymbol{\theta}_{\text{hyper}}$ using the maximum likelihood method.

Stack the observations of all the AUVs in the ℓ-th epoch into a vector $\mathbf{y}_\ell$ of length MK. Stack all the observations up to the ℓ-th epoch into a long vector $\mathbf{y}_{\text{cum},\ell} := [\mathbf{y}_0^{\text{T}}, \cdots, \mathbf{y}_\ell^{\text{T}}]^{\text{T}}$. The observation vector follows a Gaussian distribution,

$$\mathbf{y}_{\text{cum},\ell} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_\ell), \tag{8}$$

where $\mathbf{C}_\ell$ is the covariance matrix. The (i,j)-th element of $\mathbf{C}_\ell$ is computed as:

$$[\mathbf{C}_\ell]_{i,j} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}\sigma_w^2, \tag{9}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the sampling location of the i-th element and the j-th element in $\mathbf{y}_{\text{cum},\ell}$, respectively, and δ_{ij} denotes the Dirac delta function.

We stack the field values at target locations within the set $\mathcal{X}$ into a vector $\mathbf{z}$ of length $|\mathcal{X}|$. Its posterior distribution based on the observations collected up to the ℓ-th epoch is:

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell), \tag{10}$$

with:

$$\boldsymbol{\mu}_\ell = \mathbf{C}_{\mathbf{z},\mathbf{y}_{\text{cum},\ell}} \mathbf{C}_\ell^{-1} \mathbf{y}_{\text{cum},\ell}, \tag{11}$$

$$\boldsymbol{\Sigma}_\ell = \mathbf{C}_{\mathbf{z}} - \mathbf{C}_{\mathbf{z},\mathbf{y}_{\text{cum},\ell}} \mathbf{C}_\ell^{-1} \mathbf{C}_{\mathbf{z},\mathbf{y}_{\text{cum},\ell}}^{\text{T}}, \tag{12}$$

where $\mathbf{C}_{\mathbf{z}}$ is the covariance matrix of $\mathbf{z}$, its (i,j)-th element is obtained as $[\mathbf{C}_{\mathbf{z}}]_{i,j} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ with $\mathbf{x}_i$ and $\mathbf{x}_j$ being the location of the i-th and the j-th element in $\mathbf{z}$, respectively, $\mathbf{C}_{\mathbf{z},\mathbf{y}_{\text{cum},\ell}}$ is the covariance matrix between $\mathbf{z}$ and $\mathbf{y}_{\text{cum},\ell}$ and its (i,j)-th element is obtained as $[\mathbf{C}_{\mathbf{z},\mathbf{y}_{\text{cum},\ell}}]_{i,j} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ with $\mathbf{x}_i$ and $\mathbf{x}_j$ being the location of the i-th element in $\mathbf{z}$ and the j-th element in $\mathbf{y}_{\text{cum},\ell}$, respectively.

Based on the observation vector $\mathbf{y}_{\text{cum},\ell}$, the field hyper-parameters $\boldsymbol{\theta}_{\text{hyper}} = \{\sigma_f^2, \boldsymbol{\Lambda}\}$ can be estimated by maximizing the log likelihood function [26],

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{hyper}} &= \max_{\boldsymbol{\theta}_{\text{hyper}}} \ln f(\mathbf{y}_{\text{cum},\ell}; \boldsymbol{\theta}_{\text{hyper}}), \\ &= \max_{\boldsymbol{\theta}_{\text{hyper}}} \left\{ -\frac{1}{2} \mathbf{y}_{\text{cum},\ell}^{\text{T}} \mathbf{C}_\ell^{-1} \mathbf{y}_{\text{cum},\ell} - \frac{1}{2} \log \det(\mathbf{C}_\ell) \right\},\end{aligned} \tag{13}$$

where $f(\mathbf{y}_{\text{cum},\ell}; \boldsymbol{\theta}_{\text{hyper}}) = \mathcal{N}(\mathbf{0}, \mathbf{C}_\ell)$ is the probability density function (also known as the likelihood function) of vector $\mathbf{y}_{\text{cum},\ell}$ (cf. Equation (8)) and $\mathbf{C}_\ell$ is related to the hyper-parameters $\boldsymbol{\theta}_{\text{hyper}}$ through Equations (6) and (9). The optimization problem Equation (13) can be solved using a quasi-Newton method, i.e., the L-BFGS-B (limited memory Broyden–Fletcher–Goldfarb–Shannon algorithm for bound constrained optimization) method [27]. The optimization Equation (13) is used to estimate the field spatial correlation parameters, i.e., the field hyper-parameters, based on the collected field measurements at the end of each epoch. The estimated field spatial correlation parameters will then be used for trajectory planning.

3.2. Problem Formulation for Optimal Trajectory Planning

The field uncertainty can be computed based on the posterior distribution of $\mathbf{z}$. Specifically, we define $\mathbf{u}_\ell := \text{diag}(\boldsymbol{\Sigma}_{\ell-1})$ to describe the uncertainty of all the target points in $\mathcal{X}$ based on the observations up to the $(\ell-1)$-th epoch. Denote $\mathbf{p}_\ell := \{\mathbf{x}_{1,\ell}(0), \mathbf{x}_{2,\ell}(0), \cdots, \mathbf{x}_{M,\ell}(0)\}$ as the locations of all the AUVs at the beginning of the ℓ-th epoch. Denote $\mathbf{s}(\ell) := \{\mathbf{p}_\ell, \mathbf{u}_\ell\}$ as the system state at the beginning of the ℓ-th epoch. Denote $\mathbf{a}(\ell)$ as the action in the ℓ-th epoch, which consists of the planned waypoints for all the AUVs in the ℓ-th epoch.

For the long-term AUV deployment, the AUV trajectories in all the future time epochs can be optimized to maximize the overall system performance. Mathematically, given the the randomness of the field values (cf. Equations (5) and (7)) and the uncertainty during the AUV navigation (cf. Equation (4)), the desired trajectories for all the AUVs can be determined to maximize the expected total discounted reward,

$$\max_{\{\mathbf{a}(\ell)\}_{\ell=0}^{\infty}} \mathbb{E}\left\{ \sum_{\ell=0}^{\infty} \gamma^{\ell} R(\mathbf{s}(\ell), \mathbf{a}(\ell)) \right\}, \tag{14}$$

where $\gamma \in (0,1]$ is a discount factor, $R(\mathbf{s}(\ell), \mathbf{a}(\ell))$ is an application-dependent reward function and $\mathbb{E}\{\cdot\}$ denotes the statistical expectation of a random variable. Here, the discount factor is introduced to give more preference to the reward in the near future. In this work, the reward function takes into account the field uncertainty reduction, the AUV mobility cost based on the planned trajectories and the constraints in Section 2.2, as defined in the following.

3.2.1. Reward Function

For the ease of exposition, denote the current state as $\mathbf{s} = \{\mathbf{p}, \mathbf{u}\}$ and the planned trajectories as $\mathbf{a}$. Denote the next state as $\mathbf{s}' = \{\mathbf{p}', \mathbf{u}'\}$. The reward, costs and penalties induced by action $\mathbf{a}$ under the current state $\mathbf{s}$ and the next state $\mathbf{s}'$ are in the following.

- Uncertainty reduction reward: Given the system mission objective of minimizing the field uncertainty over target locations in $\mathcal{X}$, the reward associated with the reduction of the field uncertainty by performing action $\mathbf{a}$ at the system state $\mathbf{s}$ is defined as:

  $$R_{\mathrm{U}}(\mathbf{s}, \mathbf{a}) := \frac{\alpha_R}{|\mathcal{X}|} \left(||\mathbf{u}||_1 - ||\mathbf{u}'||_1 \right), \tag{15}$$

  where $\alpha_{\mathrm{R}} > 0$ is a weighting factor (set as $\alpha_{\mathrm{R}} = 10$ in the simulation) and $||\mathbf{u}||_1$ is the summation of all the elements in $\mathbf{u}$, which describes the total estimation error over target locations.

- Trajectory cost: Notice that the AUV energy consumption increases with the travel distance and the turning angle. The mobility cost associated with action $\mathbf{a}$ is defined as:

  $$C_{\mathrm{T}}(\mathbf{a}) := \alpha_L L(\mathbf{a}) + \alpha_A A(\mathbf{a}), \tag{16}$$

  where $L(\mathbf{a})$ is the total distance of the planned trajectories based on $\mathbf{a}$, $A(\mathbf{a})$ is the total angle that the AUVs travel along the planned trajectories based on $\mathbf{a}$ and $\alpha_{\mathrm{L}} > 0$ and $\alpha_{\mathrm{A}} > 0$ are weighting factors and set as $\alpha_{\mathrm{L}} = 1 \times 10^{-3}$ and $\alpha_{\mathrm{A}} = 5 \times 10^{-2}$ in the simulation.

- Trajectory constraint penalty: The kinematics constraint in Equation (1) will be addressed in the algorithm design for solving the optimization problem in Equation (14) (to be clear in Section 4.2). The constraints in Equations (2) and (3) are tackled by introducing a penalty term into the objective function, where zero penalty is applied when both constraints are satisfied and an extremely large penalty is incurred when either of the two constraints cannot be satisfied. The constraint penalty is defined as:

  $$C_{\mathrm{P}}(\mathbf{a}) := \alpha_{\mathrm{p1}} I_1 + \alpha_{\mathrm{p2}} I_2, \tag{17}$$

  where α_{p1} and α_{p2} are positive values and I_1 and I_2 are indication functions for constraint Equations (2) and (3), respectively, which equal one if the corresponding constraint is not satisfied and zero otherwise.

The reward function in Equation (14) is then formulated as:

$$R(\mathbf{s}, \mathbf{a}) = R_{\mathrm{U}}(\mathbf{s}, \mathbf{a}) - C_{\mathrm{T}}(\mathbf{a}) - C_{\mathrm{P}}(\mathbf{a}). \tag{18}$$

3.2.2. Bellman Optimality Equation

Directly solving the optimization problem Equation (14) is often intractable due to the large action and state spaces. Instead, the Bellman optimality equation is used to obtain the optimal actions [22,28]. Denote $Q^*(\mathbf{s}, \mathbf{a})$ as the optimal expected reward by performing action $\mathbf{a}$ under the current state $\mathbf{s}$, which is also called the Q-value function. The Bellman optimality equation for the Q-value function is:

$$Q^*(\mathbf{s}, \mathbf{a}) = \mathbb{E}\left\{ R(\mathbf{s}, \mathbf{a}) + \gamma \max_{\mathbf{a}' \in \mathcal{A}} Q^*(\mathbf{s}', \mathbf{a}') \right\}, \tag{19}$$

where $\mathbf{s}'$ is the next state, $\mathbf{a}'$ is the action taken in the next state, $\mathcal{A}$ is the action space and the expectation $\mathbb{E}\{\cdot\}$ is performed with respect to the probability distribution of $\mathbf{s}'$ given $\mathbf{s}$ and $\mathbf{a}$. The optimal action $\mathbf{a}^*$ under the current state $\mathbf{s}$ can be obtained by maximizing the optimal expected reward,

$$\mathbf{a}^* = \arg\max_{\mathbf{a} \in \mathcal{A}} Q^*(\mathbf{s}, \mathbf{a}). \tag{20}$$

In practice, the optimal expected reward $Q^*(\mathbf{s}, \mathbf{a})$ is not directly available. In the next section, we present a reinforcement learning algorithm to approximate the Q-value function and to generate optimal actions.

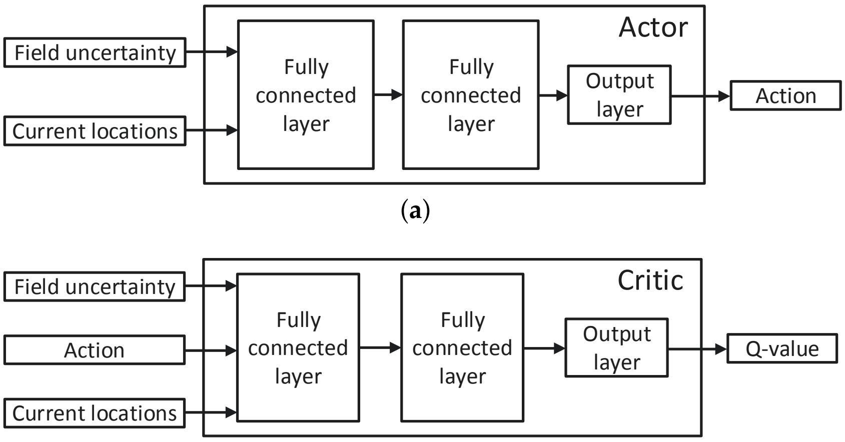
4. Reinforcement Learning-Based Adaptive Trajectory Planning

The proposed optimization problem Equation (14) is essentially an MDP if the field hyper-parameters are known a priori. It has a continuous action space and a continuous state space, which is generally difficult to solve. In this work, we adopt one type of reinforcement learning mechanism, the actor-critic method, to solve the proposed MDP [28]. Classic reinforcement learning algorithms can be categorized into two types. One type is the actor-based method where an actor is trained to generate optimal actions to maximize the Q-value function directly, while the other type is the critic-based method where a critic is trained to evaluate actions, i.e., to approximate the Q-value function and then select the action that yields the maximal Q-value. The actor-critic method combines the two classic types of reinforcement learning methods to achieve higher learning performance. Specifically, in actor-critic-based algorithms, the actor is trained to generate optimal actions, while the critic is trained to provide action evaluation, which helps the actor to improve its action generation strategy. Among various actor-critic-based algorithms, we employed the deep deterministic policy gradient (DDPG) algorithm [29], which deals with continuous action spaces and has high learning efficiency.

4.1. Deep Deterministic Policy Gradient Basics and Design

In the DDPG algorithm, an actor is represented by a neural network, which takes the system state $\mathbf{s}$ as the input and takes the optimal action $\mathbf{a}$ under the system state $\mathbf{s}$ as the output. A critic is also represented by a neural network, which takes the system state $\mathbf{s}$ and the action $\mathbf{a}$ as the inputs and takes a Q-value function $Q(\mathbf{s}, \mathbf{a})$ as the output. The Q-value $Q(\mathbf{s}, \mathbf{a})$ indicates the expected reward after taking action $\mathbf{a}$ under the system state $\mathbf{s}$. In the learning process, the actor network provides the action $\mathbf{a}$ to be executed under the state $\mathbf{s}$. After performing action $\mathbf{a}$, the corresponding reward $R(\mathbf{s}, \mathbf{a})$ can be obtained. Based on the obtained reward, the weights of the critic network are adjusted to better approximate the Q-value function $Q(\mathbf{s}, \mathbf{a})$. Then, the weights of the actor are adjusted using the policy gradient method such that the action obtained by the actor could result in higher expected reward, i.e., higher output of the critic network, which takes the output of the actor network as the input. For more details about the DDPG method, please refer to [29].

A critical issue of the DDPG method is how to design the actor and critic neural networks. In this work, the action for each AUV is parameterized by the moving distance within each time slot along each dimension of the area $\mathcal{X}_{\text{area}}$ for K time slots in total. The structural design of the actor and the critic is presented as follows. For the actor, as illustrated in Figure 3a, the current field uncertainty and current locations of all the AUVs go through two fully-connected layers with rectified linear units as the activation functions. The output layer takes the summation of the outputs of the second fully-connected layer and uses the tanh activation function to bound the elements of the action to be within $[-1, 1]$ (the use of the tanh function will be clear shortly). For the critic, as shown in Figure 3b, the field uncertainty and the current locations and actions of all the AUVs go through two fully-connected layers with rectified linear units as the activation functions. The output layer of the critic is the summation of the outputs of the second fully-connected layer. Consider the online application in this work. The structural design of the actor and critic networks should achieve learning efficiency to balance the system performance and the computational complexity.



(**a**)

(**b**)

Figure 3. Neural network design in deep deterministic policy gradient (DDPG). (**a**) The forward structure of the actor network; (**b**) the forward structure of the critic network.

In each training iteration, the weights of the actor and the critic networks are updated based on one iteration of the backpropagation algorithm [30].

4.2. Training for Actions under Constraints

Consider that the action for each AUV is described by the moving distance within each time slot along each dimension of the area and that each element in the output of the actor network in Figure 3a is constrained within $[-1, 1]$ through the employment of the tanh activation function. The kinematics constraint Equation (1) can be met through multiplying each element in the actor output by $\kappa_{\text{up}}/\sqrt{\text{Dim}}$ with $\text{Dim} = 2$ or 3 being the dimension of the area, such that the distance that an AUV travels in each time slot is guaranteed to be no greater than κ_{up}.

For the constraint Equations (2) and (3), we introduce a technique called experience replay used in the DDPG algorithm [29]. Experience replay is a technique to train the agent by transition samples drawn from a buffer, which consists of historical transitions in previous training experience. Denote the transition from one epoch to the next by a quadruple $(\mathbf{s}, \mathbf{a}, \mathbf{s}', R)$, which consists of the current state $\mathbf{s}$, the action $\mathbf{a}$ performed under the state $\mathbf{s}$, the next state $\mathbf{s}'$ by performing $\mathbf{a}$ based on $\mathbf{s}$, the immediate reward R collected by performing $\mathbf{a}$. All the historical transitions are stored in a replay buffer denoted by $\mathcal{B}$ and will be used for training the actor network and the critic network. Specifically, in each training iteration, the parameters in the actor and critic networks are adjusted by a mini-batch of samples of transitions, which are randomly taken from $\mathcal{B}$. With mini-batch samples, the neural networks can be trained more efficiently compared to the case with one sample per learning iteration. By training based on samples from the replay buffer rather than sequentially-obtained samples, the correlation among training samples can be removed, which improves the convergence performance of the neural networks.

To better learn the actions that satisfy the constraint Equations (2) and (3), we propose a modified DDPG (MDDPG) algorithm where two replay buffers are used for training. Denote $\mathcal{B}_1$ and $\mathcal{B}_2$ as two buffers where $\mathcal{B}_1$ consists of transitions whose actions satisfy the constraint Equations (2) and (3) and $\mathcal{B}_2$ consists of transitions whose actions do not. By randomly drawing a sufficient amount of transition samples from the buffers $\mathcal{B}_1$ and $\mathcal{B}_1$, the actor network and critic network can learn from both "good" and "bad" transition samples with high learning efficiency.

Denote the actor network as μ and the critic network as Q. The MDDPG algorithm to obtain the optimal trajectories with the known field hyper-parameters is described in Algorithm 1. In the training process, one training episode refers to a process that begins from the initial state when all the AUVs are at their initial positions (the beginning epoch) and ends at the final state when the whole sampling task is completed (the last epoch). In each epoch, action $\mathbf{a}$ is randomly adjusted based on the output of the actor according to Algorithm 2. Specifically, an exploration noise is added to generate travel distances, which yield potential higher rewards, and then, the travel distance along a randomly-selected dimension (e.g., longitude, latitude or depth) of the area will be set to zero to introduce trajectories that have less travel angles and also to prevent curling of AUVs. After performing action $\mathbf{a}$, the immediate reward R and the next state $\mathbf{s}'$ can be obtained based on Equation (18) and the Gaussian process regression. Instead of learning from the transition quadruple $\{\mathbf{s}, \mathbf{a}, \mathbf{s}', R\}$ immediately, the quadruple is stored in the replay buffers $\mathcal{B}_1$ or $\mathcal{B}_2$ based on the condition whether $\mathbf{a}$ satisfies the constraint Equations (2) and (3) or not. We will train the actor and the critic by a mini-batch of transitions drawn from the buffers $\mathcal{B}_1$ or $\mathcal{B}_2$. To ensure that the actor and the critic learn from sufficient samples in both $\mathcal{B}_1$ and $\mathcal{B}_2$, the transition samples from $\mathcal{B}_1$ and $\mathcal{B}_2$ are drawn, respectively, in two consecutive learning iterations. With the transition samples, the weights of the actor network are updated to minimize the prediction error of the Q-value function, and the weights of the critic network are updated to maximize the Q-value. The stochastic gradient descent method and the target networks are used to update those weights. The target networks are updated by learning the weights of the critic and actor networks with a relatively low learning rate. They provide the action evaluation and generation to update the critic network. The introduction of the target networks improves the learning stability [29]. At the end of the training iteration, the target critic and actor networks are updated.

When the field hyper-parameters are known a priori, the MDDPG algorithm can be used to learn the optimal actions offline. The obtained trajectories can serve as the performance upper bound for the proposed online learning strategy when the hyper-parameters are unknown prior to the system deployment.

Algorithm 1 Modified deterministic policy gradient (MDPPG) algorithm: MDDPG$(N_{\text{initial}}, N_{\text{episode}}, N_{\text{epoch}}, N_{\text{batch}}, \gamma, \tau, \eta, \beta^2, Q, \mathcal{W}_Q, \mu, \mathcal{W}_\mu, Q', \mathcal{W}_{Q'}, \mu', \mathcal{W}_{\mu'}, \boldsymbol{\theta}_{\text{hyper}}, \mathbf{s})$.

Input: Initial epoch N_{initial}, total training episodes N_{episode}, total epochs in an episode N_{epoch}, mini-batch size N_{batch}, discount factor γ, learning rate of the target networks τ, threshold value η, action adjust variance β^2, the critic network Q with its weights $\mathcal{W}_Q$, the actor network μ with its weights $\mathcal{W}_\mu$, the target critic network Q' with its weights $\mathcal{W}_{Q'}$, the target actor network μ' with its weights $\mathcal{W}_{\mu'}$, the field hyper-parameters $\boldsymbol{\theta}_{\text{hyper}}$ and the current system state $\mathbf{s}$
Output: Optimal action set $\mathcal{T}_{\text{opt}}$ for future epochs, the critic and actor networks Q and μ with weights $\mathcal{W}_Q$ and $\mathcal{W}_\mu$, the target critic and actor networks Q' and μ' with weights $\mathcal{W}_{Q'}$ and $\mathcal{W}_{\mu'}$

```
Initialize replay buffers $\mathcal{B}_1$ and $\mathcal{B}_2$. Set $i_{\text{iter}} = 0$ and $R_{\text{opt}} = -\infty$
for episode $= 1$ to $N_{\text{episode}}$ do
    Set $R_{\text{tot}} = 0$
    Set the initial state $\mathbf{s}$
    for epoch $= N_{\text{initial}}$ to $N_{\text{epoch}}$ do
        Perform action $\mathbf{a}_{\text{epoch}} = \text{RandomAdjust}(\mu(\mathbf{s}), \eta, \beta^2)$ according to Algorithm 2
        Obtain the immediate reward $R$ based on Equation (18), and observe the next state $\mathbf{s}'$
        if $\mathbf{a}_{\text{epoch}}$ satisfies the constraint Equations (2) and (3) then
            Store the transition sample $\{\mathbf{s}, \mathbf{a}_{\text{epoch}}, \mathbf{s}', R\}$ into the buffer $\mathcal{B}_1$
        else
            Store the transition sample $\{\mathbf{s}, \mathbf{a}_{\text{epoch}}, \mathbf{s}', R\}$ into the buffer $\mathcal{B}_2$
        if $i_{\text{iter}}$ mod 2 then
            Sample a random mini-batch of $N_{\text{batch}}$ transition sample from $\mathcal{B}_1$
        else
            Sample a random mini-batch of $N_{\text{batch}}$ transition sample from $\mathcal{B}_2$
        For the $i$-th transition sample from the mini-batch $\{\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, R_i\}$, $1 \leq i \leq N_{\text{batch}}$, compute $\xi_i \leftarrow R_i + \gamma Q'(\mathbf{s}'_i, \mu'(\mathbf{s}'_i))$ based on the weights $\mathcal{W}'_Q$ and $\mathcal{W}'_\mu$
        Update $\mathcal{W}_Q$ by minimizing the error: $L = \frac{1}{N_{\text{batch}}} \sum_i ||(\xi_i - Q(\mathbf{s}_i, \mathbf{a}_i))||_2$ using the backpropagation algorithm [30]
        Update $\mathcal{W}_\mu$ by the deterministic policy gradient theorem to maximize $Q(\mathbf{s}_i, \mu(\mathbf{s}_i))$ using the backpropagation algorithm [30]
        Update target networks: $\mathcal{W}_{Q'} \leftarrow \tau \mathcal{W}_Q + (1-\tau)\mathcal{W}_{Q'}$ and $\mathcal{W}_{\mu'} \leftarrow \tau \mathcal{W}_\mu + (1-\tau)\mathcal{W}_{\mu'}$
        $R_{\text{tot}} \leftarrow R_{\text{tot}} + R$, $i_{\text{iter}} \leftarrow i_{\text{iter}} + 1$, and $\mathbf{s} \leftarrow \mathbf{s}'$
    if $R_{\text{opt}} < R_{\text{tot}}$ then
        Set $\mathcal{T}_{\text{opt}} = \{\mathbf{a}_{\text{epoch}}\}_{\text{epoch}=N_{\text{initial}}}^{N_{\text{epoch}}}$ and $R_{\text{opt}} \leftarrow R_{\text{tot}}$
Return $(\mathcal{T}_{\text{opt}}, Q, \mu, \mathcal{W}_Q, \mathcal{W}_\mu, Q', \mu', \mathcal{W}_{Q'}, \mathcal{W}_{\mu'})$
```

Algorithm 2 Random action adjust: RandomAdjust$(\mathbf{a}, \eta, \beta^2)$.

Input: Action $\mathbf{a}$, threshold value η and action adjust variance β^2
Output: Adjusted action $\mathbf{a}$

```
Draw $\mathbf{w}$ from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \beta^2 \mathbf{I}_{\text{Dim}})$ where $\mathbf{I}_{\text{Dim}}$ is an identity matrix of size Dim with Dim $= 2$ or 3 being the dimension of the area
$\mathbf{a} \leftarrow \mathbf{a} + \mathbf{w}$ and drawn $u$ from a uniform distribution $\mathcal{U}[0, 1]$
if $u < \eta$ then
    Uniformly select and set to zero the travel distance along one dimension of the area, and adjust $\mathbf{a}$ accordingly
Clip elements in $\mathbf{a}$ to be within $[-\kappa_{\text{up}}/\sqrt{\text{Dim}}, \kappa_{\text{up}}/\sqrt{\text{Dim}}]$ to meet the kinematics constraint Equation (1).
Return $\mathbf{a}$
```

4.3. Online Learning for Trajectory Planning with Unknown Field Hyper-Parameters

In practice, perfect knowledge of the field hyper-parameters is often unavailable. In this work, we propose an online estimation of the hyper-parameters based on sequentially-collected field samples. An online trajectory planning algorithm, which incorporates the MDDPG algorithm and the online estimation of the field hyper-parameters is described in Algorithm 3. Specifically, after the collection of field samples in each epoch, the unknown field hyper-parameters in the covariance function Equation (6) can be estimated by solving the optimization problem Equation (13) based on all the field observations. After obtaining the estimated hyper-parameters, the previous learned knowledge, including the critic network $Q(\mathbf{s}, \mathbf{a})$ with its weights $\mathcal{W}_Q$, the actor network $\mu(\mathbf{s})$ with its weights $\mathcal{W}_\mu$, the target critic network $Q'(\mathbf{s}, \mathbf{a})$ with its weights $\mathcal{W}_{Q'}$ and the target actor network $\mu'(\mathbf{s})$ with its weights $\mathcal{W}_{\mu'}$ in the previous epoch, is transferred to the current epoch. The MDDPG algorithm then takes the available knowledge of the actors and the critics and the estimated field hyper-parameters as inputs to learn what will be the optimal trajectories for future epochs. In this way, the optimal trajectories for each epoch can be learned online according to the online estimated field hyper-parameters.

Algorithm 3 Online trajectory planning algorithm in each epoch.

Input: Current epoch N_{curr}, total training episodes N_{episode}, total epochs in an episode N_{epoch}, mini-batch size N_{batch}, discount factor γ, learning rate of the target network τ, threshold value η, action adjust variance β^2, the critic network Q with its weights $\mathcal{W}_Q$, the actor network μ with its weights $\mathcal{W}_\mu$, the target critic network Q' with its weights $\mathcal{W}_{Q'}$ and the target actor network μ' with its weights $\mathcal{W}_{\mu'}$

```
All autonomous underwater vehicles (AUVs) take samples of the field according to their planned trajectories
The fusion center receives the field samples from all the AUVs
The field hyper-parameters $\boldsymbol{\theta}_{\text{hyper}}$ are estimated based on Equation (13)
The fusion center obtained the updated system state $\mathbf{s}$ of all the AUVs based on $\hat{\boldsymbol{\theta}}_{\text{hyper}}$
$(\mathcal{T}_{\text{opt}}, Q, \mu, \mathcal{W}_Q, \mathcal{W}_\mu, Q', \mu', \mathcal{W}_{Q'}, \mathcal{W}_{\mu'})$
        $\leftarrow \text{MDDPG}(N_{\text{curr}}, N_{\text{episode}}, N_{\text{epoch}}, N_{\text{batch}}, \gamma, \tau, \eta, \beta^2, Q, \mathcal{W}_Q, \mu, \mathcal{W}_\mu, Q', \mathcal{W}_{Q'}, \mu', \mathcal{W}_{\mu'}, \boldsymbol{\theta}_{\text{hyper}}, \mathbf{s})$
Start to perform the action for the next epoch according to $\mathcal{T}_{\text{opt}}$
```

4.4. Computational Complexity

The main computational load of the proposed MDDPG algorithm is for neural network training and the hyper-parameter estimation. Denote by I the total number of layers in both the actor network and the critic network. Denote by N_i the number of nodes within the i-th layer. In each epoch, the computational complexity of the MDDPG algorithm is $\mathcal{C}_{\text{MDDPG}} = \mathcal{O}(N_{\text{episode}} N_{\text{batch}} (\sum_{i=1}^{I-1} N_{i+1} N_i))$. To estimate the hyper-parameters in each epoch, the computational complexity is $\mathcal{C}_{\text{est}} = \mathcal{O}(\ell^3)$. Hence, the total computational complexity in each epoch for trajectory planning is $\mathcal{C}_{\text{total}} = \mathcal{C}_{\text{MDDPG}} + \mathcal{C}_{\text{est}}$.

In practice, the computational complexity $\mathcal{C}_{\text{total}}$ can be reduced by employing low-complexity algorithms for the matrix multiplication [31] during the neural network training and for the matrix inversion during the Gaussian process regression [32]. Furthermore, the parameters of neural networks N_{episode}, N_{batch} and N_i can be set to relatively small values, which will release the pressure on the computational time. One reason that we can set those parameter to be small is due to the fact that the weights of the neural networks are transferred from epoch to epoch for a warm start. Finally, by using the parallel computation [33], the execution time of the proposed trajectory planning algorithm can be reduced to an acceptable level for practical applications, e.g., less than two minutes.

5. Algorithm Evaluation

We consider an under-ice field in a 2D area of interest with size 15 km × 15 km. The target set $\mathcal{X}$ consists of 16 × 16 grid points where the latitude and longitude distances between any two consecutive locations are 1 km. The 2D water parameter field is generated based on the circulant embedding method [34] with the field hyper-parameters $\sigma_f^2 = 1$ and $\mathbf{\Lambda} = \text{diag}([0.3, 0.3])$.

The duration of one time slot is 1000 s (16.7 min), and one epoch consists of three time slots, leading to an epoch duration of 50 min. We consider a total of nine epochs in the sampling process, which yields a deployment time duration of 7.5 h in total. The simulated system consists of four AUVs and four access points. The four access points are located at (4 km, 4 km), (4 km, 11 km), (11 km, 4 km) and (11 km, 11 km), respectively. Those four locations are also the initial deployment sites of the four AUVs. The maximal navigation error is $\epsilon = 5$ m [35]. The maximal speed of each AUV is 1 m/s [36], and the maximal distance an AUV can travel within one time slot is therefore $\kappa_{\text{up}} = 1$ km. The communication range for underwater acoustic links between an AUV and an access point is $\kappa_{\text{comm}} = 3.5$ km. The discounted factor is $\gamma = 0.99$. The weights in the reward function Equation (18) are $\alpha_R = 10$, $\alpha_L = 1 \times 10^{-3}$, $\alpha_A = 5 \times 10^{-2}$, $\alpha_{p1} = 2$ and $\alpha_{p2} = 4$.

For both the actor network and the critic network in the proposed MDDPG algorithm, the number of units in the first hidden layer and in the second hidden layer is 400 and 300, respectively. The activation functions of the hidden layers are rectified linear units. The batch normalization is used in the actor network. The learning rate for the actor network and for the critic network is 1×10^{-3} and 1×10^{-4}, respectively. The learning rate for target networks is $\tau = 1 \times 10^{-3}$. The mini-batch size for training is $N_{\text{batch}} = 10$. The threshold value is $\eta = 0.2$. The action adjust variance is $\beta^2 = 0.5$ km^2.

We evaluate the field estimation performance of three schemes.

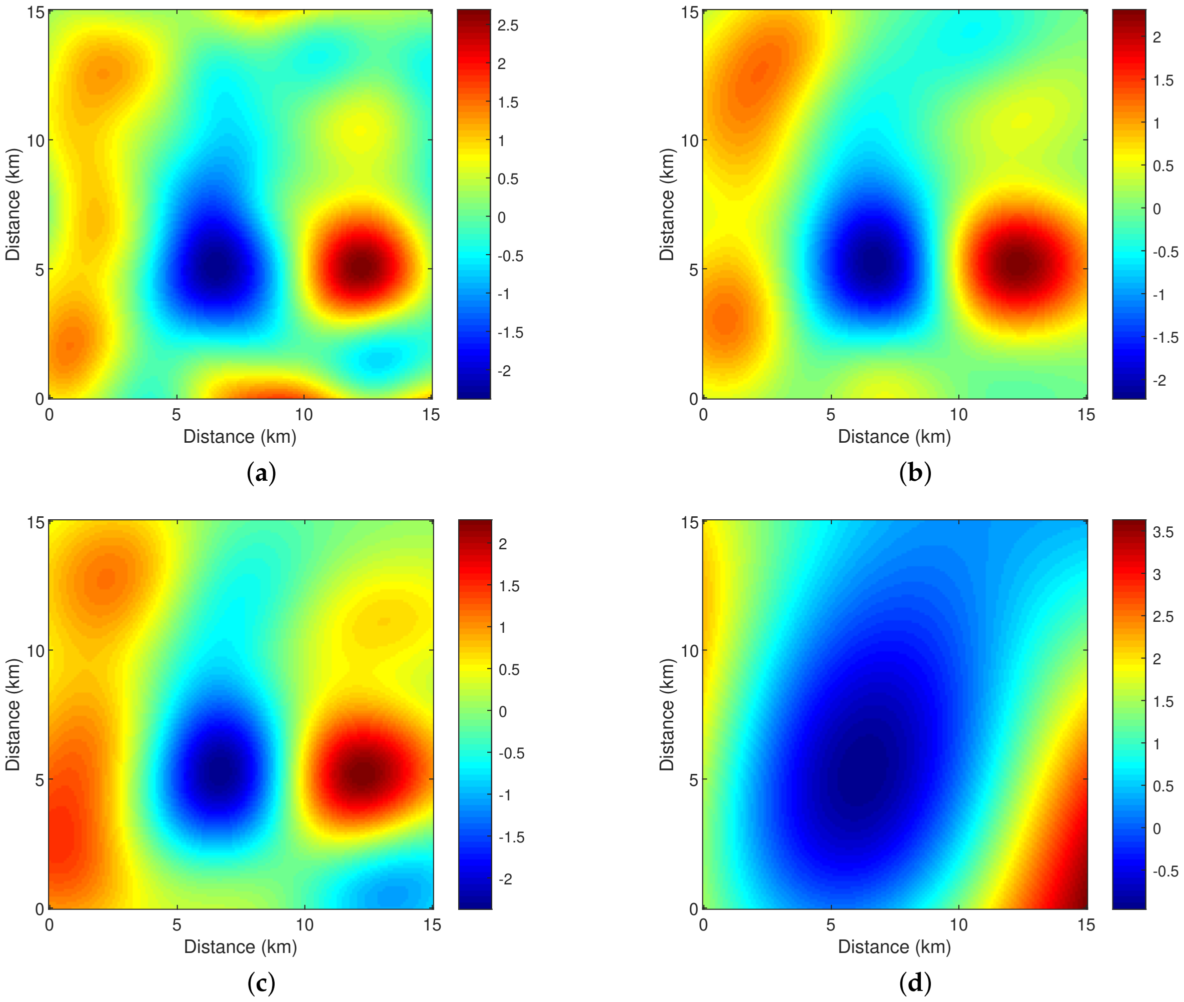
- Scheme 1: A clairvoyant method that determines the sampling trajectories through the offline MDDPG algorithm based on the perfect knowledge of the field hyper-parameters, according to Algorithm 1;
- Scheme 2: The proposed online reinforcement learning algorithm that determines the sampling trajectories epoch-by-epoch through the MDDPG algorithm where the field hyper-parameters are online estimated in each epoch based on the collected samples, according to Algorithm 3;
- Scheme 3: All the AUVs sample the water parameter field via a random walk. Here, the simulation result to be presented is selected among 10,000 Monte Carlo runs, which yields the maximal total reward.

We take the normalized mean square error (NMSE) as a performance metric for the field estimation, which describes the normalized difference between the true field values and the estimated field values over the target points in $\mathcal{X}$,

$$\text{NMSE} := \frac{\|\mathbf{z} - \hat{\mathbf{z}}\|_2^2}{\|\mathbf{z}\|_2^2}, \tag{21}$$

where $\mathbf{z}$ is the vector of field values at target points in $\mathcal{X}$ and $\hat{\mathbf{z}}$ is the estimation based on the Gaussian process regression.

Corresponding to the simulated true field in Figure 4a, the trajectories obtained by the three schemes are shown in Figure 5. To explore the area with high uncertainty, the trajectories determined by Scheme 1 spread out more than those of Schemes 2 and 3, which results in the largest sensed area. Without prior knowledge of the field spatial correlation, the sensed area in the early epochs of Scheme 2 is small, primarily due to the inaccurate estimation of the field hyper-parameters based on limited field samples. With more field samples collected and consequently more accurate estimation of the field hyper-parameters, the trajectory pattern obtained by Scheme 2 is similar to the pattern obtained by Scheme 1, which tends to explore the area with high uncertainty.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 4. The true field and the estimated fields obtained by the three schemes. (**a**) True field; (**b**) Estimated field by Scheme 1; (**c**) Estimated field by Scheme 2; (**d**) Estimated field by Scheme 3.

In Table 1, the three trajectories are compared in the aspects of the AUV total traveled distance, the AUV total traveled angle and the NMSE of field estimation. Scheme 1 achieves the least total traveled distance and the least total traveled angle, while Scheme 2 has a similar total traveled distance, but greater total traveled angle. The performance gap is due to the fact that Scheme 2 does not assume prior knowledge of the field spatial correlation and performs online estimation of the spatial correlation parameters. The total traveled distance and the total traveled angle obtained by Scheme 3 are similar to those of Scheme 2. However, Schemes 1 and 2 achieve much more accurate field estimation (i.e., significantly less NMSEs) than Scheme 3, and a marginal difference of the NMSEs between Schemes 1 and 2 can be observed. The estimated fields by the three schemes are presented in Figure 4. One can see that Schemes 1 and 2 can capture important features of the true field and the estimated field by Scheme 3 is significantly different from the true field.

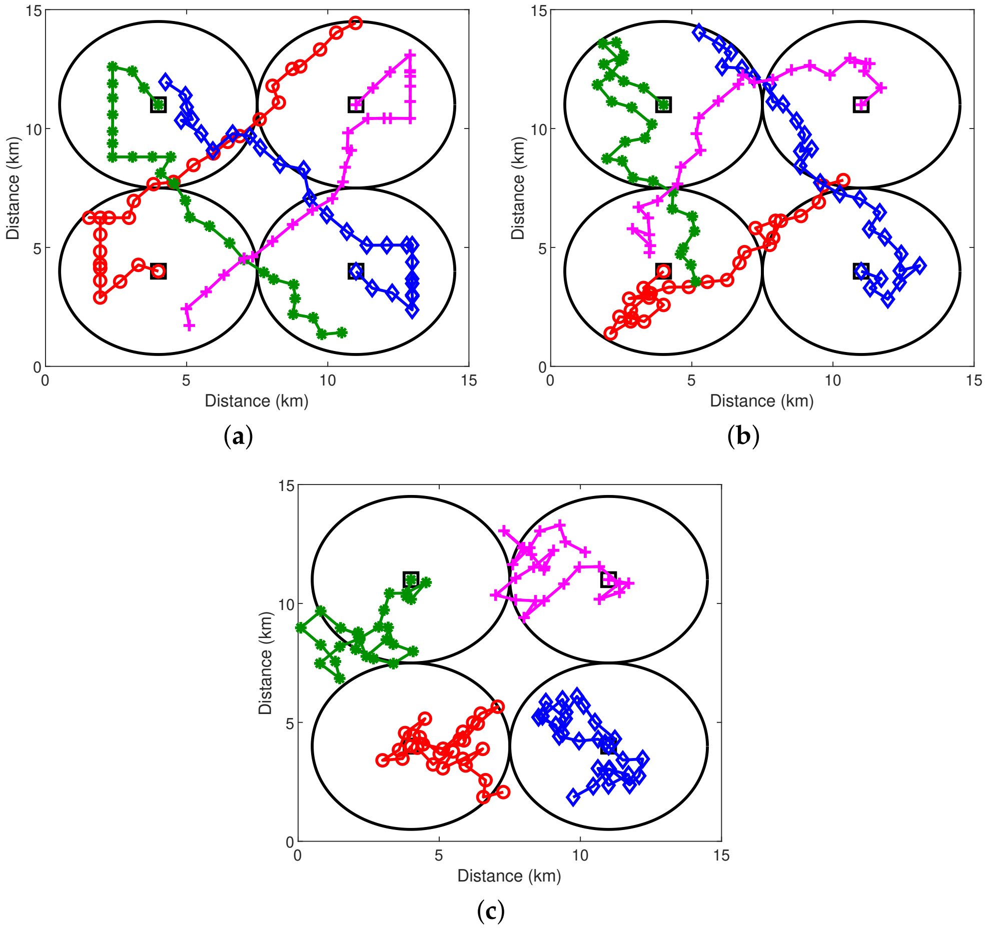


Figure 5. Trajectories of four AUVs obtained by three schemes. The squares indicate the positions of four access points and the initial deployment locations of four AUVs. The circles indicate the acoustic communication coverage of four access points. (**a**) Scheme 1; (**b**) Scheme 2; (**c**) Scheme 3.

Table 1. Performance comparison of the three schemes.

	Scheme 1	**Scheme 2**	**Scheme 3**
Total traveled distance (km)	74.4	77.9	78.1
Total traveled angle (rad)	76.6	117.4	131.5
Normalized mean square error (NMSE)	0.17	0.26	1.35

The above results reveal that although without prior knowledge of the field spatial correlation, the proposed method in this work is able to perform online estimation of the field spatial correlation parameters based on collected field samples and adaptively adjust the trajectories of AUVs while they are on the go. It achieves a performance close to the clairvoyant method assuming perfect knowledge of the field spatial correlation.

Specifically about the proposed method, we further examine the field estimation performance by varying the time epoch duration (namely, the AUV reporting frequency to the fusion center for trajectory adaptation). The number of time slots in each epoch is fixed to be three. The simulation results in Table 2 show that the NMSE of the field estimation decreases as the epoch duration decreases (namely, as the trajectory updating rate increases). Furthermore, the rate of performance improvement becomes less for smaller epoch durations. Consider that such performance improvement is at the cost of more frequent wireless communications and computation at the fusion center for trajectory planning. In real missions, the appropriate choice of epoch duration needs to consider the tradeoff between the field estimation performance and the cost of communications and computation.

Table 2. Field estimation performance of Scheme 2 with different values of the epoch duration.

Epoch Duration (minutes)	30	40	50
NMSE	0.22	0.23	0.26

6. Conclusions

This work studied the online adaptive trajectory planning of multiple AUVs for the water parameter field estimation in the under-ice environment. An online learning-based trajectory planning algorithm was proposed to determine the trajectories of AUVs adaptively. The field of interest was modeled as a Gaussian process with unknown hyper-parameters. The field hyper-parameters and the field posterior distribution were estimated online based on the collected samples. The adaptive trajectory planning problem was formulated as an MDP with the goal of maximizing a long-term reward that is defined based on the field uncertainty reduction and the AUV mobility cost, subject to the kinematics constraint, the communication constraint and the sensing area constraint. A reinforcement learning-based method was designed to solve the above MDP with a constrained action space. The simulation results showed that the proposed reinforcement learning-based adaptive trajectory planning algorithm achieved a performance close to a benchmark method that assumes perfect knowledge of the field hyper-parameters.

Author Contributions: Conceptualization, C.W., L.W., Z.W., M.S. and N.M. Methodology, C.W., L.W., Z.W., M.S. and N.M. Software, C.W. Validation, C.W. and Z.W. Formal analysis, C.W., L.W., Z.W., M.S. and N.M. Investigation, C.W., L.W., Z.W., M.S. and N.M. Resources, Z.W. Data curation, C.W. Writing, original draft preparation, C.W., L.W. and Z.W. Writing, review and editing, C.W, L.W., Z.W., M.S. and N.M. Supervision, Z.W., M.S. and N.M. Project administration, Z.W., M.S. and N.M. Funding acquisition, Z.W., M.S. and N.M.

Funding: This work was funded in part by the National Science Foundation under Grants CNS-1453886 (CAREER), CNS-1551067 and ECCS-1651135 (CAREER) and by the ICCPaul William Seed Grant of Michigan Technological University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Thompson, D.; Caress, D.; Thomas, H.; Conlin, D. MBARI Mapping AUV Operations in the Gulf of California 2015. In Proceedings of the Conference on MTS/IEEE OCEANS, Washington, DC, USA, 19–22 October 2015; pp. 1–7.
2. Thompson, D.; Caress, D.; Clague, D.; Conlin, D.; Harvey, J.; Martin, E.; Paduan, J.; Paull, C.; Ryan, J.; Thomas, H.; et al. MBARI Dorado AUV's Scientific Results. In Proceedings of the Conference on MTS/IEEE OCEANS, San Diego, CA, USA, 23–27 September 2013; pp. 1–9.
3. Kukulya, A.; Plueddemann, A.; Austin, T.; Stokey, R.; Purcell, M.; Allen, B.; Littlefield, R.; Freitag, L.; Koski, P.; Gallimore, E.; et al. Under-ice operations with a REMUS-100 AUV in the Arctic. In Proceedings of the IEEE/OES Autonomous Underwater Vehicles, Monterey, CA, USA, 1–3 September 2010.
4. Leonard, N.E.; Paley, D.A.; Lekien, F.; Sepulchre, R.; Fratantoni, D.M.; Davis, R.E. Collective motion, sensor networks, and ocean sampling. In *Proceedings of the IEEE*; IEEE: Piscataway, NJ, USA, 2007; Volume 95, pp. 48–74.
5. Yilmaz, N.; Evangelinos, C.; Lermusiaux, P.; Patrikalakis, N. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE J. Ocean. Eng.* **2008**, *33*, 522–537. [CrossRef]
6. Zhu, D.; Huang, H.; Yang, S.X. Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace. *IEEE Trans. Cybern.* **2013**, *43*, 504–514. [PubMed]

7. Szwaykowska, K.; Zhang, F. Trend and bounds for error growth in controlled Lagrangian particle tracking. *IEEE J. Ocean. Eng.* **2014**, *39*, 10–25. [CrossRef]
8. Xu, Y.; Choi, J.; Oh, S. Mobile sensor network navigation using Gaussian processes with truncated observations. *IEEE Trans. Robot.* **2011**, *27*, 1118–1131. [CrossRef]
9. Marchant, R.; Ramos, F. Bayesian optimisation for informative continuous path planning. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6136–6143.
10. Nguyen, L.; Kodagoda, S.; Ranasinghe, R.; Dissanayake, G. Information-driven adaptive sampling strategy for mobile robotic wireless sensor network. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 372–379. [CrossRef]
11. Martinez-Cantin, R.; Freitas, N.; Brochu, E.; Castellanos, J.; Doucet, A. A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Auton. Robot.* **2009**, *27*, 93–103. [CrossRef]
12. Singh, A.; Krause, A.; Kaiser, W. Nonmyopic adaptive informative path planning for multiple robots. In Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09), Pasadena, CA, USA, 11–17 July 2009; pp. 1843–1850.
13. Marchant, R.; Ramos, F.; Sanner, S. Sequential Bayesian optimization for spatial-temporal monitoring. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), Quebec City, QC, Canada, 23–17 July 2014; pp. 553–562.
14. Morere, P.; Marchant, R.; Ramos, F. Sequential Bayesian optimization as a POMDP for environment monitoring with UAVs. In Proceedings of the Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 6381–6388.
15. Binney, J.; Krause, A.; Sukhatme, G.S. Informative path planning for an autonomous underwater vehicle. In Proceedings of the Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010; pp. 4791–4796.
16. Hollinger, G.; Englot, B.; Hover, F.; Mitra, U.; Sukhatme, G. Uncertainty-driven view planning for underwater inspection. In Proceedings of the Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 4884–4891.
17. Marino, A.; Antonelli, G.; Aguiar, A.; Pascoal, A.; Chiaverini, S. A decentralized strategy for multirobot sampling/patrolling: Theory and experiments. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 313–322. [CrossRef]
18. Kemna, S.; Rogers, J.G.; Nieto-Granda, C.; Young, S.; Sukhatme, G.S. Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments. In Proceedings of the Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2124–2130.
19. Marino, A.; Antonelli, G. Experimental results of coordinated sampling/patrolling by autonomous underwater vehicles. In Proceedings of the Conference on IEEE Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 4141–4146.
20. Rasmussen, C.E. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2004; pp. 63–71.
21. Williams, C.K.; Rasmussen, C.E. Gaussian processes for regression. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1996; pp. 514–520.
22. Bellman, R. A Markovian decision process. *J. Math. Mech.* **1957**, *6*, 679–684. [CrossRef]
23. Karl, H.; Willig, A. *Protocols and Architectures for Wireless Sensor Networks*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2005.
24. Brito, M.P.; Lewis, R.S.; Bose, N.; Griffiths, G. Adaptive autonomous underwater vehicles: An assessment of their effectiveness for oceanographic applications. *IEEE Trans. Eng. Manage.* **2018**, 1–14. [CrossRef]
25. Mertikas, S.P. *Error Distributions and Accuracy Measures in Navigation: An Overview*; Technical Report; Geodesy and Geomatics Engineering: Fredericton, NB, Canada, 1985.
26. Kay, S.M. *Fundamentals of Statistical Signal Processing: Estimation Theory*; Prentice Hall: Upper Saddle River, NJ, USA, 1993; Volume 2.
27. Byrd, R.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190–1208. [CrossRef]
28. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: Cambridge, MA, USA, 2017.

29. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016.
30. Bishop, C.M. *Pattern Recognition and Machine Learning*, 6th ed.; Springer: New York, NY, USA, 2006.
31. Le Gall, F. Powers of tensors and fast matrix multiplication. In Proceedings of the Symposium on Symbolic and Algebraic Computation, Kobe, Japan, 23–25 July 2014; pp. 296–303.
32. Leithead, W.E.; Zhang, Y. $O(N^2)$-operation approximation of covariance matrix inverse in Gaussian process regression based on quasi-Newton BFGS Method. *Commun. Stat. Simul. Comput.* **2007**, *36*, 367–380. [CrossRef]
33. Takefuji, Y. *Neural Network Parallel Computing*; Springer Science & Business Media: Berlin, Germany, 2012.
34. Kroese, D.P.; Botev, Z.I. Spatial process simulation. In *Stochastic Geometry, Spatial Statistics and Random Fields: Models and Algorithms*; Schmidt, V., Ed.; Springer: Berlin, Germany, 2015; pp. 369–404. [CrossRef]
35. McEwen, R.; Thomas, H.; Weber, D.; Psota, F. Performance of an AUV navigation system at Arctic latitudes. *IEEE J. Ocean. Eng.* **2005**, *30*, 443–454. [CrossRef]
36. Norgre, P.; Skjetne, R. Using autonomous underwater vehicles as sensor platforms for ice-monitoring. *Model. Identif. Control* **2014**, *35*, 269–277.