Journal of
Marine Science and Engineering

MDPI

Article

Dynamic Data-Driven Application System for Flow Field Prediction with Autonomous Marine Vehicles

Qianlong Jin [1,2,3], Yu Tian [1,2,4,*], Weicong Zhan [1,2,3], Qiming Sang [1,2,3], Jiancheng Yu [1,2] and Xiaohui Wang [1,2]

[1] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; jinqianlong@sia.cn (Q.J.); zhanweicong@sia.cn (W.Z.)
[2] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
[4] China-Portugal Belt and Road Joint Laboratory on Space & Sea Technology Advanced Research, Shanghai 201304, China
* Correspondence: tiany@sia.cn

Abstract: Efficiently predicting high-resolution and accurate flow fields through networked autonomous marine vehicles (AMVs) is crucial for diverse applications. Nonetheless, a research gap exists in the seamless integration of data-driven flow modeling, real-time data assimilation from flow sensing, and the optimization of AMVs' sensing strategies, culminating in a closed-loop dynamic data-driven application system (DDDAS). This article presents a novel DDDAS that systematically integrates flow modeling, data assimilation, and adaptive flow sensing using networked AMVs. It features a hybrid data-driven flow model, uniting a neural network for trend prediction and a Gaussian process model for residual fitting. The neural network architecture is designed using knowledge extracted from historic flow data through tidal harmonic analysis, enhancing its capability in flow prediction. The Kriged ensemble transform Kalman filter is introduced to assimilate spatially correlated flow-sensing data from AMVs, enabling effective model learning and accurate spatiotemporal flow prediction, while forming the basis for optimizing AMVs' flow-sensing paths. A receding horizon strategy is proposed to implement non-myopic optimal path planning, and a distributed strategy of implementing Monte Carlo tree search is proposed to solve the resulting large-scale tree searching-based optimization problem. Computer simulations, employing underwater gliders as sensing networks, demonstrate the effectiveness of the proposed DDDAS in predicting depth-averaged flow in nearshore ocean environments.

Keywords: dynamic data-driven application system; autonomous marine vehicles; flow field prediction; data assimilation; adaptive sampling

check for updates

Citation: Jin, Q.; Tian, Y.; Zhan, W.; Sang, Q.; Yu, J.; Wang, X. Dynamic Data-Driven Application System for Flow Field Prediction with Autonomous Marine Vehicles. *J. Mar. Sci. Eng.* **2023**, *11*, 1617. https://doi.org/10.3390/jmse11081617

Academic Editor: Sergei Chernyi

Received: 28 July 2023
Revised: 16 August 2023
Accepted: 16 August 2023
Published: 18 August 2023

1. Introduction

The advancement of affordable, long-range autonomous marine vehicle (AMV) technologies, including unmanned surface vehicles (USVs), autonomous underwater vehicles (AUVs), underwater gliders, and wave gliders [1], has presented opportunities for the deployment of persistent, robotic, and autonomous Lagrangian networks in ocean environment sensing. In contrast to conventional ocean observation networks such as the ARGO array [2], the utilization of mobile and autonomous networks composed of a collection of AMVs offers a distinctive technical capability: controllable maneuverability in complex and dynamic ocean environments. Therefore, effectively leveraging their controllable maneuverability to enable the networks to acquire cost-effective and information-rich sensing data within nonuniform and dynamic ocean environments, aligning with the requirements of observation missions, is a key issue within the research community focused on autonomous ocean sensing with AMV networks.

The tracking of dynamic ocean features to gather in situ data for characterizing and predicting the evolution of these features, such as thermoclines [3], plumes [4], and mesoscale eddies [5], is an aspect of applying AMV networks for ocean sensing. Besides feature tracking, a foundational application objective in utilizing AMV networks is to autonomously and adaptively collect a variety of informative Lagrangian data streams, such as temperature, salinity, and flow velocity of seawater, that can be assimilated into predictive ocean models. The assimilation of these data streams can lead to significant improvement in the accuracy of temporal predictions for spatial fields in ocean environments [6]. Consequently, the real-time optimization of the data streams for assimilation using numerical ocean models is crucial in maximizing model prediction performance, and it plays an essential role in both research and practical applications of ocean sensing with AMV networks. One key approach to addressing this optimization problem is through the utilization of feedback control theory and the formulation of a sensor management technical framework [7]. This framework establishes a closed-loop system by integrating a numerical ocean model with an AMV sensing network [8] and uses iterative feedback and optimization processes to continuously adapt the network's sensing strategies and improve prediction performance. Different from conventional feedback control of dynamic systems, the feedback control in the closed-loop adaptive ocean sensing and prediction system involves cooperative control of the sensing actions of the networked AMVs, taking into account the constraints imposed by ocean environments. And the primary objective of this feedback control is to collect informative sensing data of ocean environments, which can be utilized in ocean models to improve prediction performance. The AOSN II and ASAP projects have made significant contributions to the research on autonomous and adaptive ocean sensing with AMV networks, particularly regarding the utilization of underwater gliders as part of the sensing network [9,10]. These projects have laid an important foundation for the further advancements in this field.

In the context of ocean sensing and prediction systems, predictive ocean models play an important role. Alongside traditional methods based on numerically solving partial differential equations of geophysical fluid dynamics, data-driven techniques have emerged as powerful and complementary tools for ocean modeling and prediction. Leveraging machine learning and statistical approaches, data-driven methods extract patterns and relationships from observed data, enabling them to make high-resolution and accurate predictions with reduced computational costs. Extensive research efforts have been dedicated to developing and applying data-driven techniques for ocean modeling and spatial and temporal predictions. Notable approaches include objective analysis (OA) [11,12], Gaussian process (GP) [13], Kriging geostatistical prediction [14], compressive sensing [15], radial basis functions (RBFs) [16], proper orthogonal decomposition [17,18], dynamic mode decomposition (DMD) [19], and neural networks [20,21]. Compared to conventional ocean models, data-driven techniques rely more heavily on the information content within the data for their implementation. Building upon the aforementioned data-driven techniques, researchers have devised methodologies to optimize sensing strategies for both Eulerian and Lagrangian platforms, as demonstrated in [11–13,16–18]. These optimized strategies enable the collection of informative data, leading to enhancements in data-driven prediction performance. Through the integration of data-driven techniques and adaptive observation strategies within the closed-loop system, the ocean sensing and prediction capabilities of AMVs can be significantly improved. This approach holds great potential in achieving high-resolution and accurate ocean prediction with AMV sensing networks.

High-resolution and accurate flow field prediction is a fundamental component in ocean prediction. The spatiotemporal flow information not only offers insights into the multi-scale dynamic behavior of ocean currents but also plays a crucial role in path planning [22,23], navigation [24,25], feature prediction and tracking [26], and motion control for both individual and swarms of AMVs [11,27] implementing ocean sensing missions. Many studies have been conducted on data-driven flow prediction [22,24,25,28–30], addressing flow prediction in diverse oceanic environments where tidal or non-tidal flow is dominant. However, a research gap exists in the specific area of integrating data-driven flow modeling, online data assimilation of flow sensing data into data-driven flow models, and optimizing AMVs' flow sensing strategy, based on the closed-loop feedback ocean prediction and adaptive sensing technical framework.

Based on the above review and discussions, as well as recognizing the significance of high-resolution and accurate flow field prediction in ocean prediction and AMVs' operations, this article is dedicated to exploring this data-driven closed-loop approach and systematically developing methods for data-driven flow modeling, data assimilation, and optimizing AMVs' observations. The systematic development establishes a dynamic data-driven application system (DDDAS) [31] for flow prediction and sensing with AMVs. The DDDAS seamlessly integrates data-driven modeling, data assimilation, and an optimal decision-making process, as depicted in Figure 1, thereby enhancing high-resolution and accurate flow field prediction capabilities and enabling adaptive and efficient flow sensing using AMVs.

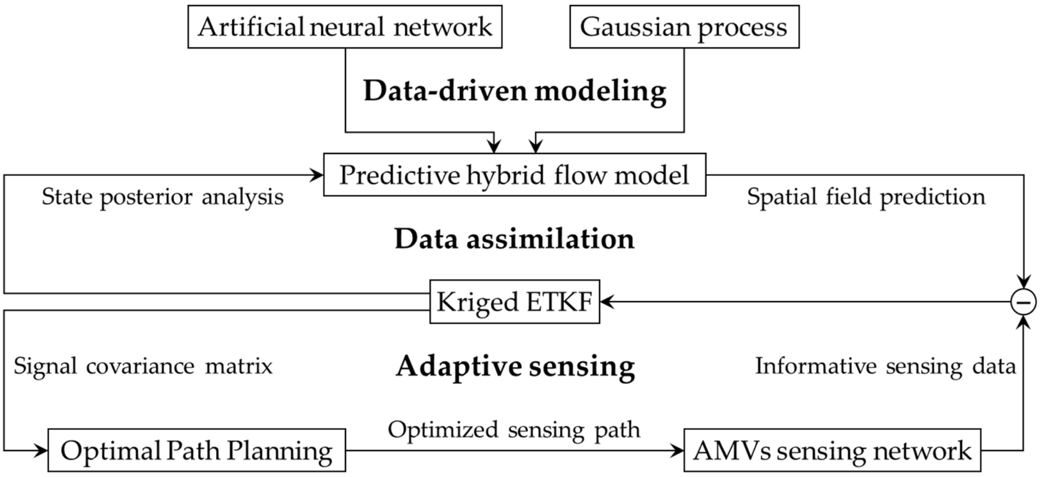


Figure 1. Illustration of the DDDAS for flow field prediction and sensing with AMVs.

The main contributions and novelties of this article are as follows: (1) Pioneering closed-loop DDDAS development: This article represents an advancement by introducing the first-ever closed-loop data-driven DDDAS designed for predicting flow fields. The system seamlessly integrates data-driven flow modeling, data assimilation, and adaptive sensing with AMVs. (2) A hybrid data-driven flow model: This article proposes an innovative hybrid data-driven flow model that combines the predictive power of a neural network with the statistical estimation of a Gaussian process model. Notably, the neural network architecture is designed to incorporate prior knowledge of tidal components, contributing to accurate flow predictions. This hybrid model not only captures flow trends accurately but also effectively handles residuals. (3) The Kriged ensemble transform Kalman filter (ETKF) data assimilation: This article introduces a pioneering application of the Kriged ETKF method for assimilating spatially correlated flow-sensing data from AMVs. This methodology demonstrates exceptional efficacy in model learning and spatiotemporal flow prediction. It also enhances the optimization of AMVs' sensing paths. (4) A receding horizon strategy for optimal path planning: Addressing the challenge of non-myopic optimal path planning for a network of AMVs, this article introduces a strategic receding horizon approach. To tackle the resulting computational complexity of large-scale tree-searching-based optimization, this article proposes a novel distributed Monte Carlo Tree Search (MCTS) method. This innovative approach harnesses the collective computational capabilities of networked AMVs to explore the search space and derive effective planning solutions.

The rest of this paper is structured as follows. Section 2 presents the data-driven hybrid flow model, including the neural network model and the Gaussian process model. Section 3 elaborates on the state and observation models for data assimilation and the implementation of the Kriged ETKF method for model learning and flow prediction.

Section 4 presents the optimization of the AMVs' paths for collecting informative flow-sensing data to enhance flow prediction accuracy. Section 5 presents simulation studies to demonstrate the performance of the DDDAS. Finally, Section 6 presents the conclusions.

2. Data-Driven Flow Prediction Model

The flow model serves as a fundamental component within the developed DDDAS. The section introduces the developed hybrid data-driven model, which leverages the machine learning model of neural networks and the statical model of the Gaussian process. Section 2.1 provides an overview of the hybrid model. Section 2.2 presents the utilization of the tidal harmonic analysis method to extract essential prior knowledge of the flow, supporting the design of the neural network model. Section 2.3 elaborates on the development of the neural network model and Section 2.4 presents the Gaussian process model.

2.1. Model Overview

To enable the prediction of the flow speed at specific spatiotemporal locations $\boldsymbol{x} = (\boldsymbol{s}, t)$, where $\boldsymbol{s} \in R^2$ represents the spatial location and $t \in R$ represents the time, a data-driven spatiotemporal flow field model is established. This model comprises two submodels:

$$u(\boldsymbol{x}) = \mu_u(\boldsymbol{x}) + r_u(\boldsymbol{x}) \tag{1}$$

$$v(\boldsymbol{x}) = \mu_v(\boldsymbol{x}) + r_v(\boldsymbol{x}) \tag{2}$$

Equation (1) represents the submodel for the latitudinal flow speed $u(\boldsymbol{x})$, and Equation (2) represents the submodel for the longitudinal flow speed $v(\boldsymbol{x})$, at the given spatiotemporal locations $\boldsymbol{x}$.

The flow model consists of two fundamental components: the mean component, and the residual component. The mean component, denoted as $\mu_u(\boldsymbol{x})$ and $\mu_v(\boldsymbol{x})$, captures the average or expected flow speeds, providing the trends in flow at the given spatiotemporal location. In this article, a feedforward neural network is employed to approximate the continuous and nonlinear function of $\mu_u(\boldsymbol{x})$ and $\mu_v(\boldsymbol{x})$. The choice is motivated by the neural network's universal approximation capabilities, which enable it to effectively learn and represent the complex relationships between input parameters including longitude, latitude, time, and the corresponding flow speed. To implement the flow model, a specific neural network is designed, and its detailed architecture is described in Section 2.3. On the other hand, the residual component, denoted as $r_u(\boldsymbol{x})$ and $r_v(\boldsymbol{x})$, accounts for variations from the mean component. These residuals may arise due to unmodelled influences or other factors affecting the flow speed. In this article, the Gaussian process is employed to model $r_u(\boldsymbol{x})$ and $r_v(\boldsymbol{x})$, which will be described in detail in Section 2.4. The composition of the flow prediction model, take $u(\boldsymbol{x})$ for example, is depicted in Figure 2. By decomposing the flow into mean and residual components, the spatiotemporal flow model could effectively capture both the average behavior and the deviations from the average, which enhance the accuracy and reliability of the predictions.

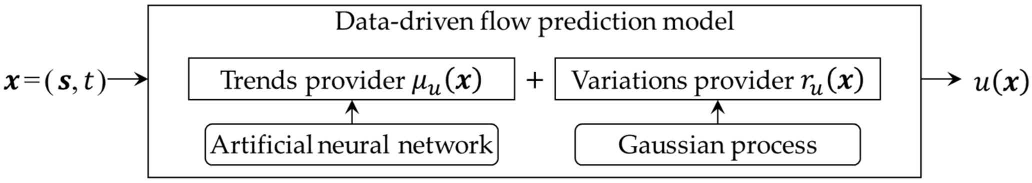

Data-driven flow prediction model
$\boldsymbol{x}=(\boldsymbol{s},t)$
Trends provider $\mu_u(\boldsymbol{x})$
+
Variations provider $r_u(\boldsymbol{x})$
$u(\boldsymbol{x})$
Artificial neural network
Gaussian process

Figure 2. The composition of the data-driven flow prediction model.

Throughout the subsequent sections, the model for $u(\boldsymbol{x})$ will be focused on as an example for description and discussion. And to simplify expression and notation, the subscript u in $\mu_u(\boldsymbol{x})$ and $r_u(\boldsymbol{x})$ will be omitted. It is important to note that both $u(\boldsymbol{x})$ and

$v(\boldsymbol{x})$ share the same structural components and underlying principles. Therefore, the descriptions, methodologies, and discussions related to $u(\boldsymbol{x})$ can be directly applied to $v(\boldsymbol{x})$.

2.2. Decomposition of Flow Using Tidal Harmonic Analysis

Driven by tidal forces, flow fields in nearshore ocean environments often exhibit spatiotemporal periodic features, which can be characterized by the presence of multiple periodic components. These components possess distinct frequencies and amplitudes, playing significant roles in shaping the overall flow dynamics. Incorporating information about the tidal components in the design of the neural network architecture allows for capturing and learning about the relationships between the tidal components and the flow speed more effectively. Therefore, this article implements this integration of knowledge of tidal components into the neural network design, leveraging the periodic patterns present in the flow field to enhance the accuracy of flow prediction.

Tidal harmonic analysis is an effective approach for decomposing the flow into different constituents, represented by trigonometric functions with distinct frequencies. It processes the training data of a temporal sequence of flow fields to identify the dominant tidal frequencies and their corresponding amplitudes. The tidal frequencies can be calculated using methods such as least-squares harmonic estimation (LS-HE) [32] and multivariate LS-HE [33]. Once the frequencies are determined, the amplitudes and phases of the corresponding trigonometric function for each tidal constituent can be calculated using methods such as least-squares estimation. These constituents capture the fundamental periodic components of the flow.

In this article, the tidal harmonic analysis is implemented to obtain the tidal constituents. Each obtained tidal constituent corresponds to two orthogonal temporal basis functions, given by:

$$\begin{cases} \psi_i(t) = cos\left(\frac{2\pi t}{\mathscr{T}_i}\right) \\ \psi_{i+N_{tc}}(t) = sin\left(\frac{2\pi t}{\mathscr{T}_i}\right) \end{cases} \quad i = 1, \cdots, N_{tc} \tag{3}$$

where $\mathscr{T}_i$ represents the period of the i-th tidal constituent, and N_{tc} represents the total number of the chosen dominant constituents. These basis functions form a basis set that captures the periodic variations in the flow over time.

For a specific spatial location $\boldsymbol{s}$, the flow contributed by the i-th tidal constituent can be expressed as follows:

$$\mu_i(\boldsymbol{s}, t) = \beta_i(\boldsymbol{s})cos\left(\frac{2\pi t}{\mathscr{T}_i}\right) + \beta_{i+N_{tc}}(\boldsymbol{s})sin\left(\frac{2\pi t}{\mathscr{T}_i}\right) \quad i = 1, \cdots, N_{tc} \tag{4}$$

where the coefficients $\beta_i(\boldsymbol{s})$ and $\beta_{i+N_{tc}}(\boldsymbol{s})$ represent the amplitudes of the basis functions $\psi_i(\cdot)$ and $\psi_{i+N_{tc}}(\cdot)$ at location $\boldsymbol{s}$, respectively. Considering the contributions of all tidal constituents, the overall flow at the spatial location $\boldsymbol{s}$ can be expressed as:

$$\mu(\boldsymbol{s}, t) = \beta_0(\boldsymbol{s}) + \textstyle\sum_{i=1}^{N_{tc}} \mu_i(\boldsymbol{s}, t) = \boldsymbol{\beta}^T(\boldsymbol{s})\boldsymbol{\psi}(t) \tag{5}$$

where $\beta_0(\boldsymbol{s})$ represents the non-tidal constant component of the flow at spatial location $\boldsymbol{s}$, $\boldsymbol{\beta}(\cdot) = [\beta_0(\cdot), \beta_1(\cdot), \cdots, \beta_{2N_{tc}}(\cdot)]^T$ varies with the locations, and $\boldsymbol{\psi}(\cdot) = [1, \psi_1(\cdot), \cdots, \psi_{2N_{tc}}(\cdot)]^T$.

2.3. Neural Network Model for the Mean Component

Based on the flow model in Equation (5), a feedforward neural network is designed to model $\mu(\boldsymbol{x})$ in this article, and its architecture is depicted in Figure 3. The main objective of the network is to learn about the relationship between $\mu(\boldsymbol{x})$ and $\boldsymbol{x} = (\boldsymbol{s}, t)$, enabling the prediction of the flow field.

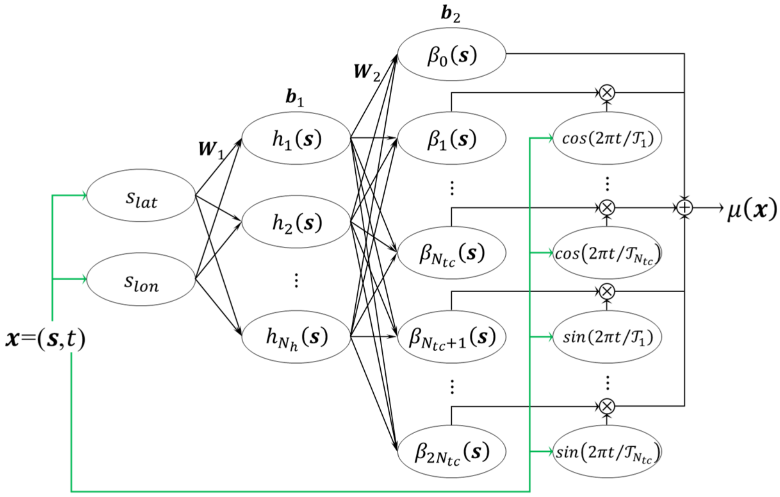

$\boldsymbol{b}_2$
$\beta_0(\boldsymbol{s})$
$\boldsymbol{W}_2$
$\boldsymbol{b}_1$
$\boldsymbol{W}_1$
$h_1(\boldsymbol{s})$
$\beta_1(\boldsymbol{s})$
$cos(2\pi t/\mathcal{T}_1)$
s_{lat}
$h_2(\boldsymbol{s})$
$\beta_{N_{tc}}(\boldsymbol{s})$
$cos(2\pi t/\mathcal{T}_{N_{tc}})$
$\mu(\boldsymbol{x})$
s_{lon}
$\boldsymbol{x}$=($\boldsymbol{s}$,t)
$h_{N_h}(\boldsymbol{s})$
$\beta_{N_{tc}+1}(\boldsymbol{s})$
$sin(2\pi t/\mathcal{T}_1)$
$\beta_{2N_{tc}}(\boldsymbol{s})$
$sin(2\pi t/\mathcal{T}_{N_{tc}})$

Figure 3. The architecture of the neural network.

In the model, the neural network takes the latitudinal coordinate s_{lat} and the longitudinal coordinate s_{lon} of the spatial location $\boldsymbol{s}$, along with the time t as input parameters. It then outputs the predicted flow speed $\mu(\boldsymbol{x})$ at $\boldsymbol{x} = (\boldsymbol{s}, t)$. The neural network consists of three hidden layers: the h-layer, the β-layer, and the ψ-layer. The h-layer consists of N_h nodes that are fully connected to the model inputs s_{lat} and s_{lon}. The β-layer has $2N_{tc} + 1$ nodes that are fully connected to the h-layer. And the ψ-layer is a hidden multiplicative layer with $2N_{tc}$ nodes, which is connected to the model input t. The model output is obtained by calculating the inner product of the outputs from the β-layer and the ψ-layer. The parameters of the neural network are denoted as follows: the connection weight matrix between the h-layer nodes and the input nodes is denoted as $\boldsymbol{W}_1$, and the bias vector of the h-layer nodes is denoted as $\boldsymbol{b}_1$. Similarly, the connection weight matrix between the β-layer nodes and the h-layer nodes is denoted as $\boldsymbol{W}_2$, and the bias vector of the β-layer nodes is denoted as $\boldsymbol{b}_2$. Additionally, the connection weight between the node $\beta_i(\cdot)$ and the node $h_j(\cdot)$ is denoted as $w_{2,i,j}$, and the bias of the node $\beta_i(\cdot)$ is denoted as $b_{2,i}$, where i ranges from 0 to $2N_{tc}$, and j ranges from 1 to N_h.

To introduce nonlinearity into the neural network, the S-type hyperbolic tangent function $\mathscr{g}_h(\cdot)$ is selected as the activation function for the h-layer, i.e., $\boldsymbol{h}(\boldsymbol{s}) = \left[h_1(\boldsymbol{s}), \cdots, h_{N_h}(\boldsymbol{s})\right]^T = \mathscr{g}_h(\boldsymbol{W}_1\boldsymbol{s} + \boldsymbol{b}_1) = tanh(\boldsymbol{W}_1\boldsymbol{s} + \boldsymbol{b}_1)$. For the β-layer, the linear activation function $\mathscr{g}_\beta(\cdot)$ is chosen, i.e., $\boldsymbol{\beta}(\boldsymbol{s}) = [\beta_0(\boldsymbol{s}), \cdots, \beta_{2N_{tc}}(\boldsymbol{s})]^T = \mathscr{g}_\beta(\boldsymbol{W}_2\boldsymbol{h}(\boldsymbol{s}) + \boldsymbol{b}_2) = \boldsymbol{W}_2\boldsymbol{h}(\boldsymbol{s}) + \boldsymbol{b}_2$. The temporal basis functions are chosen as the corresponding activation functions for the ψ-layer.

For the offline learning of the network parameters using the data of flow and its learned tidal constituents, the Levenberg–Marquardt backpropagation optimization method is employed. This optimization technique efficiently adjusts the connection weights and biases of the neural network to minimize the mean squared error function, which is chosen as the objective function. Once the network is trained, it can effectively predict the flow field at new spatiotemporal locations using the learned relationships from the training data.

2.4. Gaussian Process Model for the Residual Component

In this article, the residual component $r(\boldsymbol{x})$ of the flow model is modeled using a Gaussian process model. GPs, renowned in machine learning and statistics, prove invaluable for representing complex data relationships and providing predictions along with uncertainty estimations. In this article, the Gaussian process model adopts a zero mean and an assumption is made that in the model the residuals do not exhibit temporal dependencies, meaning that they are temporally uncorrelated, thus $r(\boldsymbol{x}) \sim GP\left(\boldsymbol{0}, \mathscr{k}\left((\boldsymbol{s}_i, t_k), (\boldsymbol{s}_j, t_k)\right)\right)$. This assumption allows the model to capture the spatial correlation between flow measurements at different spatial locations.

In the model, the kernel function $\mathscr{k}\big((\boldsymbol{s}_i, t_k), (\boldsymbol{s}_j, t_k)\big) = E\big[(r(\boldsymbol{s}_i, t_k))(r(\boldsymbol{s}_j, t_k))\big]$ quantifies the covariance between the spatiotemporal locations $(\boldsymbol{s}_i, t_k)$ and $(\boldsymbol{s}_j, t_k)$, thereby measuring the similarity or dissimilarity between the flow at different spatiotemporal locations. Here, $E[\cdot]$ represents the expectation operator. In this article, the anisotropic squared exponential kernel function is selected to describe the spatial correlation of the residual component of the model at time t_k:

$$\begin{aligned} & \mathscr{k}\big((\boldsymbol{s}_i, t_k), (\boldsymbol{s}_j, t_k)\big) \\ = & \; COV\big(r(\boldsymbol{s}_i, t_k), r(\boldsymbol{s}_j, t_k)\big) \\ = & \; \theta_{k,0}^2 exp\left\{ -\frac{1}{2}\big(\boldsymbol{s}_i - \boldsymbol{s}_j\big)^T \begin{bmatrix} \theta_{k,1}^2 & 0 \\ 0 & \theta_{k,2}^2 \end{bmatrix} \big(\boldsymbol{s}_i - \boldsymbol{s}_j\big) \right\} \end{aligned} \tag{6}$$

Here, $\theta_{k,0}$, $\theta_{k,1}$, and $\theta_{k,2}$ are the model hyper-parameters at time t_k, where $\theta_{k,0}^2$ is the overall variance, $\theta_{k,1}$ is the latitudinal length scale, and $\theta_{k,2}$ is the longitudinal length scale. And $\boldsymbol{\theta}_k = [\theta_{k,0}, \theta_{k,1}, \theta_{k,2}]^T$ represents the vector of the model hyper-parameters.

The accuracy of the hyper-parameters $\boldsymbol{\theta}_k$ significantly impacts the prediction accuracy of the model. To estimate the model hyper-parameters $\boldsymbol{\theta}_k$, the maximum likelihood estimation method is employed. This estimation method uses the available data at time t_k to find the hyper-parameters that maximize the likelihood of the observed residuals. The implementation details of this estimation process are provided in Section 3.5.

It should be noted that the modeling performance can be further enhanced by employing more advanced and sophisticated kernel functions. This article, however, focuses on the widely used anisotropic squared exponential kernel function. Comparisons involving different kernel functions fall beyond the scope of this article.

3. Data Assimilation with the Kriged ETKF

To achieve accurate predictions of flow speeds using the presented flow model, it is essential to accurately estimate the model parameters, including the weights and biases in the neural network model, as well as the hyper-parameters in the residual model. While the parameters of the flow model can be learned from historical data, there are limitations and uncertainties associated with the data used for training. These challenges include data sparsity, noise, and discrepancies in the actual flow conditions that the model aims to predict. To address these issues and improve the accuracy of flow predictions, this article implements online real-time learning of the parameters of the model using in situ sensing data of flow speeds provided by AMVs, employing the sequential Bayesian filtering method. This real-time learning enables the model to adapt to changing conditions to closely align with the actual flow conditions and to refine its predictions based on the most recent observations, thereby leading to more accurate and reliable predictions of flow in nearshore ocean environments.

This section delves into the implementation of data assimilation, a crucial aspect of the developed DDDAS. Sections 3.1 and 3.2 provide the state model and observation model, respectively, which serve as the foundation for the subsequent implementation of the sequential data assimilation process. Section 3.3 presents the learning of the neural network model using the ETKF method, and Section 3.4 presents the flow field prediction using contributions from both the learned neural network model and the Gaussian process model. Finally, Section 3.5 addresses the estimation of the hyper-parameters within the Gaussian process model, thereby completing the comprehensive data assimilation process with the Kriged ETKF method.

3.1. State Model

For the neural network model, all the connection weights $\{w_{2,i,j} | i = 0, \cdots, 2N_{tc}; j = 1, \cdots, N_h\}$, as well as all the biases $\{b_{2,i} | i = 0, \cdots, 2N_{tc}\}$, play a crucial role in establishing the nonlinear regression relationship between $\boldsymbol{\beta}(\boldsymbol{s})$ and $\boldsymbol{s}$ for the spatiotemporal flow field model. These model parameters are selected as model state variables. And the model state vector, denoted as $\boldsymbol{\alpha}$, consists of all model state variables, represented as follows:

$$\boldsymbol{\alpha} = \left[w_{2,0,1}, \cdots, w_{2,2N_{tc},N_h}, b_{2,0}, \cdots, b_{2,2N_{tc}}\right]^T \tag{7}$$

And the temporal state evolution is assumed to be driven by white noise, and it conforms to the following equation:

$$\boldsymbol{\alpha}_k^f = \boldsymbol{\alpha}_{k-1}^a + \boldsymbol{\eta}_k \tag{8}$$

In this equation, $\boldsymbol{\alpha}_{k-1}^a$ represents the posterior analysis state at the previous time step t_{k-1}, which is the estimated state based on the available observations up to time step t_{k-1}, $\boldsymbol{\alpha}_k^f$ represents the forecast state at time step t_k based on $\boldsymbol{\alpha}_{k-1}^a$, and $\boldsymbol{\eta}_k$ represents the Gaussian white process noise at time step t_k.

3.2. Observation Model

This section describes the observation model for the implementation of the online learning with the sequential Bayesian filtering method, which establishes the relationship between the observed flow speeds from AMVs and the underlying state variables.

Assuming that there are N_s flow speed observations $\left\{y_1, \cdots, y_{N_s}\right\}$ at N_s spatial locations $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{N_s}\}$ by N_s AMVs, and the N_s-dimensional observation vector $\boldsymbol{y}_k = \left[y_1, \cdots, y_{N_s}\right]^T$ at time t_k for all the N_s spatial locations is given by:

$$\boldsymbol{y}_k = \boldsymbol{u}_k + \boldsymbol{\varepsilon}_k \tag{9}$$

where $\boldsymbol{u}_k = [u(\boldsymbol{s}_1, t_k), \cdots, u(\boldsymbol{s}_{N_s}, t_k)]^T$ is the true flow speed at time t_k for all N_s spatial locations, and $\boldsymbol{\varepsilon}_k$ is the observation noise at t_k. $\boldsymbol{\varepsilon}_k$ follows a Gaussian distribution with zero-mean vector and covariance matrix $\sigma_\varepsilon^2\boldsymbol{I}$, where σ_ε^2 is the observation noise variance, and $\boldsymbol{I}$ is the N_s-dimensional identity matrix.

In Equation (9), the mean component $\boldsymbol{\mu}_k$ of $\boldsymbol{u}_k$ can be expressed as follows:

$$\boldsymbol{\mu}_k = \boldsymbol{H}_k\boldsymbol{\alpha}_k \tag{10}$$

where $\boldsymbol{H}_k$ is the observation matrix. The observation matrix maps the flow from the model state space to the observation space at t_k for all N_s spatial observation locations and is defined based on the architecture of the neural network as follows:

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{\hbar}(\boldsymbol{s}_1) & \boldsymbol{\mathscr{H}}_1(\boldsymbol{s}_1, t_k) & \cdots & \boldsymbol{\mathscr{H}}_{2N_{tc}}(\boldsymbol{s}_1, t_k) & \boldsymbol{\psi}^T(t_k) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{\hbar}(\boldsymbol{s}_{N_s}) & \boldsymbol{\mathscr{H}}_1(\boldsymbol{s}_{N_s}, t_k) & \cdots & \boldsymbol{\mathscr{H}}_{2N_{tc}}(\boldsymbol{s}_{N_s}, t_k) & \boldsymbol{\psi}^T(t_k) \end{bmatrix} \tag{11}$$

where $\boldsymbol{\hbar}(\boldsymbol{s}_i) = \left[h_1(\boldsymbol{s}_i), \cdots, h_{N_h}(\boldsymbol{s}_i)\right]$, and $\boldsymbol{\mathscr{H}}_j(\boldsymbol{s}_i, t_k) = \boldsymbol{\hbar}(\boldsymbol{s}_i)\psi_j(t_k)$, in which $i = 1, \cdots, N_s$ and $j = 1, \cdots, 2N_{tc}$.

Based on the above equations, the measurement equation can be expressed as follows:

$$\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{\alpha}_k + \boldsymbol{r}_k + \boldsymbol{\varepsilon}_k \tag{12}$$

3.3. Filtering with ETKF

Based on the structure of the presented flow model and the need for model learning and flow prediction using flow speed data from AMVs, this article employs the Kriged Kalman filter method [34] to implement the data assimilation and flow prediction. The Kriged Kalman Filter integrates Kriging's spatial interpolation capabilities and the Kalman filter's optimal state estimation to effectively combine spatial correlations and temporal dynamics. In the DDDAS system, the data assimilation process not only improves flow predictions but also supports the optimization of AMVs' observation strategies. To achieve this, the Kriged ETKF is introduced. The implementation of the Kriged ETKF is detailed as follows, and the optimization strategy which further enhances the model's predictive capabilities will be elaborated on in Section 4.

In the ensemble-based filtering approach, a total of N_{em} ensemble members are considered. The initial model state vector is perturbed to generate the initial ensemble of the model state vectors $\left\{\boldsymbol{\alpha}_0^{(i)} \middle| i = 1, \cdots, N_{em}\right\}$. These perturbations introduce variability and diversity in the initial state estimates, which is crucial for robust data assimilation. The forecast equation then drives the evolution of each ensemble member over time as follows:

$$\boldsymbol{\alpha}_k^{f,(i)} = \boldsymbol{\alpha}_{k-1}^{a,(i)} + \boldsymbol{\eta}_k^{(i)} \quad i = 1, \cdots, N_{em} \tag{13}$$

where $\left\{\boldsymbol{\alpha}_k^{f,(i)} \middle| i = 1, \cdots, N_{em}\right\}$ represents the forecast ensemble at t_k, $\left\{\boldsymbol{\alpha}_{k-1}^{a,(i)} \middle| i = 1, \cdots, N_{em}\right\}$ is the perturbed posterior analysis ensemble at t_{k-1}, and $\left\{\boldsymbol{\eta}_k^{(i)} \sim \mathcal{N}_{N_\alpha}(\mathbf{0}, \boldsymbol{\tau}) \middle| i = 1, \cdots, N_{em}\right\}$ is the ensemble of forecast noise at t_k, where $N_\alpha = (2N_{tc} + 1) \times (N_h + 1)$ is the total number of model state variables. This noise follows a Gaussian distribution with a zero-mean vector and a covariance matrix $\boldsymbol{\tau}$.

By subtracting the forecast ensemble mean from each ensemble member, the forecast ensemble perturbations matrix at t_k can be obtained as follows [35]:

$$\boldsymbol{Z}_k^f = \frac{1}{\sqrt{N_{em} - 1}} \left[\boldsymbol{\alpha}_k^{f,(1)} - \overline{\boldsymbol{\alpha}}_k^f, \cdots, \boldsymbol{\alpha}_k^{f,(N_{em})} - \overline{\boldsymbol{\alpha}}_k^f\right] \tag{14}$$

where $\overline{\boldsymbol{\alpha}}_k^f$ is the mean forecast state at t_k, which is defined as the ensemble mean of the forecast state as follows:

$$\overline{\boldsymbol{\alpha}}_k^f = \frac{1}{N_{em}} \sum_{i=1}^{N_{em}} \boldsymbol{\alpha}_k^{f,(i)} \tag{15}$$

The forecast error covariance matrix at t_k can be obtained as follows:

$$\begin{aligned} \boldsymbol{P}_k^f &= \tfrac{1}{N_{em}-1} \sum_{i=1}^{N_{em}} \left(\boldsymbol{\alpha}_k^{f,(i)} - \overline{\boldsymbol{\alpha}}_k^f\right)\left(\boldsymbol{\alpha}_k^{f,(i)} - \overline{\boldsymbol{\alpha}}_k^f\right)^T \\ &= \boldsymbol{Z}_k^f \boldsymbol{Z}_k^{fT} \end{aligned} \tag{16}$$

where $\boldsymbol{P}_k^f$ represents the forecast error covariance matrix at time t_k, and $\boldsymbol{Z}_k^f$ is the forecast ensemble perturbations matrix as defined in Equation (14). The forecast error covariance matrix provides a measure of uncertainty in the forecast state, which is essential for accurately estimating the flow field and making reliable predictions.

The mean posterior analysis state at time t_k can be obtained as follows:

$$\overline{\boldsymbol{\alpha}}_k^a = \overline{\boldsymbol{\alpha}}_k^f + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{H}_k \overline{\boldsymbol{\alpha}}_k^f\right) \tag{17}$$

where $\boldsymbol{K}_k$ represents the Kalman gain matrix at time t_k as follows:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^f \boldsymbol{H}_k^T \left(\boldsymbol{H}_k \boldsymbol{P}_k^f \boldsymbol{H}_k^T + \boldsymbol{\Sigma}_k\right)^{-1} \tag{18}$$

and $\boldsymbol{\Sigma}_k$ is the observation error covariance matrix at time t_k given by:

$$\boldsymbol{\Sigma}_k = COV(\boldsymbol{r}_k, \boldsymbol{r}_k) + \sigma_\varepsilon^2 \boldsymbol{I} \tag{19}$$

where $\boldsymbol{r}_k = [r(\boldsymbol{s}_1, t_k), \cdots, r(\boldsymbol{s}_{N_s}, t_k)]^T$ is the residual vector at time t_k, and $\boldsymbol{I}$ is the N_s-dimensional identity matrix.

The ETKF [35] introduces a transformation matrix that efficiently transforms the forecast ensemble perturbations into the posterior analysis ensemble perturbations, thereby reducing the complexity of the posterior analysis state calculation. At time t_k, the transformation matrix is given by:

$$\boldsymbol{T}_k = \boldsymbol{C}_k(\boldsymbol{\Gamma}_k + \boldsymbol{I})^{-1/2} \tag{20}$$

where $\boldsymbol{C}_k$ is a matrix containing all the orthogonal eigenvectors of the matrix $\boldsymbol{Z}_k^{f\,T}\boldsymbol{H}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{H}_k\boldsymbol{Z}_k^f$, $\boldsymbol{\Gamma}_k$ is a diagonal matrix with all corresponding eigenvalues as its main diagonal elements, and $\boldsymbol{I}$ is the N_{em}-dimensional identity matrix. This transformation matrix plays a crucial role in the data assimilation process by efficiently adjusting the forecast ensemble perturbations to be consistent with the observations, leading to more accurate and reliable posterior analysis state estimates.

The posterior analysis ensemble perturbations matrix at time t_k can be obtained by applying the transformation matrix $\boldsymbol{T}_k$ as follows [35]:

$$\boldsymbol{Z}_k^a = \boldsymbol{Z}_k^f\boldsymbol{T}_k \tag{21}$$

By adding the mean posterior analysis state to each column of the posterior analysis ensemble perturbation matrix $\boldsymbol{Z}_k^a$, the perturbed posterior analysis ensemble at time t_k can be obtained as follows:

$$\left\{\boldsymbol{\alpha}_k^{a,(i)} = \overline{\boldsymbol{\alpha}}_k^a + \sqrt{N_{em}-1}\boldsymbol{Z}_k^{a,(i)}\middle| i = 1,\cdots,N_{em}\right\} \tag{22}$$

The posterior analysis error covariance matrix at t_k can be obtained as follows:

$$\begin{aligned}\boldsymbol{P}_k^a &= \tfrac{1}{N_{em}-1}\textstyle\sum_{i=1}^{N_{em}}\left(\boldsymbol{\alpha}_k^{a,(i)} - \overline{\boldsymbol{\alpha}}_k^a\right)\left(\boldsymbol{\alpha}_k^{a,(i)} - \overline{\boldsymbol{\alpha}}_k^a\right)^T \\ &= \boldsymbol{Z}_k^a\boldsymbol{Z}_k^{aT}\end{aligned} \tag{23}$$

Based on the above calculations, the posterior analysis of the flow and the corresponding uncertainty for observed spatial locations under given observations at time t_k can be obtained, respectively, as follows:

$$\boldsymbol{u}_k^a = \boldsymbol{H}_k\overline{\boldsymbol{\alpha}}_k^a \tag{24}$$

$$COV(\boldsymbol{u}_k^a|\boldsymbol{y}_k) = \boldsymbol{H}_k\boldsymbol{P}_k^a\boldsymbol{H}_k^T + \boldsymbol{\Sigma}_k \tag{25}$$

3.4. Flow Prediction

After obtaining the posterior analysis state by assimilating the sensing data of flow speeds, the Kriging method is employed to further estimate the flow speeds at unobserved locations.

Let $\{\boldsymbol{s}_i^*|i = 1,\cdots,N_{s^*}\}$ represent the collection of spatial locations without sensing data, and $\boldsymbol{r}_k^* = \left[r(\boldsymbol{s}_1^*,t_k),\cdots,r\left(\boldsymbol{s}_{N_{s^*}}^*,t_k\right)\right]^T$ be the corresponding residual components of the model at time t_k, where N_{s^*} is the total number of unobserved locations considered for flow prediction. Define $\boldsymbol{\Sigma}_k^* = COV(\boldsymbol{r}_k,\boldsymbol{r}_k^*)$ as the $N_s \times N_{s^*}$-dimensional matrix of covariances of residual components of the model evaluated at all pairs of spatial observed and unobserved locations at t_k, and $\boldsymbol{\Sigma}_k^{**} = COV(\boldsymbol{r}_k^*,\boldsymbol{r}_k^*)$ as the $N_{s^*} \times N_{s^*}$-dimensional matrix of covariances of residual components of the model evaluated at all pairs of spatial unobserved locations at time t_k. The observation matrix at time t_k for all spatial unobserved locations is defined as follows:

$$\boldsymbol{H}_k^* = \begin{bmatrix} \boldsymbol{\hbar}(\boldsymbol{s}_1^*) & \boldsymbol{\mathscr{H}}_1(\boldsymbol{s}_1^*,t_k) & \cdots & \boldsymbol{\mathscr{H}}_{2N_{tc}}(\boldsymbol{s}_1^*,t_k) & \boldsymbol{\psi}^T(t_k) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{\hbar}\left(\boldsymbol{s}_{N_{s^*}}^*\right) & \boldsymbol{\mathscr{H}}_1\left(\boldsymbol{s}_{N_{s^*}}^*,t_k\right) & \cdots & \boldsymbol{\mathscr{H}}_{2N_{tc}}\left(\boldsymbol{s}_{N_{s^*}}^*,t_k\right) & \boldsymbol{\psi}^T(t_k) \end{bmatrix} \tag{26}$$

Based on the abovementioned definition, the posterior analysis of the flow and the corresponding covariance for all unobserved spatial locations under given observations at time t_k can be obtained, respectively, as follows [34]:

$$\boldsymbol{u}_k^{*,a} = \boldsymbol{H}_k^{*}\overline{\boldsymbol{\alpha}}_k^{a} + \boldsymbol{\Sigma}_k^{*T}\boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{H}_k\overline{\boldsymbol{\alpha}}_k^{a}\right) \tag{27}$$

$$COV\left(\boldsymbol{u}_k^{*,a}\middle|\boldsymbol{y}_k\right) = \boldsymbol{\Sigma}_k^{**} - \boldsymbol{\Sigma}_k^{*T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Sigma}_k^{*} + \boldsymbol{\Phi}_k\boldsymbol{P}_k^{a}\boldsymbol{\Phi}_k^{T} \tag{28}$$

where $\boldsymbol{u}_k^{*} = \left[u\left(\boldsymbol{s}_1^{*}, t_k\right), \cdots, u\left(\boldsymbol{s}_{N_s^{*}}^{*}, t_k\right)\right]^T$, and $\boldsymbol{\Phi}_k = \boldsymbol{H}_k^{*} - \boldsymbol{\Sigma}_k^{*T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{H}_k$.

3.5. Estimation of Hyper-Parameters

To achieve flow prediction, it is essential to estimate the hyper-parameters in the residual model at each data assimilation time. In this article, the maximum likelihood estimation method is employed to estimate the model hyper-parameters using the flow speed data collected by the AMVs at time t_k.

According to Equation (12), the observed flow speed vector $\boldsymbol{y}_k$ follows a multivariate normal distribution, i.e., $\boldsymbol{y}_k \sim \mathcal{N}_{N_s}\left(\boldsymbol{H}_k\boldsymbol{\alpha}_k, \boldsymbol{H}_k\boldsymbol{P}_k\boldsymbol{H}_k^T + \boldsymbol{\Sigma}_k\right)$. The likelihood function, denoted as $\mathscr{L}(\cdot)$, with respect to the model hyper-parameters $\boldsymbol{\theta}_k$ at t_k is expressed as follows:

$$\begin{array}{rl} & \mathscr{L}\left(\boldsymbol{\theta}_k\middle|\boldsymbol{y}_k, \overline{\boldsymbol{\alpha}}_k^{f}, \boldsymbol{P}_k^{f}\right) \\ = & \wp\left(\boldsymbol{y}_k\middle|\boldsymbol{\theta}_k, \overline{\boldsymbol{\alpha}}_k^{f}, \boldsymbol{P}_k^{f}\right) \\ = & \frac{1}{(2\pi)^{N_s/2}|\boldsymbol{D}_k|^{1/2}} exp\left\{-\frac{1}{2}\boldsymbol{\xi}_k^T\boldsymbol{D}_k^{-1}\boldsymbol{\xi}_k\right\} \end{array} \tag{29}$$

where $\boldsymbol{D}_k = \boldsymbol{H}_k\boldsymbol{P}_k^{f}\boldsymbol{H}_k^T + \boldsymbol{\Sigma}_k$, $\boldsymbol{\xi}_k = \boldsymbol{y}_k - \boldsymbol{H}_k\overline{\boldsymbol{\alpha}}_k^{f}$, and $\wp(\cdot)$ is the probability density function of the observations. The likelihood function $\mathscr{L}\left(\boldsymbol{\theta}_k\middle|\boldsymbol{y}_k, \overline{\boldsymbol{\alpha}}_k^{f}, \boldsymbol{P}_k^{f}\right)$ quantifies the likelihood of observing the data $\boldsymbol{y}_k$ given the model hyper-parameters $\boldsymbol{\theta}_k$, the mean forecast state $\overline{\boldsymbol{\alpha}}_k^{f}$, and the forecast error covariance matrix $\boldsymbol{P}_k^{f}$. By maximizing this likelihood function, the optimal values for the model hyper-parameters $\boldsymbol{\theta}_k$ could be obtained. The logarithmic likelihood function, denoted as $ln\mathscr{L}$, is commonly used for numerical optimization to solve the optimal values of $\boldsymbol{\theta}_k$, as follows:

$$ln\mathscr{L} = -\frac{1}{2}ln|\boldsymbol{D}_k| - \frac{1}{2}\boldsymbol{\xi}_k^T\boldsymbol{D}_k^{-1}\boldsymbol{\xi}_k - \frac{N_s}{2}ln(2\pi) \tag{30}$$

To find the numerical solution for the model hyper-parameters $\boldsymbol{\theta}_k$, the genetic algorithm is employed to solve the following objective function:

$$\boldsymbol{\theta}_k = \underset{\boldsymbol{\theta}_k \in R^3}{arg\ min}\left(-ln\mathscr{L}\left(\boldsymbol{\theta}_k\middle|\boldsymbol{y}_k, \overline{\boldsymbol{\alpha}}_k^{f}, \boldsymbol{P}_k^{f}\right)\right) \tag{31}$$

By using the genetic algorithm, the numerical solution for the model hyper-parameters $\boldsymbol{\theta}_k$ can be efficiently obtained, enabling accurate estimation of the model parameters and improving the flow prediction performance.

4. Optimization of AMVs' Sensing Paths

In the context of the DDDAS for flow prediction, the effective optimization of the AMVs' sensing paths is of utmost importance. The assimilation of the informative flow-sensing data streams from AMVs enables improved estimation of model parameters, leading to enhanced accuracy in flow prediction. The neural network model serves as the backbone of the flow model, and this section focuses on the optimization of AMVs sensing paths to improve the estimation of the neural network parameters. Through this optimization process, the developed DDDAS is completed, enabling adaptive flow sensing using AMVs.

In the Kriged ETKF data assimilation, the ETKF method provides an effective means to quantify the reduction in error variance of the forecast state variables and to access the impact of different observation schemes. In the article, the optimization is built upon the ETKF method, and two optimization scenarios are investigated. In the first scenario, the optimization focuses on finding optimal AMVs' sensing locations for improving the estimation performance only in the subsequent data assimilation time step. This strategy is relatively straightforward to implement. However, it has a myopic nature, as it only considers the immediate impact on the subsequent time step. On the other hand, the second scenario investigated extends the optimization to consider multiple data-assimilation time steps over future time horizon. Compared to the greedy optimization in the first scenario, the non-myopic optimization problem in the second scenario presents more complexities. This article places a greater emphasis on non-myopic optimization.

This section elaborates on the optimization. Section 4.1 states the optimization problems for the two scenarios. Section 4.2 presents the strategies for solving the optimization problems, with a particular emphasis on the proposed receding horizon strategy for the non-myopic optimization scenario. And Section 4.3 presents a proposed distributed strategy for implementing the MCTS method to solve the receding horizon optimization.

4.1. Problem Statement

In the Kriged ETKF data assimilation described in Section 3, the signal covariance matrix at time t_k, denoted as $\boldsymbol{S}_k$ and defined in Equation (32), as follows [35]

$$\boldsymbol{S}_k = \boldsymbol{P}_k^f - \boldsymbol{P}_k^a = \boldsymbol{Z}_k^f \boldsymbol{C}_k \boldsymbol{\Gamma}_k (\boldsymbol{\Gamma}_k + \boldsymbol{I})^{-1} \boldsymbol{C}_k^T \boldsymbol{Z}_k^{fT} \tag{32}$$

provides a means to assess the impact of assimilating observational information from various observation schemes $\boldsymbol{H}_k(\boldsymbol{\mathcal{S}}_k)$ at time t_k on the reduction in error variance of the forecast state variables and posterior analysis, where $\boldsymbol{\mathcal{S}}_k = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{N_s}\}$ represents the spatial locations of N_s AMVs. This difference represents the information gained from the observations, which leads to an improvement in state estimation.

The trace of $\boldsymbol{S}_k$, denoted as $tr\{\boldsymbol{S}_k\}$, serves as a measure of the improvement in the model's state estimation when assimilating the flow-sensing data $\boldsymbol{y}_k = \left[y_1, \cdots, y_{N_s}\right]^T$ obtained at $\boldsymbol{\mathcal{S}}_k$. A larger value of $\boldsymbol{S}_k$ indicates a more significant reduction in error variance, suggesting that the assimilated flow-sensing data have a substantial positive impact on enhancing the accuracy of the model estimation. Therefore, in this article, the optimization of the AMVs' flow-sensing paths is based on maximizing the trace of the signal covariance matrix.

For the first scenario of greedy optimization performed at time t_k, the optimization of AMVs' sensing locations to be reached at time t_{k+1} is expressed as follows:

$$\boldsymbol{\mathcal{S}}_{LO}^* = \underset{\boldsymbol{\mathcal{S}}_{LO} \in RS_{k+1}(\boldsymbol{\mathcal{S}}_k)}{arg\ max} \quad tr(\boldsymbol{S}_{k+1}) \tag{33}$$

In this expression, $\boldsymbol{\mathcal{S}}_{LO}^*$ represents the optimized locations, and $RS_{k+1}(\boldsymbol{\mathcal{S}}_k)$ represents the set of possible sensing locations for the AMVs at time t_{k+1}, which are constrained to be within the reachable region determined by the previous sensing locations $\boldsymbol{\mathcal{S}}_k$ at time t_k. The objective of the optimization is to find $\boldsymbol{\mathcal{S}}_{LO}^*$ to maximize the data assimilation performance at time t_{k+1}. By implementing the optimization continuously at each data assimilation time step, the DDDAS forms the optimal paths of AMVs that consistently improve the data assimilation performance.

For the second scenario of non-myopic optimization performed at time t_k, the objective is to find the optimal paths of AMVs, covering multiple time steps from time t_{k+1} to time

t_{k+N_T}, with the aim to enhance the data assimilation performance during the time step t_{k+1} to a user-defined future time step t_{k+N_T}. The optimization problem is expressed as follows:

$$\boldsymbol{\mathcal{S}}_{PA}^{*} = \underset{\boldsymbol{\mathcal{S}}_{PA} \in RS_{k+N_T}(\boldsymbol{\mathcal{S}}_k)}{arg\ max} \sum_{i=1}^{N_T} tr(w_{k+i}\mathbf{S}_{k+i}) \tag{34}$$

In this expression, $\boldsymbol{\mathcal{S}}_{PA}^{*}$ represents the optimized paths from time t_{k+1} to time t_{k+N_T}, where N_T is the number of time steps in the optimization horizon. The weight w_{k+i} reflects the user's preference for each time step from t_{k+1} to t_{k+N_T}. $RS_{k+N_T}(\boldsymbol{\mathcal{S}}_k)$ represents the set of possible sensing paths for the AMVs from time t_{k+1} to time t_{k+N_T}. These paths are constrained to be within the reachable region determined by the sensing locations $\boldsymbol{\mathcal{S}}_k$ at time t_k.

In the optimization process of AMVs' sensing paths, $\mathbf{S}_{k+1}$ may consist of two components, one for the latitudinal direction and the other for the longitudinal direction, calculated using the two submodels. This can be expressed as $\mathbf{S}_{k+1} = w_{lat}\mathbf{S}_{k+1}^{lat} + w_{lon}\mathbf{S}_{k+1}^{lon}$, where w_{lat} and w_{lon} are user-defined weights with the constraint that $w_{lat} + w_{lon} = 1$. The choice of these weights depends on the specific objectives and requirements of the mission.

4.2. Strategies for Solving the Optimization Problems

In the implementation of the optimization in this article, the AMVs are assumed to navigate at a constant speed relative to the sea bottom, and the optimization focuses on finding the optimal commanded heading angles for the AMVs, i.e., the commanded heading angles are selected as the decision variables. With the optimized commanded heading angles, the AMVs could navigate to desired flow-sensing locations using guidance and control algorithms [36]. To solve the optimization problems, the commanded heading angle of an AMV is discretized by dividing the range of heading angles into N_H equal parts. The heading angle $i \times 360°/N_H$ is selected as one of the possible decision choices, where i ranges from 1 to N_H. The user defined constant value of N_H determines the level of discretization. Selecting a higher value will achieve fine-grained exploration of heading angle options, with an increase in computational costs.

For the optimization problem in Equation (33), the optimal solution can be obtained by evaluating the objective function for each combination of the discretized heading angles of AMVs and selecting the one that maximizes the $tr(\mathbf{S}_{k+1})$. For a small number of possible combinations, an exhaustive search could be performed. For a large number of possible combinations, an exhaustive search becomes computationally expensive and time-consuming. In such cases, this article employs the genetic algorithm to efficiently implement the optimization.

For the optimization problem in Equation (34), the objective is to find the optimal sequences of commanded heading angles for the AMVs, which will determine the paths of the AMVs over multiple time steps. The temporal sequence of the possible combinations of commanded heading angles of AMVs forms a tree structure, where each node in a layer represents a possible combination of commanded heading angles of the AMVs at a specific time step. The tree expands from one time step to the next, representing the different possibilities for AMVs' paths over the optimization horizon. To efficiently search for the optimal solution in the tree structure, this article employs the MCTS method [37]. MCTS is a heuristic search method commonly used in decision-making problems, particularly in scenarios with large search spaces. It is suitable for the optimization problem addressed in this article.

Considering the presence of uncertainties in the flow model, real flow conditions, and the AMVs' navigation and control, this article employs the receding horizon optimization to implement the path planning considering multiple time steps. In this strategy, the optimization is performed at time t_k using the MCTS method, and the computed commanded heading angles for time t_{k+1} are provided to the AMVs for implementation. This approach implements the optimization process repeatedly at each time step t_k. By this approach, the

AMVs continuously update their paths based on the most recent information, making the flow sensing and prediction capabilities more adaptive and robust.

It should be noted that the receding horizon strategy could be implemented in two different ways to achieve continuous AMV paths and optimize the data assimilation performance over a future time window, based on different mission requirements. In the first approach, the process involves repeating the optimization with the same time horizon at each data assimilation–optimization time step, i.e., at each time t_k, the optimization of $\mathcal{S}_{PA}^*$ in Equation (34) is implemented. This results in continuous AMV paths, where the AMVs' locations at each time step are chosen to be optimal for the future N_T time steps. In contrast, in the second approach, the paths of the AMVs are optimized for a fixed user-defined time window, and at each time step the optimization only considers the remaining time steps within the window. Specifically, at one time t_k, the optimization of maximization of $\sum_{i=1}^{N_T} tr(w_{k+i}\mathbf{S}_{k+i})$ is implemented, whereas at the subsequent time t_{k+1}, the optimization of $\sum_{i=2}^{N_T} tr(w_{k+i}\mathbf{S}_{k+i})$ is implemented, which considers the optimization of the remaining time steps from t_{k+2} to t_{k+N_T}. This optimization process terminates at t_{k+N_T-1}, resulting in continuous AMVs paths terminated at t_{k+N_T}. And the AMVs' locations at each time step are chosen to be optimal for the future remaining time steps within the time window.

4.3. A Distributed Strategy for Implementing MCTS

When implementing MCTS on a single processor with limited computational capability to solve large search problems within a constrained decision-making time frame, obtaining an optimal solution may be difficult or impractical. To tackle this issue, and considering the operation scenario of the DDDAS where the computation is implemented on all N_s AMVs without the support of onshore stations, this article proposes a distributed strategy to implement the optimization process. This strategy involves partitioning the search tree into subtrees and assigning each subtree to a different computing device on AMVs for simultaneous exploration. By conducting parallel searches on individual subtrees simultaneously and combining their results, the strategy can leverage the collective computational power of multiple devices of the networked AMVs, facilitating an efficient and optimal decision-making process.

The proposed distributed strategy partitions the original search tree into N_s subtrees. Each subtree is then assigned to an AMV for independent execution. The final result is obtained by combining the N_s outcomes of all subtrees. The initial state of the root nodes $node_{sub}^{i}$ of the subtrees, $i = 1, 2, \cdots, N_s$, is the same as that of the root node $node_r$ in the original tree. The number of child nodes for each $node_{sub}^{i}$ is equal to N_a / N_s, where N_a denotes the number of child nodes of each node in the original search tree.

After splitting the tree, all subtrees are executed simultaneously on the corresponding computing devices. Once all the subtree searches are completed, the final optimal decision is determined by selecting the decision with the highest quality among the best child nodes of each subtree, as follows:

$$a^* = \underset{a^{i^*}}{arg\,max} q^{i^*} \quad i = 1, 2, \cdots, N_s \tag{35}$$

where a^{i^*} represents the decision of the best child of $node_{sub}^{i}$, and q^{i^*} represents the quality of that child.

Existing distributed MCTS methods often necessitate frequent information interactions between multiple threads [38,39], and thus may not be suitable for scenarios like distributed AMV networks with wireless communication, where frequent communication among network nodes is unfavorable or impractical. The proposed distributed strategy in this article requires minimal communication between devices, limited to allocating subtrees and collecting results for determining the final optimal decision. With low data transmission needs, it is well-suited for scenarios with limited wireless communication capabilities, such as the AMV networks studied in this article.

In the implementation of the MCTS, to balance exploitation and exploration in the search process, nodes are selected based on the UCB1 (upper confidence bound 1) method [40] commonly used in MCTS. By using the UCB1, the MCTS method ensures a balance between exploring promising nodes and exploiting nodes with high estimated quality.

5. Simulation Results

In this article, a comprehensive simulation study is conducted to validate and evaluate the performance of the developed DDDAS, which is presented in this section.

Section 5.1 describes the simulation scenario, where four autonomous underwater gliders are utilized to predict the depth-averaged flow in nearshore ocean environments with tidal features. These gliders serve as the flow-sensing network, and their paths are optimized to collect informative flow-sensing data. Section 5.2 presents a comparison of three flow prediction approaches: (1) using the neural network alone; (2) incorporating the data assimilation into the neural network; and (3) further incorporating the Gaussian process. This comparison enables an assessment of the improvement in flow prediction accuracy with each successive approach. Section 5.3 focuses on comparing three glider sensing strategies: virtually moored at fixed locations, random motion, and optimized motion. The accuracy of flow prediction is evaluated under each strategy, revealing the impact of gliders' flow-sensing locations on the performance of flow prediction. Lastly, in Section 5.4, the results of implementing the receding horizon optimization with the distributed MCTS are presented, comparing the results of greedy optimization and non-myopic optimization and demonstrating the utilization of multiple AMVs' computational power for decision-making, when the computation of the DDDAS is implemented on the AMVs without support from onshore stations.

5.1. Simulation Scenario

Underwater gliders are highly valuable autonomous mobile platforms for large-scale and long-term ocean observation. Driven by buoyancy engines, gliders commonly navigate at relatively slow speeds, thus their motion can be significantly influenced by ocean flow. Therefore, accurate depth-averaged flow prediction information is crucial for navigation and control of underwater gliders. In the simulation study of the developed DDDAS in this article, the underwater gliders are employed as the sensing network, and their paths are optimized to collect flow-sensing data for predicting the depth-averaged flow in the nearshore ocean environment.

In the simulation, a designated region of $1^\circ \times 1^\circ$ in the South China Sea is chosen as the operational area. Within this operational area, four underwater gliders are deployed to navigate and perform the sensing mission. For the simulation, the depth-averaged flow data predicted by a numerical ocean model of POM (Princeton Ocean Model) during March and June 2019 are employed as the true flow. This flow dataset serves as the basis for supporting the glider sensing and acts as the ground truth for comparison and evaluation. The flow data have a temporal resolution of 1 h and a horizontal spatial resolution of $1/15^\circ$ in both the latitudinal and longitudinal directions. In Figure 4a, the flow field at 31 March 2019, 18:00 (UTC + 08:00) is depicted, where the flow is predicted on the 16 (latitudinal direction) $\times 16$ (longitudinal direction) $= 256$ uniformly distributed grid points. For the initialization of the neural network model, the flow data covering the time range from 1 March 2019, 00:00 to 31 March 2019, 23:00 (UTC + 08:00) is utilized. The tidal analysis method is then applied to this dataset to extract the tidal constituents and a total of primary $N_{tc} = 29$ tidal constituents are selected, resulting in $2N_{tc} = 58$ temporal basis functions and 59 corresponding coefficients taking into account the non-tidal component for the mean flow. Based on these tidal constituents, the neural network model is constructed, with $N_h = 7$ nodes in the hidden layer, leading to a total number of $N_\alpha = 472$ model state variables. It should be noted that the flow data used for the tidal initialization of the neural network with the gradient descent method is subsampled, with a spatial resolution of 0.2°, as shown in Figure 4b. This approach aims to demonstrate the capability of predicting high-resolution flow fields using a model taught with relatively sparse training data.

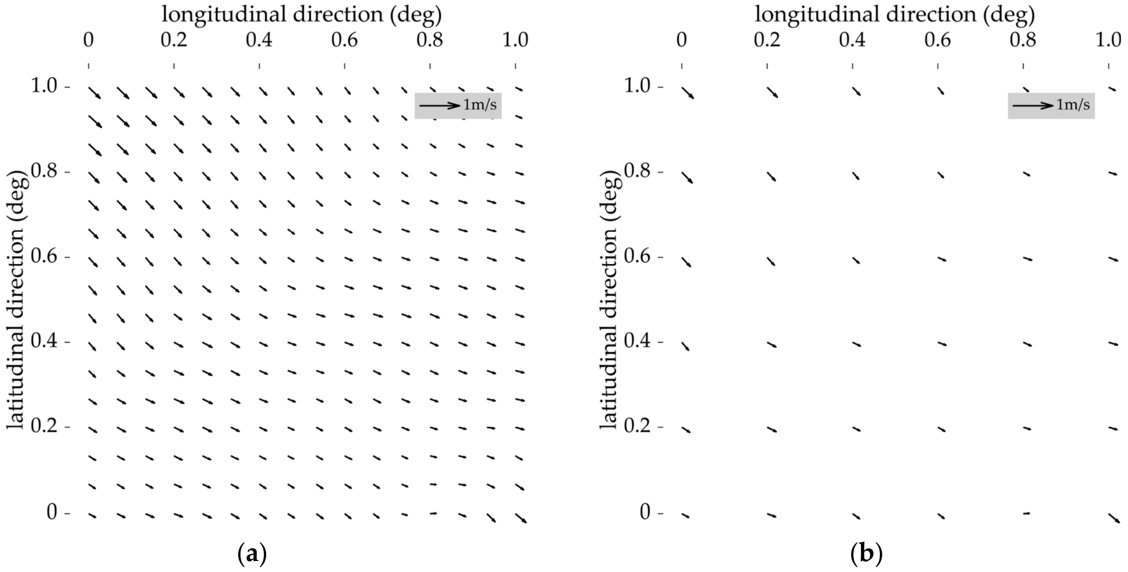


Figure 4. (**a**) The flow field predicted by the POM. (**b**) The subsampled flow field for model initialization.

In the simulation, four underwater gliders are deployed in the operational area, i.e., $N_s = 4$. The gliders are set to navigate at a constant speed of 0.7 kn relative to the sea bottom, and the influence of the flow on the motion of the glider is not considered. Each glider completes one saw-tooth dive cycle in one hour and reports its sensed depth-averaged flow speed upon surfacing. The sensed depth-averaged flow is calculated by linearly interpolating the four predicted flow values around the glider's surfacing location, with the addition of sensing noise having a variance of $\sigma_\varepsilon^2 = 10^{-4}$. The model assimilates the sensing data from the four gliders every hour. During the assimilation, the perturbation covariance matrix of the initial ensemble of the model state vector is set as $10^{-6}\boldsymbol{I}$, and the total number of ensemble members is set as $N_{em} = 100$.

For the optimization of the gliders' paths, both $\boldsymbol{S}_{k+1}^{lat}$ and $\boldsymbol{S}_{k+1}^{lon}$ are considered, and the weights w_{lat} and w_{lon} are both set as 0.5, implying an equal contribution from both components in determining the optimized sensing paths. And two sets of heading angles of $\{0, 45, 90, 135, 180, 225, 270, 325\}$ and $\{0, 90, 180, 270\}$ are used as the decision choices for the greedy and the non-myopic optimization, respectively. In the simulation, the gliders' motion is constrained within the operational area. If a heading angle leads a glider out of the operational area, that specific heading angle is not considered during the optimization process, and the optimal heading angle is selected from the remaining feasible choices to ensure that the glider's navigation remains within the operational area.

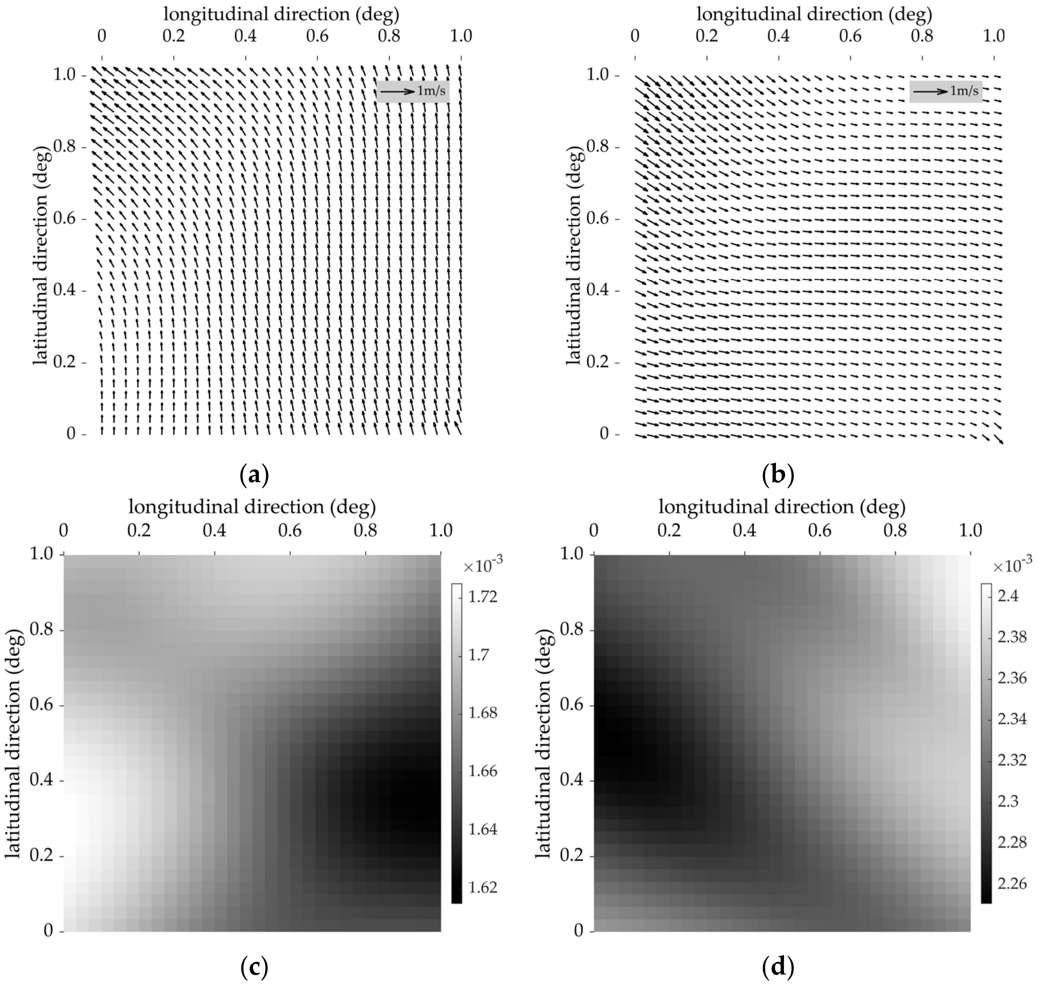
During the simulation, the flow prediction, data assimilation, and adaptive sampling processes are conducted for a total of 129 consecutive hours, specifically from 1 June 2019, 00:00 to 6 June 2019, 08:00 (UTC + 08:00). To simplify the description, the time steps are labeled starting from 0 h, which corresponds to the 1 June 2019, 00:00, and each subsequent time step represents one hour of the simulation. The predicted flow values from the DDDAS are then compared with the true flow values at the 256 grid points in the designated operational area, as shown in Figure 4a. To evaluate the prediction performance, the root mean squared error (RMSE) is calculated for the 256 grid points at each hour, as follows:

$$RMSE(t) = \sqrt{\frac{1}{256}\sum_{i=1}^{256}\left(u^{pre}(\boldsymbol{s}_i, t) - u^{true}(\boldsymbol{s}_i, t)\right)^2} \tag{36}$$

where $u^{pre}(\cdot)$ represents the model's prediction of the flow speed, and $u^{true}(\cdot)$ represents the corresponding true flow speed from the POM dataset at the same spatiotemporal location. The RMSE serves as a metric to measure the accuracy of the flow field prediction.

5.2. Comparision of Three Flow Prediction Approaches

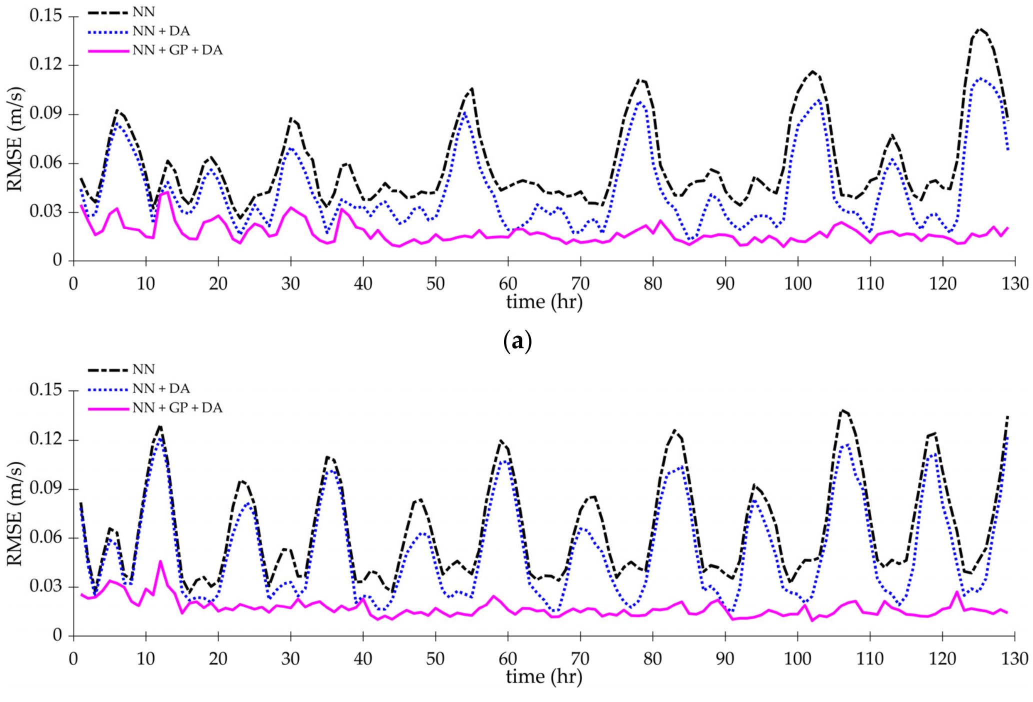
In the comparison study described in this section, the paths of the gliders are optimized using the greedy optimization strategy. Over the 129 consecutive hours, continuous paths of the gliders are formed, and these trajectories will be depicted in Section 5.3. By assimilating the optimized sensing data at each time step, the flow field is predicted and updated at each time step. Figure 5 presents examples of the predicted flow field at the 1st hour and the 9th hour, along with the corresponding variance of the predicted flow speeds in both the latitudinal and longitudinal directions. The variance information is important for many tasks such as the robust path planning or motion control of AMVs. Thanks to the capabilities of the data-driven flow model, flow field predictions could be obtained with any desired level of spatial resolution.



(**a**) (**b**) (**c**) (**d**)

Figure 5. (**a**) Predicted flow field at the 1st hour. (**b**) Predicted flow field at the 9th hour. (**c**) Variance of predicted flow speed along the latitudinal direction at the 9th hour. (**d**) Variance of predicted flow speed along the longitudinal direction at the 9th hour.

To demonstrate the performance of the hybrid data-driven flow model and the data assimilation, the RMSE is compared for the results predicted by the three approaches: (1) using the neural network alone; (2) incorporating the data assimilation into the neural network; and (3) further incorporating the Gaussian process. Figure 6 demonstrates the variations in RMSE over the 129 time steps, where NN, NN + DA, and NN + GP + DA represent the three flow prediction approaches, respectively (NN: neural network alone, NN + DA: neural network with data assimilation, NN + GP + DA: neural network with Gaussian process and data assimilation). Figure 6a presents the RMSE results of the flow speed component in the latitudinal direction, and Figure 6b presents the corresponding RMSE results for the flow speed component in the longitudinal direction. The unit of measurement is meters per second (m/s).



(**a**)

(**b**)

Figure 6. RMSE of the flow field prediction using three different flow prediction approaches. (**a**) RMSE results of the flow speed component along the latitude direction. (**b**) RMSE Results of the flow speed component along the longitude direction.

Table 1 shows the maximum, the minimum, and the maximum and the mean reduction (compared with the NN approach) values of the RMSE shown in Figure 6 for each respective flow prediction approach.

Table 1. Performance of RMSE for three flow prediction approaches, corresponding to Figure 6.

Approach	Direction	Maximum Value	Minimum Value	Max Reduction Value	Mean Reduction Value
NN	Latitudinal Longitudinal	0.143 0.139	0.026 0.027	NA NA	NA NA
NN + DA	Latitudinal Longitudinal	0.112 0.124	0.013 0.015	0.037 0.028	0.016 0.013
NN + GP + DA	Latitudinal Longitudinal	0.043 0.046	0.009 0.010	0.128 0.121	0.042 0.048

The results presented in Figure 6 and Table 1 highlight the significant impact of data assimilation and the Gaussian process on improving flow field predictions. When compared to the predictions without data assimilation, the inclusion of data assimilation leads to a notable reduction in RMSE for both latitudinal and longitudinal flow field predictions. This reduction indicates that data assimilation effectively incorporates observed data from the gliders into the model. Furthermore, the integration of the residual component Gaussian process model with data assimilation yields more accurate predictions. The combination of these techniques contributes to a substantial reduction in RMSE and enhances the model's ability to achieve accurate flow predictions. This study suggests that the hybrid data-driven modeling approach, together with data assimilation, provides an effective framework for obtaining high-resolution and accurate flow field predictions in nearshore ocean environments.

5.3. Comparision of Three Flow Sensing Strategies

In this section, three glider sensing strategies are compared to evaluate their performance in flow prediction. The first strategy involves virtually mooring the gliders at fixed locations to sense the temporal variations in the flow. In the second strategy, the commanded heading angle of each glider is randomly selected from the set of $\{0, 45, 90, 135, 180, 225, 270, 325\}$ at each time step, representing a more exploratory approach. In contrast, the third strategy optimizes the paths of gliders using the greedy optimization method, allowing for adaptive and optimized sampling based on the current situations. In Figure 7, the paths of the gliders generated by the second (random heading angles) and the third (optimized paths) sensing strategies are depicted. The starting locations of the optimized paths are set as the same locations where the gliders are virtually moored for the first sensing strategy.

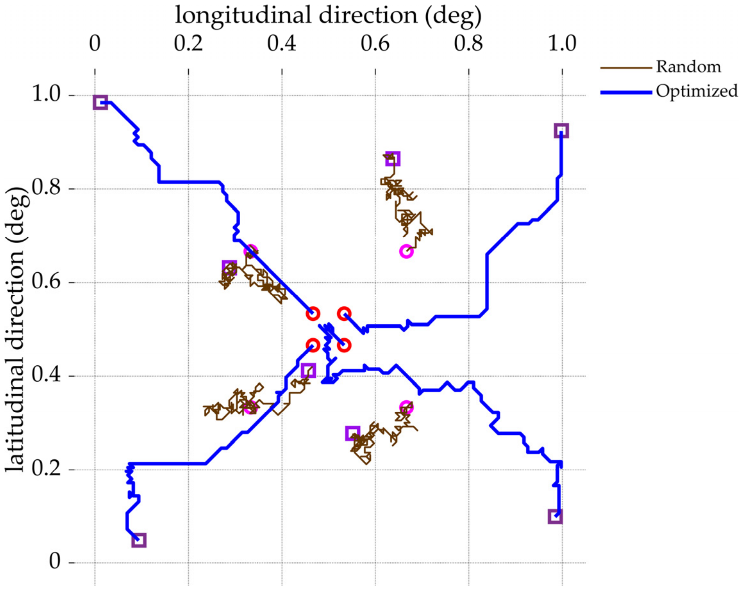
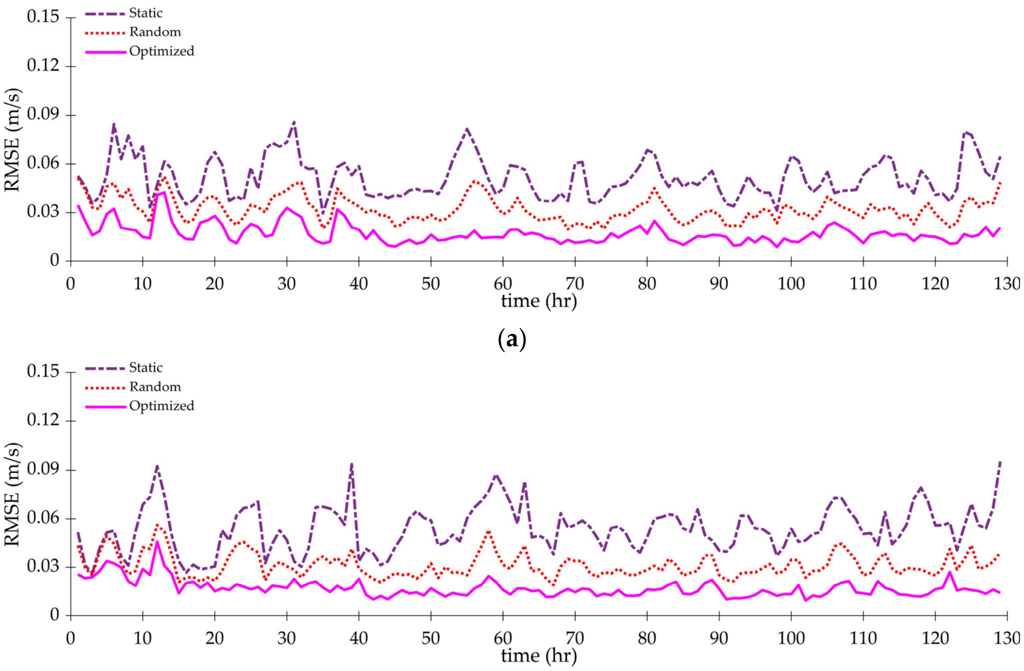


Figure 7. Paths of underwater gliders for the three sensing strategies. The circles represent the start locations, and the rectangles represent the end locations of the glider paths.

The data collected along these paths from the three sensing strategies are then assimilated into the hybrid data-driven flow model, enabling flow field predictions for each strategy at every time step. Figure 8 illustrates the variations in RMSE over the course of 129 consecutive hours, with "static", "random", and "optimized" representing the three sensing strategies, respectively. Specifically, Figure 8a plots the RMSE results of the flow component in the latitudinal direction, while Figure 8b plots the RMSE results of the flow component in the longitudinal direction. The unit of measurement is meters per second (m/s).

Table 2 shows the maximum, the minimum, and the maximum and the mean reduction (compared with the "static" strategy) values of the RMSE shown in Figure 8 for each respective flow sensing strategy.



(**a**)

(**b**)

Figure 8. RMSE of the flow field prediction using three different flow sensing strategies. (**a**) RMSE results of the flow speed component along the latitudinal direction. (**b**) RMSE results of the flow speed component along the longitudinal direction.

Table 2. Performance of RMSE for three flow prediction approaches, corresponding to Figure 8.

Strategy	Direction	Maximum Value	Minimum Value	Max Reduction Value	Mean Reduction Value
Static	Latitudinal Longitudinal	0.086 0.101	0.030 0.027	NA NA	NA NA
Random	Latitudinal Longitudinal	0.052 0.056	0.020 0.019	0.043 0.066	0.019 0.023
Optimized	Latitudinal Longitudinal	0.043 0.046	0.009 0.010	0.067 0.081	0.034 0.037

The results above highlight the significant impact of using mobile and optimized observations in the flow prediction process. When compared to the fixed observation locations, the RMSE of the prediction results using observations from different spatiotemporal locations is notably reduced. More notably, the accuracy of the prediction results corresponding to the optimal observation locations selected by the optimization method is the most promising.

By optimizing data collection locations, more relevant and informative flow-speed data are provided for the data assimilation process, which, in turn, accelerates the optimization of model state variables and enhances the accuracy of continuous flow field predictions. The simulation study validates the utility of the adaptive sensing approach, demonstrating its potential to improve the overall performance of the hybrid data-driven flow model in predicting flow fields.

5.4. Non-Myopic Optimization with Distributed MCTS

In certain applications of the DDDAS, the main objective may be to provide flow information solely for the AMVs themselves to facilitate tasks such as path planning and cooperative control in dynamic flow fields. These scenarios may not involve onshore stations or external infrastructures for support, and as a result, the DDDAS is directly integrated into the networked AMVs. When implementing the DDDAS on AMVs, the optimization considering multi-step information gain using the MCTS approach on a single AMV can become time-consuming. To achieve timely decision-making, this article proposes a distributed strategy to leverage the collective computational power of AMVs to reduce tree-searching time. This section conducts a simulation study on non-myopic optimization using the distributed MCTS approach.

In the simulation, three underwater gliders ($N_s = 3$) are deployed, and the data assimilation–optimization process occurs at a time step of 6 h. The optimization process involves planning gliders' paths within a fixed time window of 18 h, i.e., 3 time steps, with each step aiming to maximize the weighted sum of predicted information gain for future remaining time steps within the fixed time window. This setup corresponds to the second approach discussed in Section 4.2. Specifically, at time t_k, the optimization aims to maximize the weighted sum of $1/6 \times tr(\boldsymbol{S}_{k+6}) + 2/6 \times tr(\boldsymbol{S}_{k+12}) + 3/6 \times tr(\boldsymbol{S}_{k+18})$, prioritizing flow prediction accuracy at time t_{k+18}. Subsequently, at the time t_{k+6}, the optimization shifts to $1/3 \times tr(\boldsymbol{S}_{k+12}) + 2/3 \times tr(\boldsymbol{S}_{k+18})$, and at time t_{k+12}, the optimization considers $tr(\boldsymbol{S}_{k+18})$. The weights are set accordingly to prioritize improving flow prediction accuracy at the targeted time t_{k+18}. At each time step, the distributed MCTS is utilized to solve for the optimal solution, where three subtrees are allocated to the three gliders. Then the receding horizon strategy is implemented, and this optimization process concludes at time t_{k+18}. This process results in continuous paths over 18 h, facilitating the accurate flow field prediction at time t_{k+18}. Figure 9 shows the simulation result of the paths of three gliders from a simulation run.

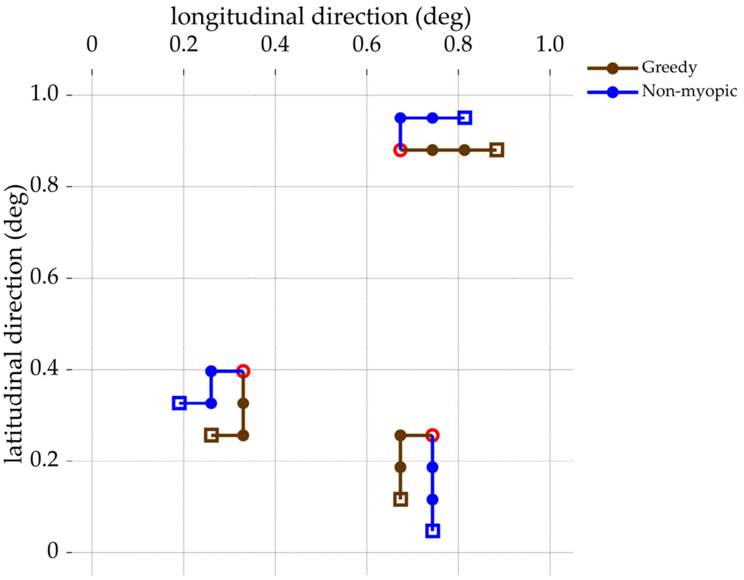


Figure 9. The paths of three underwater gliders correspond to the ID #1 simulation run in Table 3. The circles represent the start locations, the solid dots represent the locations of gliders at the second and the third data assimilation time steps, and the rectangles represent the end locations of paths.

In this simulation, ten simulation runs are performed, each starting from the time t_0. The initial starting locations of the three gliders are randomly set within the operational area. For each run, both the non-myopic optimization and the greedy optimization are performed with the same simulation setup. Figure 9 also shows the simulation result of the paths of three gliders from a greedy optimization. After each simulation run, the $tr\left(\boldsymbol{P}_0^a - \boldsymbol{P}_{18}^a\right)$ is calculated for both the cases of optimization considering greedy and non-myopic information gain. The results are presented in Table 3. Additionally, in the last column of Table 3, the percentage improvement in $tr\left(\boldsymbol{P}_0^a - \boldsymbol{P}_{18}^a\right)$ achieved through the non-myopic optimization compared to the greedy optimization is also provided.

Table 3. The $tr\left(\boldsymbol{P}_0^a - \boldsymbol{P}_{18}^a\right)$ values calculated in ten simulation runs.

ID of Runs	Greedy ($\times 10^{-5}$)	Non-Myopic ($\times 10^{-5}$)	Improvement
1	5.714	6.519	14.1%
2	5.939	6.751	13.7%
3	4.904	5.378	9.7%
4	5.772	6.346	9.9%
5	4.396	4.835	10.0%
6	4.864	5.391	10.8%
7	4.551	5.074	11.5%
8	4.661	5.059	8.5%
9	4.654	5.198	11.7%
10	5.835	6.759	15.8%

Upon analysis of the results, it can be observed that the non-myopic optimization consistently yields higher $tr\left(\boldsymbol{P}_0^a - \boldsymbol{P}_{18}^a\right)$ values compared to the greedy optimization for all ten simulation runs. The improvement percentages range from approximately 9.7% to 15.8%, with an average improvement of around 11.6% across the ten simulation runs. By considering information gains from future time steps, the gliders can better plan their paths and collect more informative data, leading to more accurate flow field predictions at a future targeted time.

The results also validate the effectiveness of the distributed strategy for implementing MCTS, enabling effective decision-making with about 1/3 of computation time compared to the implementation of the MCTS on a single vehicle. This reduction in computation time is crucial for the real-time decision-making of AMVs. It is worth noting that while the distributed MCTS method has demonstrated promising results in this simulation study, there is great potential for further performance improvement. One area of future research work for the authors is to explore adaptively allocating computational budgets to the subtrees, to further enhance the capability of distributed MCTS in obtaining optimal and reliable decisions.

Overall, the consideration of multiple future time steps, the receding horizon optimization, and the distributed MCTS demonstrate a promising approach to optimize the non-myopic paths of AMVs and achieve more accurate flow field predictions for a targeted time.

6. Conclusions

This research article presents a novel data-driven closed-loop dynamic data-driven application system that effectively addresses the challenge of high-resolution and accurate flow field prediction in ocean environments. The DDDAS leverages a hybrid data-driven flow model for trend prediction, along with a Gaussian process model for residual fitting. The assimilation of spatially correlated flow-sensing data from AMVs using the Kriged ETKF further enhances model online learning and achieves accurate spatiotemporal flow prediction. The proposed receding horizon strategy and distributed strategy of implementing Monte Carlo tree search is demonstrated to be effective in optimizing AMVs' coordinated flow-sensing paths, enabling non-myopic path planning, and solving large-scale tree searching-based optimization problems in a timely manner. The proposed DDDAS offers a comprehensive and effective solution for achieving high-resolution and accurate flow field prediction in ocean environments using AMV networks.

As future research directions, several potential avenues can be explored. Firstly, investigating the integration of other machine learning models, such as deep neural networks, and data-driven techniques into the DDDAS may lead to further improvements in computational efficiency and prediction accuracy of complex oceanic processes. Additionally, the application of the DDDAS can be extended to incorporate a wider range of ocean data, such as ocean temperature and salinity, to achieve a more comprehensive prediction of ocean environments. Furthermore, optimizing the number of AMVs for environmental sensing is crucial, as it can lead to ocean prediction with the required level of accuracy with a minimum number of AMVs needed. This optimization has the potential to reduce costs and resource requirements for the system, considering the high costs associated with deploying and operating AMVs in ocean environments.

Author Contributions: Conceptualization, Q.J. and Y.T.; methodology, Q.J. and Y.T.; software, Q.J., W.Z. and Q.S.; validation, Q.J. and Y.T.; formal analysis, Q.J. and Y.T.; writing—original draft preparation, Y.T.; writing—review and editing, Y.T.; visualization, Q.J. and Y.T.; supervision, Y.T., J.Y. and X.W.; project administration, Y.T. and J.Y.; funding acquisition, Y.T. and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (Grant No. 52271353), the Liaoning Revitalization Talents Program, China (Grant No. XLYC2007035), the Natural Science Foundation of Shenyang, China (Grant No. 22-315-6-08), the Program of the State Key Laboratory of Robotics at Shenyang Institute of Automation, Chinese Academy of Sciences (Grant No. 2022-Z11), the International Partnership Program of the Chinese Academy of Sciences (Grant No. 173321KYSB20180011), the Fundamental Research Program of Shenyang Institute of Automation, Chinese Academy of Sciences (Grant No. 2022JC3K05), and the National Key R&D Program of China (Grant No. 2022YFE0204600).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Chen, Z.; Yu, J.; Zhang, A. Overview on observation-oriented unmanned marine vehicles with high cruising ability: Development status and prospect. *J. Ocean Technol.* **2016**, *35*, 122–130.
2. Riser, S.C.; Freeland, H.J.; Roemmich, D.; Wijffels, S.; Troisi, A.; Belbéoch, M.; Gilbert, D.; Xu, J.; Pouliquen, S.; Thresher, A.; et al. Fifteen years of ocean observations with the global Argo array. *Nat. Clim. Chang.* **2016**, *6*, 145–153. [CrossRef]
3. Han, G.; Zhou, Z.; Zhang, T.; Wang, H.; Liu, L.; Peng, Y.; Guizani, M. Ant-Colony-based complete-coverage path-planning algorithm for underwater gliders in ocean areas with thermoclines. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8959–8971. [CrossRef]
4. Petillo, S.; Schmidt, H.; Balasuriya, A. Constructing a distributed AUV network for underwater plume-tracking operations. *Int. J. Distrib. Sens. Netw.* **2012**, *8*, 191235. [CrossRef]
5. Zamuda, A.; Hernández Sosa, J.D. Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures. *Appl. Soft. Comput.* **2014**, *24*, 95–108. [CrossRef]
6. Peng, S.; Zhu, Y.; Li, Z.; Li, Y.; Xie, Q.; Liu, S.; Luo, Y.; Tian, Y.; Yu, J. Improving the real-time marine forecasting of the northern South China Sea by assimilation of glider-observed T/S profiles. *Sci. Rep.* **2019**, *9*, 17845. [CrossRef] [PubMed]
7. Hero, A.O.; Cochran, D. Sensor management: Past, present, and future. *IEEE Sens. J.* **2011**, *11*, 3064–3075. [CrossRef]
8. Zhang, F. Cyber-maritime cycle: Autonomy of marine robots for ocean sensing. *Found. Trends Robot.* **2016**, *5*, 1–115. [CrossRef]
9. Leonard, N.E.; Paley, D.A.; Davis, R.E.; Fratantoni, D.M.; Lekien, F.; Zhang, F. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *J. Field Robot.* **2010**, *27*, 718–740. [CrossRef]
10. Curtin, T.B.; Bellingham, J.G. Progress toward autonomous ocean sampling networks. *Deep Sea Res. Part II Top. Stud. Oceanogr.* **2009**, *56*, 62–67. [CrossRef]
11. Leonard, N.E.; Paley, D.A.; Lekien, F.; Sepulchre, R.; Fratantoni, D.M.; Davis, R.E. Collective motion, sensor networks, and ocean sampling. *Proc. IEEE* **2007**, *95*, 48–74. [CrossRef]
12. L'Hévéder, B.; Mortier, L.; Testor, P.; Lekien, F. A glider network design study for a synoptic view of the oceanic mesoscale variability. *J. Atmos. Ocean. Technol.* **2013**, *30*, 1472–1493. [CrossRef]
13. Fossum, T.O.; Eidsvik, J.; Ellingsen, I.; Alver, M.O.; Fragoso, G.M.; Johnsen, G.; Mendes, R.; Ludvigsen, M.; Rajan, K. Information-driven robotic sampling in the coastal ocean. *J. Field Robot.* **2018**, *35*, 1101–1121. [CrossRef]
14. Gregório Ramos, P.A. Geostatistical prediction of ocean outfall plume characteristics based on an autonomous underwater vehicle. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 289. [CrossRef]
15. Sun, J.; Liu, S.; Zhang, F.; Song, A.; Yu, J.; Zhang, A. A Kriged compressive sensing approach to reconstruct acoustic fields from measurements collected by underwater vehicles. *IEEE J. Ocean. Eng.* **2021**, *46*, 294–306. [CrossRef]
16. Munafò, A.; Simetti, E.; Turetta, A.; Caiti, A.; Casalino, G. Autonomous underwater vehicle teams for adaptive ocean sampling: A data-driven approach. *Ocean Dyn.* **2011**, *61*, 1981–1994. [CrossRef]
17. Grasso, R.; Braca, P.; Fortunati, S.; Gini, F.; Greco, M.S. Dynamic underwater glider network for environmental field estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 379–395. [CrossRef]
18. Yildirim, B.; Chryssostomidis, C.; Karniadakis, G.E. Efficient sensor placement for ocean measurements using low-dimensional concepts. *Ocean Model.* **2009**, *27*, 160–173. [CrossRef]

19. Heuss, J.P.; Haley, P.J.; Mirabito, C.; Coelho, E.; Schönau, M.C.; Heaney, K.; Lermusia, P.F.J. Reduced order modeling for stochastic prediction onboard autonomous platforms at sea. In Proceedings of the Global Oceans 2020: Singapore—U.S. Gulf Coast, Biloxi, MS, USA, 5–30 October 2020.
20. Arcucci, R.; Zhu, J.; Hu, S.; Guo, Y.-K. Deep data assimilation: Integrating deep learning with data assimilation. *Appl. Sci.* **2021**, *11*, 1114. [CrossRef]
21. Yang, Y.; Dong, J.; Sun, X.; Lima, E.; Mu, Q.; Wang, X. A CFCC-LSTM model for sea surface temperature prediction. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 207–211. [CrossRef]
22. Ma, Y.; Mao, Z.; Wang, T.; Qin, J.; Ding, W.; Meng, X. Obstacle avoidance path planning of unmanned submarine vehicle in ocean current environment based on improved firework-ant colony algorithm. *Comput. Electr. Eng.* **2020**, *87*, 106773. [CrossRef]
23. Lolla, T.; Lermusiaux, P.F.J.; Ueckermann, M.P.; Haley, P.J. Time-optimal path planning in dynamic flows using level set equations: Theory and schemes. *Ocean Dyn.* **2014**, *64*, 1373–1397. [CrossRef]
24. Liang, X.; Wu, W.; Chang, D.; Zhang, F. Real-time modelling of tidal current for navigating underwater glider sensing networks. *Procedia Comput. Sci.* **2012**, *10*, 1121–1126. [CrossRef]
25. Chang, D.; Zhang, F.; Edwards, C.R. Real-time guidance of underwater gliders assisted by predictive ocean models. *J. Atmos. Ocean. Technol.* **2015**, *32*, 562–578. [CrossRef]
26. Smith, R.N.; Chao, Y.; Jones, B.H.; Caron, D.A.; Li, P.P.; Sukhatme, G.S. Trajectory design for autonomous underwater vehicles based on ocean model predictions for feature tracking. In *Field and Service Robotics: Results of the 7th International Conference*; Howard, A., Iagnemma, K., Kelly, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 62, pp. 263–273.
27. Lekien, F.; Mortier, L.; Testor, P. Glider coordinated control and Lagrangian coherent structures. *IFAC Proc. Vol.* **2008**, *41*, 125–130. [CrossRef]
28. Chang, D.; Wu, W.; Edwards, C.R.; Zhang, F. Motion tomography: Mapping flow fields using autonomous underwater vehicles. *Int. J. Robot. Res.* **2017**, *36*, 320–336. [CrossRef]
29. Saha, D.; Deo, M.C.; Joseph, S.; Bhargava, K. A combined numerical and neural technique for short term prediction of ocean currents in the Indian Ocean. *Environ. Syst. Res.* **2016**, *5*, 4. [CrossRef]
30. Thongniran, N.; Vateekul, P.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P. Spatio-temporal deep learning for ocean current prediction based on HF radar data. In Proceedings of the 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), Chonburi, Thailand, 10–12 July 2019.
31. Darema, F. Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In *Computational Science—ICCS 2004*; Bubak, M., Albada, G.D., Sloot, P.M.A., Dongarra, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3038, pp. 662–669.
32. Mousavian, R.; Hossainali, M.M. Detection of main tidal frequencies using least squares harmonic estimation method. *J. Geod. Sci.* **2012**, *2*, 224–233. [CrossRef]
33. Amiri-Simkooei, A.R.; Zaminpardaz, S.; Sharifi, M.A. Extracting tidal frequencies using multivariate harmonic analysis of sea level height time series. *J. Geod.* **2014**, *88*, 975–988. [CrossRef]
34. Mardia, K.V.; Goodall, C.; Redfern, E.J.; Alonso, F.J. The Kriged Kalman filter. *Test* **1998**, *7*, 217–282. [CrossRef]
35. Bishop, C.H.; Etherton, B.J.; Majumdar, S.J. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Mon. Weather Rev.* **2001**, *129*, 420–436. [CrossRef]
36. Tian, Y.; Zhang, A. Development of a guidance and control system for an underwater plume exploring AUV. In Proceedings of the 2010 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010.
37. Browne, C.B.; Powley, E.; Whitehouse, D.; Lucas, S.M.; Cowling, P.I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S. A survey of Monte Carlo tree search methods. *IEEE Trans. Comput. Intell. AI Games* **2012**, *4*, 1–43. [CrossRef]
38. Graf, T.; Lorenz, U.; Platzner, M.; Schaefers, L. Parallel Monte-Carlo tree search for HPC systems. In *Euro-Par 2011 Parallel Processing*; Jeannot, E., Namyst, R., Roman, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6853, pp. 365–376.
39. Schaefers, L.; Platzner, M. Distributed Monte Carlo tree search: A novel technique and its application to Computer Go. *IEEE Trans. Comput. Intell. AI Games* **2015**, *7*, 361–374. [CrossRef]
40. Auer, P.; Cesa-Bianchi, N.; Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* **2002**, *47*, 235–256. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.