A Modular Control Aid for Profiling Floats with a Gulf Stream Case Study

Joseph Tolone[1], Trevor Harrison[2], Thomas Curtin[3], Zoltan Szuts[2], and Derek A. Paley[1]
[1]*Dept. Aerospace Engineering and Institute for Systems Research, University of Maryland, College Park, MD*
[2]*Ocean Physics Dept., Applied Physics Laboratory, University of Washington, Seattle, WA*
[3]*Environmental Information Systems, Applied Physics Laboratory, University of Washington, Seattle, WA*

***Abstract*—This work presents a conceptual framework, called FloatCast, for the control of a small fleet of buoyancy-controlled ocean profiling floats. The control objective is to maximize sampling coverage in a given region of interest. The framework optimizes park depth and park duration commands for each float in the fleet. FloatCast uses an Echo State Network to make a sea level anomaly forecast, which is converted into a surface flow forecast. This flow forecast informs a Lagrangian particle model of drifting vehicle dynamics. The state-space model of the float dynamics uses candidate sets of commands to predict float trajectories, which are evaluated using a mapping error scoring metric. Stochastic analysis illustrates a risk-reward trade-off between uncertainty and potential coverage for candidate float commands. This paper introduces each of these components of FloatCast and presents initial simulation results using float data from a deployment in the Gulf Stream from July 2024.**

***Index Terms*—Echo State Network, flow forecasting, Lagrangian particle model, float trajectory forecasting, adaptive sampling, mapping error, fleet optimization.**

I. Introduction

Autonomous ocean sampling vehicles improve the capabilities of researchers to take targeted measurements that can enrich our understanding of ocean behavior. For example, buoyancy-controlled floats like those used in the Argo project [1] provide a relatively cheap option for this purpose. These floats commonly use fixed sampling patterns that lead to random dispersion of the fleet. An alternate approach is to dynamically change the sampling pattern to meet control objectives. To that end, float depth can be manipulated strategically to produce movement in the desired direction. Float deployment management software like FlowPilot [2] enables floats to take advantage of favorable currents to affect desired horizontal movement. Such software automates control of a fleet of floats that can collectively measure regional circulation or other processes that occur over days or weeks. These tools therefore have applications in the monitoring of ocean acoustics, marine heatwaves [3], climate change, and in the study of marine plastics [4].

This work describes a conceptual framework to increase the ability of a small fleet of buoyancy-controlled floats to take targeted measurements via data-driven forecasting. In principle, this is done by strategically selecting float dive commands. However, controlling the lateral movement of floats is challenging because it is situated at the intersection two complex research topics: (1) adaptive ocean sampling and (2) application of numerical prediction models. This work addresses both of these components with a focus on lightweight forecasting. The technical approach to reach the control objective is to combine machine learning for ocean forecasting with a motion prediction model in order to select optimal fleet commands; the custom software that seeks to solve this problem is called *FloatCast.* The architecture of FloatCast is outlined in Fig. 1.

Profiling floats are used most intensively in the Argo project, an international array of floats that covers the entire global ocean [1]. This program chose a fixed sampling pattern, which meets their sampling need to randomly sample the ocean to obtain statistically meaningful observations of changes over decades. Recent studies have investigated the best locations to deploy floats to maintain spatial coverage [5], [6], [7]. A probabilistic approach for understanding expected post-deployment float trajectories has been developed [5]. This methodology informs deployment decisions in areas with incomplete satellite coverage [5]. Similarly, different sampling patterns have been evaluated in a specific region from the simulation of a large number of virtual floats [6]. Lastly, an optimal spatial spread of advanced Argo floats has been proposed [7]. Strategic float deployment and replenishment is emphasized in [7]. For the project goals, Argo has no need for real-time control of floats.

Various techniques have been applied to predict marine vehicle trajectories. A Lagrangian particle model based on typical Argo profiling patterns was used in the probability-based methodologies in [5]. Some fundamental principles of Lagrangian particle models are reviewed in [8]. A transition matrix prediction method similar to the Perron-Frobenius operator[1] has been used to predict the global distribution of Argo floats [10]. Various machine learning techniques have been applied to make trajectory predictions for surface floats based on historical data [11]. Our work uses a Lagrangian particle model in order to be computationally lightweight.

The problem of directing the lateral movement of buoyancy-controlled floats has been addressed [12], [2]. Early work on this float control problem is presented in [12]. Here, a motion planning approach is taken where control decisions are based on the probability of success [12]. Our work focuses on persistence rather than waypoint navigation as in [12]. FlowPilot is foundational to this work, and is introduced in

This work is supported by ONR grant number N000142112434.

[1]For more information on the Perron-Frobenius operator, see [9].

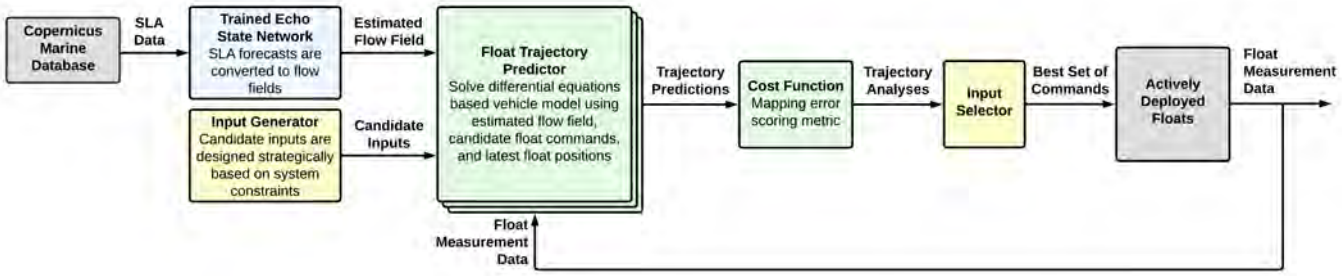

Fig. 1. FloatCast workflow block diagram. There are two main components: the surface flow forecast and the float trajectory predictions. The float trajectory predictor is fed ocean surface flow information from the trained Echo State Network (see Fig. 2) as well as candidate float commands from the input generator. Predicted float trajectories are then evaluated according to the mapping error scoring metric. The commands generating the best set of trajectories are sent to the active floats in the water. Float location data is then fed back into the system.

[2]. FlowPilot is being used in real-world deployments to accomplish float control objectives similar to the ones we discuss here. This is performed using multiple float trajectory prediction models to select optimal commands [2]. In the future, some FloatCast capabilities could be absorbed into FlowPilot.

In concert with our float model, we use an Echo State Network (ESN) to predict ocean surface velocity. ESNs are a data-driven prediction method, and have a variety of uses—some of which are highlighted in the review article by Sun et al. [13]. In one application, an ESN has been used to forecast sea-surface temperature in order to make predictions about El Niño Southern Oscillation [14]. An ESN was also used in the pipeline for predicting the travel of a network of electric scooters in cities [15]. These two examples highlight the versatility of ESNs. Taking a different approach, a system based on a deep neural network has been developed for forecasting the upper ocean [16]. This work forecasts additional ocean features and depths which are out of scope for our current interests.

FloatCast is a modular software that addresses the multifaceted challenge of adaptive ocean sampling with underactuated vehicles. This paper details major components of this software. In particular, the contributions of this paper are (1) an ocean surface velocity forecasting system that uses a custom-trained Echo State Network; (2) a float trajectory forecasting tool based on a Lagrangian particle model of the vehicle dynamics; and (3) a method for generating risk-reward assessments for candidate fleet commands based on mapping error. These components work together to accomplish the objective of controlling underactuated vehicles in a volatile environment. This work uses sea level anomaly and absolute geostrophic velocity data from E.U. Copernicus Marine Service Information [17], [18]. The University of Washington Applied Physics Laboratory managed the float deployment in the Gulf Stream in the central North Atlantic in July 2024.

This paper is organized as follows. Section II provides background information related to key components of the FloatCast workflow. Section III details the float trajectory prediction methodology. Section IV illustrates how the the float trajectory predictions are used to optimize float command selection for a case study in July 2024. Section V summarizes the paper and details directions of ongoing and future work.

II. Background

This section provides background information for major components of FloatCast: the Echo State Network (ESN), float trajectory predictor, and mapping error scoring metric.

A. Echo State Networks

The first step in the float control workflow (see Fig. 1) is the prediction of the ocean surface flow field. In our implementation, we use an Echo State Network (ESN) to forecast Sea Level Anomaly (SLA) based on daily-updated historical data. An ESN is a recurrent neural network used to predict data in a time series [19]. There are several advantages to using an ESN for ocean forecasting. First, ESN parameters do not have to be finely tuned to achieve good performance [19], which increases the versatility of this system for forecasting in various ocean locations. ESN parameters are obtained here by following the tuning guidelines proposed in [19]. Secondly, we take advantage of the computationally efficient training of ESNs to train multiple at once, and then use the forecast from the best one (see Section III-A). This retraining can be done each time the daily ocean data is updated. ESN structure, training, and usage is discussed below based on the work of [19]. Fig. 2 summarizes this description.

The heart of an ESN are the $\mathbf{W}_{\text{in}}$, $\mathbf{W}$, and $\mathbf{W}_{\text{out}}$ matrices. These are the input, reservoir, and output weights matrices, respectively [19]. Following the same order, the dimensionality of these matrices is $N_x \times (1 + N_u)$, $N_x \times N_x$, and $N_y \times (1 + N_u + N_x)$; N_u is the size of an input, N_x is the size of the reservoir, and N_y is the size of an output [19]. For our purposes, N_u and N_y are equivalent, and are equal to the number of SLA data points forecasted. Before training, the $\mathbf{W}_{\text{in}}$ and $\mathbf{W}$ matrices are randomly initialized generally following the guidelines proposed in [19]. The $\mathbf{W}_{\text{out}}$ matrix is trained following the process outlined below. The reader is referred to [19] for more detailed information.

In the first phase of training, daily SLA data is iteratively fed in [19]

$$\mathbf{x}_i = (1 - \varrho)\mathbf{x}_{i-1} + \varrho f(\mathbf{W}\mathbf{x}_{i-1} + \mathbf{W}_{\text{in}}[1; \mathbf{u}_i]) \qquad (1)$$

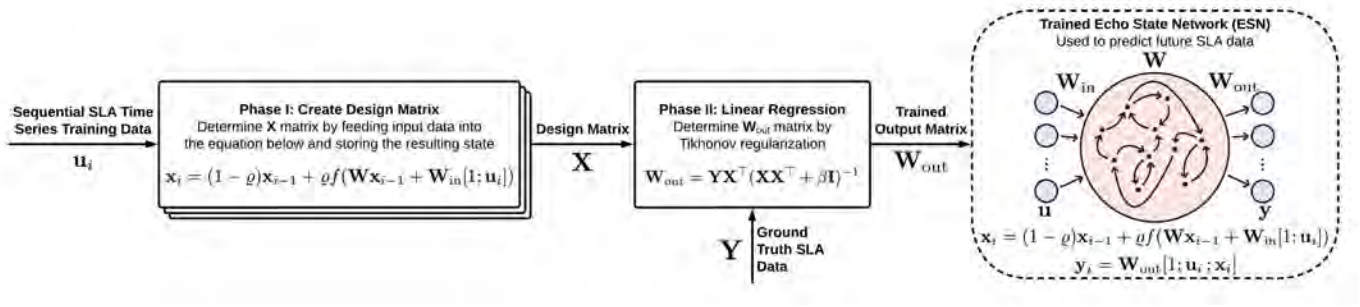

Fig. 2. Method for training the Echo State Network (ESN). The trained ESN is used in the FloatCast workflow to make the Sea Level Anomaly (SLA) forecast, which is converted to a surface flow field (see Fig. 1). The $\mathbf{W}_{\text{in}}$ and $\mathbf{W}$ matrices are randomly initialized prior to training; ϱ and β are scalars; f is the $\tanh$ function. In phase I, the design matrix $\mathbf{X} = \{[1;\mathbf{u}_i\,;\mathbf{x}_i]\}$ is constructed iteratively using input SLA data $\mathbf{u}_i$ and computed reservoir states $\mathbf{x}_i$, for $i = 1, ..., n$. In phase II, Tikhonov regularization is used to solve for $\mathbf{W}_{\text{out}}$ using the design matrix and ground truth SLA data, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ...\mathbf{y}_n\}$. To make predictions with the trained ESN, the equations in the dashed box are used in conjunction with each other. Note that the vector subscript i indicates the time step t_i. This graphic is based on information and illustrations in [15], [19], [20].

Here, $\mathbf{u}_i$ and $\mathbf{x}_i$ are the vectorized SLA data and the reservoir state vector at time step i, respectively. The leaking rate ϱ is a scalar and f is the element-wise $\tanh$ function. Information from each use of (1) is stored in the design matrix, $\mathbf{X} = \{[1;\mathbf{u}_i\,;\mathbf{x}_i]\}$ for $i = 1, ..., n$. In the first use of (1), $\mathbf{x}_0$ is the zero vector. Note that the bracket notation used here indicates a column vector.

The second phase of training directly solves for $\mathbf{W}_{\text{out}}$. Since the states in $\mathbf{X}$ have been organized column-wise according to time step, then (2) must be true for the ideal $\mathbf{W}_{\text{out}}$ matrix, i.e., [19]

$$\mathbf{Y} = \mathbf{W}_{\text{out}}\mathbf{X} \tag{2}$$

Here, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ...\mathbf{y}_n\}$ denotes ground truth output SLA data. In other words, it should ideally be the case that $\mathbf{u}_{i+1} = \mathbf{y}_i$, for $i = 1, ..., n$. Using the design matrix $\mathbf{X}$ and ground truth SLA data $\mathbf{Y}$, the $\mathbf{W}_{\text{out}}$ matrix can be obtained from (2). This is done via Tikhonov regularization as follows:

$$\mathbf{W}_{\text{out}} = \mathbf{Y}\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \beta\mathbf{I})^{-1} \tag{3}$$

where β is a scalar and $\mathbf{I}$ is an identity matrix [19].

With $\mathbf{W}_{\text{in}}$, $\mathbf{W}$, and $\mathbf{W}_{\text{out}}$ now defined, ESN predictions are made by chaining together the output of (1) with the vector form of (2), which yields [19]

$$\mathbf{y}_i = \mathbf{W}_{\text{out}}[1;\mathbf{u}_i\,;\mathbf{x}_i] \tag{4}$$

The process begins by seeding (1) with the newest piece of SLA data $\mathbf{u}_i$, and the last state from the training stage $\mathbf{x}_{i-1}$ to calculate the new state $\mathbf{x}_i$. The output of (1) is then fed into (4), where the resulting $\mathbf{y}_i$ becomes the SLA forecast for the next day. Consequently, this forecast $\mathbf{y}_i$ is then used as input $\mathbf{u}_{i+1}$ in (1) for the next time step forecasted. The training and usage of the ESN are illustrated in Fig. 2.

B. Autonomous Profiling Floats for Ocean Sampling

We seek to improve the collective sampling performance of small fleets of Alto and EM-APEX floats. These floats have no direct horizontal actuation; they change their vertical velocity by adjusting their buoyancy. Ideally, floats will follow the dive routine sketched in Fig. 3, which is similar to the Argo project [1]. On their ascent to the surface, floats measure quantities such as temperature, salinity, and pressure. FloatCast seeks to maximize collective sampling performance by strategically selecting new dive commands for each time a float surfaces.

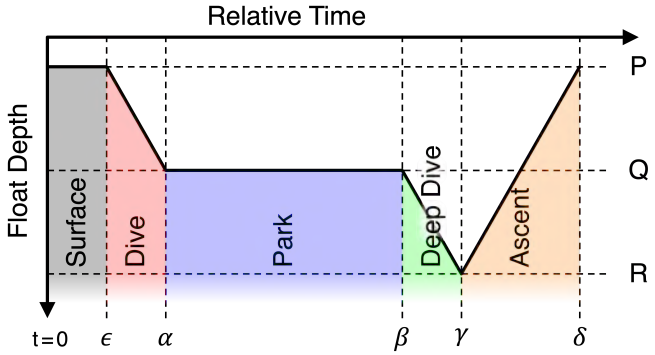

Fig. 3. Typical float profile pattern. All timing variables here are relative to the start of the current cycle. The float remains on the surface at depth P until time ϵ to receive new commands. The float then dives to the parking depth Q between time ϵ and α. The float remains at depth Q until time β. The float does its deep dive to depth R between time β and γ. Finally, the float ascends to the surface from depth R while taking a full profile between time γ and δ. Note that the similar timing notation in Section III-B is for absolute time. See [1], [5] for similar visualizations.

Referring back to Fig. 3, a float will wait on the surface until it receives new dive commands from FloatCast: park depth and park duration. Following these commands, a float begins its initial dive down to the parking depth Q. It remains at depth Q between times α and β. Typical park durations $(\beta - \alpha)$ considered are no longer than 5 days. The float then begins its deep dive to depth R, nominally 2000 m. Upon reaching depth R, the float begins its ascent and takes a full profile sample along the way. This routine repeats as long as the float is active.

This research specifically seeks to determine optimal park depth and park duration commands to relay to floats in the communication window. The mission objective is to collec-

tively maximize sampling coverage for a given region. The performance of candidate float commands is assessed via the mapping error scoring metric.

C. Mapping Error Scoring Metric

To maximize float coverage in a given region, we use a mapping error scoring metric based on the time and location of float samples. This metric encourages floats to spread out in the region of interest by limiting the reward from samples that are closely clustered in time or space. This metric is similar to a scoring methodology that has been used for underwater gliders [21], [22]. Mapping error describes user confidence in a predicted scalar field of some quantity of interest based on measurements [21], [22], [23]. We use this confidence metric in isolation—without the scalar field estimate—for analyzing performance, as was done in [21], [22]. A brief explanation of this mapping error scoring metric is given below based on [21], [22], [23].

Mapping error is fundamentally an analysis of the covariance between measurements and a grid of reference points. The reference point grid defines the locations where the scalar field predictions and associated confidence can be calculated based on accumulated measurements. This reference point grid subsequently defines the region of interest that the floats seek to cover. To quantify the covariance between two data points, the following function is used [22].

$$C(R,t,R',t') \triangleq \sigma_0 e^{-\frac{\Gamma(R,R')}{\sigma} - \frac{|t-t'|}{\tau}} \tag{5}$$

Here, R and t represent spatial coordinates and time of a data point, respectively. The covariance of the scalar measurement field about its mean is σ_0. The distance function is denoted by Γ. Moreover, σ and τ are the regional spatial and temporal decorrelation scales, respectively. When a pair of data points (R,T) and (R',t') are close to each other in space and time, the resulting covariance is high. Conversely, the covariance is low when this pair of data points are far apart in space and time.

Using (5), the covariance between all measurements is quantified via the measurement covariance matrix. The element in the kth row, jth column of the measurement covariance matrix $\tilde{C}$ is [22]

$$[\tilde{C}]_{kj} \triangleq \left(\tilde{\sigma}_0 \delta_{kj} + C(R_k,t_k,R_j,t_j)\right) \tag{6}$$

The $\tilde{C}$ matrix is $B \times B$ dimensional, where B is the number of measurements. The measurement noise variance constant is denoted by $\tilde{\sigma}_0$. Note that δ_{kj} equals 1 if $k = j$, and 0 otherwise.

Using the inverse of the measurement covariance matrix, the variance of the error for the scalar field estimate is defined by [21], [22], [23]

$$\hat{C}(R,t,R',t') := C(R,t,R',t') - \sum_{k=1}^{P}\sum_{j=1}^{P} C(R,t,R_k,t_k)[\tilde{C}^{-1}]_{kj} C(R_j,t_j,R',t') \tag{7}$$

This quantity defines confidence in the estimates of the scalar field that would be produced by these measurements [23]. Lower values for (7) indicate lower error and better performance. Note that $\hat{C} \in [0,1]$, as given in [22].

Equation (7) is foundational to the process used to analyze relative optimality of commands. In particular, the following equation is used to score candidate trajectories [21], [22]:

$$\psi(t) := -\log_{10}\left(\frac{1}{\sigma_0|\mathcal{B}|}\int_{\mathcal{B}} \hat{C}(R,t,R,t)dR\right) \tag{8}$$

where $\mathcal{B}$ is the region of interest. Equation (8) collectively assesses the error from (7). The desired outcome is to minimize the area integral of the error variance. The negative logarithm inverts the integral such that $\psi(t)$ is maximized when the error is minimized. We therefore seek to maximize $\psi(t)$ with our choice of float commands.

III. Float Trajectory Prediction

This section describes three components of our float trajectory prediction procedure. First, the Echo State Network (ESN) usage is discussed from a practical standpoint. Second, the float state-space model and trajectory prediction methodology are described. Lastly, the method to select candidate float commands is discussed. Recall that Fig. 1 illustrates where these three components are situated in the FloatCast workflow.

A. Daily ESN Predictions of Ocean Surface Flow

To generate daily predictions of ocean surface flow, an ESN is trained on Sea Level Anomaly (SLA) data from E.U. Copernicus Marine Service Information [17], [18]. The SLA forecasts from the ESN are converted to surface flow forecasts using a numerical method [24]. The temporal resolution of the training data is one day. The spatial resolution is $0.125° \times 0.125°$, but we downsample this to $0.25° \times 0.25°$. Doing so significantly reduces the memory required to train and use the ESN. For a similar reason, we train the ESN on SLA rather than geostrophic surface flow directly because it takes less memory to forecast SLA scalars than surface flow vectors. These design choices conserve computation time which can be put to use elsewhere in the FloatCast pipeline.

Larger ESNs ($\mathbf{W}$ matrices, see equation (1)) generally perform better than smaller ones [19]. Thus, it is reasonable to assume that the best way to improve our flow forecasts is to maximize ESN size. However, doing so increases the required memory quadratically with matrix size. Moreover, predicting float trajectories from ocean flow is challenging enough that devoting the maximum amount of computational resources to flow prediction may be counterproductive. We propose an alternative strategy to improve overall system performance. Namely, several ESNs are trained and the one that produces the best hindcast is selected for use. This is determined based on the Mean Square Error (MSE), which is discussed below. This approach is feasible because of the inherent randomness of ESN generation from a given set of parameters. Training multiple ESNs in this way increases the memory requirement linearly rather than quadratically. Having taken measures to

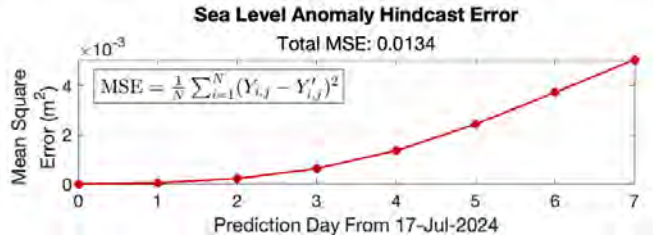

Fig. 4. Sea level anomaly hindcast error over time for the Echo State Network flow prediction from 17 July 2024 to 24 July 2024. Dates are indexed to midnight.

guard against computational burden, new ESNs can be trained each day as updates are made available to the near real time data [17].

To evaluate the performance of an ESN hindcast, the MSE is computed for a central subset of the prediction SLA data according to

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Y_{i,j} - Y'_{i,j})^2 \tag{9}$$

where N is the number of data points inside the central box. The ith data point on hindcast day j is denoted as $Y_{i,j}$. The corresponding reference data point is denoted as $Y'_{i,j}$. In this work, the region of interest is given by the area enclosed by $34°\mathrm{N}$ to $43°\mathrm{N}$, and $-68°\mathrm{E}$, to $-59°\mathrm{N}$. This region is marked off into a 3×3 grid. To prioritize prediction accuracy in the most important region, the MSE is computed only inside the middle box of the grid, defined from $37°\mathrm{N}$ to $40°\mathrm{N}$, and $-65°\mathrm{E}$, to $-62°\mathrm{N}$. Consequently, the $\mathbf{Y}_j$ and $\mathbf{Y}'_j$ vectors only contain SLA data inside this middle box. The mean square error over time for the surface flow hindcast used in Section IV-A is given in Fig. 4. Note that the performance metric used to rank ESN performance is the total MSE over the prediction window. Next, we discuss the float trajectory prediction model and how it makes use of the ESN flow forecasts.

B. Float Trajectory Predictions Using State-Space System

As alluded to above, we use a Lagrangian particle model for the prediction of float trajectories. Similar to one of the FlowPilot methods [2], this model uses an exponential decay of surface flow velocity with depth. The state-space model for the lateral movement of float j in its kth dive is given by

$$\dot{x}_{j,k}(t) = uc(a\exp[-z_{j,k}(t)/\zeta] + b) \tag{10}$$
$$\dot{y}_{j,k}(t) = vc(a\exp[-z_{j,k}(t)/\zeta] + b) \tag{11}$$

where $\dot{x}_{j,k}(t)$ is the zonal velocity of float j in its kth dive at time t. The analogous meridional velocity is $\dot{y}_{j,k}(t)$. The depth of float j in cycle k at time t is written as $z_{j,k}(t)$. Moreover, $x_{j,k}(t)$, $y_{j,k}(t)$, and $z_{j,k}(t)$ are all scalars. Consequently, $x_{j,k}(t)$ and $y_{j,k}(t)$ are referenced to the origin point of the first dive of float j, and can be used to obtain updated coordinates. For the Gulf Stream deployment, we use $a = 0.11$, $b = 0.04$, and $\zeta = 500\ \mathrm{m}^{-1}$ in (10) and (11). These constants were chosen based on prior float deployments. The c parameter was chosen by trial and error such that a float on the surface moves at 75% of the altimetric surface velocity. Thus, $c = 0.75/(a+b) = 5$. Moreover, $u \triangleq u(x_{j,k}(t), y_{j,k}(t), t)$ and $v \triangleq v(x_{j,k}(t), y_{j,k}(t), t)$ are measured zonal and meridional surface flows, respectively, at the coordinate point nearest to the current float location, $(x_{j,k}(t), y_{j,k}(t))$, at time t. Note that u and v are measured in $\mathrm{m/s}$. FloatCast is designed so that either measured or ESN forecast flow data can be used to set the values of u and v.

Recall the typical float cycle sequence sketched in Fig. 3. The vertical velocity of float j in its kth dive at time t is given by $\dot{z}_{j,k}(t)$ defined as

$$\dot{z}_{j,k}(t) = \begin{cases} 0 & \text{if } t_{j,k-1} \le t < \epsilon_{j,k}, \text{ or } \alpha_{j,k} \le t < \beta_{j,k} \\ w_{j,k} & \text{if } \epsilon_{j,k} \le t < \alpha_{j,k}, \text{ or } \beta_{j,k} \le t < \gamma_{j,k} \\ -w_{j,k} & \text{if } \gamma_{j,k} \le t \le \delta_{j,k} \end{cases} \tag{12}$$

The constant dive speed for the current float dive is given by $w_{j,k}$. In our model, all floats use the same vertical velocity when evaluating different park depth and park duration commands. The $\epsilon_{j,k}$, $\alpha_{j,k}$, $\beta_{j,k}$, $\gamma_{j,k}$ and $\delta_{j,k}$ quantities all describe absolute times of interest for float j in its kth cycle. Float j begins its descent for dive k at time $\epsilon_{j,k}$. Similarly, it begins the park phase at time $\alpha_{j,k}$. The deep dive then begins at $\beta_{j,k}$. The ascent to the surface begins at $\gamma_{j,k}$. Finally, the float surfaces at $\delta_{j,k} = t_{j,k}$. Note that $t_{j,k-1} \le \epsilon_{j,k} \le \alpha_{j,k} \le \beta_{j,k} \le \gamma_{j,k} \le \delta_{j,k}$. The float cycle sketch in Fig. 3 uses relative times instead of the absolute times used here.

To understand the prediction error of this model over time, we compare float trajectory hindcasts with ground truth historical float trajectory data. This analysis is based on the following error definitions:

$$\tilde{\mathbf{x}}_j^i \triangleq \{\hat{x}_{j,k} - x_{j,k} \mid t_{j,k} - t_{j,0} < i\} \tag{13}$$
$$\tilde{\mathbf{y}}_j^i \triangleq \{\hat{y}_{j,k} - y_{j,k} \mid t_{j,k} - t_{j,0} < i\} \tag{14}$$
$$\tilde{X}^i \triangleq \{\tilde{\mathbf{x}}_j^i \mid j = 1, ..., N\} \tag{15}$$
$$\tilde{Y}^i \triangleq \{\tilde{\mathbf{y}}_j^i \mid j = 1, ..., N\} \tag{16}$$

Here, $x_{j,k}$ and $y_{j,k}$ denote the actual eastings and northings with respect to the origin point of the first dive for float j at the end of cycle k. The analogous predicted eastings and northings are written as $\hat{x}_{j,k}$ and $\hat{y}_{j,k}$, respectively. These easting and northing predictions use commands determined heuristically from measured float depth data. Consequently, $\tilde{\mathbf{x}}_j^i$ and $\tilde{\mathbf{y}}_j^i$ represent the set of all easting and northing prediction errors for float j that occurred within i days of $t_{j,0}$. Thus, the superset of easting and northing errors for all floats that occurred within i days of $t_{j,0}$ are written as $\tilde{X}^i$ and $\tilde{Y}^i$.

Using historical data from 10 floats in the Gulf Stream in the North Atlantic from July 2024 as ground truth, we make trajectory hindcasts from each surfacing point. These hindcasts extend in time until the last measured surfacing of the given float. This effectively transforms our 10-float data set into a much larger data set of virtual floats. The commands used for

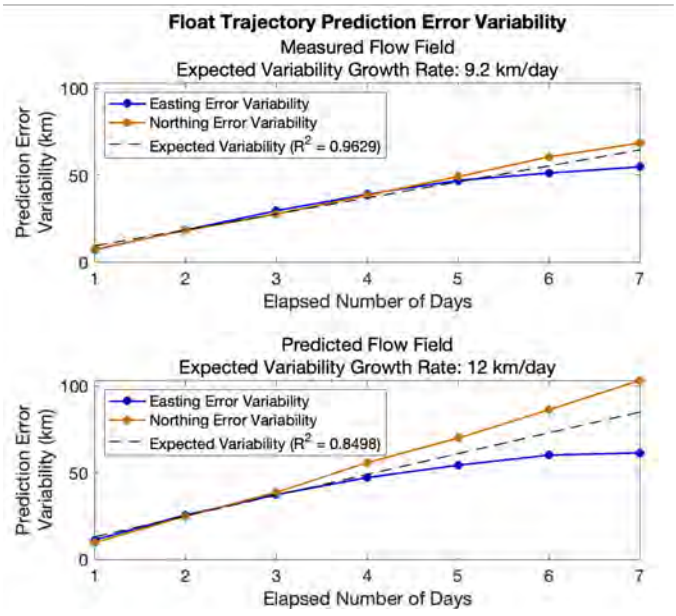

Fig. 5. Float trajectory prediction variability over the course of 7 days. Prediction errors here are grouped based on the total elapsed time of the prediction. The float trajectory hindcast predictions that used the predicted flow data had a higher variability growth rate compared to the predictions that used the measured flow field.

each prediction are based on measured depth values specific to each dive cycle.

We then analyze the error of the predicted eastings and northings for each virtual float (see Fig. 5) by grouping prediction error based on the elapsed number of days, as in (15)–(16). The mean prediction error for the virtual float experiment is roughly zero for the first few days because predictions deviate relatively equally in the north/south and east/west directions. To understand the variability of predictions, we analyze the standard deviation of the error of each cycle prediction. The standard deviation of the error with respect to elapsed time is plotted in Fig. 5. As this plot shows, the expected variability in our predictions grows at a rate of $9.2\ \mathrm{km/day}$ and $12\ \mathrm{km/day}$ when using the measured flow data and our ESN prediction, respectively. The ESN forecast used here was for a 14-day window.

Multiple factors contribute to the error in Fig. 5. Generally speaking, our model makes the sweeping assumption that floats precisely experience forces causing them to move in the same direction of the measured or predicted surface flow. It is unlikely that this surface flow accurately replicates this phenomena, especially considering resolution restrictions. Moreover, this model assumes that subsurface currents uniformly follow the direction of the surface flow, which is not realistic. However, these types of assumptions are necessary for the lightweight forecasting method used here.

As discussed below, it is critical for the optimization component of FloatCast that solutions to (10)–(12) are computed quickly. We therefore use the piecewise analytic solutions of this system of equations to make float trajectory predictions. To further ensure that the float command optimization is efficient, candidate float commands are selected judiciously; see Section III-C.

C. Strategic Selection of Candidate Float Commands

A key component of FloatCast is that it should operate in real-time, which becomes challenging when running Monte Carlo trials (see Section IV-B). The choice of command sets—park depth and park duration—to evaluate is critical because Monte Carlo trials magnify time loss from redundant calculations. In order to improve computational efficiency, we seek to maximize the variety of predicted outcomes for the number of float commands tested.

To maximize prediction variety, we select commands based on a new metric, introduced below. Consider the uniform flow field, $(u, v) = (1, 0),\ \forall\ x_{j,k}(t),\ y_{j,k}(t)$, and t. Recall that u and v are functions of float position and time, and are measured in $\mathrm{m/s}$. For this specific flow field, the total distance traveled for an arbitrary float is equal the distance traveled in the horizontal direction (easting). Moreover, since the flow field is uniform, the float trajectory for an arbitrary set of commands can be calculated outright. We integrate (10) to solve for the position of the float $100\ \mathrm{days}$ into the future, given that it followed fixed dive commands. The total distance traveled by float j is given by

$$A = \sum_{k=1}^{K} \left| x_{j,k}(t_{j,k-1}, t_{j,k}) \right| + \left| x_{j,K+1}(t_{j,K}, t_{j,K} + \mathbb{T}_j) \right| \quad (17)$$

where $x_{j,k}(\cdot,\cdot)$ is the solution to (10) between the two specified time arguments. Recall that $t_{j,k}$ is the time that float j ends cycle k. The total number of complete cycles in the fixed interval of time for a given set of commands is K. The duration of time between the last complete cycle of float j and the end of the time interval is denoted by $\mathbb{T}_j$.

The total distance traveled for a number of park depth and park duration commands are calculated using (17). The results are normalized by the maximum distance traveled, and plotted in Fig. 6. As this plot shows, it is potentially wasteful to test short dive durations at several park depths because they result in similar normalized distance traveled values. Thus, rather than testing all permutations of a set of park depth and park duration commands, we elect to test the commands sketched on Fig. 6 with black dots. To maximize variety, no commands are chosen from the same color ribbon in the plot. We believe this maximizes computational resources by eliminating similar trajectories from the calculations. Moreover, the multiple park durations considered ensure that there can be a timing offset between nearby floats, which factors into the mapping error calculations.

IV. Float Trajectory Optimization

This section presents hindcast results from a case study of three floats deployed in the Gulf Stream in the North Atlantic near the New England Seamount region during July 2024. First, results are presented from a deterministic FloatCast

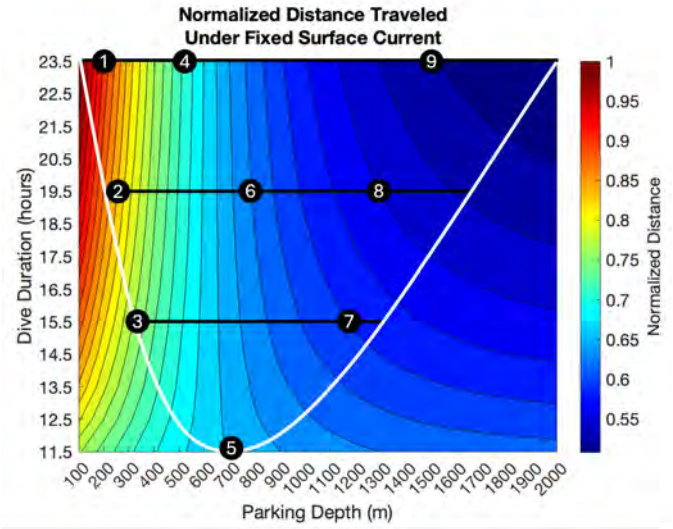

Fig. 6. Contour plot characterizing the normalized distance traveled by a float in a uniform flow field for assorted command combinations. The total distance traveled for a float using each command set was calculated over 100 days using a fixed dive velocity, $w_j = 0.12$ m/s. The surface time between dives is fixed at 30 min, and the deep dive depth is uniformly 2000 m for all command combinations. Distances are normalized in the plot. To maximize the expected variety of prediction outcomes according to this plot, we select the command combinations approximately marked by the black dots. The black and white lines are sketched to illustrate the pattern used to select commands.

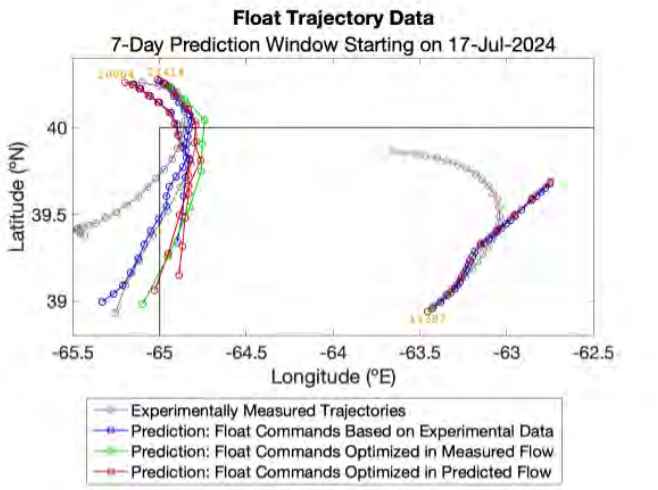

Fig. 7. Example float trajectory data. The gray points are experimentally measured float surfacing locations. The blue, green, and red trajectory forecasts each use the measured flow data to inform the state-space model predictions. The blue trajectory uses commands based on experimentally measured float depths. The green and red trajectories use float command sets optimized in the measured and predicted flows, respectively. The selected command from the optimization in the measured and predicted flows are the same for floats 10004 and 11387. The border of the region of interest is drawn with a black line (40°N, and −65°E).

optimization. Then, a risk-reward assessment for candidate command sets is given using results from a stochastic analysis.

A. *FloatCast Demonstration for Deterministic Predictions*

Three floats in the North Atlantic are selected for this case study of a deterministic FloatCast optimization. The sampling objective is to maximize coverage in the box defined from 37°N to 40°N, and −65°E, to −62°N. Both measured and predicted surface flows for this region are used (separately) to inform float trajectory predictions. These predictions are made for a 7-day window beginning on 17 July 2024. Trajectory predictions are scored using the mapping error scoring metric parameterized as follows. The reference point grid is uniformly distributed within the box specified above using a 6×6 resolution such that each reference point covers a $0.5° \times 0.5°$ area. The spatial and temporal decorrelation scales used are 25 km and 1.25 days, respectively. Moreover, we set the covariance of the scalar measurement field about its mean (σ_0) to 1, and measurement noise variance ($\tilde{\sigma}_0$) to 0.1. Measurements older than 3 temporal decorrelation scales are excluded from the mapping error calculation as they have minimal impact. The performance score for generated float trajectory predictions is computed as the sum of the mapping performance $\psi(t)$ from midnight of each day from 17–21 July 2024. This timing prioritizes scoring when prediction accuracy is at its highest. Note that the surface flow hindcast error for the predicted flow is given by Fig. 4.

Fig. 7 shows various trajectories for the three floats considered in this analysis. The blue, green, and red trajectories in Fig. 7 come from predictions made in the measured (known) flow field [18]. The float commands used to make each color trajectory were determined in different ways. The blue trajectories use commands that are meant to mimic the actual behavior of floats in the deployment. These blue trajectories illustrate our predictive capabilities under the best circumstances: when the flow field is known and float commands are inferred. The green and red trajectories use commands resulting from the FloatCast optimization in measured and predicted flows, respectively. The selected commands for the green and red optimization data are identical, except for float number 11414. The 9 command sets used for these optimizations (green, red) are sketched on Fig. 6 with black dots. For the green and red trajectories in Fig. 7, each float followed its selected command repeatedly throughout the 7-day prediction window. Moreover, the vertical dive velocity of each float was fixed 0.12 m/s, the surface time between dives was fixed at 30 min, and the deep dive depth was uniformly 2000 m.

The performance score for the predicted trajectories based on experimental data (blue) was better than the suggested commands (green, red). There are several possible explanations. Within the context of the simulation framework, the main cause is likely due to assumptions related to uniform float behavior that were not true for the deployment data. For example, the estimated vertical velocity for each float varied relative to the fixed 0.12 m/s velocity used in the simulation. While floats 10004 and 11414 had estimated vertical velocities close to this value, float 11387 had an estimated vertical

velocity of almost twice this value[2]. This is crucial because as shown in Fig. 7, float 11387 stays inside the box for the duration of the analysis and is likely the highest points-scorer. The faster vertical velocity allowed this float to make sub 6-hour dives, which is faster than the shortest dive duration considered in the optimization. Another contributing factor is variability in experimental deep dive depths. Crucially, float 11414 consistently used a deep dive depth near 1200 m. As a result, this float also consistently completed dives in under 6 hr. For these reasons, it is important that our optimization results presented in Fig. 7 are framed as suggested rather than experimentally validated optimal values. In practice, it is impossible to know whether using one set of commands would outperform another on the actual hardware.

While the optimized values did not outperform the hindcast in simulation, there is still insight to be gained by analyzing the suggested values. In particular, the selected command set was almost the same between the optimization performed in the measured and predicted data, inspiring confidence that this flow forecasting method is viable. In practice, a user selecting the next batch of float commands would have to base their decision on the trajectories predicted in the flows forecasted by the Echo State Network alone.

Some other practical considerations are worth noting. The last set of actual flow data used is centered on midnight, 17 July 2024. The first ESN prediction is centered on midnight, 18 July 2024. Moreover, the float trajectory predictions begin on 17 July 2024 shortly after midnight. In practice, it is likely that there would be a larger buffer between when the actual flow data ends and when the float trajectory predictions begin. In addition, it is likely that float commands will need to be computed one cycle ahead using an estimate for the next surfacing location. These are topics of ongoing work.

B. Risk-Reward Assessments of Fleet Commands

In Section IV-A, all float trajectories considered were deterministic. However, given the variable nature of ocean currents and other sources of error, our study benefits from a stochastic analysis. To address this, we run several Monte Carlo trials for the float trajectory predictions for each set of commands. In each Monte Carlo trial, the predicted end point of each dive is perturbed by a random amount based on a Gaussian normal distribution. The mapping error scoring metric is computed for each Monte Carlo trial, giving a distribution characterizing the performance of each command set. This allows users to compare performance distributions and select between commands with varying risk-reward levels between uncertainty and expected performance.

As a basis for comparison, we perform the following simulation. Using measured flow data and the same three floats from Section IV-A, predictions are made over a 7-day window using the 9 different commands specified in Fig. 6. For each command set, 100 Monte Carlo trials are performed following the format described above. The end point of each predicted dive is randomly perturbed in its horizontal position using a normal distribution with mean $0\ \mathrm{km/day}$ and standard deviation $10\ \mathrm{km/day}$. This standard deviation was chosen to reflect the expected prediction error variability growth rate for float trajectory predictions in a measured flow field, as illustrated in Fig. 5.

To establish a reference, the mapping error is then computed for each trial using the same parameters outlined in Section IV-A. Note that the mapping error is calculated for the whole group of floats under a given set of commands. This is computationally intensive, as there are $100(9^3) = 72,900$ sets of mapping error calculations to be performed. The performance score of each trial is obtained by summing the mapping error over the first 4 days of the prediction. The distribution of average performance scores from the exhaustive search is plotted with a gray line in Fig. 8.

Computing the mapping error scoring metric for many Monte Carlo trials is computationally intensive. At the expense of accuracy, one could speed up this calculation by reducing the mapping error reference point resolution and shrinking the time window for accepted measurements. However, more can be done to address the computational efficiency of scoring many Monte Carlo trials. In particular, we use a two-stage optimization to limit the number of mapping error calculations required to obtain a viable command set. At a high level, we do a random sparse search of candidate command sets, and then evaluate command sets similar to the best-performing ones from the random search.

The two-stage optimization begins by randomly taking a sparse subset of all possible command sets. The mapping performance is then computed for all Monte Carlo trials for each of these command sets. Relative performance is ranked based on the average mapping performance of all Monte Carlo trials for a given command set. The best-performing command sets from this sparse search are used in the next stage of the optimization. The second stage of the optimization is a greedy search, which computes the mapping performance for command sets that are similar to the best ones from the sparse search. The relative performance is then ranked, and the best command set is selected for use.

To determine which command sets are most similar to each other for the greedy search, we use the normalized distance traveled metric introduced in Section III-C. The command numbering in Fig. 6 reflects a ranking for which commands are expected to produce the most lateral movement, down to the least. For example, command 1 is expected to generate the most movement, and is therefore more similar to command 2 than it is to command 9. This manner of thinking is used to generate new permutations of float commands that are similar to the reference command sets from the sparse search.

For the initial sparse search (stage I), 100 command sets were selected for further analysis. The 3 best-performing command sets were then selected for further analysis for the greedy search (stage II). Using the similarity metric described

[2]There is inherent imprecision in the heuristic used to estimate float vertical velocity from deployment depth data. Floats 11387 and 11414 followed sawtooth-like dive patterns (park duration was minimal) which likely disrupted the heuristic.

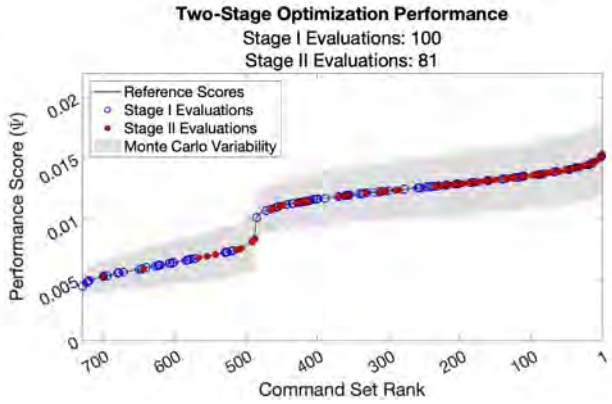

Fig. 8. Results of the two-stage optimization referenced to the exhaustive search for the simulation described in Section IV-B. This optimization is for 3 floats and 9 candidate command sets, which makes for $9^3 = 729$ possible combinations. Command sets are ranked and sorted above based on average performance score from the 100 Monte Carlo trials for the given set. The reference scores are from the full exhaustive search. The command sets evaluated in the two-stage optimization are indicated with red and blue markers. The performance score variability of the 100 Monte Carlo trials for each command set is plotted as the mean performance score plus/minus one standard deviation.

above, 81 command sets were selected for this next evaluation. All told, the two-stage optimization method requires $100(100 + 81) = 18,100$ sets of mapping error calculations, as opposed to $72,900$ for the full exhaustive search.

The results of the full exhaustive and two-stage optimization methods are presented in Fig. 8. The data in this plot is sorted based on the average performance score from all Monte Carlo trials for each command set. The standard deviation of the performance scores for each command set is also shown with light gray shading. This indicates the overall uncertainty associated with each command set. Overlaid on this plot are points indicating which command sets were selected in each stage of the two-stage optimization. As Fig. 8 shows, the methodology behind the greedy search resulted in command sets being tested that performed well. Interestingly, the command sets tested in the greedy search did not all come from the top-performing group. In any case, the two-stage search was able to correctly identify the best command set using fewer mapping error calculations. This result is promising because it shows that more complex optimizations with more floats and/or commands could be feasible using the two-stage optimization approach. The best performing command set from the sparse search was ranked second, which especially helped the greedy search find the top command set. The performance uncertainty highlighted in Fig. 8 can be used to establish the risk associated with a given command set. Fig. 8 shows that command sets with poor performance generally have lower uncertainty, and that those with better performance have higher uncertainty.

V. Conclusion

This paper presents a conceptual framework called FloatCast, which is an aid to the control of autonomous ocean sampling floats. The purpose of FloatCast is to maximize sampling coverage of a given region through intelligent selection of fleet commands. This tool forecasts float trajectories using predictions of ocean surface flow and various dive command combinations. The ocean surface flow prediction is generated from an Echo State Network that is trained to forecast sea level anomaly with a one-day temporal resolution. Command combinations tested are chosen strategically to maximize computational resources. The predicted trajectories from command sets are analyzed according to the mapping error scoring metric. This metric is used to encourage floats to stay within a predefined region without getting too close to neighboring floats—in space and/or time. Lastly, a stochastic risk-reward analysis shows the expected uncertainty and potential performance of candidate commands.

In ongoing and future work, this system shall be tested in the field during a float deployment. This could give further insight into the flow field and float trajectory prediction skill. Successful implementation could mean that FloatCast is absorbed as a module of FlowPilot [2]. Moreover, FloatCast could also be adjusted to achieve different mission objectives. For example, instead of remaining fixed, mapping error reference points could be dynamically updated. On a finer level, the state-space model of the float vehicle dynamics could interpolate flow data in time to potentially improve accuracy. Currently, the flow field constants for specific coordinates are fixed for a full day. In the future, these constants can be adjusted based on the time of day and the flow at those coordinates for the next day as a first-order smoothing method. The performance of this method can be tested against the current workflow.

Acknowledgment

This work was supported by the Office of Naval Research Grant number N000142112434. The authors thank Debdipta Goswami for providing help and guidance on Echo State Networks. They also thank Barry Ma for modeling how to present daily-updated ocean data. Lastly, the authors also thank Beth Kirby for organizing and providing the float deployment sample data analyzed here.

References

[1] A. P. S. Wong, S. E. Wijffels, S. C. Riser, S. Pouliquen, S. Hosoda, D. Roemmich, J. Gilson, G. C. Johnson, K. Martini, D. J. Murphy, M. Scanderbeg, T. V. S. U. Bhaskar, J. J. H. Buck, F. Merceur, T. Carval, G. Maze, C. Cabanes, X. André, N. Poffa, I. Yashayaev, P. M. Barker, S. Guinehut, M. Belbéoch, M. Ignaszewski, M. O. Baringer, C. Schmid, J. M. Lyman, K. E. McTaggart, S. G. Purkey, N. Zilberman, M. B. Alkire, D. Swift, W. B. Owens, S. R. Jayne, C. Hersh, P. Robbins, D. West-Mack, F. Bahr, S. Yoshida, P. J. H. Sutton, R. Cancouët, C. Coatanoan, D. Dobbler, A. G. Juan, J. Gourrion, N. Kolodziejczyk, V. Bernard, B. Bourlès, H. Claustre, F. D'Ortenzio, S. Le Reste, P.-Y. Le Traon, J.-P. Rannou, C. Saout-Grit, S. Speich, V. Thierry, N. Verbrugge, I. M. Angel-Benavides, B. Klein, G. Notarstefano, P.-M. Poulain, P. Vélez-Belchí, T. Suga, K. Ando, N. Iwasaska, T. Kobayashi, S. Masuda, E. Oka, K. Sato, T. Nakamura, K. Sato, Y. Takatsuki, T. Yoshida, R. Cowley, J. L. Lovell, P. R. Oke, E. M. van Wijk, F. Carse, M. Donnelly, W. J. Gould, K. Gowers, B. A. King, S. G. Loch, M. Mowat, J. Turton, E. P. Rama Rao, M. Ravichandran, H. J. Freeland, I. Gaboury, D. Gilbert, B. J. W. Greenan, M. Ouellet, T. Ross, A. Tran, M. Dong, Z. Liu, J. Xu, K. Kang, H. Jo, S.-D. Kim, and H.-M. Park, "Argo data 1999–2019: Two million

temperature-salinity profiles and subsurface velocity observations from a global array of profiling floats," *Frontiers in Marine Science*, vol. 7, 2020. [Online]. Available: https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2020.00700

[2] Z. Szuts, T. Harrison, T. Curtin, B. Kirby, and B. Ma, "FlowPilot: Shoreside autonomy for profiling floats," in *OCEANS 2023 - MTS/IEEE U.S. Gulf Coast*, Sep. 2023, pp. 1–6.

[3] N. J. Holbrook, A. Sen Gupta, E. C. J. Oliver, A. J. Hobday, J. A. Benthuysen, H. A. Scannell, and D. A. Smale, "Keeping pace with marine heatwaves," *Nature Reviews Earth & Environment*, vol. 1, no. 9, pp. 482–493, 2020.

[4] E. van Sebille, S. Aliani, K. L. Law, N. Maximenko, J. M. Alsina, A. Bagaev, M. Bergmann, B. Chapron, I. Chubarenko, A. Cózar, P. Delandmeter, M. Egger, B. Fox-Kemper, S. P. Garaba, L. Goddijn-Murphy, B. D. Hardesty, M. J. Hoffman, A. Isobe, C. E. Jongedijk, M. L. A. Kaandorp, L. Khatmullina, A. A. Koelmans, T. Kukulka, C. Laufkötter, L. Lebreton, D. Lobelle, C. Maes, V. Martinez-Vicente, M. A. M. Maqueda, M. Poulain-Zarcos, E. Rodríguez, P. G. Ryan, A. L. Shanks, W. J. Shim, G. Suaria, M. Thiel, T. S. van den Bremer, and D. Wichmann, "The physical oceanography of the transport of floating marine debris," *Environmental Research Letters*, vol. 15, no. 2, p. 023003, feb 2020. [Online]. Available: https://dx.doi.org/10.1088/1748-9326/ab6d7d

[5] T. Wang, Z. Liu, and Y. Du, "A synthetic autonomous profiling float array in a Lagrangian particle tracking system," *Acta Oceanologica Sinica*, vol. 43, pp. 34–46, 2024. [Online]. Available: https://doi.org/10.1007/s13131-024-2395-7

[6] X. Liu, J. P. Dunne, E. J. Drenkard, and G. C. Johnson, "Simulating Argo float trajectories and along-track physical and biogeochemical variability in the California Current System," *Frontiers in Marine Science*, vol. 12, 2025. [Online]. Available: https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2025.1481761

[7] P. Chamberlain, L. D. Talley, B. Cornuelle, M. Mazloff, and S. T. Gille, "Optimizing the biogeochemical Argo float distribution," *Journal of Atmospheric and Oceanic Technology*, vol. 40, no. 11, pp. 1355 – 1379, 2023. [Online]. Available: https://journals.ametsoc.org/view/journals/atot/40/11/JTECH-D-22-0093.1.xml

[8] E. van Sebille, S. M. Griffies, R. Abernathey, T. P. Adams, P. Berloff, A. Biastoch, B. Blanke, E. P. Chassignet, Y. Cheng, C. J. Cotter, E. Deleersnijder, K. Döös, H. F. Drake, S. Drijfhout, S. F. Gary, A. W. Heemink, J. Kjellsson, I. M. Koszalka, M. Lange, C. Lique, G. A. MacGilchrist, R. Marsh, C. G. Mayorga Adame, R. McAdam, F. Nencioli, C. B. Paris, M. D. Piggott, J. A. Polton, S. Rühs, S. H. Shah, M. D. Thomas, J. Wang, P. J. Wolfram, L. Zanna, and J. D. Zika, "Lagrangian ocean analysis: Fundamentals and practices," *Ocean Modelling*, vol. 121, pp. 49–75, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1463500317301853

[9] A. Lasota and M. C. Mackey, *Probabilistic properties of deterministic systems*. Cambridge University Press, 1985.

[10] P. Chamberlain, L. D. Talley, M. Mazloff, E. van Sebille, S. T. Gille, T. Tucker, M. Scanderbeg, and P. Robbins, "Using existing Argo trajectories to statistically predict future float positions with a transition matrix," *Journal of Atmospheric and Oceanic Technology*, vol. 40, no. 9, pp. 1083 – 1103, 2023. [Online]. Available: https://journals.ametsoc.org/view/journals/atot/40/9/JTECH-D-22-0070.1.xml

[11] M. D. Grossi, S. Jegelka, P. F. Lermusiaux, and T. M. Özgökmen, "Surface drifter trajectory prediction in the Gulf of Mexico using neural networks," *Ocean Modelling*, vol. 196, p. 102543, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1463500325000460

[12] R. N. Smith and V. T. Huynh, "Controlling buoyancy-driven profiling floats for applications in ocean observation," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 3, pp. 571–586, July 2014.

[13] C. Sun, M. Song, D. Cai, B. Zhang, S. Hong, and H. Li, "A systematic review of echo state networks from design to application," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 23–37, Jan 2024.

[14] F. Hassanibesheli, J. Kurths, and N. Boers, "Long-term ENSO prediction with echo-state networks," *Environmental Research: Climate*, vol. 1, no. 1, p. 011002, jul 2022. [Online]. Available: https://dx.doi.org/10.1088/2752-5295/ac7f4c

[15] D. Goswami, A. Riggins, and D. A. Paley, "Data-driven prediction of urban micromobility: A study of dockless electric scooters [applications of control]," *IEEE Control Systems Magazine*, vol. 42, no. 5, pp. 18–31, Oct 2022.

[16] Y. Cui, R. Wu, X. Zhang, Z. Zhu, B. Liu, J. Shi, J. Chen, H. Liu, S. Zhou, L. Su, Z. Jing, H. An, and L. Wu, "Forecasting the eddying ocean with a deep neural network," *Nature Communications*, vol. 16, 2025. [Online]. Available: https://doi.org/10.1038/s41467-025-57389-2

[17] E.U. Copernicus Marine Service Information (CMEMS). Marine Data Store (MDS), "Global Ocean Gridded L 4 Sea Surface Heights And Derived Variables Nrt," DOI: https://doi.org/10.48670/moi-00149, (Accessed on 7 July 2025).

[18] E.U. Copernicus Marine Service Information (CMEMS). Marine Data Store (MDS), "Global Ocean Gridded L 4 Sea Surface Heights And Derived Variables Reprocessed 1993 Ongoing," DOI: https://doi.org/10.48670/moi-00148, (Accessed on 7 July 2025).

[19] M. Lukoševičius, *A Practical Guide to Applying Echo State Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 659–686. [Online]. Available: https://doi.org/10.1007/978-3-642-35289-8_36

[20] M. Cucchi, S. Abreu, G. Ciccone, D. Brunner, and H. Kleemann, "Hands-on reservoir computing: a tutorial for practical implementation," *Neuromorphic Computing and Engineering*, vol. 2, no. 3, p. 032002, aug 2022. [Online]. Available: https://dx.doi.org/10.1088/2634-4386/ac7db7

[21] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, 2007.

[22] D. A. Paley, "Cooperative control of collective motion for ocean sampling with autonomous vehicles," Ph.D. dissertation, Princeton University, 2007.

[23] F. P. Bretherton, R. E. Davis, and C. Fandry, "A technique for objective analysis and design of oceanographic experiments applied to MODE-73," *Deep Sea Research and Oceanographic Abstracts*, vol. 23, no. 7, pp. 559–582, 1976. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0011747176900012

[24] J. geun Choi (2025), "Physical oceanography: geostrophic current (numerical)," URL: https://www.mathworks.com/matlabcentral/fileexchange/59557-physical-oceanography-geostrophic-current-numerical, MATLAB Central File Exchange. (Accessed on 1 July 2024).