

“© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

3D Ensemble-Based Online Oceanic Flow Field Estimation for Underwater Glider Path Planning

Felix H. Kong¹, K. Y. Cadmus To¹, Gary Brassington², Stuart Anstee³, and Robert Fitch¹

Abstract—Estimating ocean flow fields in 3D is a critical step in enabling the reliable operation of underwater gliders and other small, low-powered autonomous marine vehicles. Existing methods produce depth-averaged 2D layers arranged at discrete vertical intervals, but this type of estimation can lead to severe navigation errors. Based on the observation that real-world ocean currents exhibit relatively low vertical velocity components, we propose an accurate 3D estimator that extends our previous work in estimating 2D flow fields as a linear combination of basis flows. The proposed algorithm uses data from ensemble forecasting to build a set of 3D basis flows, and then iteratively updates basis coefficients using point measurements of underwater currents. We report results from experiments using actual ensemble forecasts and synthetic measurements to compare the performance of our method to the direct 3D extension of the previous work. These results show that our method produces estimates with dramatically lower error metrics, with and without measurement noise.

I. INTRODUCTION

Oceanic flow field estimation is an important precursor to path planning for autonomous marine vehicles. Underwater gliders are particularly dependent on accurate estimates because of their comparatively limited thrust, requiring the aid of ocean currents to reach an intended destination. Conversely, the advantage of underwater gliders is that they consume little power and can achieve months-long deployments relevant to applications in ocean science [1], defence [2], and other industries.

While 3D flow field estimation has long been a topic of study in oceanography, obtaining and digesting meteorological flow field estimates in a form useful for practical robotics applications is less well understood. The goal of this paper is to develop a 3D estimator that builds on the idea of augmenting available forecasts with online measurement data.

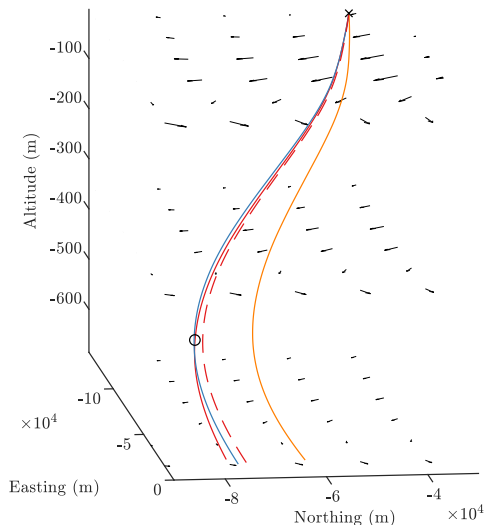
The standard tool in meteorological forecasting is the *ensemble forecast*, a collection of multiple predictions of future oceanic flow fields generated by simulation of ocean models. Our previous work showed how data from an ensemble forecasting can be used to build a continuous flow

This research is supported by an Australian Government Research Training Program (RTP) Scholarship, Australia’s Defence Science and Technology Group, Australian Bureau of Meteorology, and the University of Technology Sydney.

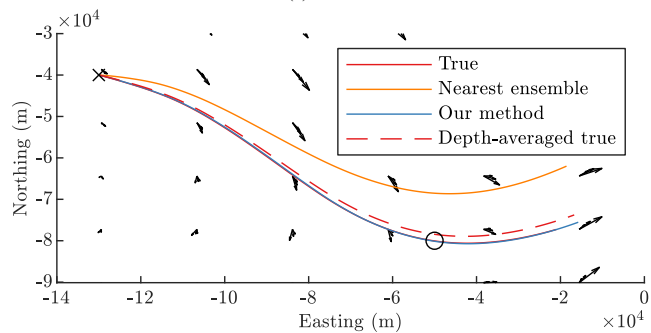
¹Authors are with the University of Technology Sydney, NSW 2007, Australia Cadmus.To@student.uts.edu.au and {Felix.Kong,Robert.Fitch}@uts.edu.au

² Author is with the Australian Bureau of Meteorology gary.brassington@bom.gov.au

³ Author is with the Defence Science and Technology Group, Department of Defence, Australia stuart.anstee@dst.defence.gov.au



(a) 3D view



(b) Top view

Fig. 1. Simulated glider trajectories when path-planning is performed using various flow field estimates. True flow field (black) is downsampled for ease of visualization. Departure location (at surface) marked by an ‘x’, target destination (at 500m depth) marked by circle.

field estimate in 2D [3]. However, for the purpose of path-planning for underwater gliders, which oscillate through a wide depth range, 2D estimates appear to be insufficient. As an illustration, Figure 1 shows glider trajectories planned based on various estimation approaches, including the common “depth-averaged” approach. Even when using the depth-averaged *true* flow field for path planning, the error to the desired destination is visibly worse than the planning over our 3D-aware estimate generated by noisy measurement. This is due to the inability of the depth-averaged representation to account for depth-varying variations in the flow field. Hence accurate 3D flow field estimates are crucial to reliable performance of underwater gliders.

Our existing algorithm [3] attempts to estimate an unknown 2D flow field as a linear combination of 2D “basis flow fields” generated from the ensemble forecast. The direct extension to the 3D case would result in 3D basis flow fields, which are limited to the span of the ensemble forecast. Crucially, the true flow field is unlikely to lie entirely within the span of the ensemble, which is a source of estimation error. To address this, a useful insight we make use of in this paper comes from noticing that 3D ocean flow sufficiently far from the coastline has negligible vertical velocity [4–6]. This fact allows 3D flow fields to be seen as collections of 2D horizontal flow fields. Roughly speaking, building the basis directly from existing 3D flow fields would enforce the existing correlations in depth. By breaking them apart into 2D flow fields and sharing these 2D flow fields between depths, the number of degrees of freedom increase, and greater expressivity of the basis is achieved. The contribution of this paper is to not only to simply extend [3] to 3D, but to adjust the “naive” 3D extension using the fact of negligible vertical velocity in order to increase the span of the basis, and hence improve estimation accuracy.

We describe the resulting algorithm in detail in Sections III and IV, and evaluate its performance using ensemble forecast data produced by the Australian Bureau of Meteorology for an area of the Tasman Sea between Australia and New Zealand. We report results from experiments that compare the output of our algorithm with the “naive” 3D extension of [3], with and without measurement noise, and illustrate the accuracy of our method in Section V. In several cases, the root-mean-square (RMS) error of the achieved error of the proposed algorithm fell below the RMS of the lower bound of the “naive” 3D extension.

The algorithms and results we contribute are a promising step forward in understanding how forecasts of underwater current can be used to produce useful flow field estimates for planning. Although this work focuses on the case of time-invariant flows, it lays a foundation to begin to address the challenges of the time-varying case.

II. RELATED WORK

As stated previously, this work is an extension of a previous paper [3], which is the closest related work. This paper builds upon [3] by extending to the case of 3D flow fields, and more importantly, by altering the “naive” 3D extension of algorithm to take advantage of the layered structure of 3D flow fields, resulting in improved accuracy.

This paper also joins a large body of work in the algorithmic estimation of flow fields. A standard tool for the estimation of flow fields in meteorology is the ensemble Kalman filter (e.g. [7–9]). Similar to this algorithm, the ensemble Kalman filter takes as input an ensemble forecast [10, 11], and makes use of the well-known Kalman filter for prediction. However, the proposed algorithm in this paper is lightweight enough to directly use the Kalman filter update equations, instead of requiring a still-expensive Monte Carlo approximation of the covariance matrix update.

Another class of flow field estimation techniques rely on direct simulation of a dynamical model of the ocean (e.g. [12, 13]). Many such models result in accurate estimates, not just of ocean currents, but also salinity, sea surface temperature, and many other important physical quantities. In fact, these methods are used to produce the very ensembles the present work considers as an input. However, they are especially computationally intensive, and may be unsuitable for robotics applications that require relatively fast, up-to-date estimates.

In contrast to the above model-based techniques, some data-driven techniques have also previously been used for flow field estimation. The “incompressible Gaussian Process (GP)” was introduced in [14], where a novel kernel was developed to enforce incompressibility of 2D flow field estimates. Though it was developed independently, the incompressible GP was in fact an example of a linearly constrained GP [15]. Another related method, the kernel observer, combines a kernel embedding with an observer such as a Kalman filter [16–18]. While the present work makes use of a kernel embedding and a Kalman filter, it additionally uses an ensemble forecast and a singular value decomposition (SVD). The use of the SVD allow intuitive interpretability; also, these methods are not directly compatible with the ensemble format of meteorological forecasts.

Another class of data-driven methods used in estimation of fluid flows is the Dynamic Mode Decomposition (DMD) [19–21], and its extensions (e.g. [22, 23]). These methods are based on a clever use of the SVD to obtain an approximate linear system model for fluid flows. The use of observers has also been proposed in conjunction with DMD [24, 25]. However, in these methods, the SVD is computed across samples in time, while in our method, the SVD is computed across ensemble members. Additionally, DMD only estimates fluid flow at the grid points (i.e. data locations); our method uses a kernel embedding to “interpolate” between grid points in a physically consistent way.

III. PROBLEM FORMULATION

We consider the problem of producing the “best” estimate of an unknown 3D time-invariant flow field. The inputs to our problem are an ensemble forecast, and a sequence of point measurements of the unknown flow field.

Suppose the ensemble forecast \mathcal{E} is composed of E “members”. Each ensemble member e_i , with $i \in \{1, \dots, E\}$ is a flow field on a grid $\mathcal{G} \subset \mathbb{R}^3$ with $X \times Y \times Z$ points, so that $\mathcal{E} = \{e_i | i = 1, \dots, E\}$. Within each e_i , associated with each grid point $x_g \in \mathcal{G}$ is a 2D velocity vector $u_i(x_g)$ representing the ocean’s current in the horizontal direction, and assume the vertical velocity is negligible. Indeed, measurements and best estimates place the vertical velocity of the ocean’s current on the order of 10^{-5} m/s or less [4–6]. In contrast, the horizontal current in our dataset is on the order 10^{-1} to 10^0 m/s. Notice that the ensemble members e_i are *not* measurements of the true flow field, but predictions based on simulations of ocean models.

The data from the true flow field takes the form of point measurements of the ocean current. The true flow field is modeled as a continuous flow field $f_{true} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. Then, the point measurements take the form:

$$z(x) = f_{true}(x) + n, \quad (1)$$

where $x \in \mathbb{R}^3$ is a position in 3D space, $n \in \mathbb{R}^2$ is a random variable representing sensor noise drawn from some sensor model, e.g. $n \sim N(0, \Sigma_n)$ for some covariance Σ_n . Let the set of all measurements be written \mathcal{Z} . Notice that $z(x) \in \mathbb{R}^2$ describes the *horizontal* velocity at a 3D point x in the ocean, since the vertical component of the velocity is negligible.

Additionally, a common assumption when modeling ocean flow is that the flow field should be incompressible [13, 26], often referred to as the *Boussinesq approximation* (e.g. [27, §7.7]). Hence, a desirable property of a flow field estimate $\hat{f}(x)$ is $\nabla \cdot \hat{f}(x) = 0$, i.e. $\hat{f}(x)$ is incompressible.

Then, given a sequence of measurements $z(x_k) \in \mathcal{Z}$ at locations x_k , we would like to find “the” flow field function $\hat{f}^*(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ such that

$$\begin{aligned} \hat{f}^*(x) = \arg \min_{\hat{f}(x) \in C^\infty} & \sum_{k=1}^{|\mathcal{Z}|} \|z(x_k) - \hat{f}(x_k)\|_2^2, \\ \text{subject to} & \quad \nabla \cdot \hat{f}(x) = 0 \text{ for all } x \in \mathbb{R}^2 \end{aligned} \quad (2)$$

where $\|\cdot\|_2$ is the Euclidean norm of a vector, $C^\infty(x)$ is the set of smooth 2-vector valued functions on x , and $|\mathcal{Z}|$ is the number of elements in the set \mathcal{Z} .

However, it is apparent that the optimisation problem (2) is severely underconstrained, as there are infinitely many smooth, nondivergent vector fields that can fit a finite number of measurements \mathcal{Z} . To address this, our previous work [3] imposed the form

$$\hat{f}(x) = \mathbf{H}(x)w, \quad (3)$$

where $\mathbf{H}(x)$ is a $2 \times N$ matrix function and a weight vector $w \in \mathbb{R}^N$. That paper constructed the basis of incompressible flow fields $\mathbf{H}(x)$ from ensemble data, so that $\hat{f}^*(x)$ is constrained to the span of $\mathbf{H}(x)$.

However, the choice of $\mathbf{H}(x)$ in [3] restricts its span to be equal to the span of \mathcal{E} , which can result in “out-of-span” error since the true flow field is unlikely to lie in the span of \mathcal{E} . Hence the problem in this paper is to choose $\mathbf{H}(x)$ to have an increased span compared to the method in [3] in a physically meaningful way in order to reduce “out-of-span” error.

Judicious selection of the basis $\mathbf{H}(x)$ is critical; it is certainly possible to add random basis vectors to increase the span of $\mathbf{H}(x)$, but this would be inefficient. Our approach is to share flow field information across depths in the ensemble data as a source of likely useful basis flows.

IV. APPROACH

In this section we describe the proposed algorithm, which is summarised in Figure 2. As in [3], $\mathbf{H}(x)$ is computed offline based on ensemble data, and w is updated recursively based on online measurements of the true flow field.

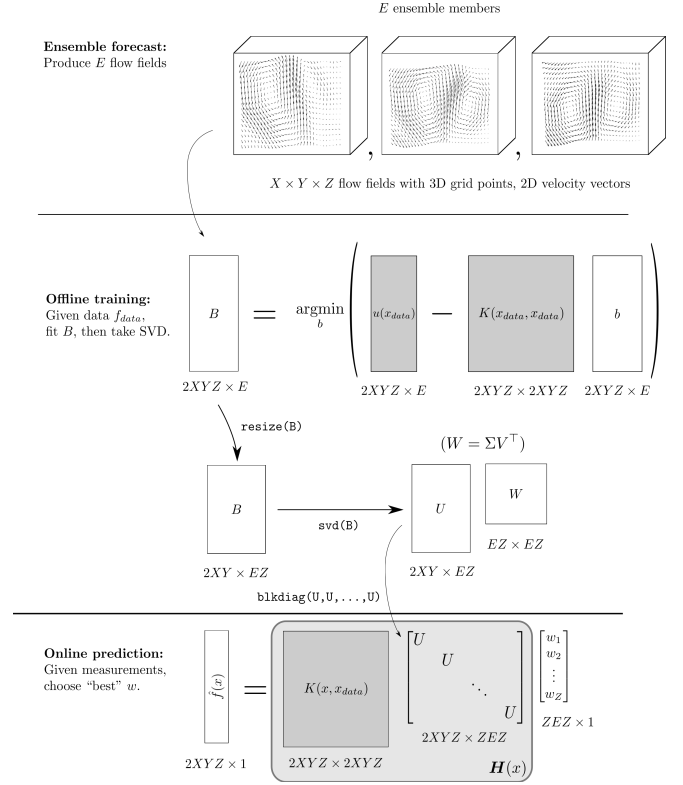


Fig. 2. A sketch of the different components of $\hat{f}(x)$, showing the case of the thin SVD and where $S = ZE$. The key algorithmic contribution of this paper is the reshape, which allows many more modes (i.e. left singular vectors) in U , resulting in a large span of the basis $\mathbf{H}(x) = \mathbf{K}(x, x_{data})U$. Shaded gray boxes indicate mathematical objects that are fixed at each stage.

In the proposed algorithm, $\mathbf{H}(x)$ is selected offline based on the ensemble \mathcal{E} , while the weight vector w is recursively estimated online based on the measurements \mathcal{Z} .

A. Offline step 1: Kernel embedding

Kernel methods are a common machine learning tool to allow linear methods to be applied to nonlinear patterns in data, and have previously been used to represent flow fields [3, 14, 16]. Kernels can be used to build a continuous spatial model of the flow field from data samples at discrete locations. The *incompressible kernel* allows this “interpolation” to respect incompressibility [14] in 2D; in this work we apply the incompressible kernel to describe 3D oceanic flow fields.

Given any kernel $k(x, x') : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$, the corresponding incompressible kernel is given by the following:

$$K(x, x') = D(x)k(x, x')D(x')^\top, \quad (4)$$

where $D(x) = [\frac{\partial}{\partial x_2}, -\frac{\partial}{\partial x_1}]^\top$, and x_1, x_2 are the first and second coordinate in the 3-dimensional vector x . Notice that while $x, x' \in \mathbb{R}^3$, $K(x, x') \in \mathbb{R}^{2 \times 2}$, as it describes the similarity between *two-dimensional* flow vector components, and enforces 2D incompressibility. Due to the assumption of negligible vertical velocity, 2D incompressibility is sufficient for 3D incompressibility.

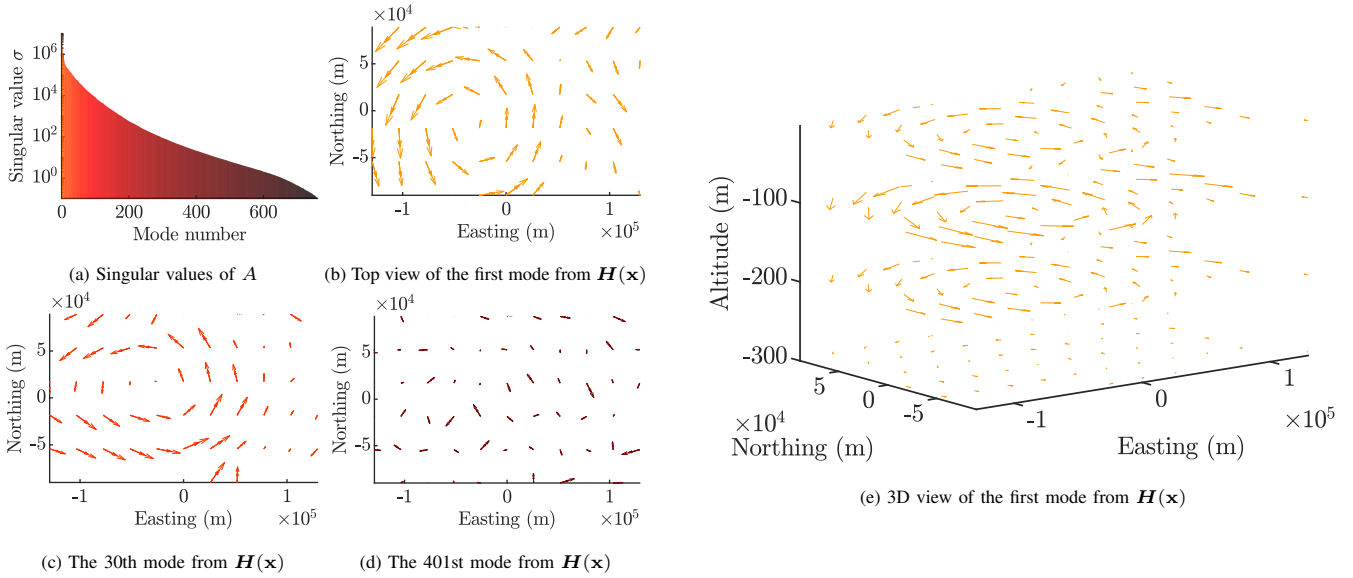


Fig. 3. Example SVD of A . (a) shows the singular values corresponding to the modes of $\mathbf{H}(\mathbf{x})$. Notice there are $S = 760$ modes, which is many more than $E = 95$ (the number that would have been produced by the “naive 3D” method, which is the direct 3D extension of [3]). (b-e) shows some examples of the “2.5D” basis flow fields, corresponding to singular values of 1.0×10^7 , 1.16×10^5 , and 2.09×10^1 , respectively.

The matrix $K(x, x')$ can be used to infer, or “query”, the flow at x . This can be done *en masse* by concatenating many $K(x, x')$ with varying x, x' . Let \mathbf{x}_q be a vector of coordinates of Q “query locations” x_q in \mathbb{R}^3 with total length $3Q$, and \mathbf{x}_{data} be the vector of all grid points $x_{data} \in \mathcal{G}$. Then, we write the kernel matrix

$$\mathbf{K}(\mathbf{x}_q, \mathbf{x}_{data}) = [K(x_q, x_{data})] \in \mathbb{R}^{2Q \times 2|\mathcal{G}|} \quad (5)$$

for all x_q in \mathbf{x}_q and x_{data} in \mathbf{x}_{data} . The number of query points Q is not required to match the number of points in \mathcal{G} .

Using this kernel matrix \mathbf{K} , inference can be performed. For any $\beta \in \mathbb{R}^{2|\mathcal{G}|}$, the flow field

$$\hat{f}(x) = \mathbf{K}(x, \mathbf{x}_{data})\beta \quad (6)$$

has the property that $\nabla \cdot \hat{f}(x) = 0$ for all $x \in \mathbb{R}^3$, i.e. it is incompressible.

Each ensemble member e_i can be represented by a latent representation β_i by setting $\mathbf{x}_q = \mathbf{x}_{data}$ and performing the following linear regression:

$$\beta_i = \arg \min_{\beta} \|\mathbf{u}_i(\mathbf{x}_{data}) - \mathbf{K}(\mathbf{x}_{data}, \mathbf{x}_{data})\beta\|_2^2, \quad (7)$$

where $\mathbf{u}_i(\mathbf{x}_{data})$ is the vector containing all the 2D velocity vectors $u(x_{data})$ for all $x_{data} \in \mathcal{G}$ in the i -th ensemble member e_i . Then, the set of (finite) flow vectors $\mathbf{K}(\mathbf{x}_{data}, \mathbf{x}_{data})\beta_i$ is the incompressible flow field that best fits e_i at the grid points in a least-squares sense.

In preparation for the next subsection, let B be the $2|\mathcal{G}| \times E$ matrix

$$B = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_E]. \quad (8)$$

B. Offline step 2: Singular value decomposition (SVD)

The other major component of $\mathbf{H}(\mathbf{x})$ comes from performing a singular value decomposition (SVD), which allow the columns of $\mathbf{H}(\mathbf{x})$ to be easily interpretable as basis

flow fields, which we call *modes*. The major contribution of this paper is to exploit the fact that 3D flow fields can be seen as a collection of depth-varying 2D flow fields, or equivalently, their 2D latent representations. Then, each 2D latent representation is used to generate a 3D flow field via $\mathbf{K}(x, \mathbf{x}_{data})$ that is self-similar in the depth dimension, which is added to the set of basis flow fields (Figure 3e shows an example of one such basis flow field). This results in up to Z times the basis flow fields compared to using the 3D flow fields in the ensemble directly, thereby increasing the span, and hence the expressiveness of $\mathbf{H}(\mathbf{x})$.

To do this, we collect the horizontal slices of each β_i in B for the SVD for a shared “library” of basis vectors. Each β_i is a $2XYZ \times 1$ vector of latent variables (recall that the grid \mathcal{G} has size $X \times Y \times Z$), which can be reshaped into columns of horizontal slices

$$\alpha_i = [\beta_{i,z_1} \quad \cdots \quad \beta_{i,z_Z}] \in \mathbb{R}^{2XY \times Z}, \quad (9)$$

where $\beta_{i,z}$ is the $2XY \times 1$ vector containing the elements of β_i that correspond to a depth of z , and $\{z_1, \dots, z_Z\}$ are the depth values present in the grid points \mathcal{G} . Then, let the matrix A be:

$$A = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_E] \in \mathbb{R}^{2XY \times ZE}. \quad (10)$$

Throughout this paper we refer exclusively to the thin SVD, where only the entries corresponding to nonzero singular values are calculated. The SVD of A is:

$$A = U\Sigma V^T, \quad (11)$$

since e_i , $K(x, x')$, and hence also A are real-valued. Notice that the number of columns in U is $S = \min(ZE, 2XY)$. Typically, E is limited, since producing ensemble forecasts is computationally expensive, but they are often computed over large grids, resulting in large XY and Z . Hence

in many cases, both $2XY$ and ZE will be greater than $\min(E, 2XYZ)$, the number of columns of U resulting from the SVD of B .

Importantly, the SVD induces an ordering of the left singular vectors according to their singular values. In some cases, it can be advantageous to truncate U , Σ , and V to reduce the data required to store U by removing elements associated with small singular values. Let \tilde{U} , $\tilde{\Sigma}$, and \tilde{V} be the rank r truncations of U , Σ , and V , respectively, with $r \leq S$. Then, let \tilde{U} be:

$$\tilde{U} = \begin{bmatrix} \tilde{U} & & & \\ & \tilde{U} & & \\ & & \ddots & \\ & & & \tilde{U} \end{bmatrix} \in \mathbb{R}^{2XYZ \times rZ}, \quad (12)$$

with blank elements being equal to the appropriately sized zero matrices.

Finally, as in [3], the basis $\mathbf{H}(\mathbf{x})$ is constructed in the following way:

$$\mathbf{H}(x) = \mathbf{K}(x, \mathbf{x}_{data})\mathbf{U}. \quad (13)$$

Then, the weight vector $w \in \mathbb{R}^{rZ}$ selects a linear combination of the columns of $\mathbf{H}(x)$, which can be seen as 3D basis flow fields “generated” from a 2D flow field at a certain depth. Figure 3e shows a visualization of $\mathbf{H}(x)w_1$, where $w_1 = [1, 0, \dots, 0]^T$. The rest of Figure 3 shows visualizations of several examples of the modes (i.e. columns) of $\mathbf{H}(x)$, and Figure 3a shows the relative contribution of each of these modes to the ensemble.

However, in cases where the number of modes is relatively high, significant truncation can be performed with surprisingly small loss in expressive power of $\mathbf{H}(x)$. Obviously, choosing $r < S$ results in a reduction of the span of U , and hence $\mathbf{H}(x)$, so there is a tradeoff between compactness and expressibility.

Importantly, at each depth in \mathcal{G} , each \tilde{U} will contain basis flow fields constructed using flow field information from *all* depths. In addition, the 2D flow field at each depth has its own set of weights, which may be distinct from other depths. This results in an increased span compared to the naive 3D extension of [3], where \mathbf{U} is the left singular vectors of B , which would only have E modes.

C. Online update of w via Kalman Filtering

We now address the remaining part of (3), finding w^* , which we determine based on online measurement data. For the case of time-invariant flow fields considered in this paper, the Kalman Filter (KF) is not strictly necessary; however, it will be an invaluable tool when extending to the time-varying case.

The well-known Kalman Filter recursively estimates the maximum likelihood state w of a linear dynamical system, given a sequence of measurements $z_k \in \mathcal{Z}$ experiencing Gaussian sensor noise. In the case of time-invariant flow fields, the process model F is simply the identity, and the

predict step is:

$$w_{k+1} = Fw_k, \quad (14)$$

$$\hat{z}_k = \mathbf{H}(x_k)w_k + n_k, \quad (15)$$

where the subscript k refers to the iteration number of the KF. The correction step equations are standard and omitted for brevity.

In the proposed algorithm, the initial state and covariance matrix of KF is initialised with the mean and variance of $W = \tilde{\Sigma}\tilde{V}^T \in \mathbb{R}^{S \times ZE}$ across the ensemble members. More verbosely, contiguous Z columns of W should be reshaped into new columns of SZ values in the matrix $W' \in \mathbb{R}^{SZ \times E}$. The initial state w_0 is then computed as the row-wise mean of W' . The initial covariance matrix P_0 is set to be a diagonal matrix with diagonal elements equal to the row-wise variances of W' . Initialising the KF in this way expediently assumes that the estimated flow field is a random variable described by the same probability distribution from which the ensemble members are sampled.

V. SIMULATION RESULTS AND DISCUSSION

In this section, the proposed algorithm is applied to an example flow field estimation problem, and the ramifications for underwater glider path-planning are illustrated.

A. Flow field estimation error comparison

We first demonstrate the proposed algorithm on dataset provided by the Australian Bureau of Meteorology (BOM). The ensemble forecast data provided by the BOM consists of 96 ensemble member flow fields forecasting conditions in the Tasman Sea between Australia and New Zealand on 16th of November 2018. Notably, the forecast itself does not include data for the vertical velocity of the currents. The flow field data covers a vast volume of ocean; we consider only a small region from 39.2° to 40.8° S, 151° to 154° E, and 2.5 m to 685 m deep.

The data is provided in geographic coordinates (latitude, longitude, and depth); an azimuthal projection about the center of the region was used to produce a Cartesian coordinate grid. To reduce computation time, this grid was then downsampled by a factor of 5 in the depth direction, resulting in a grid of size \mathcal{G} is $31 \times 17 \times 8$, with grid points about 10 km apart horizontally, and about 85 m in depth.

Since high-resolution sensor measurements of the region are unavailable, we extract one of the ensemble members and consider it to be the true unknown flow field. In the real world, the true flow field is unlikely to be one of the ensemble members, and may be significantly different. In an attempt to capture this, the “true” ensemble member was hand-picked through visual inspection to be qualitatively different from the rest of the ensemble members. The remaining 95 ensemble member flow fields form \mathcal{E} .

Simulated measurements were performed based on the true flow field $f_{true}(x)$, estimated via cubic interpolation. The way measurements were simulated was inspired by acoustic Doppler current profilers (ADCP), which are sensors that can measure ocean current in a column of water at regular

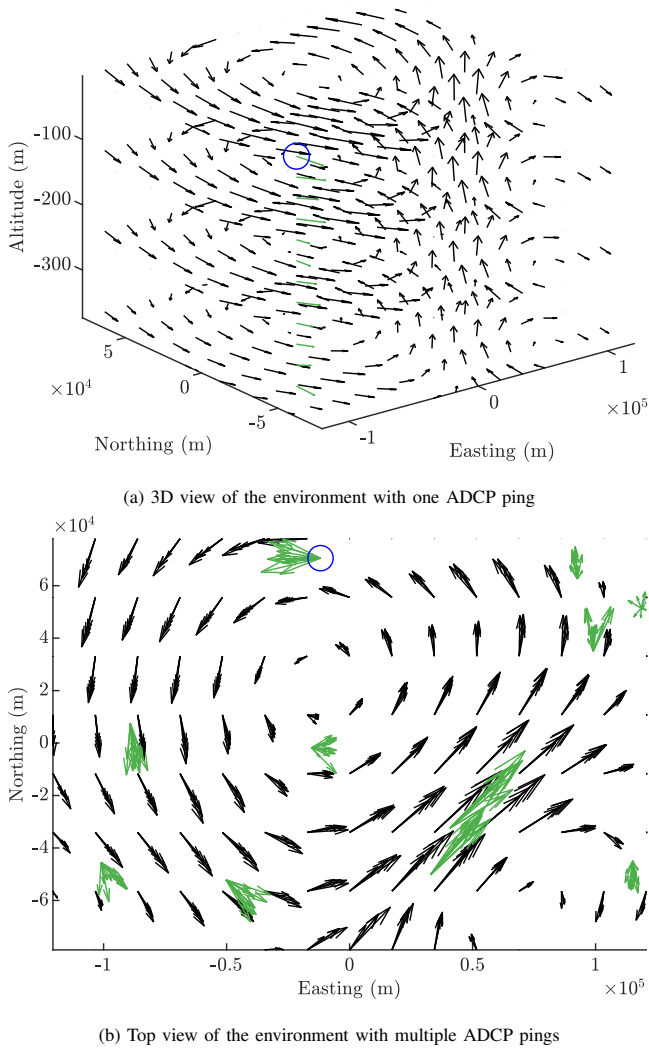


Fig. 4. Visualisation of measurements in the true flow field in black, downsampled by a factor of 2. A single ADCP “ping” results in 22 measurements (in green), only 12 of 450 are shown in (b) for readability. The location of a simulated ADCP-equipped surface vessel is shown in blue. Note the large noise on measurements.

depth intervals. The simulated ADCP in our experiments has a vertical resolution of 32 m and a precision standard deviation of 9 cm/s. It is worth noting that 9 cm/s standard deviation is quite large compared to the average speed of the current in the dataset, which was 22.8 cm/s (see the spread of the green arrows in Figure 4).

A radial basis function (RBF) is chosen as the kernel to reflect the property that in viscous fluid flow, motion on one “layer” of the fluid will exert an influence on nearby “layers”:

$$k(x, x') = \sigma_k^2 \exp(\langle x - x', [\ell_x^{-2}, \ell_y^{-2}, \ell_z^{-2}] \rangle). \quad (16)$$

The hyperparameters were tuned by hand, with $\ell_x = \ell_y = 10^4$ m, $\ell_z = 100$ m, and $\sigma_k = \mu/\ell_x = 1718.9 \text{ m}^2/\text{s}$. The symbol μ is the average flow vector magnitude at the surface, where the flow is usually fastest.

Three methods were compared in simulated experiments: the proposed method, the “naive 3D” method, and the “nearest” ensemble member. Briefly, the naive 3D method

TABLE I
ERROR PERFORMANCE AND SIZE OF DIFFERENT METHODS
IN IDEAL CONDITIONS

	No. modes	Rel error (%)	RMSE (cm/s)	RMSE of bound (cm/s)
3D basis	95	8.30	2.33	2.33
2.5D bases	760	1.42	0.39	0.077
	400	1.78	0.50	0.34
	300	2.46	0.69	0.57
	200	4.18	1.17	1.14
	95	11.00	3.08	3.07
Nearest e_i ($i = 82$)	—	45.6	12.80	—

TABLE II
ERROR PERFORMANCE AND SIZE OF DIFFERENT METHODS WITH NOISE
AND RANDOM MEASUREMENT LOCATION

	No. modes	Rel error (%)	RMSE (cm/s)	RMSE of bound (cm/s)
3D basis	95	11.11	3.11	2.33
2.5D bases	760	9.81	2.75	0.077
	400	9.91	2.78	0.34
	300	10.64	2.98	0.57
	200	11.96	3.35	1.14
	95	17.15	4.81	3.07

is the direct extension of [3] to the 3D case, performing the SVD on B instead of on A , resulting in $E = 95$ modes, each representing a 3D flow field. The nearest ensemble member was the member whose estimated current at the measurement locations was the closest to the noisy measurements in a least-squares sense.

Two sets of simulation experiments were performed. To show the potential performance of the algorithm, the first set experienced ‘ideal’ conditions with no measurement noise and measurement locations exactly at grid points. The Kalman filter noise covariance was set to $R = 0.01^2 I$, since there is no noise. The second set encountered Gaussian noise with standard deviation 9 cm/s, and measurements were taken from approximately 450 randomly uniformly distributed surface locations for a total of 10,000 measurements. The Kalman filter noise covariance was increased to $R = 0.12^2 I$ to account for both the sensor noise model and the additional noise incurred from cubic interpolation during simulation of measurements between grid points.

Table I and Table II show the results the first and second set of experiments, respectively. The error shown in the tables is the error between the estimated and true flow fields evaluate at the grid points, irrespective of where the measurements were taken. The lower bound of the RMS error of each method was obtained as the RMS error of w^* , where

$$w^* = \arg \min_{w \in \mathbb{R}^{rZ}} \sum_{x_g \in \mathcal{G}} \|u_{true}(x_g) - \mathbf{H}(x_g)w\|_2^2, \quad (17)$$

where $u_{true}(x_g)$ is the 2D flow velocity vector of the true flow field at $x_g \in \mathcal{G}$, which would be unknown in realistic situations. The RMS error of w^* serves as a measure of the error due to the true flow field being out of the span of $\mathbf{H}(x)$.

Immediately obvious from Table I is that the proposed method (see rows labeled “2.5D bases”) have consistently lower error bounds than the “naive 3D” method. This is evidence that the proposed method increases the span of $\mathbf{H}(x)$ beyond the span of \mathcal{E} in a meaningful way, allowing reduced out-of-span error.

Not only are the *bounds* of the proposed method lower than the bound of the naive 3D method, but so is the *actual achieved* RMS error when 200 or more modes were used. This illustrates that improved estimation performance can be achieved due to the increased expressiveness of $\mathbf{H}(x)$ in the proposed method.

Figure 3a clarifies the reason why so many modes can be discarded during truncation while still retaining good representational ability of $\mathbf{H}(x)$. Noticing the logarithmic scaling of the y-axis, it can be seen that the last few hundred modes have a much smaller contribution compared to the first few. For example, Figure 3d appears to be mostly noise, and its contribution in Figure 3a is approximately 10^{-4} the dominant mode; hence the 401st mode can be omitted without much loss in representational power. However, there does come a point where too many “useful” modes are truncated; we see in both Table I and II that the estimation performance deteriorates as the number of modes drops to 200 or below. Still, in both Tables I and II, a significant number of entries in $\mathbf{H}(x)$ can be discarded while suffering only a small loss in estimation performance (see rows with 760 and 400 modes).

B. Effect on glider planning performance

In this subsection we show additional simulations which highlight the importance of accurate flow field estimation in path planning applications.

A simulated underwater glider mission was conducted in a sub-region of the region considered in the previous subsection. Inspired by mesopelagic ocean science missions (e.g. [28, 29]), we consider an example glider mission where it is released from the surface and attempts to pass through a point at a depth of 500 m. A constant glider velocity relative to flow is chosen by minimising the distance of the resulting path to the target in each flow estimation method, constrained to a fixed glider speed of 0.3 m/s. Under the influence of the true flow field and this velocity, the motion of the glider was simulated using Euler integration with fixed spatial step size.

The resulting glider path for each flow field estimation method is shown in Figure 5. A glider trajectory generated by planning over the true flow field is given to show the ideal performance of the planner. “Our method” refers to the untruncated 2.5D method generated with noise and unconstrained measurements, while the “naive” case is the naive 3D case under the same noise and measurement conditions. It is clear that planning using the estimate generated by the proposed method is closest to ideal performance, followed by the naive 3D method. We can see that a small difference of about 2% in relative error between the 2.5D and 3D methods

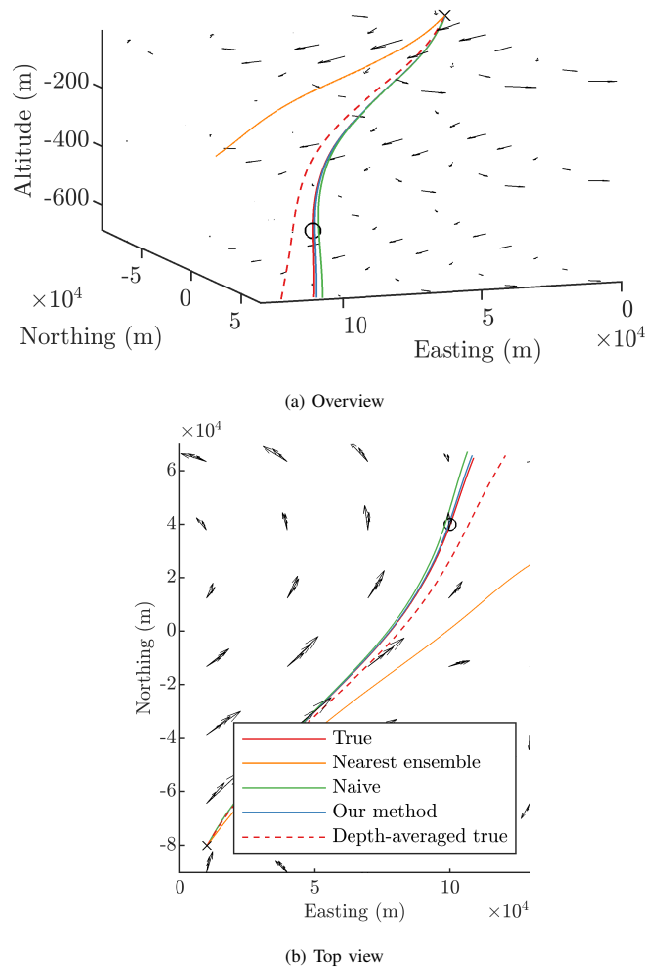


Fig. 5. Simulated glider trajectories with path-planning based on various flow field estimates. “Our method” refers the proposed approach without truncation, and “naive” refers to the “naive 3D” case.

TABLE III
GLIDER PATH RESULTS USING CONTROLS PLANNED WITH DIFFERENT METHODS

	Closest approach (m)
True	16
Nearest ensemble	29858
Depth-averaged true	6192
Naive	1808
Our method	505

shown in Table II) results in a significant difference in error of 1303 m.

Figure 5 also reinforces the idea that existing methods, i.e. depth-averaging, and choosing the nearest ensemble, are insufficient. For the purpose of glider path planning, depth-averaged flow fields are often used; its resulting trajectory suggests that even the depth-averaged *true* flow field performs significantly worse than the 2.5D or 3D estimates. In practical situations, the true flow field will be unavailable; depth-averaging will further reduce the accuracy of any estimate it is performed on, negatively affecting glider performance.

Finally, another possible source of flow field data for glider path-planning is to simply use the nearest ensemble member e_{82} , which provides a 3D flow field. However, it performs the worst of all, suggesting that for the purpose glider path-planning, using the nearest ensemble is insufficient.

VI. CONCLUSION AND FUTURE WORK

We have presented an algorithm for estimating 3D oceanic flow fields that exploits the property of negligible vertical velocity to produce highly accurate results. The most important avenue of future work is to further extend these ideas to address the time-varying case, which arises in long-duration deployments and planning long-distance paths. Another important direction for future work is to perform real-world validation of our algorithm using online observations.

REFERENCES

- [1] D. L. Rudnick, "Ocean research enabled by underwater gliders," *Annual Review of Marine Science*, vol. 8, pp. 519–541, 2016.
- [2] B. J. Nott, "Long-endurance maritime surveillance with ocean glider networks," Master's thesis, Naval Postgraduate School, 2015.
- [3] K. Y. C. To, F. H. Kong, K. M. B. Lee, C. Yoo, S. Anstee, and R. Fitch, "Estimation of spatially correlated ocean currents from ensemble forecasts and online measurements," in *Proc. of IEEE ICRA*, 2021, pp. 1–7. [Online]. Available: <https://arxiv.org/abs/2103.04036>
- [4] W. H. Munk, "Abyssal recipes," *Deep Sea Research and Oceanographic Abstracts*, vol. 13, no. 4, pp. 707–730, 1966.
- [5] M. Roughan and J. H. Middleton, "A comparison of observed upwelling mechanisms off the east coast of Australia," *Continental Shelf Research*, vol. 22, no. 17, pp. 2551–2572, 2002.
- [6] X. Liang, M. Spall, and C. Wunsch, "Global ocean vertical velocity from a dynamically consistent ocean state estimate," *Journal of Geophysical Research: Oceans*, vol. 122, no. 10, pp. 8208–8224, 2017.
- [7] P. Sakov, "EnKF-C user guide," *arXiv:1410.1233 [cs.CE]*, pp. 0–55, 2020.
- [8] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003.
- [9] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble Kalman filter technique," *Monthly Weather Review*, vol. 127, no. 3, pp. 796–811, 1999.
- [10] M. Leutbecher and T. N. Palmer, "Ensemble forecasting," *Journal of Computational Physics*, vol. 227, no. 7, pp. 3515–3539, 2008.
- [11] T. Gneiting and A. E. Raftery, "Weather forecasting with ensemble methods," *Science*, vol. 310, no. 5746, pp. 248–249, 2005.
- [12] P. R. Oke, D. A. Griffin, A. Schiller, R. J. Matear, R. Fiedler, J. Mansbridge, A. Lenton, M. Cahill, M. A. Chamberlain, and K. Ridgway, "Evaluation of a near-global eddy-resolving ocean model," *Geoscientific Model Development*, vol. 6, no. 3, pp. 591–615, 2013.
- [13] G. Madec, *NEMO ocean engine*. Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, No 27, 2008.
- [14] K. M. B. Lee, C. Yoo, B. Hollings, S. Anstee, S. Huang, and R. Fitch, "Online estimation of ocean current from sparse GPS data for underwater vehicles," in *Proc. of IEEE ICRA*, 2019, pp. 3443–3449.
- [15] C. Jidling, N. Wahlström, A. Wills, and T. B. Schön, "Linearly constrained gaussian processes," in *NIPS 2017, December 4–9, Long Beach, CA*, 2017, pp. 1215–1224.
- [16] H. A. Kingravi, H. Maske, and G. Chowdhary, "Kernel observers: Systems-theoretic modeling and inference of spatiotemporally evolving processes," in *Proc. of NIPS*, 2016, pp. 3990–3998.
- [17] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso, "The kriged Kalman filter," *Test*, vol. 7, no. 2, pp. 217–282, 1998.
- [18] J. Whitman and G. Chowdhary, "Learning dynamics across similar spatiotemporally-evolving physical systems," in *Proc. of Machine Learning Research*, vol. 78, 2017, pp. 472–481.
- [19] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.
- [20] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.
- [21] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.
- [22] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.
- [23] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, "A kernel-based method for data-driven Koopman spectral analysis," *Journal of Computational Dynamics*, vol. 2, no. 2, pp. 247–265, 2016.
- [24] T. Nonomura, H. Shibata, and R. Takaki, "Dynamic mode decomposition using a Kalman filter for parameter estimation," *AIP Advances*, vol. 8, no. 10, pp. 1–27, 2018.
- [25] J. H. Tu, J. Griffin, A. Hart, C. W. Rowley, L. N. Cattafesta, and L. S. Ukeiley, "Integration of non-time-resolved PIV and time-resolved velocity point sensors for dynamic estimation of velocity fields," *Experiments in Fluids*, vol. 54, no. 2, 2013.
- [26] Z. Kowalik and T. S. Murty, *Numerical modeling of ocean dynamics*. World Scientific, 1993, vol. 5.
- [27] R. H. Stewart, *Introduction to Physical Oceanography*. Texas A & M University College Station, 2008.
- [28] D. R. Yoerger, M. Curran, J. Fujii, C. R. German, D. Gomez-Ibanez, A. F. Govindarajan, J. C. Howland, J. K. Llopiz, P. H. Wiebe, B. W. Hobson, K. Katija, M. Risi, B. H. Robison, C. J. Wilkinson, S. M. Rock, and J. A. Breier, "Mesobot: An autonomous underwater vehicle for tracking and sampling midwater targets," *Proc. of IEEE/OES AUV Workshop*, pp. 1–7, 2018.
- [29] E. Andruszkiewicz Allan, W. G. Zhang, A. C. Lavery, and A. F. Govindarajan, "Environmental DNA shedding and decay rates from diverse animal forms and thermal regimes," *Environmental DNA*, pp. 1–23, 2020.