Active Drifters: Towards a Practical Multi-Robot System for Ocean Monitoring

Artem Molchanov, Andreas Breitenmoser and Gaurav S. Sukhatme

***Abstract*— We propose a method for controlling multiple active drifters in the presence of external forcing induced by the ocean. Our active drifters have one actuator: they can lower and raise their drogues in depth. By exploiting the vertically stratified nature of ocean currents, we show how classical multi-robot tasks (spreading out and aggregation) can be accomplished by the multi-drifter system. Tests with a realistic simulation based on an ocean model suggest that a practical implementation of active drifters which aggregate and disperse in the coastal ocean could be realized through our control method with relatively inexpensive components. Specifically, we are able to show that over a 90 day deployment a significant fraction of drifters can be made to aggregate in few clusters suitable for recovery.**

I. Introduction

Lagrangian drifters are monitoring devices that are used by oceanographers and biologists to track ocean currents and measure water characteristics. In this work, we are concerned with drifters that are composed of a surface float and a tethered drogue that acts as an "underwater sail"[1] (see Fig. 1).

Drifters are underactuated, and usually passive systems. Positioning its drogue at a fixed depth causes a drifter to travel passively with the current at that depth. This is a common technique in oceanography to tag and track currents (and all that they carry with them).

To achieve better spatial resolution, multiple drifters are commonly deployed. There are two main challenges when it comes to deploying multiple drifters in coastal regions. The first is that they do not tend to provide uniform sampling resolution close to the shore. The second is that they are expensive (in terms of ship time) to retrieve at the end of the mission (being widely dispersed).

Here we study *active drifters* [1], specifically those with a single actuator that adjusts the drogue depth. Changing the drogue depth permits the *in situ* measurement and estimation of ocean current velocity at varying depths. This opens up the possibility of gaining (limited) control of drifter motion since a drifter with a depth-adjustable drogue can actively select the best ocean current for propulsion that achieves some high level mission goal (e.g., aggregation).

Ocean currents are the drifters' main propulsive force. However, this external forcing by ocean currents—along with the underactuation of the drifter system—renders the control of active drifters difficult. The main challenge is that the forces due to currents are both significant and unknown. At best they are unpredictable or difficult to predict. Active drifters are an example of robotic systems under significant continuous external forcing. Other single and multi-robot systems, including underwater vehicles [2] and aerial robots [3], [4], are exposed to external forces in real-world applications. In such settings, tasks such as navigation, station keeping, or formation maintenance are extremely challenging.

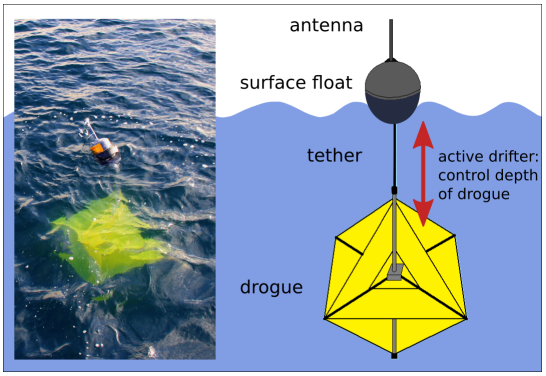

Fig. 1. Lagrangian drifters. Left: A prototype of our passive drifter system, which was deployed in the Southern California Bight to measure ocean currents. Right: The schematic shows the main components and the mode of operation of an active drifter.

In this paper, we extend our work on single active drifters from [1] and consider a multi-drifter system, with two mission objectives. The first objective is *coverage* (drifters are required to spread out) and the second is *aggregation* (drifters are required to cluster near each other). Both are well-studied in the multi-robot research community [5]. We present a control method for active drifters that offers a solution to these two classical multi-robot problems under external forcing induced by the ocean currents. Spreading out enhances the deployment process that drifters naturally experience in the ocean. Aggregation offers the practical benefit that a recovery vessel does not need to search for and pick up each individual drifter at disparate locations at the end of a monitoring mission. It can easily collect the aggregated drifters by visiting few clusters of drifters, which reduces ship operation cost. We report here on the control design and a simulation-based feasibility study to inform the design of a practical active multi-drifter system.

Our results suggest that a practical implementation of drifters with our method could be made to aggregate and

Artem Molchanov, Andreas Breitenmoser and Gaurav S. Sukhatme are with the Department of Computer Science, Viterbi School of Engineering, University of Southern California, 3710 McClintock Ave, Los Angeles, CA 90089, USA. Email: {molchano, breitenm, gaurav}@usc.edu

This work was supported in part by the Office of Naval research (ONR Grant N000141410536: Persistent Autonomy at Sea).

[1]See http://www.pacificgyre.com for an example of commercially available drifters today.

disperse in the coastal ocean with relatively inexpensive components. We are able to show that after deployment a significant fraction of drifters can be aggregated in few clusters over a 90 day period (see IV-D), which greatly facilitates the recovery of the deployed drifters.

Active drifter systems have been the subject of recent study. Regarding the mode of operation, the system we study here is closest to the system described in [6], which can raise and lower a drogue via a winch. An alternative approach uses a free-floating submerged drifter [7] wherein the entire drifter body dives to a certain depth by changing its buoyancy instead of sitting at the surface and controlling a drogue on a tether. Argo floats [8] are larger Lagrangian profilers, which are in wide use in the ocean today. They also adjust their depth by buoyancy control and could theoretically be operated as active drifter systems [9], though in practice they are not operated as such today.

In terms of active drifter control, there are two main approaches. The first is a predictive control approach which explicitly relies on predictions of the ocean currents based on an ocean model [6], [9]. The underlying assumption is that the predictions by the ocean models are reliable, which is not always the case [1]. We follow an alternative approach where each drifter uses in situ measurements of ocean currents to make control decisions.

A control strategy for rendezvous with multiple drifters is presented in [10]. Although their targeted application of aggregating multiple drifters is similar, they use a different approach, where the ocean dynamics is represented by internal wave models with known parameters. In contrast, we assume no explicit knowledge about the dynamics of the ocean, except the vertical component of the flow, which we assume to be zero. In [11] the target application is to control the absolute position of a drifter in a coastal scenario, where the drifter can anchor itself at the sea bottom if necessary. As an extension, the deep ocean scenario for a single drifter is examined under idealized conditions (the drifter is assumed to have instant estimates of currents at the present location, the ocean currents are assumed to be stationary and to span the plane positively). The external forcing by current flow fields and its impact on the control of underactuated robotic systems is furthermore addressed by [12] for a coverage task with self-propelled vehicles of bounded velocity in a river environment and by [13] for tasks of tracking coherent structures on flows with autonomous underwater vehicles in the ocean.

We state the drifter control problem formally in Section II. Section III introduces the high-level as well as the low-level control methods. The system parameters are defined, and the feasibility of the multi-robot active drifter system is tested by simulations in Section IV. Section V concludes the paper and outlines a plan for future work.

II. Problem Formulation

Our representation of the ocean relies on ocean current velocity vectors which are changing over time. This defines the time-varying flow field $\mathbf{f}\colon\ \mathbb{R}^3 \times \mathbb{R}_{\geq 0} \to \mathbb{R}^3$, with vectors

$$\mathbf{f}(\mathbf{x},\, t) = \left(f_{\mathrm{x}}(\mathbf{x},\, t),\, f_{\mathrm{y}}(\mathbf{x},\, t),\, f_{\mathrm{z}}(\mathbf{x},\, t)\right)^{\mathrm{T}}. \tag{1}$$

$f_{\mathrm{x}}(\mathbf{x},\, t)$ and $f_{\mathrm{y}}(\mathbf{x},\, t)$ are the horizontal components of the velocity vectors in east and north direction. t denotes the time in the model and $\mathbf{x} = (x, y, z)^{\mathrm{T}}$ is the 3D position. In what follows, we will neglect the vertical flow component and assume $f_{\mathrm{z}}(\mathbf{x},\, t) = 0$ everywhere.

In order to simulate realistic dynamics of the ocean, we obtain the flow field from the Regional Ocean Modeling System (ROMS) [14][2]. ROMS serves as our "ocean simulator": it generates velocity vectors for discrete time, with temporal resolution of 1 hour over a grid with spatial resolution of 3 km $\times$ 3 km. We interpolate the velocity vectors over the ROMS grid to retrieve continuity.

Given a group of N active drifters, each drifter's state is described by the 2D position of its surface float, $\mathbf{p}_i = (x_i,\, y_i)^{\mathrm{T}} \in \mathbb{R}^2$, and the position of its tethered drogue at depth $z_i,\ i \in \{1, \ldots, N\}$. In this treatment we assume that the drogue has the same horizontal position as the surface float. In practice this will not be the case, but the horizontal offset between the two will not be significant relative to the size of the coverage area. We further assume absolute localization and global communication capabilities of the drifters within a centralized network (e.g., using GPS and satellite communication via a receiver on the surface float), such that the drifters can measure their positions with sufficient accuracy and are able to exchange data among each other via a central base station if needed.

We model the drifters as first-order systems with Lagrangian dynamics,

$$\dot{\mathbf{p}}_i = \left(f_{\mathrm{x}}(\mathbf{x}_i(t),\, t),\, f_{\mathrm{y}}(\mathbf{x}_i(t),\, t)\right)^{\mathrm{T}} \text{ and } \dot{z}_i = u_i(t)\,, \tag{2}$$

with $\mathbf{x}_i(t) = \left(\mathbf{p}_i(t)^{\mathrm{T}},\, z_i(t)\right)^{\mathrm{T}}$ and control input

$$u_i(t) = \begin{cases} v & ,\, z_i^* > z_i(t) \\ 0 & ,\, z_i^* = z_i(t) \\ -v & ,\, z_i^* < z_i(t)\,, \end{cases} \tag{3}$$

where z_i^* denotes the desired depth, and v represents the constant vertical speed for actuating the drogue.

Given the above ocean and drifter models, our goal is to control the drogues of N drifters in such a way that the drifters manage to 1) spread out inside a preference area A, and 2) aggregate within the overall mission area Ω, where $A \subset \Omega \subset \mathbb{R}^2$, despite the external forcing of the ocean currents. In other words, with respect to spreading out, the drifters should finally cover A in the limit by incrementally maximizing the minimum distance between drifters contained in A,

$$\max_{\mathbf{u}(t)} \Big(\min_{\mathbf{p}_i,\, \mathbf{p}_j \in A,\ i,\, j \in \{1,\, \ldots,\, N\},\ i \neq j} \left\| \mathbf{d}_{i,\, j} \right\|_2 \Big)\,, \tag{4}$$

where $\mathbf{u}(t) = (u_1(t), \ldots, u_i(t), \ldots, u_N(t))$ and $\mathbf{d}_{i,\, j} = \mathbf{p}_j - \mathbf{p}_i$. Similarly, with respect to aggregation, the drifters

[2] See also `http://ourocean.jpl.nasa.gov/`.

are required to converge to clusters within Ω, i.e., the distances between the drifters $\|\mathbf{d}_{i,j}\|_2$ are minimized over time. This (reverse) process results in the maximization of an *aggregation metric*. We delay the introduction of this metric till Section IV where the quantitative evaluation of the active drifter system is discussed.

III. Drifter Control Methods

We design a control law that utilizes in situ measurements of ocean currents collected by the drifter. Our control method consists of two layers: a *high-level controller* that generates the desired instantaneous motion direction for each drifter and a *low-level controller* (a tracking controller) that selects a depth at which the ocean current best causes the drifter to move along the desired direction.

A. High-Level Control Law

The high-level controller is a potential field controller, which generates a unit direction vector $\tilde{\mathbf{F}}_i$ for a drifter i,

$$\tilde{\mathbf{F}}_i = \frac{\mathbf{F}_i}{\|\mathbf{F}_i\|_2} \ , \text{ with } \ \mathbf{F}_i = \sum_{j=1,\, j\neq i}^{n} \mathbf{F}_{d_{i,j}} + \mathbf{C}_i \ , \tag{5}$$

and

$$\mathbf{F}_{d_{i,j}} = \begin{cases} \xi \left(\frac{r_d}{\|\mathbf{d}_{i,j}\|_2}\right)^2 \tilde{\mathbf{d}}_{i,j} & , \|\mathbf{d}_{i,j}\|_2 > d_{\text{min}} \\ 0 & , \|\mathbf{d}_{i,j}\|_2 \leq d_{\text{min}} \end{cases} \tag{6}$$

$$\mathbf{C}_i = \left(\frac{\|\mathbf{d}_{i,c}\|_2}{r_c}\right)^{w_c} \tilde{\mathbf{d}}_{i,c} \, , \tag{7}$$

where $\mathbf{F}_{d_{i,j}}$ is the interaction force which attracts (or repels) drifter i toward (or from) drifter j, $\tilde{\mathbf{d}}_{i,j}$ and $\tilde{\mathbf{d}}_{i,c}$ are unit direction vectors, d_{min} is the user-defined minimum distance of interaction between two drifters, and r_d is the radius at which the 2-norm of the interaction force of the drifters equals 1. $\xi = 1$ holds when a drifter is in the "aggregation" mode (attraction), and $\xi = -1$ when a drifter is in the "spreading" mode (repulsion). $\mathbf{C}_i$ is the force of attraction toward the center of the preference area A defined by the central point $\mathbf{p}_c$ and the radius r_c. $\mathbf{C}_i$ should not disturb the interaction of drifters inside the preference area, but, at the same time, it should dominate outside the area. We achieve this by setting w_c to a high value. As the high-level controller generates only the direction for the motion, we are not interested in the magnitude of the force and normalize the vector $\mathbf{F}_i$ to obtain the unit direction vector $\tilde{\mathbf{F}}_i$.

B. Low-Level Control Law

The purpose of the low-level controller is to select the best depth at which the propulsion due to the ocean currents will cause the drifter to move in the direction designated by the high-level control law. In this work, we consider a discrete set of depths. The choice of these depths is non-trivial and is explained in Section IV-B; it involves *sampling*, i.e., making measurements of currents at various depths, and *decision making*, i.e., choosing a particular depth.

With the drogue positioned at a particular depth for a small duration, the GPS on the surface is able to measure a change in drifter position and hence estimate motion locally. We ascribe this motion entirely due to the propulsive force at that depth, thereby estimating the ocean current at the present drogue depth. The low-level controller projects the ocean current vector onto the desired direction $\tilde{\mathbf{F}}_i$ (generated by the high-level controller) to obtain the *sample quality* Q of the ocean current that has effectively been sampled at the present depth. The sample quality thus measures the fitness of the ocean current at a particular depth.

The condition that triggers a *control decision event* is defined as

$$|\alpha - \alpha_{\text{prev}}| > \alpha_{\text{tol}} \, , \tag{8}$$

where α is the angle between the latest estimate of the current the drifter is traveling with and the direction vector $\tilde{\mathbf{F}}_i$, α_{prev} corresponds to α for the drogue depth at which the previous control decision was made, and α_{tol} is the *tolerance angle* (a threshold value). Thus, the low-level controller makes a new decision every time the situation changes "significantly enough", which is defined by the tolerance angle.

The basic low-level controller employs a simple strategy. It periodically estimates the direction of the current it is traveling with. If a control decision event is triggered, the drogue starts sampling by cycling through a discrete range of depths. At each depth, the ocean current is estimated. Once all estimates are made, the controller picks the depth with the highest value of Q, and saves the present value of α (at the depth chosen by the controller) as α_{prev}.

This basic controller has two limitations. First, it does not reuse ocean current estimates. All the estimates from the previous control event are treated as outdated every time a new decision is required, even though, according to (8), in some cases a control decision event may be triggered due to frequent changes in the desired direction. For example, such behavior is typically observed when the drifters are aggregated in a cluster. Second, it does not leverage the aggregation of drifters. Accordingly, we extend the basic controller as follows to improve on performance.

In order to reuse previous ocean current estimates, we introduce the *trust time* T_{trust} and associate time stamps with ocean current estimates. T_{trust} defines how long an estimate can be used before it is outdated. The sampling strategy is modified as follows. Every time a drifter needs to make a new control decision, it re-samples only at the depths where its ocean current estimates are outdated. A decision is made once the estimates at all depths in the set are up to date.

The second extension uses samples acquired by other drifters in the cluster and induces *collaboration* among closely located drifters. Every time a new control decision is required, a drifter creates a pool of ocean current estimates, which consists of estimates acquired by itself and those acquired from neighboring drifters that are within the data exchange distance. We choose the data exchange distance for our system to be the same as the minimum distance of interaction d_{min}. For every depth value, the estimates

are sorted according to their time stamp (the most recently acquired estimates are put first). The drifter puts the latest estimate for every depth value into the *best set*. Next, the drifter moves the drogue to only the depths for which the estimates are outdated in the best set. The best set is updated every time a new estimate is made. Once all estimates in the best set are up to date, the new control decision is made. This process is asynchronous and stochastic. It is asynchronous in the sense that current estimates can be shared as available—there are no preset communication slots. The clocks on the drifters do need to be synchronized however. This is a reasonable requirement considering the drifters are GPS equipped. The process is stochastic in the sense that there is no centralized current estimation task assignment, and, since all drifters act asynchronously, ocean current estimation happens randomly.

For the practical application, the high-level controller can be implemented as a centralized controller at the base station and a copy of the low-level controller resides on each drifter. Each drifter communicates its coordinates to the base station, and receives a desired direction vector back from the base station. Additionally, since drifters collaborate only when they are in close proximity to each other, we believe that the exchange of ocean current estimates can be done via RF modems in a practical system.

IV. Performance Evaluation

A. Aggregation Metric

In order to numerically evaluate the performance of the control algorithms, we propose a metric similar to a normalized pairwise potential energy of points,

$$M_a(D,n,N) = \frac{(\sum_{i=1}^{n}\sum_{j=1}^{n} U(\|\mathbf{d}_{i,j}\|_2)) - n}{N^2 - N}\,, \qquad (9)$$

where N is the initial number of drifters and n denotes the number of remaining drifters still left in the mission area Ω at a given time. $D \in \mathbb{R}^{n^2}$ is a square matrix of horizontal distances between the n remaining drifters, and $\|\mathbf{d}_{i,j}\|_2$ are elements of this matrix. Thus, since the normalization factor depends on N but the sum of the potential energies of individual pairs depends on n, the aggregation metric penalizes the fact that drifters leave the mission area (such drifters are considered to be lost). The function U is given as

$$U(d) = f_s(d, a_u)\,, \qquad (10)$$

where the tuple a_u is a special instance of the parameter tuple $a = (a_{00}, a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23})$ of the logistic function f_s for the potential field. The logistic function f_s is defined as

$$f_s(x,a) = \frac{1}{a_{00}}\left(\frac{a_{11}}{1+e^{a_{12}(x-a_{13})}} + \frac{a_{21}}{1+e^{a_{22}(x-a_{23})}}\right). \qquad (11)$$

The parameters a_u are chosen such that the function has the highest drop approximately between 20 000 and 50 000, which corresponds to distance in meters. This zone is a transition area, once drifters get into this zone the metric starts growing faster, reflecting the fact that the drifters are approaching aggregation. Fig. 2 shows the graph of $U(d)$ for our choice of a_u with parameter values $a_{00} = 1.77203, a_{11} = 1, a_{12} = 0.0002, a_{13} = 35\,000, a_{21} = 1, a_{22} = 0.000035, a_{23} = 35\,000$.

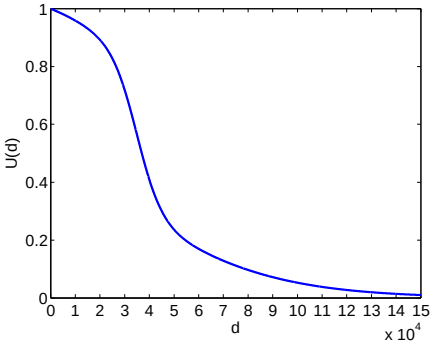

Fig. 2. The artificial potential energy function $U(d)$ for tuple a_u. The function defines the quality of aggregation of a pair of drifters.

M_a is normalized, thus it assumes 1 when all drifters are aggregated at one point (independent of the number of drifters). In order to determine the approximate number of clusters of drifters, n_{clust}, we use the heuristic rule

$$n_{\text{clust}} \approx 1/M_a\,, \qquad (12)$$

which agrees with our observations in simulations.

B. ROMS Parameters

In our simulations, we use a discrete ROMS model with interpolation to the nearest grid point in order to generate a realistic ocean flow field. Since there is a cost associated with estimating currents at each depth, in practice we want the controller to work with as few depths as possible. An active drifter is controllable if the currents in the area span the plane positively [11]. This criterion can be reformulated as follows

$$\max_{i,j}(\theta_{i,j}(\mathbf{p}_l)) < \pi\,, \qquad (13)$$

where $\theta_{i,j}(\mathbf{p}_l)$ is the angle (in the horizontal plane) between adjacent current vectors at depths i, j at the point $\mathbf{p}_l$. This means that if we project current vectors with the same horizontal initial point $\mathbf{p}_l$ onto the horizontal plane, and the largest angle between these vectors is less than π, the system is controllable. This rule also implies that the minimum number of depths necessary is 3 (i.e., two vectors cannot span the plane positively). Thus, in order to find the appropriate depths we introduce the integral fitness function

$$R = \sum_{k=1}^{T}\sum_{l=1}^{P} f_a(\theta_{\text{max1}}(f'_{k,l}),\, \theta_{\text{max2}}(f'_{k,l}))\,, \qquad (14)$$

where R depends on the ROMS dataset represented by P spatial points (in our case it is a mission area) and T time points (in our case it is a number of hours) in a given time period of the dataset. $f'_{k,l} = \left(\mathbf{f}((\mathbf{p}_l^T, z_1)^T, t_k),\, \mathbf{f}((\mathbf{p}_l^T, z_2)^T, t_k),\, \mathbf{f}((\mathbf{p}_l^T, z_3)^T, t_k)\right)$ is a set of three ocean current vectors at a given time t_k and

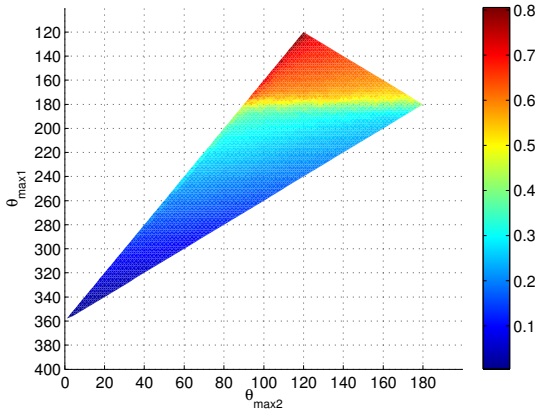

Fig. 3. ROMS point-wise fitness function $f_a(\theta_{\text{max1}}, \theta_{\text{max2}})$. The function defines how well the plane is spanned at a point by a triple of vectors, based on the angles between each pair of vectors.

given horizontal point $\mathbf{p}_l$, but at different depths z_1, z_2, z_3. θ_{max1} is the largest angle of a triple of angles formed by the current vectors at different depths at the same $\mathbf{p}_l$, θ_{max2} is the second largest angle. The point-wise fitness $f_a(\theta_{\text{max1}}, \theta_{\text{max2}})$ is now given by

$$f_a(\theta_{\text{max1}}, \theta_{\text{max2}}) = f_s(\frac{\theta_{\text{max1}}}{\pi}, a_a) + (\frac{\pi - \theta_{\text{max2}}}{\pi}) f_s(\frac{\theta_{\text{max2}}}{\pi}, a_a) - \Delta f_s\,, \quad (15)$$

where Δf_s is bias compensation, such that the point-wise fitness is 0 when all three vectors are collinear, a_a is a special instance of the parameter tuple a of the logistic function f_s presented earlier in (11). Fig. 3 depicts the point-wise fitness function $f_a(\theta_{\text{max1}}, \theta_{\text{max2}})$ in the space of its arguments. As one can see, the function has its maximum value when $\theta_{\text{max1}} = \theta_{\text{max2}} = 120°$, and has the highest derivative in the area where $\theta_{\text{max1}} \approx 180°$ with minor influence of θ_{max2}. f_a is also used to plot fitness maps, which are color maps (with a color corresponding to the value of the function at the point) acquired by summation of point-wise fitness values over time at each point. Such fitness maps allow visualization of the points where currents span the plane positively. These fitness maps are similar to the controllability heat maps introduced in [9]. The fitness function presented above allows us to compare and pick the most appropriate combination of three depths. For this, we calculate the fitness value for every possible combination of depths and pick the combination that scores the highest fitness value over the whole map. As an exemplar we calculate the best combination of depths for January 2014—the first month of data that is used in the simulations reported here. We also calculated the best combination for February and March of 2014. They yielded the same combination of depths. Based on this, for purposes of the present study, all further investigation was conducted for the January 2014 dataset. The best depths are $\{0, 75, 400\}$ m. To evaluate the results we build the fitness map for these three depths and compare it to the best map H_{12}. We build H_{12} as follows: for a given hour k, we take the time slice of the full (12 depth) ROMS flow field (i.e., a multidimensional array representing this flow field at the given hour) and for each surface position $\mathbf{p}_l$ of this slice, we compare all combinations of depth triples by their point-wise fitness value. For each position, we select the best depth triple and save its scalar fitness value in the matrix H_{12}^k. The target best map is obtained as the matrix

$$H_{12} = \sum_{k=1}^{T} H_{12}^k\,, \quad (16)$$

where T depends on the time span used for calculations (as mentioned, we used one month of data). The results are shown in Fig. 4 on the left and in the center. As one can see, three depths are not enough to approximate H_{12}. We add additional depths greedily, by adding the depth that gives the highest gain of fitness at each step, and terminate the procedure once the map is represented "reasonably well". The resulting map built on a six depth set $\{0, 10, 30, 75, 200, 400\}$ m is shown in Fig. 4 on the right. The overall fitness of this map is $\approx 85\%$ of the fitness of the 12 depth map. We treat this as a reasonable approximation; these are the six discrete depths used in the simulations reported here[3].

C. Upper Bound of Performance

We find the upper bound by evaluating the performance of the high-level algorithm for an *ideal drifter*. An ideal drifter is capable of

- instantaneously changing the drogue depth;
- instantaneously estimating ocean currents, i.e., the drifter spends no time with the drogue at a particular depth to estimate currents at that depth;
- making ideal measurements of ocean currents with no estimation noise;
- perfect communication.

Thus, at the moment of taking a control decision, the ideal drifter knows all currents at its present location instantaneously. This scenario gives us an upper bound of what can be achieved by our algorithm in the best case. All further evaluations are performed with the simulation setup parameters given in Tab. I, for 100 simulation runs.

TABLE I
SIMULATION PARAMETERS

ROMS dates	starting with January, 01, 2014
Simulation time span	90 / 180 days
Depths set	{ 0, 10, 30, 75, 200, 400 } m
Number of drifters	30
Number of runs	100
r_d	30 000 m
d_{min}	1 000 m
p_c	longitude = -124.75, latitude = 35.75
r_c	250 000 m
w_c	10
Simulation time step	300 s

[3]Note that adjusting the set of depths online during a mission presents an interesting extension of the controller and is subject to future work.

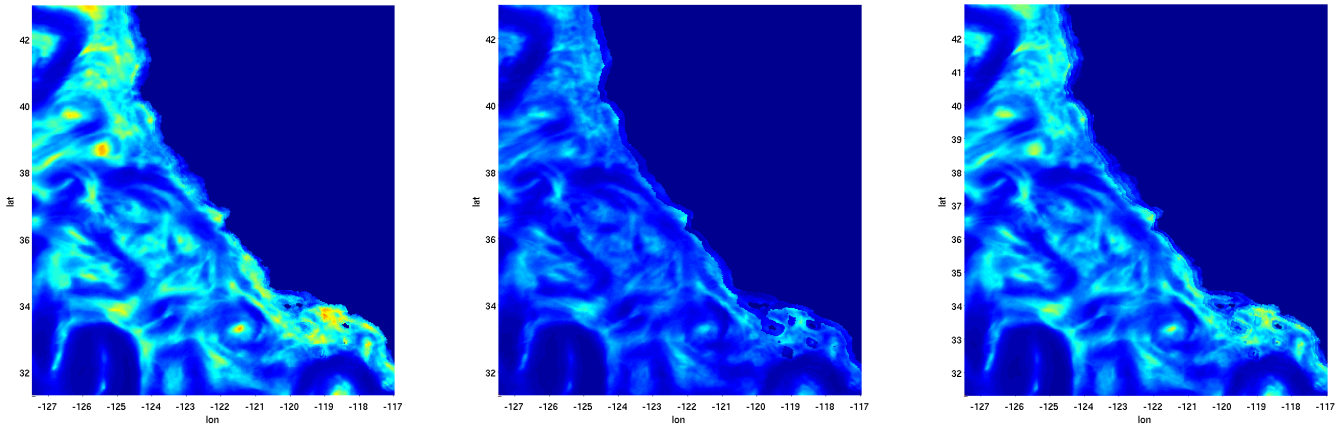

Fig. 4. Plots of ROMS fitness maps. Left: A map created from the full 12 layer set. Center: A map created from the best 3 layer set. Right: A map created from the 6 layer set used in our simulations.

TABLE II

SYSTEM PERFORMANCE OVER 100 SIMULATIONS

	Average metric		Average number of drifters lost	
	90 days	180 days	90 days	180 days
Ideal drifter	0.49	0.55	7	8.5
Realistic drifter	0.21	0.29	9	12.8

As mentioned earlier, the aggregation area in the mission space Ω is defined by the central attraction point $\mathbf{p}_c$ and the radius r_c of the aggregation high-level controller introduced in (5). Each of the 100 simulation runs performs a full aggregation scenario with random initial positions of drifters inside the preference area A. The results of the simulation for 90 and 180 days are presented in Tab. II for the ideal drifters. Fig. 5 presents the average evolution of the aggregation metric (9) over time. As one can see, the ideal drifter scenario is almost saturated after 90 days of simulation and it has $\approx 90\%$ of its final value at this point. Thus, we decide to limit simulation time to 90 days for all further evaluations. This time cutoff has some advantages. First, late aggregation is penalized, which is important since in practice drifters may have limited life time. Second, we partially account for cases where saturation may happen earlier and the system starts deteriorating by losing drifters (i.e., they leave Ω), which happens inevitably when the time span is increased.

From the simulations and applying heuristic (12), one can see that for the ideal drifter case we obtain ≈ 2 clusters after the process is finished, which is a very good result, considering the nature of external forcing and the fact that the algorithm utilizes no predictions but only in situ measurements. The choice of parameters for the *realistic drifter* scenario is explained next.

D. Performance under Estimation Limitations

In the previous section, we presented simulations that were performed under the assumption that the drifter can estimate currents instantaneously. This is never the case in reality. A drifter must sample currents at various depths before it makes an appropriate decision according to the low-level controller (see III-B).

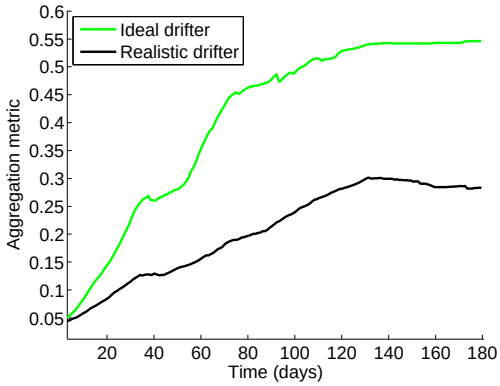

Fig. 5. Evolution of the average aggregation metric over time based on 100 simulation runs. The green curve represents the performance of ideal drifters. The black curve represents the performance of realistic drifters with a low-level controller.

For our system, we assume that, in order to acquire a current estimate, the drogue has to submerge to a corresponding depth and drift for a certain time T_{est} with the ocean current. As an initial guess, we assume the estimation time to be $T_{\text{est}} = 10$ minutes per depth.

The vertical velocity of a realistic drogue is bounded. Our model uses the constant speed v for climbing and diving. The choice is based on parameters of existing buoyancy-driven vehicles [7], [16]–[18] that report achievable maximum vertical velocities in the range of 0.1 to 0.5 m/s. For our system, we chose a reasonable value in the middle of this range, and set it to 0.3 m/s.

As can be inferred from [15], given a slowly moving object, such as the drifter, with estimation time on the order of minutes and almost constant speed, modern estimation approaches can reduce the velocity measurement error to the order of millimeters per second. Hence, given that the average velocity of the drifter is around 0.1 to 0.2 m/s, such estimation errors due to noise can be neglected even for realistic drifters at first.

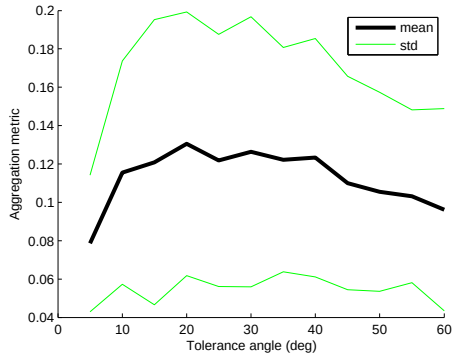

Fig. 6. Performance of the realistic drifter system with the basic low-level controller under different tolerance angles. Every point is based on 100 simulation runs. The black curve represents the mean value and the green curves represent the standard deviation of the aggregation metric at every point.

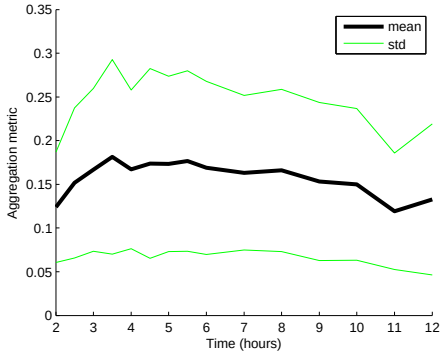

Fig. 7. Performance of the realistic drifter system with the extended low-level controller under different T_{trust}. Every point is based on 100 simulation runs. The black curve represents the mean value and the green curves represent the standard deviation of the aggregation metric at every point.

Another parameter of the basic low-level control algorithm is the tolerance angle α_{tol}. Fig. 6 presents the graph of performance of the system under different tolerance angles. From this graph we can infer that our system has the best average performance at $\alpha_{\text{tol}} = 20° \pm 5°$, thus, for the selection of the remaining parameters and further simulations we fix the value to $20°$.

The next set of parameters belongs to the extended version of the low-level control algorithm (see III-B). The first is the trust time T_{trust}. Fig. 7 depicts the graph of the performance of the system for varying T_{trust}. As one can see, the performance deterioration starts approximately after the 3.5 hour mark. Thus, for further experiments we pick $T_{\text{trust}} = 3.5$ hours.

The final addition to the low-level control algorithm is the ability to share information about ocean current estimates. We implemented it as a sample pool—a set of current estimates that every drifter collects from drifters that are within a distance d_{min} when the control decision is taken. Such collaboration helps drifters to save time by splitting the task of current estimation between nearby drifters stochastically, and, hence, taking advantage of clustering (see III-B). The performance of the system in this setting is shown in Tab. II for the realistic drifter scenario. The average performance of 0.21 and 0.29 (for 90 and 180 days of simulation) corresponds to $\approx$ 3–5 clusters (with around 6–10 drifters per cluster). Although it is lower than the average score for the ideal drifter scenario, we are encouraged by the performance and the implication for the practical use. Considering high ship operation costs, such clustering can significantly facilitate the process of collection of drifters.

Finally, we demonstrate an example of a complete deployment scenario, where drifters are dropped at one position (marked with a green cross), then spread all over the area during 90 days (Fig. 9 left) and finally aggregate over the last 90 days (Fig. 9 right). Although we have quite a good aggregation (more than a half of drifters assembled), the seemingly low score in this scenario (≈ 0.31) reflects the fact that the rest of the drifters are either lost or completely spread through the area due to external forcing (in the particular case of Fig. 9: 10 lost, 3 outliers).

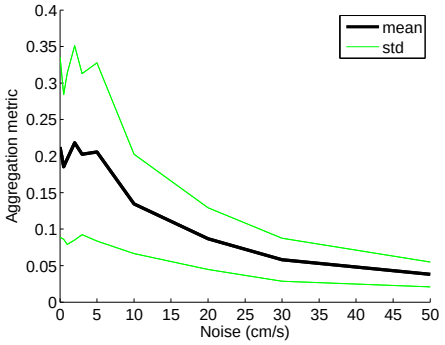

Fig. 8. Performance of the realistic drifter system with the extended controller under different estimation noise levels. The level of the noise is defined through the upper limit of a triangular distribution. The system does not exhibit significant drop in performance up to the noise level of 5 cm/s. After this mark the performance degrades gracefully. Every point is based on 100 simulation runs. The black curve represents the mean value and the green curves represent the standard deviation of the aggregation metric at every point.

E. Performance under Noisy Estimation

In this section, we finally evaluate the system's performance under estimation noise. We assume estimations of velocities are acquired from GPS data solely. As mentioned above in IV-D, modern estimation approaches allow to obtain estimation errors on the order of millimeters per second. We now calculate the errors for a worst case scenario. For that, let us consider the simple velocity estimator

$$\hat{\mathbf{f}} = \frac{\mathbf{p}_k - \mathbf{p}_{k-1}}{T_{est}}\,. \tag{17}$$

For the GPS error, we assume simple additive noise with a uniform distribution symmetric around 0 with a lower limit $-e_{\text{max}}$ and upper limit e_{max}, and independence of noise components. For (17), we get a triangular distribution of the noise (as the distribution of the sum of uniformly distributed random variables) with half the distribution width $e_{\text{tr}} = \frac{2e_{\text{max}}}{T_{\text{est}}}$. For the value of e_{max}, we take the error of a popular inexpensive GPS satellite messenger Spot Tracker[4]

[4]For more details, see `https://www.findmespot.com/downloads/SPOT2-SellSheet.pdf`

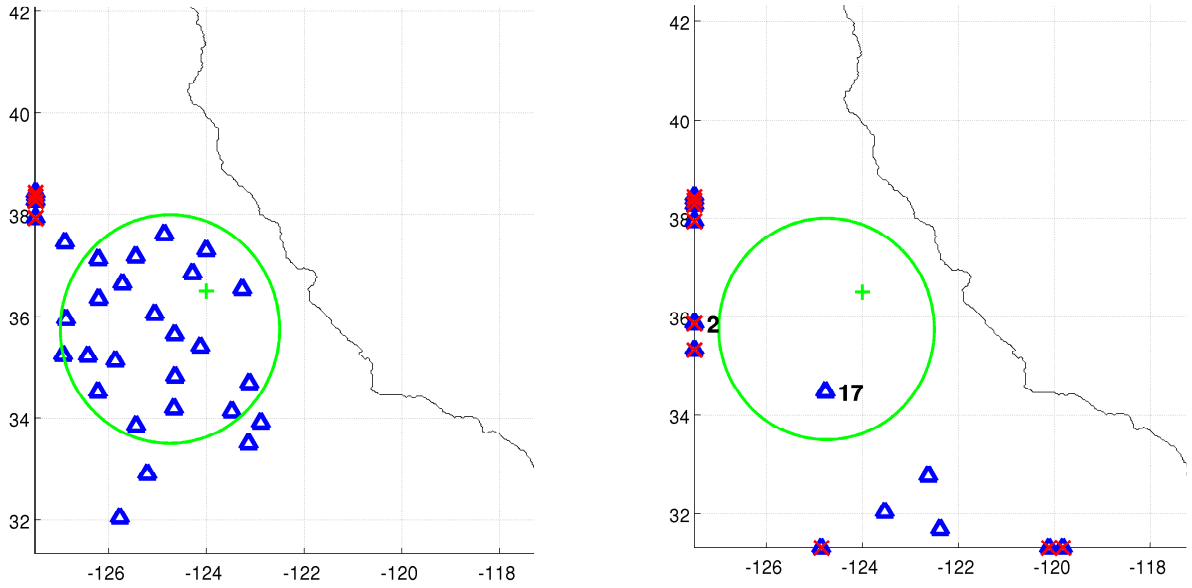

Fig. 9. Left: An example of positions of 30 drifters after 90 days of spreading. Right: An example of positions of 30 drifters after 90 days of aggregation following the spreading phase (180 days of simulation total). Blue triangles denote drifters and drifter clusters. Numbers near triangles denote cluster sizes. Triangles without numbers denote single drifters. Crossed out triangles on the borders (i.e., borders of Ω) denote lost drifters. The green cross marks the initial position of drifters and the green circle denotes the preference area A. The black thin curve denotes the coastal line.

$e_{\max} = 6.4\,\text{m}$. Thus, for the triangular distribution we have $e_{\text{tr}} \approx 0.021\,\text{m/s}$.

Fig. 8 depicts the curve of the average aggregation metric for different e_{tr}. As one can see, the system is robust to the estimation noise. In the worst case scenario with e_{tr} and even beyond (with noise up to $5\,\text{cm/s}$) there is no significant drop in performance. More than that, the system exhibits critical problems (i.e., performance lower than 0.1) only at the upper limit of the noise distribution of $20\,\text{cm/s}$, which corresponds to the average velocity of the ocean currents.

V. Future Work

In our future work, we plan to address other key aspects of an effective practical active drifter system, namely the limitations in communication and the power efficiency. Further interesting research directions include the study of other multi-robot tasks under the influence of external forcing, such as loitering and formation maintenance.

References

[1] A. Molchanov, A. Breitenmoser, S. Subbaraya and G. S. Sukhatme, "Active Drifters: Sailing with the Ocean Currents", in *RSS 2014 Workshop on Robotic Monitoring*, 2014.

[2] A. A. de Menezes Pereira, J. Binney, G. A. Hollinger and G. S. Sukhatme, "Risk-Aware Path Planning for Autonomous Underwater Vehicles using Predictive Ocean Models", in *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.

[3] V. R. Desaraju and N. Michael, "Hierarchical Adaptive Planning in Environments with Uncertain, Spatially-Varying Disturbance Forces", in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 5171–5176, 2014.

[4] M. T. Wolf, L. Blackmore, Y. Kuwata, N. Fathpour and A. Elfes, "Probabilistic Motion Planning of Balloons in Strong, Uncertain Wind Fields", in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1123–1129, 2010.

[5] R. M. Murray, "Recent Research in Cooperative Control of Multivehicle Systems", in *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[6] M. Dunbabin, "Optimal 4D Path-Planning in Strongly Tidal Coastal Environments: Application to AUVs and Profiling Drifters", in *RSS 2012 Workshop on Robotics for Environmental Monitoring*, 2012.

[7] Y. Han, R. A. de Callafon, J. Cortés and J. Jaffe, "Dynamic Modeling and Pneumatic Switching Control of a Submersible Drogue", in *Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics*, pp. 89–97, 2010.

[8] J. Gould et al., "Argo Profiling Floats Bring New Era of In Situ Ocean Observations", in *EOS Trans.*, vol. 85, no. 19, pp. 185–190, 2004.

[9] R. N. Smith and V. T. Huynh, "Controlling Buoyancy-Driven Profiling Floats for Applications in Ocean Observation", in *IEEE Journal of Oceanic Engineering*, 2013.

[10] M. Ouimet and J. Cortés, "Coordinated Rendezvous of Underwater Drifters in Ocean Internal Waves", in *Proc. of the IEEE Conf. on Decision and Control*, pp. 6099–6104, 2014.

[11] J. Jouffroy, Q. Zhou and O. Zielinski, "On Active Current Selection for Lagrangian Profilers", in *Modeling, Identification and Control*, vol. 34, no. 1, pp. 1–10, 2013.

[12] A. Kwok and S. Martínez, "A Coverage Algorithm for Drifters in a River Environment", in *Proc. of the American Control Conf.*, pp. 6436–6441, 2010.

[13] M. Michini, M. A. Hsieh, E. Forgoston and I. B. Schwartz, "Robotic Tracking of Coherent Structures in Flows", in *IEEE Transactions on Robotics*, 2014.

[14] A. F. Shchepetkin and J. C. McWilliams, "The Regional Ocean Modeling System (ROMS): A Split-Explicit, Free-Surface, Topography-Following-Coordinate Oceanic Model", in *Ocean Modeling*, vol. 9, no. 4, pp. 347–404, 2005.

[15] L. Serrano, D. Kim and R. B. Langley, "A GPS Velocity Sensor: How Accurate Can It Be? A First Look", in *Proc. of The Institute of Navigation. National Technical Meeting*, pp. 875–885, 2004.

[16] B. Reed, C. Ambler, J. Guerrero and F. Hover, "Vertical Glider Robots for Subsea Equipment Delivery", in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2356–2361, 2011.

[17] A. Schwithal and C. Roman, "Development of a New Lagrangian Float for Studying Coastal Marine Ecosystems", in *Proc. of the Oceans 2009 - Europe*, pp. 1–6, 2009.

[18] C. C. Eriksen et al., "Seaglider: A Long-Range Autonomous Underwater Vehicle for Oceanographic Research", in *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 424–436, 2001.