

Glider CT: Analysis and Experimental Validation

Dongsik Chang, Wencen Wu, and Fumin Zhang

Abstract Underwater gliders are robust ocean sensor platforms characterized by high reliability and endurance. Due to their relatively low speed, the motion of underwater gliders is strongly affected by ocean currents, which provides data to estimate the depth averaged flow velocity. The glider computerized tomography (Glider CT) algorithm reconstructs a depth-averaged flow field from the navigation errors accumulated along the glider trajectories. This paper justifies the convergence of the Glider CT algorithm as a row action method solving nonlinear equations previously used for bent-ray ultrasonic CT. The paper also validates the algorithm through experiments where the horizontal motion of underwater gliders under flow is imitated by mobile robots in a lab setting. Both theoretical analysis and experimental results suggests the Glider CT algorithm as a promising method for marine operations.

1 Introduction

The underwater glider has found a broad range of applications such as oil field surveys, military operations, and deep-sea and coastal research [10, 13]. The sampling and monitoring performance of gliders significantly relies on the navigation performance of gliders. Due to their relatively low speed, motion of gliders are sensitive to the ocean current. Therefore, control systems [2, 4, 11], control algorithms [5, 17], and path planning algorithms [2, 4, 15] have been developed to navigate gliders through ocean flow with improved performance.

The primary means of localization for underwater gliders is the global positioning system (GPS) [7]. However, since GPS signals cannot propagate through sea water, gliders estimate their underwater positions via dead-reckoning between regu-

Dongsik Chang · Fumin Zhang
Georgia Institute of Technology, e-mail: {dsfrancis3,fumin}@gatech.edu

Wencen Wu
Rensselaer Polytechnic Institute e-mail: wuw8@rpi.edu

lar surfacing events for GPS updates. Because underwater gliders swim at relatively low speed, their trajectories are strongly perturbed by ocean currents. Hence, we can typically observe a difference between the dead-reckoning and actual surfacing positions of a glider. We refer to this difference as the *dead-reckoning error*. To mitigate the dead-reckoning error, an underwater glider computes an estimate of average flow velocity along the trajectory between the last and current surfacing positions, and incorporates the flow estimate into navigation until the following surfacing event [8]. However, the estimate does not account for the temporal/spatial variations of the flow field during navigation, and our recent work [2] emphasized the importance of incorporating such variations into navigation algorithms in field deployments.

In [2, 3], we proposed efficient methods for real-time ocean current modeling based on measurements from underwater gliders to improve navigation performance. The method in [2] first approximates slowly-varying non-tidal flow from glider-derived flow estimates and then adds rapidly-varying tidal flow from an tidal ocean model to the non-tidal flow. However, since the low-frequency flow of the model is empirically estimated from glider measurements only, its accuracy is limited to a local area around each glider. The method in [3] approximates ocean currents using spatial and temporal basis functions. The ocean model constructed there requires historic data, such as HF-radar measurements or general circulation model output, to initialize. Once initialized, the model is updated based on the measurements from a group of gliders in real-time and provides ocean current data at higher resolution than existing approaches.

In our previous work [16], we developed the Glider CT algorithm that reconstructs the spatial distribution of a depth-averaged flow field with no *a priori* knowledge of the field. Glider CT is named after computerized tomography (CT), which reconstructs an image of the internal structure of an object from signals (e.g., X-rays) that are projected onto the object. A typical setup of CT has the transmitters and receivers of signals around an object. The transmitters emit signals onto an object, and while penetrating the object, the signals attenuate. Then, the remaining strength of the signals is measured at the receivers. Based on the signal paths and the measured signal strengths attenuated along the paths, an image of the object is reconstructed. In a similar way to CT, Glider CT reconstructs a flow field from the trajectories and dead-reckoning errors of gliders. We draw analogies between signal paths of CT and glider trajectories of Glider CT and between measured signal strengths attenuated along the signal paths and dead-reckoning error accumulated along the glider trajectories.

The structure of the Glider CT algorithm is very similar to a general CT reconstruction algorithm for bent ultrasound-rays [14]. In this paper, we extend our work in [16] by analyzing the Glider CT algorithm and providing a convergence proof, which is applicable to the method in [14] as well. Our algorithm solves a specific type of nonlinear systems of equations by extending the Kaczmarz method for linear equations. The Kaczmarz method is one of the row-action methods [1]. Convergence results have previously been obtained [6, 9] regarding various Kaczmarz-type meth-

ods for nonlinear equations. The glider CT algorithm can be viewed as one special case of these methods, and our proof of convergence adds to the collection.

In addition to the convergence analysis, this paper provides experimental results demonstrating the Glider CT algorithm. We imitate the horizontal motion of underwater gliders under a flow field using Khepera III robots under a simulated flow field. We place a light source in a target domain and simulate a flow field based on light intensity. By applying the dead-reckoning technique of gliders, we estimate the dead-reckoning trajectories of Khepera III robots. We control the motion of Khepera III robots as if their trajectories are affected by the simulated flow field. Since the actual underwater trajectories of gliders are unknown because of unavailable GPS signals underwater, we treat the trajectories of Khepera III robots as unknown. Only the starting and ending positions of the robots are used to measure the dead-reckoning error. Then, we apply the Glider CT algorithm to reconstruct the simulated flow field. The experimental results show a promising performance of the algorithm in practical settings.

The rest of the paper is organized as follows. Section 2 provides background information about underwater glider navigation, and Sect. 3 reviews our preliminary work on the Glider CT algorithm. In Sects. 4 and 5, we analyze the details of Glider CT and prove the convergence of the algorithm, respectively. Section 6 validates the algorithm through experiments, and Sect. 7 concludes the paper.

2 Background: Underwater Glider Navigation

An underwater glider regularly comes to the surface of water for GPS updates and data transfer. Between two surfacing events, glider navigation consists of two phases: surface and subsurface. We denote the actual and dead-reckoning positions of a glider at time t by $\mathbf{r}(t)$ and $\tilde{\mathbf{r}}(t)$, respectively. We also denote the time associated with the k -th surfacing and diving events by t_k^s and t_k^d , respectively. Figure 1 illustrates glider navigation from the $(k-1)$ -th diving event to the k -th diving event.

In this paper, we deal with the glider and the flow in the horizontal plane. Suppose we navigate a glider towards a waypoint during one subsurface phase. Before it dives, the glider computes its heading towards the waypoint. Let us denote the heading of the glider during the k -th subsurface phase by θ_k . Then, the glider dives at $\mathbf{r}(t_{k-1}^d)$ and navigates underwater until it reaches the waypoint by dead-reckoning, which estimates the position $\tilde{\mathbf{r}}(t)$ of the glider using estimates of glider speed, compass heading, and flow velocity. Because of the influence of flow, when the glider comes back to the surface of water at the k -th surfacing event, the glider experiences the dead-reckoning error, which is the difference between the dead-reckoning surfacing position $\tilde{\mathbf{r}}(t_k^s)$ and the GPS surfacing position $\mathbf{r}(t_k^s)$.

Upon the k -th surfacing event, a glider computes an estimate of average flow velocity along the glider trajectory based on the dead-reckoning error accumulated over the $(k-1)$ -th subsurface phase. This glider-derived flow estimate can be either incorporated into navigation to reduce the dead-reckoning error or deactivated so

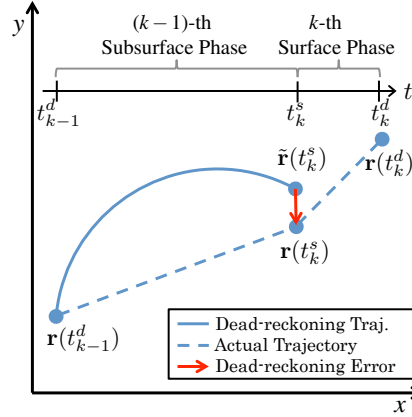


Fig. 1 Glider navigation during the $(k-1)$ -th subsurface phase followed by the k -th surface phase. The figure shows the actual (blue dashed line) and dead-reckoning (blue solid line) trajectories of a glider. The dead-reckoning error is shown as a red arrow.

that no flow estimate is used in navigation. Let us introduce a switching signal I_f to indicate whether the estimated flow is used for navigation or not. The signal $I_f = 1$ indicates that the estimated flow is used for navigation and 0, otherwise. Then, the glider-derived flow estimate at the k -th surfacing event is given by

$$\bar{\mathbf{f}}_k = \bar{\mathbf{f}}_{k-1} I_f + \frac{\mathbf{r}(t_k^s) - \tilde{\mathbf{r}}(t_k^s)}{t_k^s - t_{k-1}^d}, \quad (1)$$

which combines the previous flow estimate used to navigate over the $(k-1)$ -th subsurface phase with the new flow estimate based on the dead-reckoning error accumulated during the $(k-1)$ -th subsurface phase.

To describe the motion of a glider in the plane, we use a particle model with a constant through-water speed s_h . The position of a glider along the dead-reckoning trajectory can be predicted by integrating the following equation over time:

$$\dot{\hat{\mathbf{r}}}(t) = s_h \begin{bmatrix} \cos \theta_k \\ \sin \theta_k \end{bmatrix} + \bar{\mathbf{f}}_k I_f. \quad (2)$$

However, the real flow experienced by a glider may be different from the glider-estimated flow. Hence, the actual trajectory between the k -th and $(k+1)$ -th surfacing events can be described by integrating the following equation over time:

$$\dot{\mathbf{r}}(t) = s_h \begin{bmatrix} \cos \theta_k \\ \sin \theta_k \end{bmatrix} + \mathbf{f}(\mathbf{r}, t) = \dot{\hat{\mathbf{r}}}(t) + \mathbf{f}(\mathbf{r}, t) - \bar{\mathbf{f}}_k I_f, \quad (3)$$

which is usually unknown because of the unknown flow velocity $\mathbf{f}(\mathbf{r}, t)$.

3 Preliminary Work: Problem Formulation of Glider CT

The Glider CT problem is formulated from the fact that the dead-reckoning error accumulated along the glider trajectory is determined by a line integral of the difference between real flow experienced by the glider and glider-estimated flow incorporated in navigation. Suppose we deploy m gliders in the ocean and consider glider navigation over one subsurface phase only. Hereafter, we drop the subscript index k for the surfacing events and use the subscript index $i = \{1, \dots, m\}$ for the gliders for simplicity. After each glider finishes one subsurface phase, we will obtain a dead-reckoning error \mathbf{d}_i and a glider-derived flow estimate $\bar{\mathbf{f}}_i$ from each glider. The dead-reckoning error is accumulated over one subsurface phase, and from Eqs. (2) and (3), it is given by

$$\mathbf{d}_i = \int_{t_i^d}^{t_i^s} (\dot{\mathbf{r}}_i(\tau) - \dot{\hat{\mathbf{r}}}_i(\tau)) d\tau = \int_{t_i^d}^{t_i^s} \mathbf{f}(\mathbf{r}_i) d\tau - \bar{\mathbf{f}} \cdot I_f (t_i^s - t_i^d), \quad (4)$$

in which t_i^d and t_i^s are the diving and surfacing times of the i -th glider. We introduce arc-length parameter l of the trajectory, given by

$$dl = s_{\text{tr}} dt, \quad (5)$$

in which s_{tr} is the speed of the glider along its actual trajectory, which satisfies

$$s_{\text{tr}}(\mathbf{r}_i) = \|\dot{\mathbf{r}}_i\| = \left\| s_{\text{h}} \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix} + \mathbf{f}(\mathbf{r}_i) \right\|. \quad (6)$$

Substituting Eqs. (5) and (6) into Eq. (4), we derive

$$\mathbf{d}_i = \int_C \frac{1}{s_{\text{tr}}(\mathbf{r}_i)} \mathbf{f}(\mathbf{r}_i) dl - \bar{\mathbf{f}} \cdot I_f (t_i^s - t_i^d). \quad (7)$$

Since the second term on the right side of the equation is known, for simplicity we let $I_f = 0$ throughout the paper. However, our results apply to the general case without requiring $I_f = 0$.

Let us discretize area A into $R \times S$ grids with $A_{(r,s)}$ referring to the (r,s) -th grid. We denote the flow velocity in each cell by \mathbf{f}_j , $j = \{1, \dots, n = RS\}$. Indices j , r , and s satisfy $j = (r-1)S + s$. For the i -th glider passing through the j -th grid, we denote the length of the trajectory in the cell by $L_{(i,j)}$. Since we have a constant flow in each grid, the glider speed along the trajectory is given by

$$s_{\text{tr}}^{(i,j)}(\mathbf{f}_j) = \left\| s_{\text{h}} \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix} + \mathbf{f}_j \right\|.$$

We assume that the horizontal projection of the glider trajectory is straight and the heading θ_i is constant along the trajectory. Then, the discretized version of Eq. (7) with $I_f = 0$ is

$$\mathbf{d}_i = \sum_{j=1}^n \frac{L_{(i,j)}}{s_{\text{tr}}^{(i,j)}(\mathbf{f}_j)} \mathbf{f}_j, \quad i = \{1, \dots, m\}.$$

Now, consider the flow velocity along the x and y directions separately. For $i = \{1, \dots, m\}$, we can write

$$d_{x,i} = \sum_{j=1}^n \frac{L_{(i,j)}}{s_{\text{tr}}^{(i,j)}(\mathbf{f}_j)} f_{x,j}, \quad d_{y,i} = \sum_{j=1}^n \frac{L_{(i,j)}}{s_{\text{tr}}^{(i,j)}(\mathbf{f}_j)} f_{y,j}. \quad (8)$$

By introducing vectors $\mathbf{d}_x = [d_{x,1}, d_{x,2}, \dots, d_{x,m}]^T$, $\mathbf{d}_y = [d_{y,1}, d_{y,2}, \dots, d_{y,m}]^T$, $\mathbf{f}_x = [f_{x,1}, f_{x,2}, \dots, f_{x,n}]^T$, and $\mathbf{f}_y = [f_{y,1}, f_{y,2}, \dots, f_{y,n}]^T$, we can rewrite Eq. (8) as

$$\mathbf{d}_x = \mathbf{L}(\mathbf{f})\mathbf{f}_x, \quad \mathbf{d}_y = \mathbf{L}(\mathbf{f})\mathbf{f}_y, \quad (9)$$

where

$$\mathbf{L}(\mathbf{f}) = \begin{bmatrix} \frac{L_{(1,1)}}{s_{\text{tr}}^{(1,1)}(\mathbf{f}_1)} & \dots & \frac{L_{(1,n)}}{s_{\text{tr}}^{(1,n)}(\mathbf{f}_n)} \\ \vdots & \ddots & \vdots \\ \frac{L_{(m,1)}}{s_{\text{tr}}^{(m,1)}(\mathbf{f}_1)} & \dots & \frac{L_{(m,n)}}{s_{\text{tr}}^{(m,n)}(\mathbf{f}_n)} \end{bmatrix}, \quad (10)$$

which is nonlinear and typically underdetermined ($m < n$). By solving Eq. (9) for \mathbf{f}_x and \mathbf{f}_y , we can estimate flow.

4 Analysis of Glider CT

To solve the underdetermined and nonlinear system of equations in Eq. (9), we developed the Glider CT algorithm (Algorithm 1) that iteratively updates a solution to the equations with relaxation parameter λ . Let us omit x and y in the system for now. Given a nonlinear system

$$\mathbf{L}(\mathbf{f})\mathbf{f} = \mathbf{d}, \quad (11)$$

where

$$\mathbf{L}(\mathbf{f}) = \begin{bmatrix} \mathbf{L}_1(\mathbf{f}) \\ \vdots \\ \mathbf{L}_m(\mathbf{f}) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix},$$

the Glider CT algorithm finds a solution to the system equations in an iterative way. For the k -th iteration, let $\mathbf{f}^{(k,0)} = (f_1^{(k,0)}, f_2^{(k,0)}, \dots, f_n^{(k,0)})^T$ be the initial solution, and let us divide a new solution into m sequences given by

$$\mathbf{f}^{(k,1)} = (f_1^{(k,1)}, f_2^{(k,1)}, \dots, f_n^{(k,1)})^T, \\ \vdots$$

$$\mathbf{f}^{(k,m)} = \left(f_1^{(k,m)}, f_2^{(k,m)}, \dots, f_n^{(k,m)} \right)^T.$$

Since the system (11) is underdetermined, there may exist infinitely many solutions. Suppose given $\mathbf{f}^{(k,i-1)}$, we want to update solution $\mathbf{f}^{(k,i)}$ to the system in the form of

$$\mathbf{f}^{(k,i)} = \mathbf{f}^{(k,i-1)} + \alpha_i \left(\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T, \quad i = \{1, \dots, m\}, \quad (12)$$

which updates the solution by adding $\left(\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T$ weighted by α_i sequentially from $i = 1$ to m for the k -th iteration.

Algorithm 1: Glider CT

Data: Dead-reckoning errors $\mathbf{d}_i, i = \{1, \dots, m\}$

```

1 Set  $k = 0$ . Make an initial guess of the solution  $\mathbf{f}_x^{(k+1,0)}$  and  $\mathbf{f}_y^{(k+1,0)}$ .
2 repeat
3   Let  $k = k + 1$ .
4   for  $i = 1$  to  $m$  do
5     Update the solution by
        
$$\mathbf{f}_x^{(k,i)} = \mathbf{f}_x^{(k,i-1)} + \lambda^{(k,i-1)} \frac{d_{x,i} - \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}_x^{(k,i-1)}}{\|\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\|^2} \left( \mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T,$$

        
$$\mathbf{f}_y^{(k,i)} = \mathbf{f}_y^{(k,i-1)} + \lambda^{(k,i-1)} \frac{d_{y,i} - \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}_y^{(k,i-1)}}{\|\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\|^2} \left( \mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T.$$

6   end
7   Let  $\mathbf{f}_x^{(k+1,0)} = \mathbf{f}_x^{(k,K)}$  and  $\mathbf{f}_y^{(k+1,0)} = \mathbf{f}_y^{(k,K)}$ .
8 until a stopping condition is met

```

Let us define a residual term

$$r^{(k,i-1)}(\mathbf{f}) = \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f} - d_i. \quad (13)$$

To find α_i in Eq. (12), we let $r^{(k,i-1)}(\mathbf{f}) = 0$ at $\mathbf{f}^{(k,i)}$ and substitute Eq. (12) into Eq. (13), which yields

$$\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^{(k,i-1)} + \mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \left(\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T \alpha_i - d_i = 0.$$

Assuming $\mathbf{L}_i\mathbf{L}_i^T \neq 0$, the equation has the unique solution

$$\alpha_i = \frac{d_i - \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^{(k,i-1)}}{\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \left(\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T} = \frac{d_i - \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^{(k,i-1)}}{\|\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\|^2}. \quad (14)$$

Substituting Eq. (14) into Eq. (12), we have

$$\mathbf{f}^{(k,i)} = \mathbf{f}^{(k,i-1)} + \frac{d_i - \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^{(k,i-1)}}{\|\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\|^2} \left(\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T, \quad (15)$$

which is used to update the solution in Algorithm 1 without relaxation parameter $\lambda^{(k,i-1)}$. For simplicity, we assume $\lambda^{(k,i-1)} = 1$ here. Once we have $\mathbf{f}^{(k,m)}$, we obtain the initial sequence for the $(k+1)$ -th iteration by $\mathbf{f}^{(k+1,0)} = \mathbf{f}^{(k,m)}$.

5 The Convergence of the Glider CT Algorithm

Given a nonlinear system (11), we claim that the solution to the system equations derived from the Glider CT algorithm $\mathbf{f}^{(k,i)} = (f_1^{(k,i)}, f_2^{(k,i)}, \dots, f_n^{(k,i)})$, $k = \{1, 2, \dots\}$, $i = \{1, \dots, m\}$ converges to the true solution $\mathbf{f}^* = (f_1^*, f_2^*, \dots, f_n^*)$. Suppose there exists a ball $\mathcal{B}(\mathbf{f}^*, \delta)$ around \mathbf{f}^* with radius $\delta > 0$ where the following two assumptions hold for all $\mathbf{f} \in \mathcal{B}(\mathbf{f}^*, \delta)$:

Assumption 1. $\mathbf{L}_i(\mathbf{f})$ is Lipschitz continuous for all i with the largest Lipschitz constant γ , i.e., given Lipschitz constant $\gamma_i > 0$ for $\mathbf{L}_i(\mathbf{f})$, $i = \{1, \dots, m\}$, $\gamma = \max_i \gamma_i$.

Assumption 2. There exists $\varepsilon > 0$ that the following are satisfied:

- 1) $\lambda_{\max}(I - \mathbf{L}_i^+(\mathbf{f})\mathbf{L}_i(\mathbf{f})) < 1 - \varepsilon$ for all i , where $\lambda_{\max}(\cdot)$ is the largest eigenvalue,
- 2) $\frac{\gamma\|\mathbf{f}^*\|}{\|\mathbf{L}_i(\mathbf{f})\|} < \sqrt{\varepsilon}$ for all i .

Let us define $\mathbf{L}_i^+(\mathbf{f}) = \frac{\mathbf{L}_i(\mathbf{f})^T}{\|\mathbf{L}_i(\mathbf{f})\|^2}$, referred to as the pseudoinverse of $\mathbf{L}_i(\mathbf{f})$ in this paper.

Lemma 1. $\mathbf{L}_i^+(\mathbf{f})$ satisfies the following four conditions for the Moore-Penrose pseudoinverse [12]:

1. $\mathbf{L}_i(\mathbf{f})\mathbf{L}_i^+(\mathbf{f})\mathbf{L}_i(\mathbf{f}) = \mathbf{L}_i(\mathbf{f})$
2. $\mathbf{L}_i^+(\mathbf{f})\mathbf{L}_i(\mathbf{f})\mathbf{L}_i^+(\mathbf{f}) = \mathbf{L}_i^+(\mathbf{f})$
3. $(\mathbf{L}_i(\mathbf{f})\mathbf{L}_i^+(\mathbf{f}))^T = \mathbf{L}_i(\mathbf{f})\mathbf{L}_i^+(\mathbf{f})$
4. $(\mathbf{L}_i^+(\mathbf{f})\mathbf{L}_i(\mathbf{f}))^T = \mathbf{L}_i^+(\mathbf{f})\mathbf{L}_i(\mathbf{f})$

Proof. By simply substituting $\mathbf{L}_i^+(\mathbf{f})$ into the above four conditions, we can show that Lemma 1 holds. \square

Let us denote the Euclidean distance between \mathbf{p} and \mathbf{q} by $\text{dist}(\mathbf{p}, \mathbf{q})$. In the following theorem, we prove the convergence of the Glider CT algorithm.

Theorem 1. Suppose there exists a ball $\mathcal{B}(\mathbf{f}^*, \delta)$ around \mathbf{f}^* with radius $\delta > 0$ where Assumptions 1 and 2 hold. Starting from any initial point $\mathbf{f}^{(1,0)}$ within the ball, e.g., $\text{dist}(\mathbf{f}^*, \mathbf{f}^{(1,0)}) < \delta$, the sequence $\mathbf{f}^{(k,i)}$ generated by Algorithm 1 converges to \mathbf{f}^* .

Proof. Let us define an error term $\mathbf{e}^{(k,i)} = \mathbf{f}^{(k,i)} - \mathbf{f}^*$. By subtracting \mathbf{f}^* from the both sides of Eq. (15) and substituting $\mathbf{L}_i^+(\mathbf{f})$ and $r^{(k,i-1)}(\mathbf{f}^{(k,i-1)})$, we have

$$\mathbf{e}^{(k,i)} = \mathbf{e}^{(k,i-1)} - \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})r^{(k,i-1)}(\mathbf{f}^{(k,i-1)}), \quad i = \{1, \dots, m\},$$

The square of the Euclidean norm of the error is

$$\begin{aligned} \langle \mathbf{e}^{(k,i)}, \mathbf{e}^{(k,i)} \rangle &= \langle \mathbf{e}^{(k,i-1)}, \mathbf{e}^{(k,i-1)} \rangle - 2 \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})r^{(k,i-1)}(\mathbf{f}^{(k,i-1)}), \mathbf{e}^{(k,i-1)} \rangle \\ &\quad + \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})r^{(k,i-1)}(\mathbf{f}^{(k,i-1)}), \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})r^{(k,i-1)}(\mathbf{f}^{(k,i-1)}) \rangle. \end{aligned} \quad (16)$$

Since $d_i = \mathbf{L}_i(\mathbf{f}^*)\mathbf{f}^*$, we can express residual $r^{(k,i-1)}(\mathbf{f}^{(k,i-1)})$ as

$$\begin{aligned} r^{(k,i-1)}(\mathbf{f}^{(k,i-1)}) &= \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^{(k,i-1)} - d_i \\ &= \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^{(k,i-1)} - \mathbf{L}_i(\mathbf{f}^*)\mathbf{f}^* \\ &= \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^{(k,i-1)} - \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^* + \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{f}^* - \mathbf{L}_i(\mathbf{f}^*)\mathbf{f}^* \\ &= \mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)} + \mathbf{h}^{(k,i-1)}\mathbf{f}^*, \end{aligned} \quad (17)$$

where we define $\mathbf{h}^{(k,i-1)} = \mathbf{L}_i(\mathbf{f}^{(k,i-1)}) - \mathbf{L}_i(\mathbf{f}^*)$. By substituting $r^{(k,i-1)}(\mathbf{f}^{(k,i-1)})$ in Eq. (17) into Eq. (16), we have

$$\begin{aligned} \langle \mathbf{e}^{(k,i)}, \mathbf{e}^{(k,i)} \rangle &= \langle \mathbf{e}^{(k,i-1)}, \mathbf{e}^{(k,i-1)} \rangle - 2 \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*, \mathbf{e}^{(k,i-1)} \rangle \\ &\quad - 2 \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)}, \mathbf{e}^{(k,i-1)} \rangle \\ &\quad + \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)}, \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)} \rangle \\ &\quad + 2 \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*, \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)} \rangle \\ &\quad + \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*, \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^* \rangle. \end{aligned} \quad (18)$$

By the property of the inner product and Lemma 1, the fourth and fifth terms on the right side of Eq. (18) become

$$\begin{aligned} &\langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)}, \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)} \rangle \\ &= \left\langle \left(\mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)}, \mathbf{e}^{(k,i-1)} \right\rangle \\ &= \langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)}, \mathbf{e}^{(k,i-1)} \rangle \\ &\langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*, \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\mathbf{e}^{(k,i-1)} \rangle \\ &= \left\langle \left(\mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right)^T \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*, \mathbf{e}^{(k,i-1)} \right\rangle \end{aligned} \quad (19)$$

$$= \left\langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*, \mathbf{e}^{(k,i-1)} \right\rangle. \quad (20)$$

By substituting Eqs. (19) and (20) into Eq. (18), we have

$$\begin{aligned} \left\langle \mathbf{e}^{(k,i)}, \mathbf{e}^{(k,i)} \right\rangle &= \left\langle \left(I - \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right) \mathbf{e}^{(k,i-1)}, \mathbf{e}^{(k,i-1)} \right\rangle \\ &\quad + \left\langle \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*, \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^* \right\rangle. \end{aligned} \quad (21)$$

Let us define $\langle x, y \rangle_G = \langle Gx, y \rangle$ as an inner product of $x, y \in \mathbb{R}^n$ induced by matrix G and $\|x\|_G = \sqrt{\langle x, x \rangle_G}$ as a norm of $x \in \mathbb{R}^n$ induced by $\langle \cdot, \cdot \rangle_G$. Then, Eq. (21) can be rewritten as

$$\|\mathbf{e}^{(k,i)}\|^2 = \|\mathbf{e}^{(k,i-1)}\|_{(I - \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)}))}^2 + \|\mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{h}^{(k,i-1)}\mathbf{f}^*\|^2.$$

Since $\mathbf{L}_i(\mathbf{f})$ is Lipschitz continuous,

$$\|\mathbf{h}^{(k,i-1)}\| = \|\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) - \mathbf{L}_i(\mathbf{f}^*)\| \leq \gamma \|\mathbf{f}^{(k,i-1)} - \mathbf{f}^*\| = \gamma \|\mathbf{e}^{(k,i-1)}\|,$$

in which γ is the Lipschitz constant in Assumption 1. Then, we have

$$\begin{aligned} \|\mathbf{e}^{(k,i)}\|^2 &\leq \lambda_{\max} \left(I - \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right) \|\mathbf{e}^{(k,i-1)}\|^2 \\ &\quad + \gamma^2 \|\mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\|^2 \|\mathbf{e}^{(k,i-1)}\|^2 \|\mathbf{f}^*\|^2. \end{aligned}$$

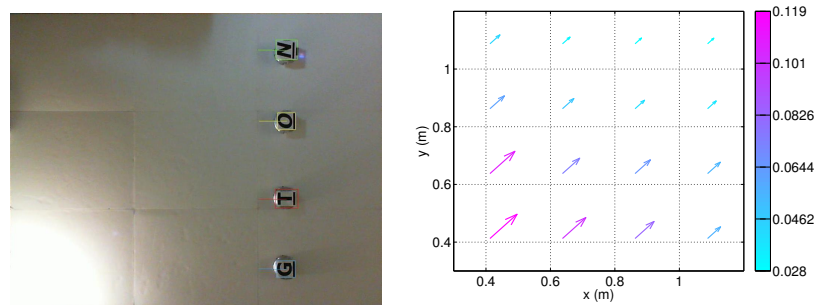
Since $\mathbf{L}_i^+(\mathbf{f}^{(k,i-1)}) = \frac{\mathbf{L}_i(\mathbf{f}^{(k,i-1)})^T}{\|\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\|^2}$, $\|\mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\|^2 = \frac{1}{\|\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\|^2}$, which gives

$$\begin{aligned} \|\mathbf{e}^{(k,i)}\|^2 &\leq \left(\lambda_{\max} \left(I - \mathbf{L}_i^+(\mathbf{f}^{(k,i-1)})\mathbf{L}_i(\mathbf{f}^{(k,i-1)}) \right) + \frac{\gamma^2 \|\mathbf{f}^*\|^2}{\|\mathbf{L}_i(\mathbf{f}^{(k,i-1)})\|^2} \right) \|\mathbf{e}^{(k,i-1)}\|^2 \\ &= \nu^{(k,i-1)} \|\mathbf{e}^{(k,i-1)}\|^2 \quad k = \{1, 2, \dots\}, \quad i = \{1, \dots, m\}, \end{aligned} \quad (22)$$

where $\mathbf{e}^{(k,i)} = \mathbf{f}^* - \mathbf{f}^{(k,i)}$. Since we set the initial solution for each iteration to be the last sequence of the solution from the previous iteration, e.g., $\mathbf{f}^{(k+1,0)} = \mathbf{f}^{(k,m)}$ for the $(k+1)$ -th iteration, we can express Eq. (22) as $\|\mathbf{e}^s\|^2 \leq \nu^{s-1} \|\mathbf{e}^{s-1}\|^2$, where $s = (k-1)m + i$ corresponds to (k, i) . Since $\mathbf{f}^{(1,0)} = \mathbf{f}^0 \in \mathcal{B}(\mathbf{f}^*, \delta)$ where Assumptions 1 and 2 hold, $\nu^0 < 1$, and therefore $\mathbf{f}^1 \in \mathcal{B}(\mathbf{f}^*, \delta)$. This applies to the following iterations sequentially, which leads to $\nu^1, \nu^2, \dots < 1$ and $\mathbf{f}^2, \mathbf{f}^3, \dots \in \mathcal{B}(\mathbf{f}^*, \delta)$. Therefore, starting from $\mathbf{f}^{(1,0)} \in \mathcal{B}(\mathbf{f}^*, \delta)$, we have $\nu^{(k,i)} < 1$ and $\mathbf{f}^{(k,i)} \in \mathcal{B}(\mathbf{f}^*, \delta)$ for all k and i . Hence, $\mathbf{e}^{(k,i)} \rightarrow 0$, and $\mathbf{f}^{(k,i)} \rightarrow \mathbf{f}^*$. \square

6 Experimental Results

We validate the Glider CT algorithm through experiments using Khepera III robots. The experimental setup is shown in Fig. 2(a). The starting and ending positions of robots are identified by a camera installed on top of the experimental domain. Even though the actual trajectory of a robot is observable through the camera, we treat the nominal trajectory of a robot as unknown to us since the underwater trajectory of a glider is unknown in glider operations. To construct a flow field, we place a light source at the left bottom corner $(x,y) = (0,0)$ of a domain and simulate a 2-D flow field such that all the flow vectors are in the direction of $[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$ and their magnitudes are scaled by the light intensity throughout the domain shown in Fig. 2(b). The intensity of ambient light around Khepera III robots is measured by 9 IR sensors located on the side of each robot. The measurements of the light intensity range from 0 to $I_{\max} = 4096$, where a lower value indicates higher light intensity.



(a) Four Khepera III robots in a light field. (b) The simulated true flow field of the domain.

Fig. 2 Experimental setup with Khepera III robots in a light field. Four Khepera III robots are differentiated using letters ‘G’, ‘T’, ‘O’, and ‘N’. National Instruments LabVIEW identifies the positions (the colored rectangle around each robot) and headings (the line attached to each robot) of the robots. A light source is located at the left bottom corner in (a) to simulate the light field in (b).

The horizontal motion of underwater gliders is imitated using Khepera III robots under a simulated flow field. Given an initial heading θ of a robot, its dead-reckoning trajectory is computed by integrating the dead-reckoning motion of gliders in Eq. (2). As discussed in Sect. 3, we set $I_f = 0$. The nominal motion of a Khepera III robot is implemented following Algorithm 2. At step k , each robot first measures the intensity of ambient light from the nine IR sensors at the current positions of the robots and computes the mean of the sensor measurements. To make a lower mean value of the sensor measurements represent a lower light intensity, we subtract the mean of the measurements from I_{\max} and compute a ratio of the mean light intensity around each robot. Then, we obtain the magnitude of the simulated flow at the current positions of the robots by scaling the ratio with scaling factor c . The simulated flow field is constructed by multiplying the magnitude by direction

vector $[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$. We compute the motion $\dot{\mathbf{r}} = [\dot{r}_x, \dot{r}_y]^T$ of a Khepera III robot under the simulated flow using Eq. (3). To implement motion $\dot{\mathbf{r}}$ in a Khepera III robot, we decompose $\dot{\mathbf{r}}$ into its magnitude and angle, i.e., $\dot{\mathbf{r}} = \|\dot{\mathbf{r}}\| \angle \dot{\mathbf{r}}$. We define the speed of a robot and the change of its heading that are affected by the simulated flow field as $s_h^k = \|\dot{\mathbf{r}}\|$ and $\Delta\theta_k = \angle \dot{\mathbf{r}} = \arctan(\dot{r}_y/\dot{r}_x)$, respectively. Then, with time step size Δt , we rotate each robot by $\Delta\theta_k \Delta t$ and move them forward by $s_h^k \Delta t$. We repeat this process until the length of the dead-reckoning trajectory of each robot reaches a predetermined travel distance D (i.e., $ks_h \Delta t < D$). For our experiment, we used $c = \frac{3\sqrt{2}}{10}$, $\Delta t = 0.1$, $s_h = 0.3 m/s$, and $D = 1.4 m$.

Algorithm 2: Nominal motion of a Khepera III robot

Data: Initial heading θ_0 of the robot

- 1 Set $k = 0$.
- 2 **repeat**
- 3 Let $k = k + 1$.
- 4 **for** $i = 1$ **to** 9 **do**
- 5 $I_i \leftarrow$ the intensity of ambient light from the i -th IR sensor
- 6 **end**
- 7 $I = \frac{\sum_i I_i}{9}$
- 8 $\mathbf{f} = \frac{c(I_{\max} - I)}{I_{\max}} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$
- 9 $\dot{\mathbf{r}} = s_h \begin{bmatrix} \cos(\theta_0) \\ \sin(\theta_0) \end{bmatrix} + \mathbf{f}$
- 10 $\Delta\theta_k = \arctan\left(\frac{\dot{r}_y}{\dot{r}_x}\right)$
- 11 $s_h^k = \|\dot{\mathbf{r}}\|$
- 12 Rotate the robot by $\Delta\theta_k \Delta t$
- 13 Move the robot forward by $s_h^k \Delta t$
- 14 **until** $k < \frac{D}{s_h \Delta t}$

We ran multiple sets of experiment using four Khepera III robots and chose ten navigation data sets of the robots – five from the right side of the domain to the left and five from the top to the bottom – shown in Fig. 3(a). Given the collected navigation data sets, we reconstructed the simulated flow field from the trajectories and dead-reckoning errors of the robots by running the Glider CT algorithm (Algorithm 1). Because of the unknown trajectories of the robots, we assume the actual robot trajectories are straight lines between their starting and final positions. For the algorithm, the true field is unknown. We chose $\lambda^{(k,i)} = 0.01$ for all k, i where $k = \{1, 2, \dots\}$, $i = \{1, \dots, m\}$, and for the k -th iteration, the iteration ended when both $\|\mathbf{f}_x^{(k,m)} - \mathbf{f}_x^{(k,0)}\|$ and $\|\mathbf{f}_y^{(k,m)} - \mathbf{f}_y^{(k,0)}\|$ are less than 10^{-3} . Figure 3(b) shows the reconstructed flow $\mathbf{f}^{\text{reconst}}$. Compared to the true field \mathbf{f}^{true} in Fig. 2(b), the reconstructed field suffers from noise. The magnitude of flow in the true field ranges from 0.028 to 0.119 m/s , and that in the reconstructed field ranges from 0.0212

to 0.0725 m/s . We compute the error between the true and reconstructed fields by $\mathbf{e} = \mathbf{f}^{\text{true}} - \mathbf{f}^{\text{reconst}}$. The root-mean-square errors in the x and y components are $\mathbf{e}_x^{\text{rms}} = 0.0182 \text{ m/s}$ and $\mathbf{e}_y^{\text{rms}} = 0.0169 \text{ m/s}$, respectively. We analyze that the error is partially due to the limitation of motor control for the differential wheels of Khepera III robots. The motor is controlled by pulse signals, and one pulse signal sent to the motors of a Khepera III robot rotates the robot by 0.06° . That is, the rotation angle is a multiple of 0.06° . Accumulated errors along the trajectories by the limitation of motor control may significantly affect the reconstruction of the field.

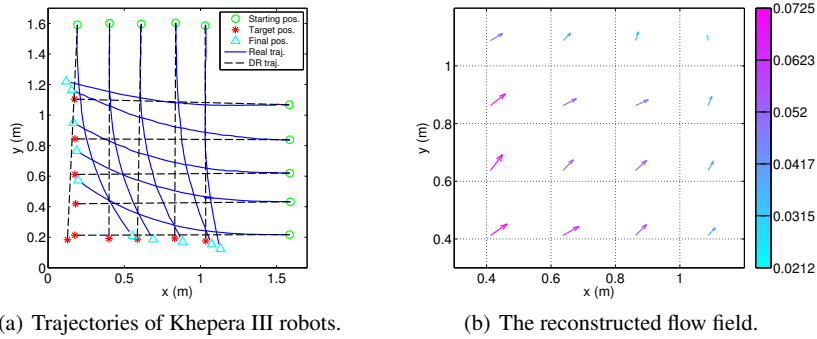


Fig. 3 Experimental results. (a) The green circles, red stars, and cyan triangles represent the starting positions, target positions, and ending positions of the robots, respectively. Solid lines connecting starting positions and ending positions are real trajectories, and dashed lines connecting starting positions and target positions are dead-reckoning trajectories. (b) The 4×4 flow field is reconstructed from navigation data of Khepera III robots shown in (a).

7 Conclusion

Glider CT reconstructs a depth-averaged flow field from the dead-reckoning errors of gliders. The Glider CT algorithm is a row-action iterative numerical method that converges to the solution of a set of nonlinear system equations sequentially. This paper proved the convergence of the Glider CT algorithm and demonstrated the effectiveness of the algorithm through experiments using Khepera III robots that imitate the horizontal motion of underwater gliders under a simulated flow field. The experimental results suggest that the Glider CT algorithm may be applied to real gliders in future ocean sensing deployments.

Acknowledgements The research work is supported by ONR grants N00014-09-1-1074 and N00014-10-10712 (YIP), and NSF grants ECCS-0841195 (CAREER), CNS-0931576, OCE-1032285, and IIS-1319874.

References

1. Censor, Y.: Row-action methods for huge and sparse systems and their applications. *SIAM Review* **23**(4), 444–466 (1981)
2. Chang, D., Edwards, C.R., Zhang, F.: Real-time guidance of underwater gliders assisted by predicted ocean models. *Journal of Atmospheric and Oceanic Technology* (Accepted) (2014)
3. Chang, D., Liang, X., Wu, W., Edwards, C.R., Zhang, F.: Real-time modeling of ocean currents for navigating underwater glider sensing networks. In: *Cooperative Robots and Sensor Networks, Studies in Computational Intelligence*, vol. 507, pp. 61–75. Springer Berlin Heidelberg (2014)
4. Fernández-Perdomo, E., Cabrera-Gómez, J., Hernández-Sosa, D., Isern-González, J., Domínguez-Brito, A.C., Redondo, A., Coca, J., Ramos, A.G., Álvarez Fanjul, E., García, M.: Path planning for gliders using regional ocean models: Application of Pinzón path planner with the ESEOAT model and the RU27 trans-Atlantic flight data. In: *Proceedings of OCEANS 2010*, pp. 1–10 (2010)
5. Leonard, N.E., Paley, D.A., Davis, R.E., Fratantoni, D.M., Lekien, F., Zhang, F.: Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *Journal of Field Robotics* **27**(6), 718–740 (2010)
6. Martínez, J.M., Sampaio, R.J.D.: Parallel and sequential Kaczmarz methods for solving underdetermined nonlinear equations. *Journal of computational and applied mathematics* **15**(3), 311–321 (1986)
7. Meldrum, D.T., Haddrell, T.: GPS in autonomous underwater vehicles. In: *Proceedings of the Sixth International Conference on Electronic Engineering in Oceanography*, pp. 11–17 (1994)
8. Merckelbach, L.M., Briggs, R.D., Smeed, D.A., Griffiths, G.: Current measurements from autonomous underwater gliders. In: *Proceedings of the IEEE/OES/CMTC Ninth Working Conference on Current Measurement Technology*, pp. 61–67 (2008)
9. Meyn, K.H.: Solution of underdetermined nonlinear equations by stationary iteration methods. *Numerische Mathematik* **42**(2), 161–172 (1983)
10. Nicholson, J.W., Healey, A.J.: The present state of autonomous underwater vehicle (auv) applications and technologies. *Marine Technology Society Journal* **42**(1), 44–51 (2008)
11. Paley, D.A., Zhang, F., Leonard, N.E.: Cooperative control for ocean sampling: The glider coordinated control system. *IEEE Transactions on Control Systems Technology* **16**(4), 735–744 (2008)
12. Penrose, R.: A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society* **51**(3), 406–413 (1955)
13. Schofield, O., Kohut, J., Aragon, D., Creed, L., Graver, J., Haldeman, C., Kerfoot, J., Roarty, H., Jones, C., Webb, D., Glenn, S.: Slocum gliders: Robust and ready. *Journal of Field Robotics* **24**(6), 473–485 (2007)
14. Schomberg, H.: An improved approach to reconstructive ultrasound tomography. *Journal of Physics. D, Applied Physics* **11**(15), 181–185 (1978)
15. Smith, R.N., Pereira, A., Chao, Y., Li, P.P., Caron, D.A., Jones, B.H., Sukhatme, G.S.: Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study. In: *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, pp. 4770–4777 (2010)
16. Wu, W., Chang, D., Zhang, F.: Glider CT: Reconstructing flow fields from predicted motion of underwater gliders. In: *Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems*, pp. 47:1–8 (2013)
17. Zhang, F., Fratantoni, D.M., Paley, D.A., Lund, J.M., Leonard, N.E.: Control of coordinated patterns for ocean sampling. *International Journal of Control* **80**(7), 1186–1199 (2007)