Journal of Intelligent & Robotic Systems (2024) 110:29
https://doi.org/10.1007/s10846-023-02044-5

REGULAR PAPER

Surface Tracking Controls of an Unmanned Underwater Vehicle with Fixed Sonar Ray Measurements in Tunnel-Like Environments

Jonghoek Kim[1]

Received: 26 February 2023 / Accepted: 16 December 2023 / Published online: 3 February 2024

Abstract

This paper introduces 3-D surface tracking control of an Unmanned Underwater Vehicle (UUV) in tunnel-like environments. Consider the case where a sonar transducer in the UUV does not rotate, and it only emits fixed sonar ray reporting a simple distance measurement. This reduces the power consumption of the UUV, while reducing the UUV's size and price. The UUV is controlled to proceed in tunnel-like environments, while maintaining a predefined distance from the tunnel boundaries. For maintaining a predefined distance from tunnel boundaries, the UUV uses fixed sonar rays surrounding it. As far as we know, our article is novel in developing 3-D surface tracking controls of tunnel-like environments utilizing an UUV with fixed sonar rays surrounding it. MATLAB simulations are used for demonstrating the performance of the proposed tracking controls.

Keywords Boundary tracking · Tunnel follow · Underwater exploration · Underwater navigation · Unmanned underwater vehicle · Fixed sonar ray · Single-beam echo-sounder

1 Introduction

Our article introduces surface tracking controls for an Unmanned Underwater Vehicle (UUV) with sampled range measurements. In underwater environments, Global Positioning Systems (GPS) cannot be used for vehicle localization. For enhancing the vehicle localization, various on-board sensors, such as Inertial Measurement Units (IMU), can be utilized [1–3]. However, using on-board sensors, it is inevitable that localization error integrates as time elapses.

Considering an UUV that cannot utilize GPS signals, this paper tackles the task of controlling the UUV so that it proceeds in tunnel-like environments, while maintaining a predefined distance from tunnel boundaries. This task is used in many scenarios, such as inspection of underwater structures, surveillance, or underwater 3-D mapping and exploration [2, 4, 5]. This task is elemental for many applications, such as inspection of bypass tunnels for dams, pipelines in sewer networks, power plants, factories, petrochemical, water supply and fluid transportation industries.

Our article considers the case where the 3-D tunnel boundary is not known in advance and the UUV cannot utilize GPS signal for localizing itself. The UUV in our paper is not equipped with full-scanning sonar sensors. In traditional scanning sonars, the transducer head inside the scanning sonar mechanically rotates via a stepper motor and moves the transducer head in an arc to build up an image on the the display software. However, the transducer rotation takes time for rotating 360 degrees. Thus, simultaneous range measurements are not feasible. Moreover, image processing requires computational burden and power consumption, compared to the processing of range measurements.

Our paper considers the case where a sonar transducer in the UUV does not rotate, and it only emits fixed sonar ray reporting a simple distance measurement. For instance, a single-beam echo-sounder in [6] can be used for emitting fixed sonar ray. This single-beam echo-sounder reduces the power consumption of the UUV, while reducing the UUV's size and price.

We consider a spherical underwater robot as our UUV [7–10]. Due to the high water pressure resistance of spherical objects, spherical UUVs can perform rotational motions with a 0 degree turn radius.

By attaching single-beam echo-sounders on the spherical UUV's body, every fixed ray can build an equi-angular configuration. In this way, the UUV can emit fixed sonar

✉ Jonghoek Kim
jonghoek@gmail.com

[1] System Engineering Department at, Sejong University, Seoul-city, South Korea

rays surrounding it, such that every fixed ray builds an equi-angular configuration. Multiple fixed sonar rays are utilized to sense 3-D tunnel boundaries surrounding the UUV. It is assumed that the maximum sensing range of a sonar ray is known in advance. Fixed sonar rays are used to make the UUV move along the 3-D tunnel, such that its information gain along the unknown tunnel is maximized.

Our paper develops surface tracking controls that utilize sonar ray measurements for maintaining a constant distance (offset) from unknown tunnel boundaries. For maintaining a predefined distance from tunnel boundaries, the UUV uses fixed sonar rays surrounding it. The UUV is controlled so that it proceeds in a 3-D tunnel which is not known in advance, while maintaining a predefined distance from tunnel boundaries.

In the literature, there are many papers on boundary tracking controls in 2-D environments [11–15]. Reference [16] addressed how to track a boundary curve considering a robot with multiple range sensors. Considering 2-D environments, [17] introduced boundary tracking controls so that a vehicle with rigidly mounted range sensors can follow obstacle boundaries while maintaining a predefined distance. The authors of [13] proposed reactive wall following controls by a 2-D robot taking the form of an actively steered sensor model that augments the robot's motion dynamics.

Our article handles surface tracking control for an UUV in 3-D underwater environments. There are many papers [18–20] on making an unmanned vehicle follow a designated 3-D curve. Considering 3-D space, [19] addressed a sliding mode prediction controller, which makes an underactuated UUV follow a desired 3-D curve under time-varying current disturbances. The authors of [18] addressed an analysis of the integral line-of-sight (ILOS) guidance method for path-following tasks of underactuated marine vehicles, operating on and below the sea surface. Considering 3-D space, [20] handled a fully actuated robot moving at unit speed. Reference [20] developed a feedback control law enabling a unit speed vehicle to follow a 3-D curve. However, [18–20] did not consider how to make the UUV proceed in 3-D tunnel environments, while maintaining a predefined distance from tunnel boundaries.

Frontier-based exploration has been widely used for exploration of unstructured environments [21–23]. In [22, 23], frontiers are defined as the border points that are calculated throughout the mapping and navigation stage between known and unknown spaces. Formally, a frontier is a set of unknown points that each has at least one open-space neighbor [22, 23]. Thus, making a robot visit frontier points can decrease unknown spaces gradually. In [21], traversal towards the unexplored map frontiers was used with a fast marching cost-to-go calculation to select a goal pose. However, finding frontiers requires that the robot has scanning sensors, which is distinct from fixed sonar rays used in our paper.

Moving toward a frontier can make the UUV move, such that its information gain along the unknown tunnel is maximized. However, moving toward a frontier does not make the UUV maintain a designated distance from tunnel boundaries. Thus, we address surface tracking controls, so that the UUV proceeds along the unknown 3-D tunnel, while maintaining a predefined distance from tunnel boundaries.

Recently, many papers [24–27] addressed exploration of underground mines using robots with Lidar sensors. Lidar has full-scanning abilities and has been widely used for building a 3-D map of the explored environment. However, Lidar cannot be used in underwater environments, since electromagnetic signal dissipates fast.

Reference [28] studied the problem of navigation in mine environments to enable collision free inspection with a micro aerial vehicle (MAV) equipped with cameras. In [28], a convolutional neural networks (CNN) was used to classify the images from the cameras to left, middle, and right. However, processing camera image based on CNN requires time-consuming training with various images and is computationally heavy. Moreover, [28] did not handle how to make the MAV navigate in a tunnel, while maintaining a predefined offset distance from tunnel boundaries.

Note that the UUV considered in our article does not have full-scanning sensors for scanning the entire space surrounding the UUV. The UUV has fixed ray sonars with limited sensing range, for detecting the tunnel boundary. Hence, the nearest point, which is a tunnel boundary point that is the nearest to the UUV, cannot be measured accurately. Our paper thus addresses a novel method for estimating the nearest point on the tunnel boundary, based on the UUV's recent path.

In summary, this paper addresses 3-D surface tracking controls, so that the UUV proceeds along the tunnel, while maintaining a predefined distance from tunnel boundaries. The UUV emits fixed sonar rays surrounding it, such that every fixed ray builds an equi-angular configuration.

As far as we know, our article is novel in developing 3-D surface tracking controls of tunnel-like environments utilizing an UUV with fixed sonar rays surrounding it. Here, every fixed sonar ray reports a simple distance measurement to nearby obstacle boundaries. MATLAB simulations are used for demonstrating the performance of the proposed surface tracking controls.

The remainder of our article is organized as follows: Section 2 discusses the control strategy addressed in this article. Section 3 addresses how to conjecture the parameters related to the UUV controls. Section 4 addresses MATLAB simulation results. Section 5 presents Conclusions.

2 Control Strategy

2.1 Definitions and Assumptions

This subsection addresses notations and assumptions utilized in our article. Let $c(*)$ define $\cos(*)$, and let $s(*)$ define $\sin(*)$. Considering an arbitrary vector $\mathbf{v}$, $unit(\mathbf{v})$ defines the unit vector given as $unit(\mathbf{v}) = \frac{\mathbf{v}}{\|\mathbf{v}\|}$.

Consider an UUV located at the entrance of the tunnel. One needs to make the UUV follow the tunnel surface, until reaching the tunnel exit. The UUV must proceed in tunnel-like environments, while maintaining a predefined distance r_0 from tunnel boundaries.

This paper considers discrete-time systems, where k defines the sample-stamp. Let $\mathbf{r}_1(k) \in \mathcal{R}^3$ define the UUV's location at sample-stamp k.

Suppose the UUV is at $\mathbf{r}_1(k)$. Let $\mathbf{f}(\mathbf{r}_1(k)) \in \mathcal{R}^3$ define the *frontier direction* measured by the UUV. In order to make the UUV at $\mathbf{r}_1(k)$ move forward along the tunnel, the UUV's heading vector is set as $\mathbf{f}(\mathbf{r}_1(k))$. One sets $\|\mathbf{f}(\mathbf{r}_1(k))\| = 1$, thus $\mathbf{f}(\mathbf{r}_1(k))$ is a unit vector. Section 3.2 presents how to determine $\mathbf{f}(\mathbf{r}_1(k))$ when the UUV is at $\mathbf{r}_1(k)$.

Moving in the direction of $\mathbf{f}(\mathbf{r}_1(k))$ can be utilized to make the UUV move forward along the unknown tunnel, while maximizing the information gain. However, moving in the direction of $\mathbf{f}(\mathbf{r}_1(k))$ does not make the UUV maintain r_0 distance from tunnel boundaries.

Since the UUV moves in underwater environments, its localization error integrates as time goes on. This paper assumes that localization error, which integrates from sample-stamp k to $k+k_d$, can be neglected. Here, $k_d > 0$ is a small constant. This short-time localization is feasible using various localization approaches [1, 3, 29–32].

In order to follow the tunnel while maintaining r_0 distance from tunnel boundaries, the UUV emits fixed sonar rays surrounding it. These fixed sonar rays meet the tunnel surface and detect points on the surface. Among all points that are detected by sonar rays from sample-stamp $k - k_d$ to k, a *nearest point* is a point that is the nearest to the UUV. Let $\mathbf{r}_2(k) \in \mathcal{R}^3$ denote the nearest point at sample-stamp k. How to conjecture $\mathbf{r}_2(k)$ utilizing the measurements of sonar rays is explained in Section 3.1.

Suppose that the UUV has N fixed sonar rays surrounding it. Suppose that each ray has a finite sensing range, say R_f. It is assumed that we can access the maximum range R_f of a sensing ray.

See Fig. 1 for an illustration of multiple fixed sonar rays generated from the spherical UUV. In the side view, one fixed ray emitted in the UUV's heading direction is plotted with a bold arrow. Other sonar rays are plotted with dotted arrows emanating from the UUV. Red arrows indicate fixed rays that are normal to the UUV's heading direction. In total, 17 fixed rays are used.

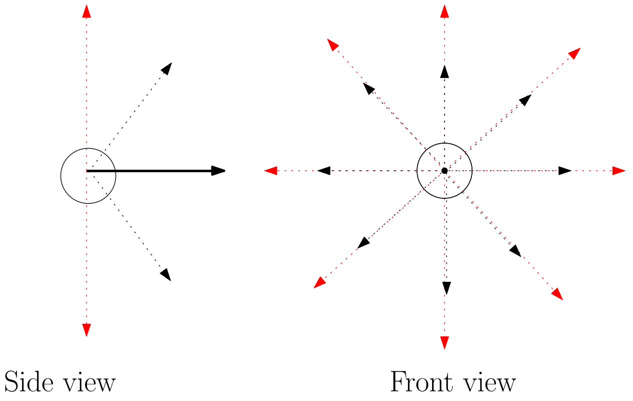

Fig. 1 An illustration of fixed sonar rays generated from the spherical UUV. In the side view, one fixed ray emitted in the UUV's heading direction is plotted with a bold arrow. Other sonar rays are plotted with dotted arrows emanating from the UUV. Red arrows indicate fixed rays that are normal to the UUV's heading direction. In total, 17 fixed rays are used

The relative position of the nearest point $\mathbf{r}_2(k)$ with respect to the UUV is

$$\mathbf{r}(k) = \mathbf{r}_2(k) - \mathbf{r}_1(k). \tag{1}$$

In addition, our article utilizes $r(k) = \|\mathbf{r}(k)\|$.

2.2 Motion Model

Let dt define the sampling interval in discrete-time systems. The UUV's process model is

$$\mathbf{r}_1(k+1) = \mathbf{r}_1(k) + dt \times v(k) \times \mathbf{u}(k). \tag{2}$$

Under (2), the UUV's speed is represented as $v(k)$. In (2), $\mathbf{u}(k) \in \mathcal{R}^3$ is the unit vector indicating the heading direction of the UUV at sample-stamp k. $\mathbf{u}(k)$ is the control input of our system. The simple dynamic model in (2) is commonly used in the literature on robotics [33–39].

This paper considers a spherical underwater robot as an UUV. The authors of [7–9] showed that by adopting vectored water-jets, a spherical underwater robot can maneuver freely in any direction. Since a spherical robot is highly maneuverable, the simple process model in (2) is feasible.

Let v_n present the nominal speed of the UUV. This implies that $v(k) = v_n$ is used for all sample-steps k.

2.3 Surface Tracking Controls

Let r_0 define the desired distance between the UUV and the tunnel surface. In addition, let $\epsilon \approx 0$ define a positive constant.

If $r(k) > r_0 + \epsilon$, then the UUV moves towards the tunnel surface while progressing along the tunnel. If $r(k) > r_0 + \epsilon$, then the UUV utilizes

$$\mathbf{u}(k) = unit(-\frac{\mathbf{r}(k)}{r(k)} \times \eta + (1 - \eta) \times \mathbf{f}(\mathbf{r}_1(k))). \tag{3}$$

In (3), η is a tuning constant indicating the weight of moving towards the boundary, compared to progression along the tunnel. η exists inside the interval [0,1]. In (3), $\mathbf{f}(\mathbf{r}_1(k))$ is a unit vector, defining the frontier direction at UUV's position $\mathbf{r}_1(k)$. Thus, $\|\mathbf{u}(k)\| = 1$, since we utilize $unit$ function. (3) implies that the UUV moves towards the nearest point, while progressing along the tunnel.

As η in (3) gets closer to 1, more weight is given to moving towards the boundary, than progressing along the tunnel.

In the case where $r_0 - \epsilon \leq r(k) \leq r_0 + \epsilon$, the UUV tracks the tunnel surface. Before addressing the controls for tracking the tunnel surface, we need to address two definitions. Let $proj(\mathbf{f}(\mathbf{r}_1(k)))$ define the projection of $\mathbf{f}(\mathbf{r}_1(k))$ onto $\mathbf{r}(k)$. We have

$$proj(\mathbf{f}(\mathbf{r}_1(k))) = \frac{\mathbf{r}(k)}{\|\mathbf{r}(k)\|^2}(\mathbf{f}(\mathbf{r}_1(k)) \cdot \mathbf{r}(k)). \tag{4}$$

Here, $\mathbf{f}(\mathbf{r}_1(k)) \cdot \mathbf{r}(k)$ indicates the inner product between $\mathbf{f}(\mathbf{r}_1(k))$ and $\mathbf{r}(k)$. Let $\mathbf{n}(k)$ be defined as

$$\mathbf{n}(k) = \mathbf{f}(\mathbf{r}_1(k)) - proj(\mathbf{f}(\mathbf{r}_1(k))). \tag{5}$$

See Fig. 2 for an illustration of $\mathbf{n}(k)$ and $proj(\mathbf{f}(\mathbf{r}_1(k)))$. In this figure, $\mathbf{r}(k)$ is depicted with a dotted arrow.

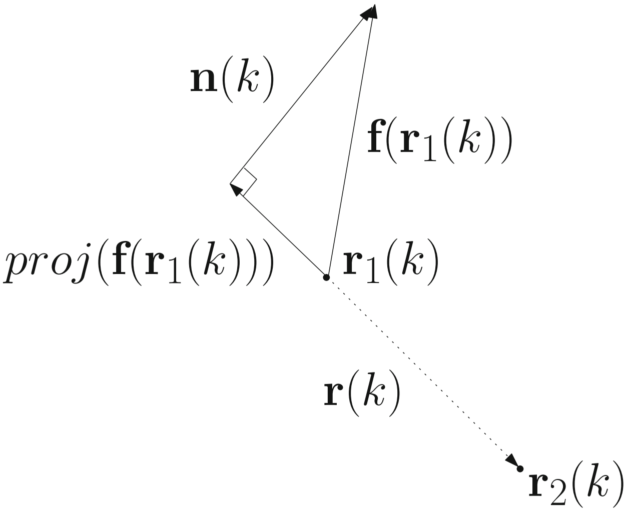
$\mathbf{n}(k)$
$\mathbf{f}(\mathbf{r}_1(k))$
$proj(\mathbf{f}(\mathbf{r}_1(k)))$
$\mathbf{r}_1(k)$
$\mathbf{r}(k)$
$\mathbf{r}_2(k)$

Fig. 2 An illustration of $\mathbf{n}(k)$ and $proj(\mathbf{f}(\mathbf{r}_1(k)))$. $\mathbf{r}(k)$ is depicted with a dotted arrow. $proj(\mathbf{f}(\mathbf{r}_1(k)))$ defines the projection of $\mathbf{f}(\mathbf{r}_1(k))$ onto $\mathbf{r}(k)$

In the case where $r_0 - \epsilon \leq r(k) \leq r_0 + \epsilon$, the UUV tracks the tunnel surface utilizing

$$\mathbf{u}(k) = \frac{\mathbf{n}(k)}{\|\mathbf{n}(k)\|}. \tag{6}$$

According to the definition of $proj(\mathbf{f}(\mathbf{r}_1(k)))$, $\mathbf{n}(k)$ is normal to $\mathbf{r}(k)$. Under (6), the UUV tracks the tunnel surface, while progressing along the tunnel.

If $r(k) < r_0 - \epsilon$, then the UUV moves away from the tunnel surface using

$$\mathbf{u}(k) = unit(\mathbf{r}(k)). \tag{7}$$

(7) imply that the UUV moves away from the nearest point.

If $r(k) \geq r_0 - \epsilon$, then we store the heading direction using

$$\mathbf{u}_s = \mathbf{u}(k). \tag{8}$$

Stored heading direction $\mathbf{u}_s \in \mathcal{R}^3$ in (8) is used for handling exceptions in Section 3.3.

3 Parameter Estimation

3.1 Conjecture the Relative Position of the Nearest Point

The UUV emanates N fixed sonar rays surrounding it. For instance, Fig. 1 plots $N = 17$ fixed sonar rays emanated from the UUV. Among all points that are detected by sonar rays from sample-stamp $k - k_d$ to k, a nearest point, $\mathbf{r}_2(k) \in \mathcal{R}^3$, is a point that is the nearest to the UUV. We address how to conjecture $\mathbf{r}(k) = \mathbf{r}_2(k) - \mathbf{r}_1(k) \in \mathcal{R}^3$ at every sample-stamp k utilizing multiple sonar rays.

Consider the body-fixed frame, say $L_{k'}$, centered at the UUV at sample-stamp k'. Since GPS cannot be used, the UUV's localization error must integrate as time goes on.

This paper assumes that localization error, which integrates from sample-stamp k' to $k' + k_d$, can be neglected. Here, $k_d > 0$ is a small constant. At every sample-stamp $k' + i$ ($i \in \{0, 1, ..., k_d\}$), we can localize the UUV in the body-fixed frame $L_{k'}$. For the UUV's localization from sample-stamp k' to $k' + k_d$, we can utilize IMU-based localization methods in [1, 3]. In the body-fixed frame $L_{k'}$, we calculate the 3-D coordinates of a *sensed point*, that is a point at which the UUV's sonar ray intersects the tunnel boundaries.

See Fig. 3 for an illustration. In Fig. 3, $\mathbf{r}_1(k') \in \mathcal{R}^3$ denotes the 3-D coordinates of the UUV at sample-stamp k'. $\mathbf{r}_1(k')$ is plotted as a circle. Moreover, $\mathbf{x}_L, \mathbf{y}_L, \mathbf{z}_L$ denote three unit vectors indicating the axis of $L_{k'}$. In Fig. 3, dotted arrows

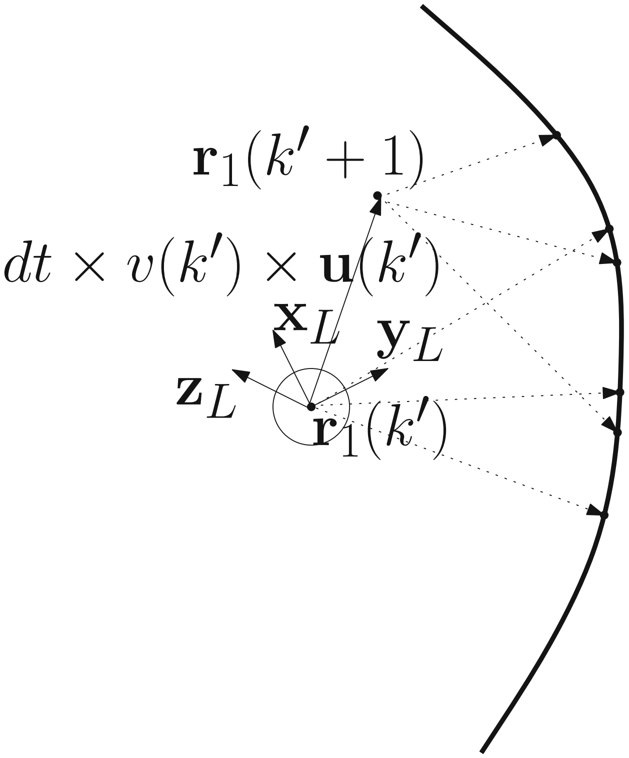

Fig. 3 The tunnel surface (bold curve) is plotted on the right side of the UUV. $\mathbf{r}_1(k') \in \mathcal{R}^3$ denotes the 3-D coordinates of the UUV at sample-stamp k'. $\mathbf{r}_1(k')$ is plotted as a circle. Moreover, $\mathbf{x}_L, \mathbf{y}_L, \mathbf{z}_L$ denote three unit vectors indicating the axis of $L_{k'}$. Dotted arrows indicate sonar rays generated from the UUV at every sample-stamp. The tunnel surface (bold curve) is plotted on the right side of the UUV

indicate sonar rays generated from the UUV at every sample-stamp. The tunnel surface (bold curve) is plotted on the right side of the UUV. See that six sensed points are generated on the tunnel surface. In our paper, sensed points represent coarse obstacle environments detected by the UUV.

Suppose that N_i sensed points are measured by the UUV's sonar rays at sample-stamp $k' + i$ ($i \in \{0, 1, ..., k_d\}$). In the body-fixed frame $L_{k'}$, $\mathbf{Co}^j(k' + i) \in \mathcal{R}^3$ defines the coordinates of the j-th ($j \in \{1, 2, ..., N_i\}$) sensed point that is measured by the UUV at sample-stamp $k' + i$. Let $\mathbf{So}(k)$ be defined as

$$\mathbf{So}(k) = [\mathbf{Co}^1(k), \mathbf{Co}^2(k), ..., \mathbf{Co}^{N_i}(k)]. \tag{9}$$

At sample-stamp $k' + k_d$, we compute the nearest point utilizing all sonar ray measurements from sample-stamp k' to $k' + k_d$. Since localization error integration from sample-stamp k' to $k' + k_d$ can be neglected, we localize the UUV at sample-stamp $k'+i$ ($i \in \{0, 1, ..., k_d\}$) in the local frame $L_{k'}$. Therefore, $[\mathbf{So}(k'), \mathbf{So}(k' + 1), ..., \mathbf{So}(k' + k_d)]$ is utilized as the set of sensed points at sample-stamp $k' + k_d$. The UUV at sample-stamp $k' + k_d$ calculates the distance to each of the sensed points in $[\mathbf{So}(k'), \mathbf{So}(k' + 1), ..., \mathbf{So}(k' + k_d)]$.

Among all sensed points, a *nearest point* is a point that is the nearest to the UUV. Therefore, the coordinates of the nearest point at sample-stamp $k' + k_d$ can be obtained in $L_{k'}$. Hence, $\mathbf{r}(k'+k_d) = \mathbf{r}_2(k'+k_d) - \mathbf{r}_1(k'+k_d)$ can be calculated accordingly. By replacing $k' + k_d$ by k, $\mathbf{r}(k) = \mathbf{r}_2(k) - \mathbf{r}_1(k)$ can be obtained at every sample-stamp k.

There may be a case where the current sample-stamp k satisfies $k < k_d$. In the case where $k < k_d$, one calculates the relative position of the nearest point utilizing all sonar ray measurements until sample-stamp k. This implies that $[\mathbf{So}(1), \mathbf{So}(2), ..., \mathbf{So}(k)]$ is utilized as the set of sensed points at sample-stamp k. Then, the UUV calculates the distance to each of the sensed points in $[\mathbf{So}(1), \mathbf{So}(2), ..., \mathbf{So}(k)]$. Among all sensed points, a nearest point is selected as a point that is the nearest to the UUV.

In our paper, sensed points represent coarse obstacle environments detected by the UUV. If the UUV is located at the center of a ball shaped obstacle boundary, then the UUV is equidistant from all sensed points. However, as the UUV moves inside a tunnel, this singular case will not happen, since the UUV detects an open space for moving forward along the tunnel.

3.2 Derive the Frontier Direction

We next address how to determine $\mathbf{f}(\mathbf{r}_1(k))$ when the UUV is at $\mathbf{r}_1(k)$. Recall that (9) presents the 3-D coordinates of N_i sensed points that are measured by the UUV's sonar rays at sample-stamp $k' + i$ ($i \in \{0, 1, ..., k_d\}$). This implies that $N - N_i$ sensing rays do not detect any obstacle at sample-stamp $k' + i$. In the case where a sensing ray does not detect an obstacle, we generate a point, called the *open point*, at the end of the ray whose length is R_f. Recall that each sensing ray has its maximum sensing range R_f. This open point is the point where the sensing ray can reach without intersecting obstacles.

Let $M_i = N - N_i$ for convenience. In the body-fixed frame $L_{k'}$, $\mathbf{Op}^j(k' + i) \in \mathcal{R}^3$ defines the 3-D coordinates of the j-th ($j \in \{1, 2, ..., M_i\}$) open point that is derived at sample-stamp $k' + i$. Let $\mathbf{Os}(k' + i)$ be defined as

$$\mathbf{Os}(k' + i) = [\mathbf{Op}^1(k' + i), \mathbf{Op}^2(k' + i), ..., \mathbf{Op}^{M_i}(k' + i)]. \tag{10}$$

At sample-stamp $k' + k_d$, the UUV derives the frontier direction utilizing $\mathbf{Os}(k'+i)$ ($i \in \{0, 1, ..., k_d\}$). Since localization error integration from sample-stamp k' to $k' + k_d$ can be neglected, we localize the UUV at sample-stamp $k' + i$ ($i \in \{0, 1, ..., k_d\}$) in $L_{k'}$. Therefore, $[\mathbf{Os}(k'), \mathbf{Os}(k' + 1), ..., \mathbf{Os}(k' + k_d)]$ is utilized as the set of open points at sample-stamp $k' + k_d$. The UUV at sample-stamp $k' + k_d$ calculates the distance to each of the open points in $[\mathbf{Os}(k'), \mathbf{Os}(k'+1), ..., \mathbf{Os}(k'+k_d)]$.

Among all open points in $[\mathbf{Os}(k'), \mathbf{Os}(k'+1), ..., \mathbf{Os}(k'+k_d)]$, a *farthest open point* is an open point that is the farthest to the UUV at sample-stamp $k' + k_d$. Therefore, the coordinates of the farthest open point at sample-stamp $k' + k_d$ can be obtained in $L_{k'}$. Let $\mathbf{r}_o(k'+k_d)$ denote the coordinates of the farthest open point at sample-stamp $k'+k_d$. By replacing $k'+k_d$ by k, $\mathbf{r}_f(k) = \mathbf{r}_o(k) - \mathbf{r}_1(k)$ can be obtained at every sample-stamp k. Then, $\mathbf{f}(\mathbf{r}_1(k))$ is set as

$$\mathbf{f}(\mathbf{r}_1(k)) = \frac{\mathbf{r}_f(k)}{\|\mathbf{r}_f(k)\|}. \tag{11}$$

Before sample-stamp k_d, the UUV maintains its initial heading $\mathbf{u}(0)$. Also, $\mathbf{f}(\mathbf{r}_1(k))$, where $k \leq k_d$, is set as the UUV's initial heading vector $\mathbf{u}(0)$. This implies that

$$\mathbf{f}(\mathbf{r}_1(k)) = \mathbf{u}(0) \tag{12}$$

when $k \leq k_d$.

3.3 Exception Handling

Consider the worst case where no open point is found at sample-stamp k. This implies that $[\mathbf{Os}(k-k_d), \mathbf{Os}(k-k_d+1), ..., \mathbf{Os}(k)]$ is empty. In this case, we use the stored heading direction $\mathbf{u}_s$ in (8) as $\mathbf{f}(\mathbf{r}_1(k))$. In other words, we use

$$\mathbf{f}(\mathbf{r}_1(k)) = \mathbf{u}_s. \tag{13}$$

If $\mathbf{f}(\mathbf{r}_1(k)) \cdot \mathbf{u}(k-1) < 0$, then this implies that the frontier direction is opposite to the previous heading direction of the UUV. This is not desirable, since the UUV needs to move towards the tunnel exit. Thus, we set the stored heading direction $\mathbf{u}_s$ in (8) as $\mathbf{f}(\mathbf{r}_1(k))$. In other words, we use (13).

3.4 Discussion on Computational Load

We discuss the computational load of the proposed control. The UUV's control uses $N \times k_d$ 3-D coordinates, which are derived using N fixed ray sensors for recent k_d sample-stamps. Moreover, stored heading direction $\mathbf{u}_s \in \mathcal{R}^3$ in (8) is used for handling exceptions in Section 3.3. In total, $(N \times k_d + 1)$ 3-D coordinates are used in our control.

4 MATLAB Simulations

We run MATLAB simulations for verifying the proposed 3-D surface tracking controls. The sampling interval is set as $dt = 0.1$ second. Furthermore, the desired distance r_0 is selected as 0.2 distance units. In Section 2.3, we use $\epsilon = \frac{r_0}{5}$. We utilize $\eta = 0.5$ in (3). The UUV's nominal speed v_n is 1 distance unit per second.

The UUV's initial location is (0,2,0). Initially, the UUV's heading vector is set as $\mathbf{u}(0) = (0, 0, 1)$. The simulation ends when the UUV reaches the tunnel exit.

At every sample-stamp, the UUV emits fixed sonar rays, and each ray has the maximum sensing range as $R_f = 30$ distance units. The UUV cannot sense a tunnel surface that is too far from the UUV. See Fig. 1 for the configuration of fixed sonar rays generated from the UUV. Each ray builds an equi-angular configuration with respect to the central axis of the UUV. In practice, there is measurement noise in sonar sensors. Whenever a sonar ray senses a tunnel surface, the associated range measurement is disturbed with Gaussian noise with mean 0 and variance $(0.01)^2$.

In MATLAB simulations, k_d is selected as 5. This implies that the localization error integration within 5 sample-stamps is neglected. In the case where the current sample-stamp k is less than k_d, we utilize all sensed points that have been measured until the sample-stamp k.

In practice, the UUV's process model (2) is perturbed by environmental effects, such as sea currents. Thus, instead of (2), we use

$$\mathbf{r}_1(k+1) = \mathbf{r}_1(k) + dt \times v \times \mathbf{u}(k) + \mathbf{e}. \tag{14}$$

Here, $\mathbf{e}$ presents the process noise with Gaussian distribution having mean 0 and variance $(0.01)^2$.

4.1 Scenario 1

We consider Scenario 1, as plotted in Fig. 4. We consider the case where $r_0 = 0.2$ distance units. The tunnel boundaries are plotted with red curves. The UUV's path is plotted with blue dots in this figure.

Recall that $r(k)$ is the relative distance between the UUV and the nearest point at sample-stamp k. Recall that r_0 is selected as 0.2 distance units. Considering Scenario 1, Fig. 5(a) describes the variation of $r(k)$ as time elapses. $r(k)$ almost converges to r_0 as time elapses. Since we use switching controls based on noisy range measurements and location measurements, $r(k)$ oscillate around r_0, but it does not converge to r_0. In addition, Fig. 5(b) describes the variation of the UUV's depth as time elapses. The UUV's depth changes from 0 to 60 as time goes on.

4.2 Scenario 2

We consider Scenario 2, as plotted in Fig. 6. We consider the case where $r_0 = 0.2$ distance units. The tunnel boundaries are plotted with red curves. The UUV's path is plotted with blue dots in this figure.

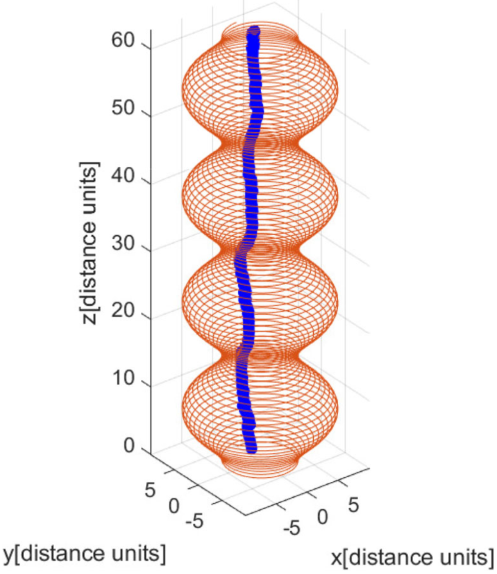

Fig. 4 Scenario 1 with $r_0 = 0.2$ distance units. The UUV's path is plotted with blue dots. The tunnel boundaries are plotted with red curves

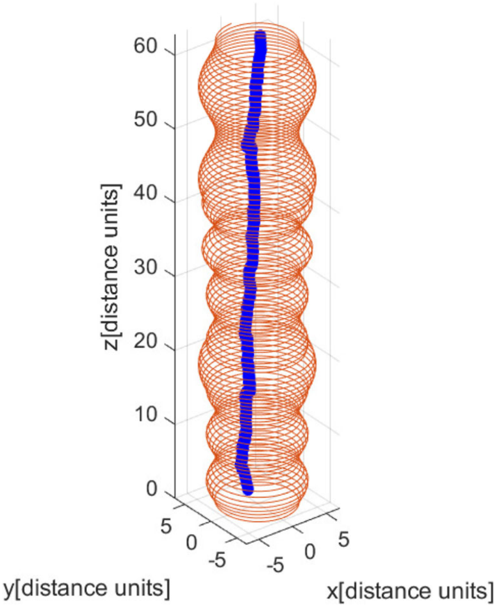

Fig. 6 Scenario 2 while setting $r_0 = 0.2$ distance units. The UUV's path is plotted with blue dots. The tunnel boundaries are plotted with red curves

Recall that $r(k)$ is the relative distance between the UUV and the nearest point at sample-stamp k. Considering Scenario 2 in Figs. 6 and 7 (a) describes the variation of $r(k)$ as time elapses. $r(k)$ almost converges to r_0 as time elapses. In addition, Fig. 7 (b) describes the variation of the UUV's depth as time elapses. The UUV's depth changes from 0 to 60 as time goes on.

Next, we simulate Scenario 2 while setting $r_0 = 0.5$ distance units. Fig. 8 (a) describes the variation of $r(k)$ as time elapses. $r(k)$ almost converges to r_0 as time elapses. Since we use switching controls based on noisy range measurements and location measurements, $r(k)$ oscillate around r_0, but it does not converge to r_0. In addition, Fig. 8 (b) describes the variation of the UUV's depth as time elapses.

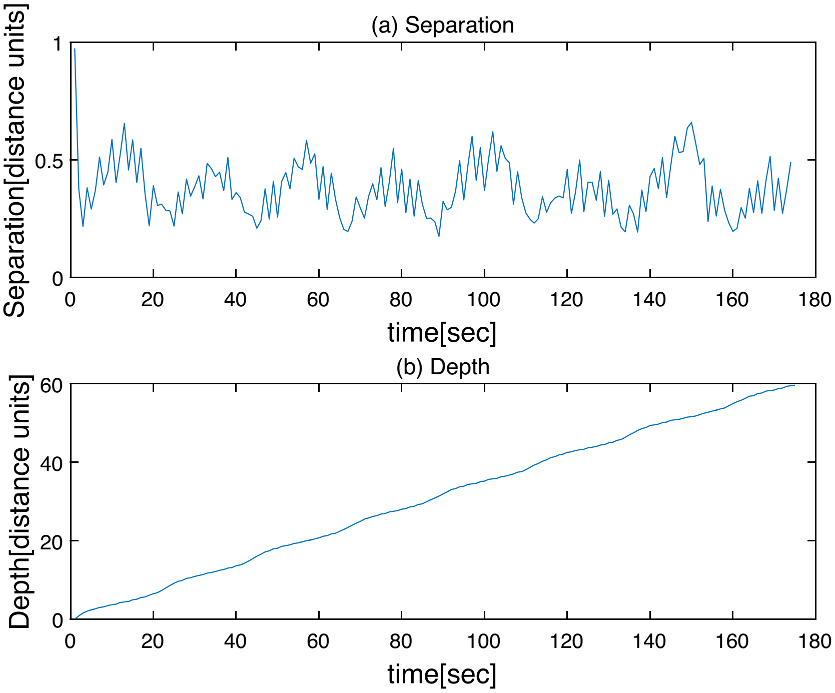

Fig. 5 Scenario 1 with the case where $r_0 = 0.2$ distance units. (a) the variation of $r(k)$ as k varies. (b) the variation of the UUV's depth as time goes on. $r(k)$ almost converges to r_0 as time elapses. In addition, the UUV's depth changes from 0 to 60 as time goes on

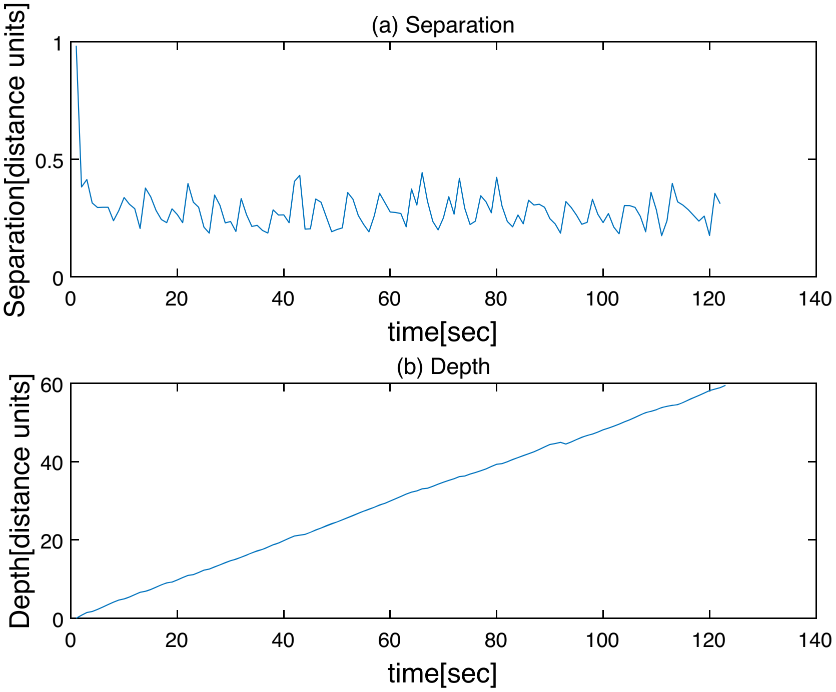

Fig. 7 Scenario 2 while setting $r_0 = 0.2$ distance units. (a) the variation of $r(k)$ as k varies. (b) the variation of the UUV's depth as time goes on. $r(k)$ almost converges to r_0 as time elapses. In addition, the UUV's depth changes from 0 to 60 as time goes on

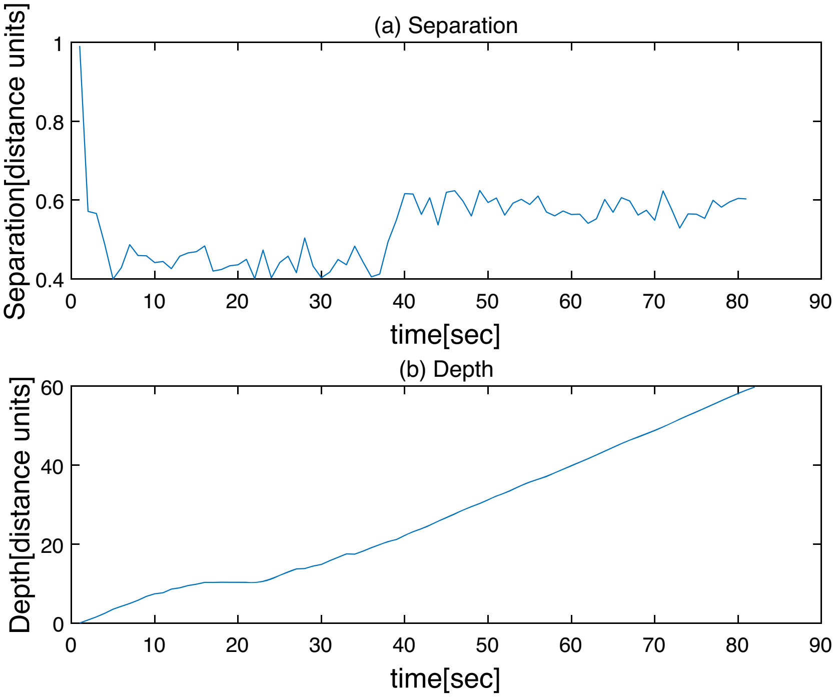

Fig. 8 Scenario 2 while setting $r_0 = 0.5$ distance units. (a) the variation of $r(k)$ as k varies. (b) the variation of the UUV's depth as time goes on. $r(k)$ almost converges to r_0 as time elapses

5 Conclusions

Our article considers an UUV with fixed sonar rays surrounding it. A sonar transducer in the UUV does not rotate, and it only emits fixed sonar ray reporting a simple distance measurement. In this paper, 3-D surface tracking controls are developed so that the UUV with fixed sonar rays moves along the tunnel, while maintaining a predefined distance to the tunnel surface. We provide MATLAB simulations for demonstrating the effectiveness of the proposed controls. In the future, we will perform experiments using a real robot, for demonstrating the effectiveness of the proposed surface tracking control.

The control law in our article can be utilized by various unmanned vehicles. For instance, aerial vehicles can utilize the proposed control to track the surface of an underground tunnel. In the future, camera sensors will be integrated with sonar ray sensors for robust surface tracking [28].

Author Contributions This paper is written by Jonghoek Kim.

Funding Not applicable.

Code or data availability Not applicable.

Declarations

Conflicts of interest/Competing interests Not applicable.

Ethics approval Not applicable.

Consent for publication Not applicable.

Consent to participate Not applicable.

References

1. Allotta, B., Costanzi, R., Fanelli, F., Monni, N., Paolucci, L., Ridolfi, A.: Sea currents estimation during auv navigation using unscented kalman filter. IFAC-PapersOnLine **50**(1), (2017)
2. Ribas, D., Ridao, P., Domingo Tardos, J., Neira, J.: Underwater slam in a marina environment. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1455–1460, (2007)
3. Brigadnov, I., Lutonin, A., Bogdanova, K.: Error state extended kalman filter localization for underground mining environments. Symmetry **15**(2), (2023)
4. Machado Jorge, V.A., de Cerqueira Gava, P.D., Belchior de França Silva, J.R., Mancilha, T.M., Vieira, W., Adabo, G.J., Nascimento, C.L.:Analytical approach to sampling estimation of underwater tunnels using mechanical profiling sonars. Sensors, **21**(5) (2021)
5. Martz, J., Al-Sabban, W., Smith, R.N.: Survey of unmanned subterranean exploration, navigation, and localisation. IET Cyber-Syst. Robot. **2**(1), 1–13 (2020)
6. cerulean sonar.:cerulean sonar. Website, (2023). https://ceruleansonar.com/products/sounder-s500
7. Gu, S., Guo, S., Zheng, L.: A highly stable and efficient spherical underwater robot with hybrid propulsion devices. Auton. Robot. **44**, 759–771 (2020)
8. Yue, C., Guo, S., Shi, L.: Hydrodynamic analysis of the spherical underwater robot sur-ii. Int. J. Adv. Robot. Syst. **10**(5), 247 (2013)
9. Li, G.S., Gou, J.: Tracking control in presence of obstacles and uncertainties for bioinspired spherical underwater robots. J. Bionic Eng. **20**, 323–337 (2023)
10. Kim, J.: Leader-based flocking of multiple swarm robots in underwater environments. Sensors **23**(11), (2023)
11. Clark, J., Fierro, R.: Cooperative hybrid control of robotic sensors for perimeter detection and tracking. In: Proc. of American Control Conference, Portland, OR, pp. 3500 – 3505 (2005)
12. Hoy, M.: A method of boundary following by a wheeled mobile robot based on sampled range information. J. Intell. Robot. Syst. **72**, 463–482 (2013)
13. De, A., Koditschek, D.E.: Toward dynamical sensor management for reactive wall-following. In Proc. of 2013 IEEE International Conference on Robotics and Automation (ICRA), Germany, pp. 2400–2406 (2013)
14. Micaelli, A., Samson, C.: Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Tech. Rep, INRIA (1993)
15. Toibero, J.M., Carelli, R., Kuchen, B.: Stable switching contour-following controller for wheeled mobile robots. In Proc. of the 2006 IEEE International Conference on Robotics and Automation, USA, pp. 3724 – 3729 (2006)
16. Hoy, M.: A method of boundary following by a wheeled mobile robot based on sampled range information. J. Intell. Robotics Syst. **72**(3–4), 463–82 (2013)
17. Kim, J.: Boundary tracking control for autonomous vehicles with rigidly mounted range sensors. J. Intell. Robot. Syst. **95**(3), 1041–1048 (2019)
18. Caharija, W., Pettersen, K.Y., Bibuli, M., Calado, P., Zereik, E., Braga, J., Gravdahl, T., SØrensen, A.J., Milovanoviç, M., Bruzzone, G.: Integral line-of-sight guidance and control of underactuated marine vehicles: Theory, simulations, and experiments. IEEE Trans. Control Syst. Technol. **24**(5), 1623–1642 (2016)
19. Zhang, L.-J., Jia, H.-M., Jiang, D.-P.: Sliding mode prediction control for 3d path following of an underactuated auv. IFAC Proc. Vol. **47**(3), 8799–8804 (2014). 19th IFAC World Congress
20. Wang, C.: Three-Dimensional Curve Tracking for Particles Using Gyroscopic Control. J. Dyn. Syst. Meas. Control **139**(12), 08 (2017)
21. Ahmad, S., Mills, A.B., Rush, E.R., Frew, E.W., Humbert, J.S.: 3d reactive control and frontier-based exploration for unstructured environments. In: IEEE/RSJ International Conference on Intelligent Robots and System (IROS) **2021**, 2289–2296 (2021)
22. Uslu, E., Çakmak, F., Balcilar, M., Akinci, A., Amasyali, M.F., Yavuz, S.: Implementation of frontier-based exploration algorithm for an autonomous robot. In: 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), pp. 1–7 (2015)

23. Quin, P., Nguyen, D.D.K., Vu, T.L., Alempijevic, A., Paul, G.: Approaches for efficiently detecting frontier cells in robotics exploration. Front. Robot. AI. **8**, 1 (2021)
24. Li, H., Savkin, A.V., Vucetic, B.: Collision free navigation of a flying robot for underground mine search and mapping. In: IEEE International Conference on Robotics and Automation (ROBIO) **2018**, 1102–1106 (2018)
25. Papachristos, C., Khattak, S., Mascarich, F., Alexis, K.: Autonomous navigation and mapping in underground mines using aerial robots. In: IEEE Aerospace Conference **2019**, 1–8 (2019)
26. Jung, S., Lee, H., Shim, D.H., akbar Agha-mohammadi, A.: Collision-free local planner for unknown subterranean navigation. ETRI J. **43**(4), 580–593, (2021)
27. Miller, I.D., Cladera, F., Cowley, A., Shivakumar, S.S., Lee, E.S., Jarin-Lipschitz, L., Bhat, A., Rodrigues, N., Zhou, A., Cohen, A., Kulkarni, A., Laney, J., Taylor, C.J., Kumar, V.: Mine tunnel exploration using multiple quadrupedal robots. IEEE Robot. Autom. Lett. **5**(2), 2840–2847 (2020)
28. Mansouri, S.S., Kanellakis, C., Georgoulas, G., Nikolakopoulos, G.: Towards mav navigation in underground mine using deep learning. In.: IEEE International Conference on Robotics and Biomimetics (ROBIO) **2018**, 880–885 (2018)
29. Arnold, S., Medagoda, L.: Robust model-aided inertial localization for autonomous underwater vehicles. In: IEEE International Conference on Robotics and Automation (ICRA) **2018**, 4889–4896 (2018)
30. Shaukat, N., Ali, A., Javed Iqbal, M., Moinuddin, M., Otero, P.: Multi-sensor fusion for underwater vehicle localization by augmentation of rbf neural network and error-state kalman filter. Sensors, **21**(4), (2021)
31. Cantelobre, T., Chahbazian, C., Croux, A., Bonnabel, S.: A real-time unscented kalman filter on manifolds for challenging auv navigation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) **2020**, 2309–2316 (2020)
32. Paull, L., Saeedi, S., Seto, M., Li, H.: Auv navigation and localization: A review. IEEE J. Ocean. Eng. **39**(1), 131–149 (2014)
33. Garcia de Marina, H., Cao, M., Jayawardhana, B.: Controlling rigid formations of mobile agents under inconsistent measurements. IEEE Trans. Robot. **31**(1), 31–39 (2015)
34. Krick, L., Broucke, M.E., Francis, B.A.: Stabilization of infinitesimally rigid formations of multi-robot networks. In 2008 47th IEEE Conference on Decision and Control, pp. 477–482 (2008)
35. Paley, D.A., Zhang, F., Leonard, N.E.: Cooperative control for ocean sampling: The glider coordinated control system. IEEE Trans. Control Syst. Technol. **16**(4), 735–744 (2008)
36. Ji, M., Egerstedt, M.: Distributed coordination control of multiagent systems while preserving connectedness. IEEE Trans. Robot. **23**(4), 693–703 (2007)
37. Kim, J.: Constructing 3d underwater sensor networks without sensing holes utilizing heterogeneous underwater robots. Appl. Sci. **11**(9), (2021)
38. Kim, J., Kim, S.: Motion control of multiple autonomous ships to approach a target without being detected. Int. J. Adv. Robot. Syst. **15**(2), 1729881418763184 (2018)
39. Luo, S., Kim, J., Parasuraman, R., Bae, J.H., Matson, E.T., Min, B.-C.: Multi-robot rendezvous based on bearing-aided hierarchical tracking of network topology. Ad Hoc Netw. **86**, 131–143 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jonghoek Kim is a Professor in the System Engineering Department at Sejong University, Republic of Korea. His research is on target tracking, control theory, robotics, multi-agent systems, and optimal estimation. He worked as a senior researcher at Agency for Defense Development in Republic of Korea from 2011 to 2018. In 2011, he earned a Ph.D. degree co-advised by Dr. Fumin Zhang and Dr. Magnus Egerstedt at Georgia Institute of Technology, USA. Jonghoek Kim received his M.S. in Electrical and Computer Engineering from Georgia Institute of Technology in 2008 and his B.S. in Electrical and Computer Engineering from Yonsei University, Republic of Korea in 2006.