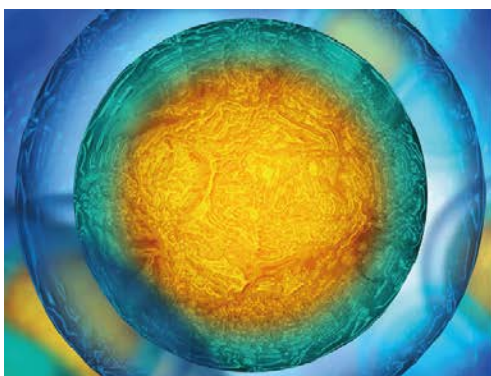


PAPER

A bio-inspired localization-free stochastic coverage algorithm with verified reachability

To cite this article: Ayesha Khan *et al* 2021 *Bioinspir. Biomim.* **16** 056009

View the [article online](#) for updates and enhancements.



Biophysical Society **IOP | ebooks™**

Your publishing choice in all areas of biophysics research.

Start exploring the collection—download the first chapter of every title for free.

Bioinspiration & Biomimetics



PAPER

A bio-inspired localization-free stochastic coverage algorithm with verified reachability

RECEIVED
2 September 2020

REVISED
8 May 2021

ACCEPTED FOR PUBLICATION
7 July 2021

PUBLISHED
29 July 2021

Ayesha Khan^{*} , Said Al-Abri, Vivek Mishra and Fumin Zhang^{*}

Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332, United States of America

^{*} Authors to whom any correspondence should be addressed.

E-mail: ayeshakhan@gatech.edu, saidalabri@gatech.edu and fumin@gatech.edu

Keywords: bio inspired algorithms, area coverage, mobile robotics, Markov chains, fish behavior, persistent turning Walker

Abstract

Dependence on constant availability to an external localization service is often unreliable and infeasible in mobile robots. In this paper, we take inspiration from a continuous fish motion model, the persistent turning Walker (PTW), to devise a strategy which is able to achieve 2D and 3D coverage in an unknown environment in the absence of a localization service, such as a global positioning system (GPS). This is achieved by converting the continuous-time dynamical system into a discrete-time Markov chain which is then shown to exhibit strongly connected properties that are verifiable through numerical methods. The aforementioned proposed framework can also be used to study the continuous-time dynamics of other biological systems and evaluate their properties. The performance of the PTW model is also compared with two existing random search strategies, simple random walks (SRW) and correlated random walks (CRW) by using analytical bounds, simulation results, and statistical tests. The simulation results show that the proposed PTW algorithm covers a given search-space at a faster rate compared to the CRW and SRW models. Hence, the PTW may be effectively used as a coverage strategy by mobile robots in underwater or underground environments where the availability of a GPS cannot be guaranteed at all times.

1. Introduction

Mobile sensing agents can be deployed in a given environment to find objects of interest. For instance, deploying a fleet of autonomous underwater vehicles (AUVs) to find a lost ship in the ocean [1–5]. A group of aerial robots can also be used to identify potential areas to execute search and rescue tasks after an earthquake [6, 7]. The environments which an agent is expected to cover can be more formally referred to as *search-spaces*. Ideally, an optimal coverage strategy should enable an agent to cover the maximum area in the least amount of time. In the literature, this is known as the *coverage* problem, and the goal is to find a trajectory for a mobile agent such that it maximizes the area visited in the search-space while avoiding existing obstacles [8–10].

In nature, animals incorporate an innate sense to search for their food or prey even in the presence of uncertainties in the environment. These uncertainties might be the presence of obstacles or an unknown environment altogether. The search strategies exhibited by animals can nonetheless be thought

of as coverage algorithms since animals tend to cover the entire environment in a constant effort to search for their potential food source or prey. Several recent works which propose coverage algorithms inspired by animal behavior are given in [11–14]. Statistical analysis for search strategies observed in animals is given in [15], where the authors compare two variants of the random walk model, correlated random walks (CRWs) and Levy walks (LWs). LWs are identical to random walks but with step sizes that follow a heavy-tailed probability distribution. The authors in [15] investigate the Levy-modulated CRW which combines the properties of correlated and LWs. Another bio-inspired search strategy, called the Levy-taxis, was also introduced in [16] which successfully located odor plumes significantly faster than CRW, LWs, and systematic zigzag patterns. In [17, 18], the authors combine LWs with gradient search to localize targets in unknown environments, while [19] compares various bio-inspired search strategies in 2D for agents with limited perception. In recent works [20], authors exploit the chaotic behavior of a bio-inspired motion controller, the

Braitenberg vehicle, to justify coverage in a search space through a combination of theoretical and simulation results.

Another category of coverage algorithms, not inspired by nature, are also devised extensively in the literature and they enable an agent or a group of agents to cover an entire search-space successfully [21–25]. These algorithms can be classified on the basis of whether the agent has prior knowledge about the global obstacle map of the search-space or not [9, 22]. Coverage algorithms in known environments can be designed to perform in an efficient and predictable manner since the optimal movement pattern of the agent can be devised beforehand, a common example being the lawn mower strategy [26, 27] and the spiraling strategy [28]. These algorithms usually rely on the availability of an external localization service (such as a global positioning system (GPS)) which enables the agent to keep track of the area covered and yet to be covered. However, the availability of a localization service cannot be relied upon in certain practical applications, such as underground or underwater environments. Therefore, it is important to evaluate coverage strategies that do not rely on GPS and are able to cover the entire search-space effectively. Bio-inspired search strategies offer the advantage that there is no dependence on the constant availability of an external localization service. A comprehensive survey reviewing the advantages of using bio-inspired behavior for AUVs is available [29].

We propose a coverage algorithm inspired by a fish motion model in [30, 31]. This model is termed as the persistent turning Walker (PTW) and is characterized by a constant speed (or a constant step size) at each instant of time. In [32], the authors show that the large-scale time dynamics of this model are of a diffusive type and provide an analytic expression for the diffusion coefficient. In this paper, we extend the PTW model to 3D and then justify that the PTW model can be used as an effective coverage strategy in a 2D and 3D environment. Extending on our preliminary work in [33], we show in this paper the coverage properties of the model by using reachability analysis based on Markov chains. The continuous-time dynamical system of the PTW model is first converted into its respective Markov chain by using the generalized cell mapping (GCM) method. This is followed by formulating the observations seen in real fish trajectory data as mathematical hypotheses. These hypotheses are then verified against the formulated Markov chain. The coverage of the PTW model is then proved by using reachability analysis and the rate of coverage is compared with existing random walk strategies. The entire proposed framework establishes a procedure which can be used to evaluate the continuous-time dynamics of other biological systems as well.

Our main contributions are as follows: (1) we propose a 3D version of the PTW model which can be used to perform coverage in a 3D environment. (2) We show that the PTW model can be used as a randomized search strategy that does not require a localization service to perform coverage in a 2D and 3D environment. In order to show this, we use GCM methods to model the movement of the fish in position space using a discrete-time Markov chain (DTMC) [34–36]. The DTMC is then verified against hypotheses using a computer program and then translated to a digraph of a position-space motif which is then shown to be strongly connected. This property is further extended to show that the entire search-space is connected. This justifies that the PTW model in 2D and 3D can be used as an effective coverage strategy in a GPS-denied environment. (3) We propose a framework that can validate the properties of a continuous-time biological system. The proposed framework converts a continuous-time system into a discrete-time system which is then converted into a DTMC. The DTMC is evaluated against specifications written in a computer program or other formalisms used in model checkers, such as linear temporal logic [37, 38]. This systemic approach enables us to verify nontrivial temporal or spatial properties of a continuous dynamical system. In addition, we also experimentally validate that the PTW outperforms two similar random search strategies that can also be used to perform coverage in GPS-denied environments. Although these experimental results are expected, they complement our theoretical results. Contrary to previous works, which mainly use simulations to evaluate the effectiveness of coverage algorithms, we use the values of transitional probabilities to show analytically that the upper bound on the expected area covered by the PTW model is always greater than or equal to the other models. We also give a quantitative analysis by comparing the mean area covered by each of the strategies using the Wilcoxon Mann Whitney test. In addition, simulation results are also provided which confirm that the PTW model performs better area coverage than the SRW and CRW schemes.

The rest of the paper is organized as follows: section 2 contains the problem formulation of the coverage task and introduces the three random walk strategies. Section 3 shows the extension of the two-dimensional PTW model to three dimensions. Section 4 introduces the framework which is used to find the backtracking probability associated with the PTW model. Section 5 justifies the coverage of the search-space using reachability analysis in 2D and 3D environments. Section 6 provides an analytical comparison of the coverage of the PTW, CRW, and SRW models. Finally, section 7 contains simulation results while section 8 concludes the paper with some future insights.

2. Problem formulation

In this section, we first present a background on three random search strategies: the simple random walk (SRW), the correlated random walk (CRW), and the PTW [30, 31]. We then formulate a coverage control problem in a bounded search-space.

2.1. Random walks

We review the random walks model in this section. At every step, the direction of motion is sampled from a probability distribution. Let γ be the constant step size and Φ be a random variable depicting the heading with respect to the positive x -axis in a 2D plane. A sample of Φ from a distribution $f(\phi)$ in the k th iteration is given by ϕ_k . The motion model for a random walk in 2D can be given by the following equations,

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \gamma \begin{bmatrix} \cos \phi_k \\ \sin \phi_k \end{bmatrix}, \quad \phi_k \sim f(\phi), \quad (1)$$

where $\mathbf{r} = [x, y]^T \in \mathbb{R}^{2 \times 1}$.

We first consider the SRW model and the CRW model. The difference between the SRW and the CRW strategies is the distribution $f(\phi)$ from which the angle of motion is sampled. In SRW, the distribution is uniform and can be given as:

$$\mathbb{P}(\phi_{k+1} = \phi_i) = \frac{1}{N_s}, \quad (2)$$

where ϕ_i is one of the N_s possible values of the heading. As can be seen in (2), the heading at the next time instant, $k + 1$, given by ϕ_{k+1} , is not correlated to the heading at the present time instant, ϕ_k . In other words, for the SRW model, ϕ_{k+1} is randomly selected from one of the N_s possible headings and then a step vh is taken in that direction. Here v denotes the constant speed while h is the discretization step size (the choice of these variables is described in detail in section 4.1).

In the CRW model [39], the heading at time step, ϕ_{k+1} is dependent on its present value, ϕ_k with a certain probability (say α). For CRW, $f_{\text{CRW}}(\phi_k)$ denotes a probability distribution given by the following

$$\mathbb{P}(\phi_{k+1}) = \begin{cases} \alpha; & \text{if } \phi_{k+1} = \phi_k \\ \frac{1 - \alpha}{N_s - 1}; & \text{if } \phi_{k+1} \neq \phi_k \end{cases}. \quad (3)$$

It is worth mentioning here that as the value of α gets closer to 1, we get a perfectly CRW (the agent always has the same heading direction) and as the value of α gets closer to 0, we get an uncorrelated random walk (subsequent heading directions of the agent not dependent on one another).

The aforementioned models can be simulated in a bounded search-space by incorporating collision avoidance behavior near the boundaries of the space. A simple approach can be to check if the next step, vh in the direction of ϕ_{k+1} lies outside the boundaries of

the search-space. If the step lies outside the boundaries, then the next heading ϕ_{k+1} can again be sampled from N_s possible headings. The process can be repeated until the next step, vh lies inside the boundaries of the search-space. This approach enables the model to remain inside a bounded space by exhibiting collision avoidance behavior near its boundaries.

2.2. Persistent turning Walker in 2D

The PTW model is a representation depicting the movement of a special species of fish, *Kuhlia Mugal* in two dimensions [30, 31]. The PTW model is characterized by a constant swimming speed v and a correlation between angular velocities, w at consecutive instances of time. Denoting the position of the fish by $\mathbf{r}$ and its heading by ϕ , the following dynamical system represents the PTW model

$$d\mathbf{r} = v \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} dt \quad (4)$$

$$d\phi = w(t)dt \quad (5)$$

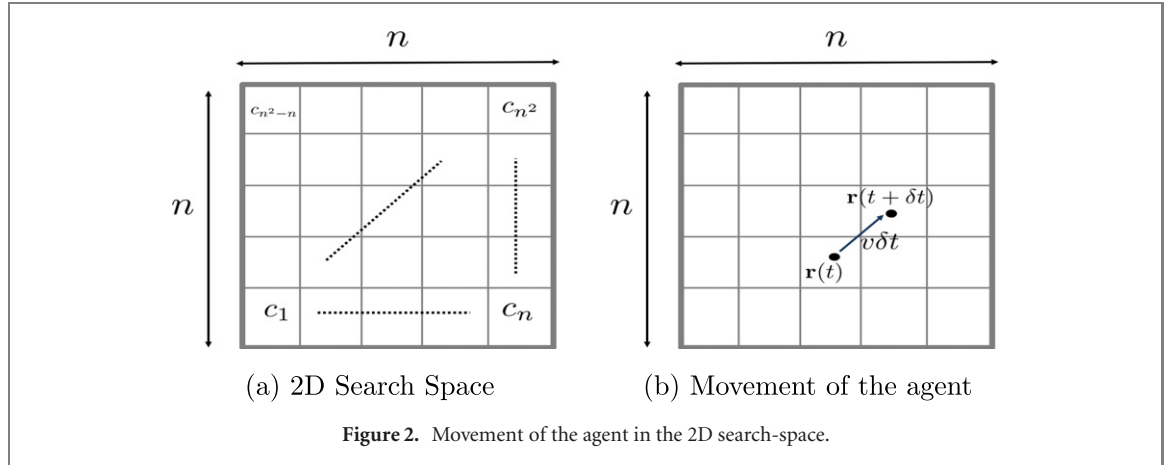
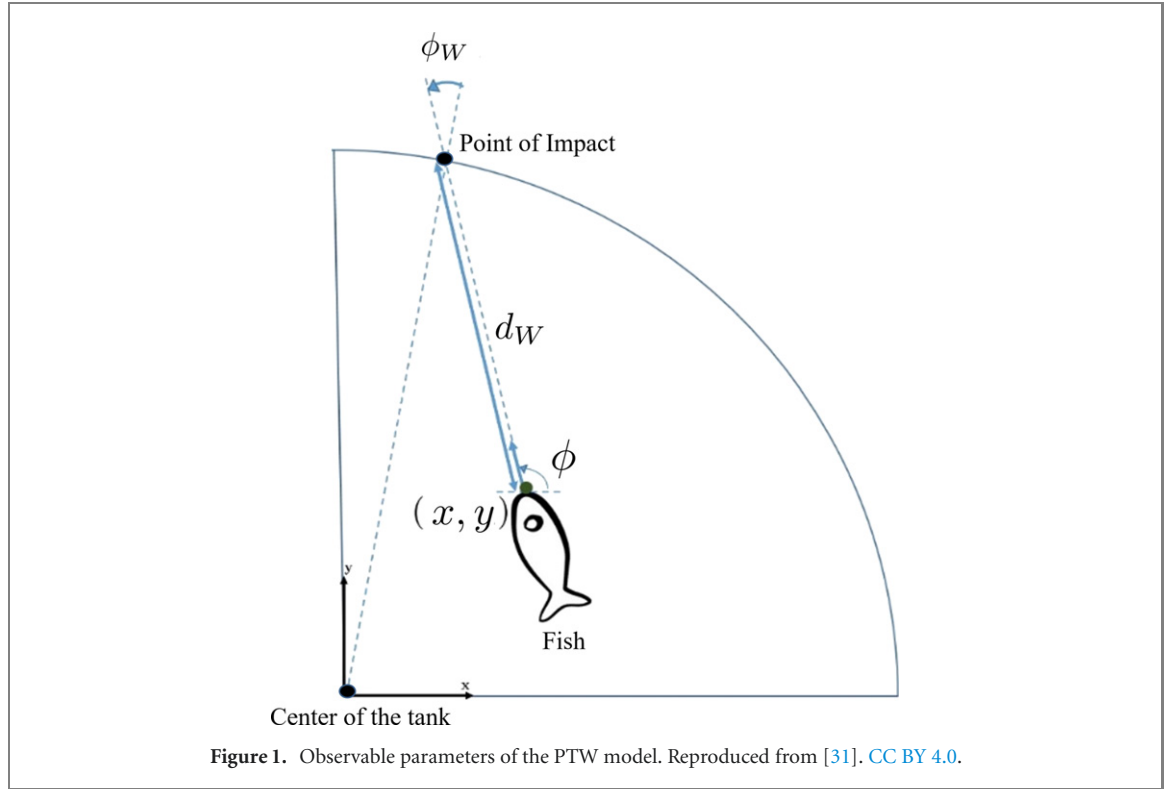
$$dw(t) = -v \left[\frac{1}{\xi} (w(t) - w^*(t))dt - \sigma dW \right], \quad (6)$$

where v is a constant speed, $w(t)$ is the angular velocity, σdW is a Wiener process of variance σ^2 and ξ is a constant. The time for the fish to hit the wall is given by $\tau_W = d_W/v$, where d_W is the distance between the current position of the fish and the point of impact on the wall (see figure 1). The point of impact refers to the point on the wall which the fish will hit if the fish keeps moving along the current heading direction. Moreover, ϕ_W denotes the angle between the heading of a fish and the normal direction that is perpendicular to the tangent line of the boundary at the point of impact. Then, the target angular velocity, w^* is given by,

$$w^*(t) = \hat{k}_W \frac{\text{sgn}(\phi_W)}{\tau_W}, \quad (7)$$

where $\hat{k}_W$ is a scalar constant.

Equations (4)–(7) denote the motion model for a single fish inside a circular tank [30, 31]. As seen in (6), the target angular velocity, $w^*(t)$ enables the fish to avoid collisions with boundaries of the tank by finding the distance from the wall ahead of itself and the relative direction of its heading angle with respect to the wall. Since these two terms are calculated at a single point of impact, they can be calculated for any smooth shaped surfaces. Equation (6) can thus be used as a general collision avoidance behavior for any arbitrarily shaped obstacle which lies ahead of the fish at any given time instant. The change in angular velocity $dw(t)$ depends on the value of $w^*(t)$ and a random component, σdW (4)–(7). When the fish is far from the wall, σdW primarily governs its motion and when the fish is relatively closer to the wall, $w^*(t)$ depicts its wall avoidance behavior. It is worth mentioning here that the given motion model results in the fish having



smooth trajectories as shown in [30, 31]. The only situation where the above model will not enable the fish to avoid the collision or have an abrupt or nonsmooth motion will be when at time $t = 0$, the value of τ_W is negligible or close to zero. In this situation, the fish might demonstrate behavior that is not predicted by the model, which is not considered in this paper.

2.3. The coverage problem

Consider a bounded and connected search-space, $\Omega \in \mathbb{R}^D$, where $D \in \{2, 3\}$ is the dimension of the search-space. We denote the search-space in $\mathbb{R}^D$ by Ω^D and discretize it into a square grid such that there are a total of $N_T = n^D$ grid cells, where n denotes the number of cells in each dimension of the search space (see figure 2(a)).

Let c_i denotes the grid cell at the i th position of the grid. Then the discretized search-space Ω^D is given by,

$$\Omega^D = \{c_i\}, \quad i = 1, \dots, N_T. \quad (8)$$

We consider a mobile robot deployed in the search-space Ω . The robot at any time t , is represented by the robot's center $\mathbf{r}(t)$, where $\mathbf{r}(t) \in \{\mathbb{R}^2, \mathbb{R}^3\}$. We consider the following assumptions.

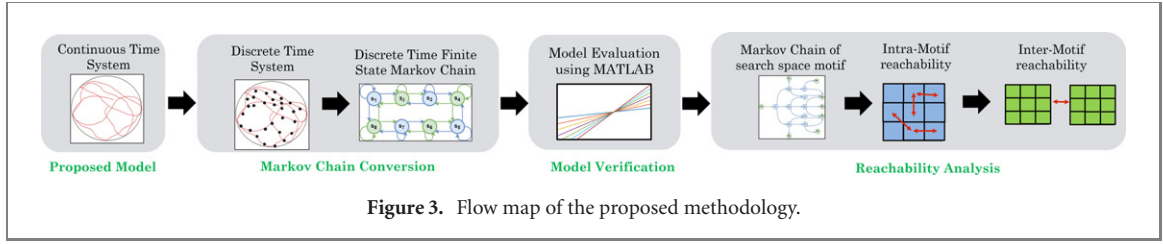
Assumption 2.1. The speed of the robot is constant.

Assumption 2.1 is to satisfy the constant speed that is dictated by the PTW model (4)–(6), where the fish is moving with a constant speed and a varying heading. This assumption can be ensured for the robot by fixing the speed v and step size h as constants.

Then, the motion dynamics of the robot at a given time instant t is as follows,

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + h\mathbf{u}, \quad (9)$$

where $\mathbf{u} \in \{\mathbb{R}^2, \mathbb{R}^3\}$ is an input velocity vector.



Define

$$\Omega_c^D(t) = \{c_i | c_i(t) \text{ has been visited till time } t\}, \quad (10)$$

and

$$\Omega_c^D(t) = \{c_i | c_i(t) \text{ has not been visited till time } t\}, \quad (11)$$

where (10) and (11) satisfy $\Omega_c^D \cup \Omega_c^D = \Omega^D$. We say that a search-space is covered if,

$$\lim_{t \rightarrow \infty} |\Omega_c^D| \rightarrow n^D, \quad (12)$$

where $|\Omega_c|$ is the cardinality of the set Ω_c , and n^D is the total number of grid cells in the D -dimensional search-space.

2.4. Problem statement

Given that the robot does not possess any localization service and has no means of knowing which cells are covered, we aim to design the control input $\mathbf{u}$ in (9) such that the robot is able to cover a given search-space in minimum possible time t . We propose to solve this problem by using the PTW model as a random search strategy. In order to apply PTW in a three-dimensional search-space, and to justify and quantify the coverage performance, we aim to solve the following four problems,

Problem 1. Extend the two-dimensional continuous-time and continuous state PTW model to a three-dimensional model.

Problem 2. Discretize the PTW model into a DTMC model suitable for GCM techniques.

Problem 3. Justify the full coverage of the search-space using the PTW model.

Problem 4. Show that the PTW model outperforms other types of random walks by providing faster coverage.

We propose solutions to each of the aforementioned problems in the subsequent sections. The proposed methodology is also shown in figure 3.

3. Extending the PTW to 3D environments

In this section, we address problem 1 which involves extending the two-dimensional PTW model to three dimensions. Let the heading of an agent on the

xy -plane and yz -plane be given by the azimuth angle ϕ_{xy} and the altitude angle ϕ_{yz} (see figure 4). Then, we can denote the heading by a vector $\phi \in S^2$ where $\phi = [\phi_{xy} \ \phi_{yz}]^T$. Likewise, if we denote the position of the agent by $\mathbf{r} \in \mathbb{R}^3$ where $\mathbf{r} = [x \ y \ z]^T$, then the dynamic equations governing the motion model in 3D can be given as follows (see figure 4),

$$d\mathbf{r} = \begin{bmatrix} v \cos \phi_{xy} \cos \phi_{yz} \\ v \sin \phi_{xy} \cos \phi_{yz} \\ v \sin \phi_{yz} \end{bmatrix} dt \quad (13)$$

$$d\phi = \begin{bmatrix} w_{xy}(t) \\ w_{yz}(t) \end{bmatrix} dt \quad (14)$$

$$dw_{xy}(t) = -v \left[\frac{1}{\xi_{xy}} (w_{xy}(t) - w_{xy}^*(t)) dt - \sigma_{xy} dW \right] \quad (15)$$

$$dw_{yz}(t) = -v \left[\frac{1}{\xi_{yz}} (w_{yz}(t) - w_{yz}^*(t)) dt - \sigma_{yz} dW \right], \quad (16)$$

where v denotes the constant speed of the agent, along the x , y and z axes. Likewise, w_{xy} is the angular velocity on the xy -plane and w_{yz} is the angular velocity on yz -plane while the scalar constants v , ξ and σ are similar to those described in equations (4)–(7). The subscripts xy or yz denote the value of that parameter on the respective plane. Similarly, the target angular velocities, w_{xy}^* and w_{yz}^* on the xy and yz planes are given by,

$$w_{xy}^*(t) = \hat{k}_{W_{xy}} \frac{\text{sgn}(\phi_{W_{xy}})}{\tau_{W_{xy}}} \quad (17)$$

$$w_{yz}^*(t) = \hat{k}_{W_{yz}} \frac{\text{sgn}(\phi_{W_{yz}})}{\tau_{W_{yz}}}. \quad (18)$$

The proposed 3D model can be visualized as the 2D model being simultaneously satisfied on two orthogonal planes. In our case, these are the xy and yz planes.

4. Discretizing the PTW model

In this section, we address problem 2 in section 2.4 by using the GCM method [35, 36]. GCM converts a dynamical system into discrete-time finite-state Markov chain (DTMC) described by a one-step transition probability matrix. To derive this transition matrix, we first convert the continuous-time PTW

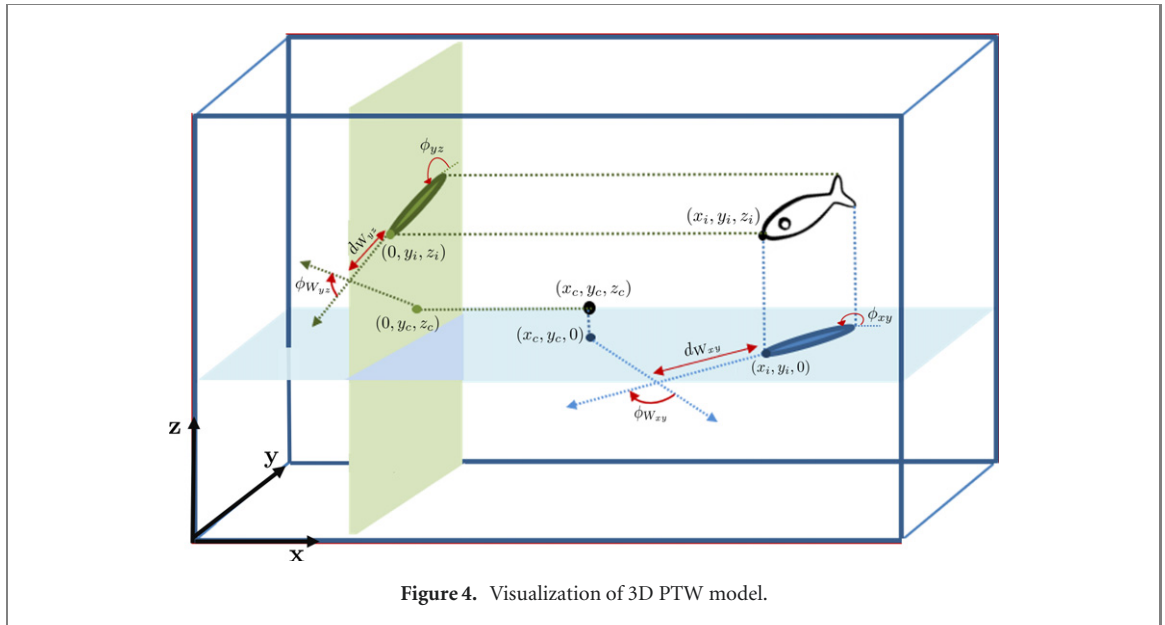


Figure 4. Visualization of 3D PTW model.

model into a discrete-time PTW model. Then, the space of the position state is discretized into geometrical cells. These cells are used to initiate simulated trajectories for the GCM using the discrete-time PTW model. The number of trajectories connecting one cell to another cell are used to formulate the one-step probability transition matrix.

4.1. Time-discretization of the PTW model

Consider the 2D PTW model given in (4)–(7). Let h denotes the Euler discretization time step. Then the discretized 2D PTW model is given by

$$\mathbf{r}_{k+1} = \mathbf{r}_k + v \begin{bmatrix} \cos \phi_k \\ \sin \phi_k \end{bmatrix} h \quad (19)$$

$$\phi_{k+1} = \phi_k + w_k h \quad (20)$$

$$w_{k+1} = w_k - v \left[\frac{1}{\xi} (w_k - w_k^*) h - \sigma \sqrt{h} W_k \right] \quad (21)$$

$$w_k^* = \hat{k}_W \frac{\text{sgn}(\phi_{W_k})}{\tau_{W_k}}, \quad (22)$$

where W_k is a zero-mean Gaussian random variable with a variance equal to 1. The Euler Maruyama method discretizes the continuous-time dynamical system to a discrete-time dynamical system with an accuracy of the order of $\sqrt{h}$, and ideally $h \ll 1$ as shown in [40]. Likewise, the discretized 3D PTW model is given by,

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \begin{bmatrix} v \cos \phi_{xy_k} \cos \phi_{yz_k} \\ v \sin \phi_{xy_k} \cos \phi_{yz_k} \\ v \sin \phi_{yz_k} \end{bmatrix} h \quad (23)$$

$$\phi_{k+1} = \phi_k + \begin{bmatrix} w_{xy_k} \\ w_{yz_k} \end{bmatrix} h \quad (24)$$

$$w_{xy_{k+1}} = w_{xy_k} - v_{xy} \left[\frac{1}{\xi_{xy}} (w_{xy_k} - w_{xy_k}^*) h - \sigma_{xy} \sqrt{h} W_k \right] \quad (25)$$

$$w_{yz_{k+1}} = w_{yz_k} - v_{yz} \left[\frac{1}{\xi_{yz}} (w_{yz_k} - w_{yz_k}^*) h - \sigma_{yz} \sqrt{h} W_k \right]. \quad (26)$$

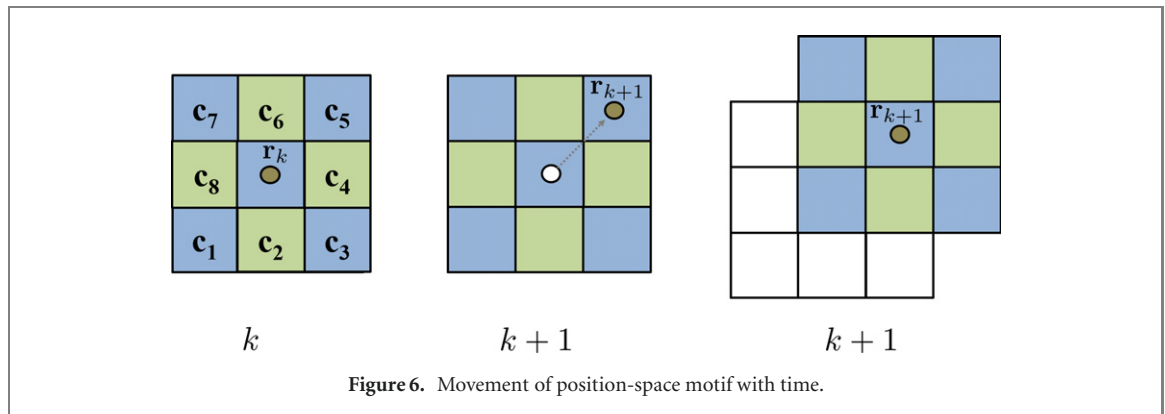
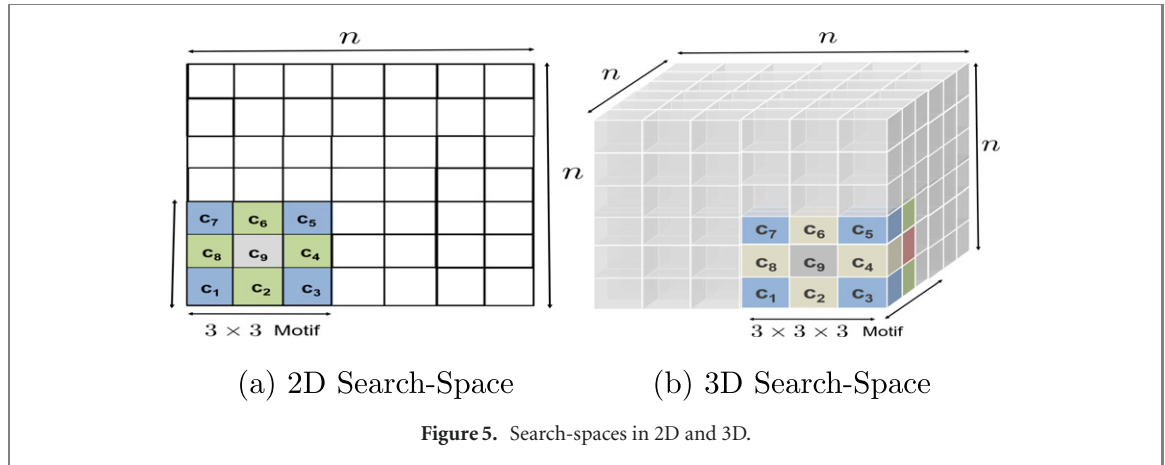
In the next section we discretize the position search space that will be used to evaluate the coverage performance of the PTW model.

4.2. Discretization of the position state-space

As shown in figure 2(a), the search space is discretized into N_T square cells, where each cell has a length denoted by l . In order to ensure that the fish can only transition to a cell which is adjacent to the current cell, we set $l = vh$ where v is the constant speed and h is the discrete time step.

Remark 4.1. Due to the constant step size vh , transitions from a cell to itself might occur. In the process of deriving the transition matrices via GCM, we ignore these transitions. Since we are deriving the transition matrices for evaluating area coverage by the PTW model, staying in the same cell implies that the cell has already been covered. However, the same cell transitions cannot be ignored if we want to theoretically analyze the rate of convergence of the coverage algorithm, which we leave it for future work.

Consider a 3×3 moving motif as shown in figure 5(a). We label each grid cell by c_i where $i \in \mathcal{I} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. At any time instant, k , the



state $\mathbf{r}_k$ can be assumed to be at cell c_9 of the moving motif. Let $\mathcal{N} = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ be the set of all adjacent cells to cell c_9 as shown in figure 6. Under the setup $l = \nu h$, at any given time instant k , we can approximate the motion of the fish by a transition from cell c_9 to any of its adjacent cells in $\mathcal{N}$.

Let $\mathbf{r}_k = [x_k \ y_k]^T$, and define $x_{c_i(\min)}$, $y_{c_i(\min)}$, $x_{c_i(\max)}$, and $y_{c_i(\max)}$ to denote the minimum and maximum xy values of the grid cell c_i , respectively. Then we define the mapping $g: [x_{c_i(\min)}, y_{c_i(\min)}, x_{c_i(\max)}, y_{c_i(\max)}] \rightarrow \mathcal{I} \setminus \{9\}$ such that $g(\mathbf{r}_{k+1}) = c_i$ if $x_{k+1} \in [x_{c_i(\min)}, x_{c_i(\max)}]$ and $y_{k+1} \in [y_{c_i(\min)}, y_{c_i(\max)}]$, i.e. any value of $\mathbf{r}_{k+1}$ can be mapped to a grid cell $c_i \in \mathcal{N}_{c_9}$. To determine c_i for $\mathbf{r}_{k+2}$, we map $\mathbf{r}_{k+1}$ to the cell c_9 , and then $c_i \in \mathcal{N}_{c_9}$ is evaluated for $k+2$ as explained above. This process can be visualized as the 3×3 position-space motif moving at each time instant such that $\mathbf{r}_k$ always maps to the center grid cell c_9 of the motif, as shown in figure 6.

In order to discretize the position variable for the 3D PTW model, we define a $3 \times 3 \times 3$ position-space motif as shown in figure 5(b). This results in a total of 27 grid cells in a given motif (see figure 7). At any time instant, $\mathbf{r}_k$ can be mapped to the grid cell at the center of the position-space motif i.e. c_{27} . At the next time instant, $\mathbf{r}_{k+1}$ can be mapped to c_i where $c_i \in \mathcal{N}_{c_{27}}$ and $|\mathcal{N}_{c_{27}}| = 26$.

4.3. Using GCM to obtain a probability transition matrix

In the position-space motif, we denote the transition between cell c_9 to cell c_i by the state s_i where $i \in \{1, \dots, 8\}$. In the 2D case, there are 8 possible states or transitions from the cell c_i to its neighboring cells. Likewise, in the 3D case, there are a total of 26 possible transitions from the center grid cell i.e. c_{18} to the neighboring cells (see figure 7). Similar to the 2D case, we define the transition from c_{18} to c_i by the state s_i where $i \in \{1, \dots, 27\} \setminus \{18\}$.

We now identify the one-step transition probability matrix $\Pr(s = s_j | s = s_i)$ for the 2D discretized PTW model where $i, j = 1, 2, \dots, 8$. We use Monte Carlo simulations (10^5 samples) of the discrete-time system given in (19)–(22). The values for ν , ξ and $\hat{k}_w$ were taken to be the same as identified in [30, 31]. However, the value of σ in (21) determines the amount of stochasticity exhibited in the PTW model and can significantly affect the amount of Gaussian noise in the model. Hence, we select the value of σ such that it satisfies certain properties seen in the real fish data from [30]. This can be achieved by using a systematic approach using the model verification techniques described in the next section (section 4.4).

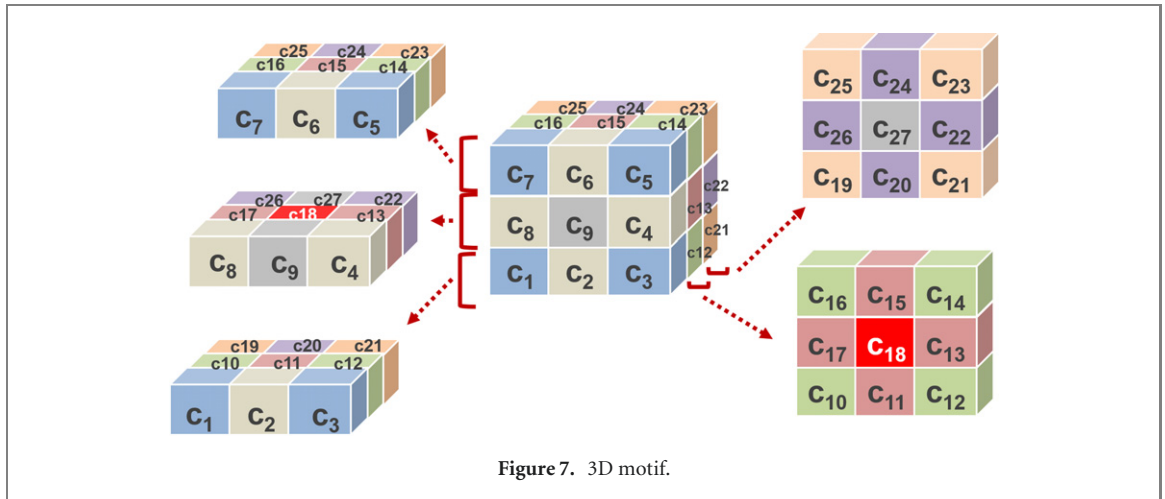


Figure 7. 3D motif.

Let the total number of simulated trajectories be denoted by $n_f \in \mathbb{N}$ and the length of each trajectory by $k_f \in \mathbb{N}$. Then, each n th trajectory can be mapped to a vector

$$\mathbf{s}^n = [s_{i,0}^n, s_{i,1}^n, \dots, s_{i,k}^n, s_{i,k+1}^n, \dots, s_{i,k_f}^n], \quad (27)$$

where $s_{i,k}^n \in \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$ represents the state at time instant k of the n th trajectory.

The transition probability is a matrix $P = [p_{ij}]$ where p_{ij} is the probability to go from state s_i to state s_j . The probability p_{ij} can be estimated from the data as,

$$p_{ij} = \frac{n_{ij}}{\sum_{l=1}^{N_s} n_{il}}, \quad (28)$$

where n_{ij} is the total number of times there is a transition from state s_i to s_j and $\sum_{l=1}^{N_s} n_{il}$ represents the total number of times there is a transition from state i to any other state l where $l \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $N_s = 8$. Equation (28) can be viewed as a maximum likelihood estimator with the probability values getting more consistent as the sample size increases [41]. Hence, it is possible to estimate a consistent transition matrix with a large enough sample size. To avoid the case where the fish might collide with the wall at the initial iterations, we consider the following assumption,

Assumption 4.1. The simulated trajectories always start sufficiently far away from the boundary of the search-space.

The transition matrix represents an incremental change in the position and heading of the agent. However, this incremental change may be infinitely large if the agent is about to have a head-on collision with an obstacle. To rule out these cases, we assume that the fish starts away from the boundary so that it can incrementally change its position in a manner to avoid collisions with the obstacles.

Remark 4.2. All possible positions in the search-space are used to generate trajectories for the GCM. Therefore, the transition matrix obtained by the GCM represents an average model for the entire

search-space. Figure 8(b) shows the structure of the transition matrix which is satisfied irrespective of whether the fish is near the boundary or far away from the boundary. The transitional probabilities colored blue are greater than zero, the ones colored red are equal to zero and the ones colored white are of negligible values.

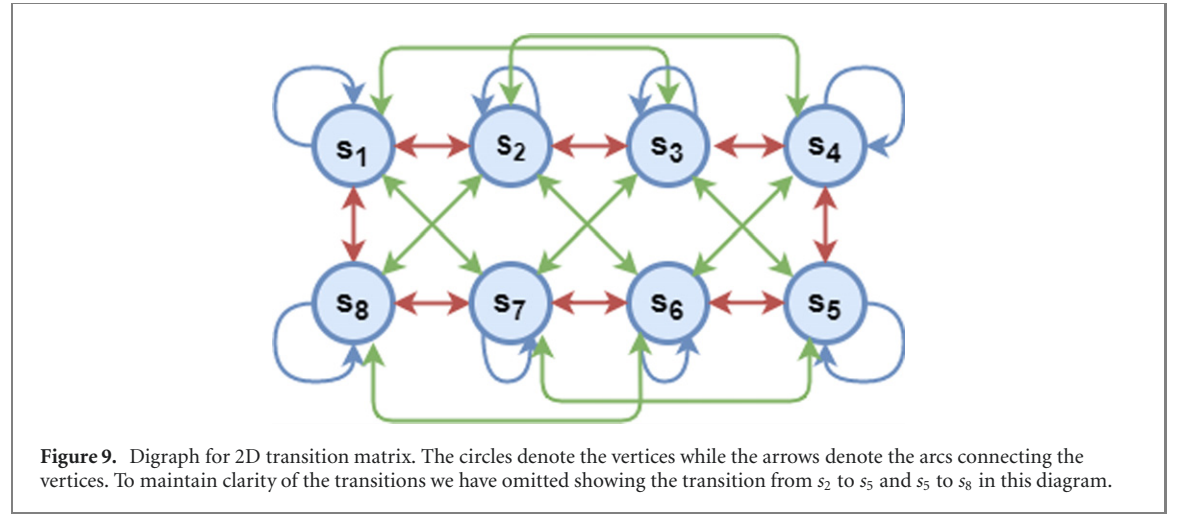
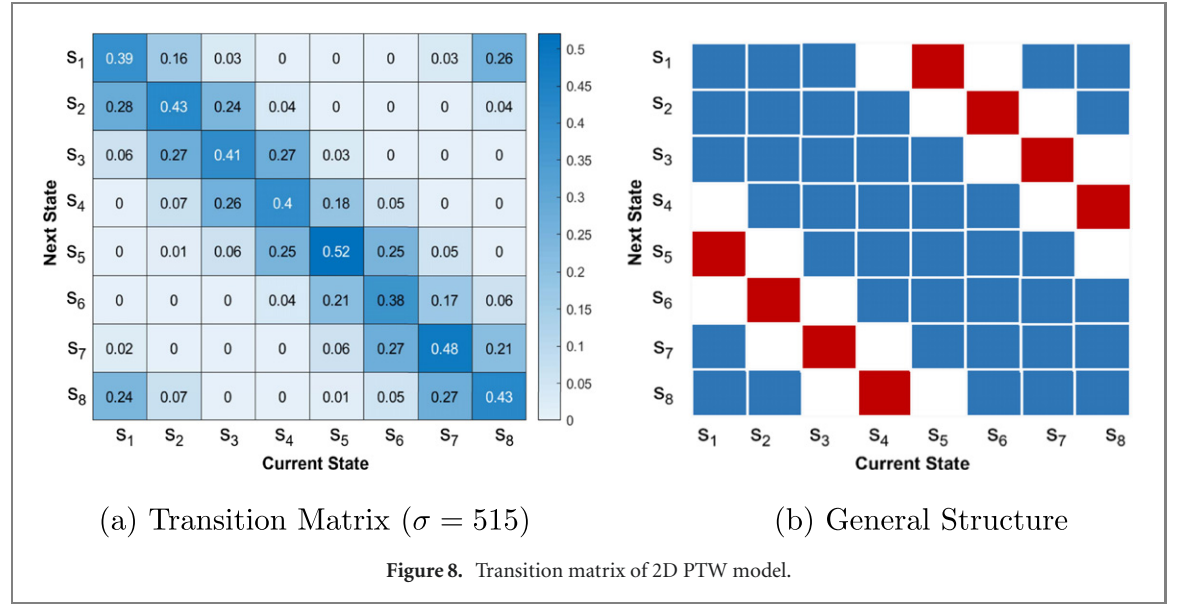
The one-step transition matrix can be converted to its associated digraph. In the GCM framework, conversion to a digraph is done in the following way: each state in the given Markov Chain can be considered as a vertex in the digraph. For $p_{ij} > 0$, there exists a directional edge connecting vertices i and j . An example of the one-step transition matrix for the 2D case is shown in figure 8(a), and the digraph for the transition matrix having the structure shown in figure 8(b) is given in figure 9.

Using the mapping mentioned for the 3D case, a given n th trajectory can be represented by a vector $\mathbf{s}^n$ as defined in (27). The probability transition matrix can similarly be estimated for the 3D PTW model by counting the number of state transitions in the simulated trajectories by using (28) where $N_s = 26$. The transition matrix of Markov Chain for $n_f = 100$ and $k_f = 50$ is given in appendix B. The characteristics of the 3D model are similar to that of the 2D model. The diagonal entries of the transition matrix being close to unity indicate that the model resists change in its current direction and only changes its direction in an incremental manner.

The DTMC model obtained depends on several choices of parameters during the discretization process. Even though we can infer properties of the identified transition probability matrix, it is difficult to guarantee that all matrices identified will satisfy the properties we have observed. In the next section, we will provide systemic guarantees by evaluating the discretized model.

4.4. Evaluation of discretized model

An important property suggested by real fish trajectories in [30] is that the backtracking probability is negligible. A couple of other properties include the



dominance of the diagonal terms along with a non-zero probability of transition to its neighbors' neighboring states. In this section, we evaluate the discretized model and tune several design parameters of the model, such that the aforementioned properties are satisfied by the DTMC. The desired system properties are first expressed in the MATLAB language and are then verified to determine whether or not the DTMC satisfies the given properties.

PTW model specifications. Let the backtracking probability, $\mathbb{P}_{BT}$, be defined as follows,

$$\mathbb{P}_{BT} = \begin{cases} \mathbb{P}(s_{k+1} = s_i | s_k = s_{i+4}), & \forall i \in \{1, 2, 3, 4\} \\ \mathbb{P}(s_{k+1} = s_i | s_k = s_{i-4}), & \forall i \in \{5, 6, 7, 8\} \end{cases}, \quad (29)$$

where $\mathbb{P}(s_{k+1} = s_i | s_k = s_{i+4})$ denotes the probability of being in state s_i at time instant $k+1$ given the state at the previous time instant is s_{i+4} . At time instant k , suppose $s_k = s_5$, then $s_{k+1} = s_1$ would imply backtracking at time instant $k+1$. Likewise, if $s_k = s_4$ and $s_{k+1} = s_8$ or if $s_k = s_2$ and $s_{k+1} = s_6$, then both of these

examples can be viewed as cases where backtracking holds true.

We now evaluate the three properties exhibited by the PTW model. The first one involves analyzing the upper bound on the backtracking probability of the model. The second one finds the lower bound on the probability of being in the same state at two consecutive instances of time. Lastly, the third property determines the lower bound on the probability to transition from a state s_i to any of its neighbors' neighboring states. These properties can be stated as follows,

- (a) $\mathbb{P}_{BT} \leq \epsilon_1$.
- (b) $\mathbb{P}(s_{k+1} = s_i | s_k = s_i) \geq \epsilon_2, \forall i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.
- (c) $\mathbb{P}(s_{k+1} = s_i | s_k \in \{(s_i^{\text{adj}})^{\text{adj}}\}) \geq \epsilon_3, \forall i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$,

where s_i^{adj} is a set containing the states adjacent to state s_i , while $(s_i^{\text{adj}})^{\text{adj}}$ are the states neighboring to those adjacent states. This implies that $s_1^{\text{adj}} = \{s_8, s_2\}$ and $s_8^{\text{adj}} = \{s_1, s_7\}$, while for any $j \notin \{1, 8\}$,

$s_j^{\text{adj}} = \{s_{j-1}, s_{j+1}\}$. This in turn implies that $s_1, (s_1^{\text{adj}})^{\text{adj}} = \{s_8, s_2, s_7, s_3\}$, $(s_2^{\text{adj}})^{\text{adj}} = \{s_1, s_8, s_3, s_4\}$ and similar logic can be followed to determine the adjacent states of neighboring states for any state s_i . The values of ϵ_1, ϵ_2 , and ϵ_3 can be found by analysis or using a software program, such as MATLAB. (Appendix A shows in detail how it can be done for a 2D state transition matrix.)

Remark 4.3. The specifications formulated for the PTW model are simple enough that they can be verified by a simple computer program. For complex specifications which have temporal or spatial correlations between them, probabilistic model checkers can be used to verify the models against the given specifications.

A careful examination of the discretized PTW model (19)–(22) shows that the properties exhibited by the discretized model largely depend on the value of σ and the dimensions of the discretized grid cell (vh) selected for GCM in section 4.2. If the value of v is taken to be constant as shown in real fish trajectories, then the selection of σ and h values prior to the discretization process can affect the values of ϵ_1, ϵ_2 and ϵ_3 considerably. We use a computer program to evaluate the range of values of σ and h such that the Markov chain of the 2D PTW model satisfies the following: (1) $\epsilon_1 \approx 0$, (2) $\epsilon_2 > 0$ and (3) $\epsilon_3 > 0$. Any combination of h and σ values in that particular range can then be used to discretize the PTW model such that it satisfies the aforementioned properties.

In order to achieve the above, the PTW model can be discretized using different values of σ and h where $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ and $h \in [h_{\min}, h_{\max}]$. After discretization, the GCM method can be used to find the corresponding Markov chain. To evaluate ϵ_1 for a particular value of σ and h , we consider the Markov chain of the 2D PTW model to start from an initial state, s_i where $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. For each s_i , the probability to transition into the state where backtracking holds true is evaluated. As seen in specification (a), ϵ_1 constitutes the upper limit on the backtracking probability for any state s_i in the DTMC (8 states). The same procedure can be repeated to evaluate ϵ_1 for the DTMC representing the 3D discretized PTW model (26 states). Likewise, to find ϵ_2 for the DTMC of 2D PTW model, the same process can be repeated but this time, ϵ_2 represents the lower bound on the probability to remain in the same state. Again, this represents 8 cases to be evaluated for the DTMC of the 2D PTW model and 26 cases for the 3D PTW model. Lastly, ϵ_3 constitutes the lower bound on the probability to transition from a given state to any of the states adjacent to its neighboring states.

In the 2D case, for $340 \leq \sigma \leq 600$, $\epsilon_1 = 0$, $\epsilon_2 > 0$ and $\epsilon_3 > 0$. For each value of σ , a range of values of h satisfies the three aforementioned properties. Let h_σ denote the maximum value of h for a particular value of σ that satisfies the three given

properties. In the 2D case, for $\sigma = 600$, $h_\sigma = 0.01$, $\epsilon_1 = 0$, $\epsilon_2 = 0.48$ and $\epsilon_3 = 0.02$. As the value of σ increases, the value of h_σ that satisfies the above conditions also tends to increase and a similar trend can be seen for smaller values of σ . This was evaluated for $h_{\min} = 0.01$, $h_{\max} = 0.2$, $\sigma_{\min} = 0$ and $\sigma_{\max} = 2000$. Similarly, the values of σ and h such that $\epsilon_1 = 0$, $\epsilon_2 > 0$ and $\epsilon_3 > 0$ can also be evaluated for the 3D case. For a given value of h , the effect of increasing σ values on the transition matrices in 3D and 2D is shown in appendix B.

5. Using PTW as a coverage strategy

In this section, we address problem 3 in section 2.4 and use reachability analysis to show that the PTW model guarantees coverage of the search-space. Our goal is to show that under the PTW model, the robot will visit every cell in the search-space with nonzero probability. This problem would seem straightforward, if we use GCM to obtain a probability transition matrix for the entire position-space. However, this will be computationally prohibitive. Instead, we use the already derived Markov chain of the moving position-space motif from section 4.3 to show that the digraph of the entire search space is connected.

5.1. Reachability analysis

In order to evaluate the reachability of the entire search-space in 2D, we consider 3×3 search-space motifs. These motifs can be placed next to one another to make the entire search-space as shown in figure 5(a). Note that these motifs are not similar to the position-space motifs, which were introduced in section 4.2. This is because the position-state motif is considered as a moving motif in time (figure 6) and its Markov chain is shown in figure 8(b). In order to evaluate reachability of the entire search-space, we first show the inter-motif reachability of the search-space motifs. We evaluate the former by using the derived Markov chain of the moving motif and use this property to show the connectivity of the entire search-space. Finally, we prove that any cell in the search-space can be reached from any other cell with a non-zero probability. We prove this for both the 2D and 3D cases.

In figure 10, we show a general representation of the inter-motif reachability in two dimensions. Suppose at time instant k the agent is at cell c_9 of the search-space motif M_9 , where M_i refers to the i th search-space motif in the 2D space. Then, in figure 10(a), we show all possible transitions to the neighboring overlapping motifs such that each transition starts from cell c_9 of the current motif at time k to cell c'_9 of the next motif at time $k + 1$. These transitions are represented by a star-digraph in figure 10(b). Finally, this representation is expanded to the entire search-space that yields a digraph $\mathcal{G}^A$ composed of the overlapping star-digraphs, as shown in figure 10(c).

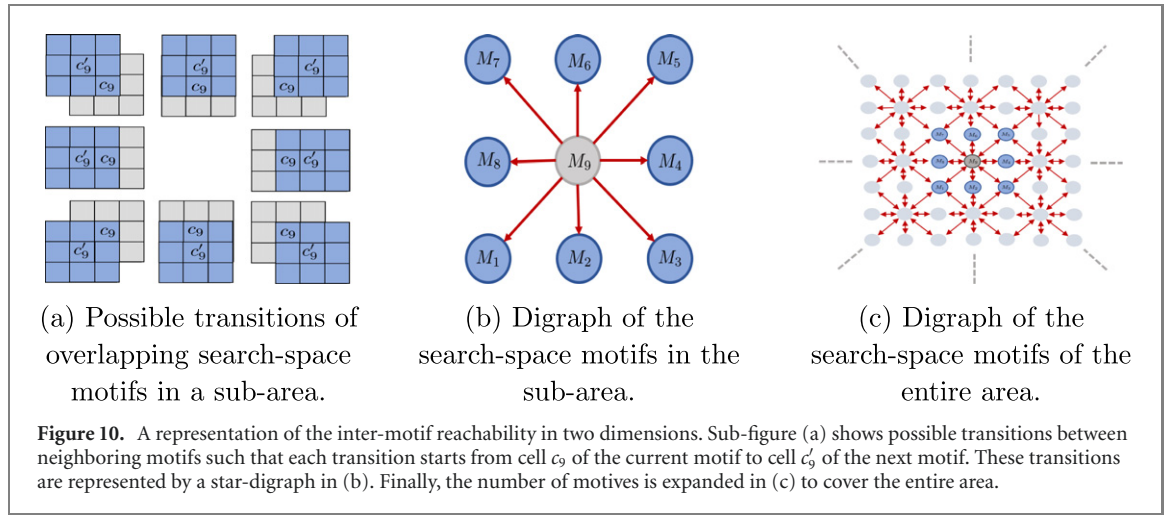


Figure 10. A representation of the inter-motif reachability in two dimensions. Sub-figure (a) shows possible transitions between neighboring motifs such that each transition starts from cell c_9 of the current motif to cell c'_9 of the next motif. These transitions are represented by a star-digraph in (b). Finally, the number of motifs is expanded in (c) to cover the entire area.

Lemma 5.1. *Under the Markov chain of the moving 3×3 motif in 2D (see figure 9), the digraph $\mathcal{G}^A$ of the 2D search-space is strongly connected.*

Proof. We need to show that the edges in the star-digraph in figure 10(b) indeed exist under the Markov chain shown in figure 9. To do so, note that, for example, cell c'_9 in motif M_7 is equivalent to cell c_7 in motif M_9 for the transition from M_9 to M_7 . Therefore, the probability of moving from the position-space motif M_9 to the position-space motif M_7 is equivalent to the probability of moving from cell c_9 to cell c_7 inside the position-space motif M_9 . Let $c_i \xrightarrow{s_a} c_j$ denote a transition from cell c_i to cell c_j under the state s_a . Let $c_i \xrightarrow{s_a} c_j \xrightarrow{s_b} c_l$ be a sequence of transitions first from cell c_i to cell c_j under the state s_a and then from cell c_j to cell c_l under the state s_b . Then, under the Markov chain shown in figure 9, there is a probability of moving from cell c_9 to cell c_7 by, for example, the following transition sequences:

$$c_1 \xrightarrow{s_5} c_9 \xrightarrow{s_7} c_7, \quad (30)$$

$$c_2 \xrightarrow{s_6} c_9 \xrightarrow{s_7} c_7, \quad (31)$$

$$c_3 \xrightarrow{s_7} c_9 \xrightarrow{s_7} c_7, \quad (32)$$

$$c_4 \xrightarrow{s_8} c_9 \xrightarrow{s_7} c_7, \quad (33)$$

$$c_5 \xrightarrow{s_1} c_9 \xrightarrow{s_7} c_7. \quad (34)$$

Note that the sequences $c_8 \xrightarrow{s_4} c_9 \xrightarrow{s_7} c_7$ or $c_6 \xrightarrow{s_2} c_9 \xrightarrow{s_7} c_7$ do not exist as there are no edges between s_4 and s_7 or between s_2 and s_7 in the Markov chain shown in figure 9. The transition sequences (30)–(34) imply that M_7 is accessible from M_9 . Following the same analysis, we can find the transition sequences that imply the accessibility of motifs $M_1 - M_8$ from motif M_9 . This implies that the star-digraph in figure 10(b) exists with a non-zero probability. Applying the same analysis in the expansion of the motifs in figure 10(c) yields a connected graph composed of union of all star-digraphs.

In order to show that the digraph of the search-space is strongly connected, we need to prove that for any two cells, c_i, c_j , if cell c_j is accessible from cell c_i , then there exists a sequence of transitions such that c_i is also accessible from cell c_j . Consider, for example, the transition from cell c_1 to cell c_9 in M_9 (see figure 11). Then, under the Markov chain shown in figure 9, the following transition sequences are possible:

$$c_1 \xrightarrow{s_5} c_9 \xrightarrow{s_k} c_k \quad \text{where } k \in \{3, 4, 5, 6, 7\}. \quad (35)$$

To prove the strong connectivity of the digraph $\mathcal{G}^A$, we need to show that for any of the transition sequences in (35), there always exists a path such that c_1 in M_9 is accessible from c_9 of M_9 .

Consider the overlapping motifs in figure 10(a). For $k = 3$, the transition sequence in (35) implies movement into the cell c_5 of the overlapping motif M_2 (see figure 11). Then, under the Markov chain given in figure 9, the following transition sequence starting from c_5 in M_2 is possible,

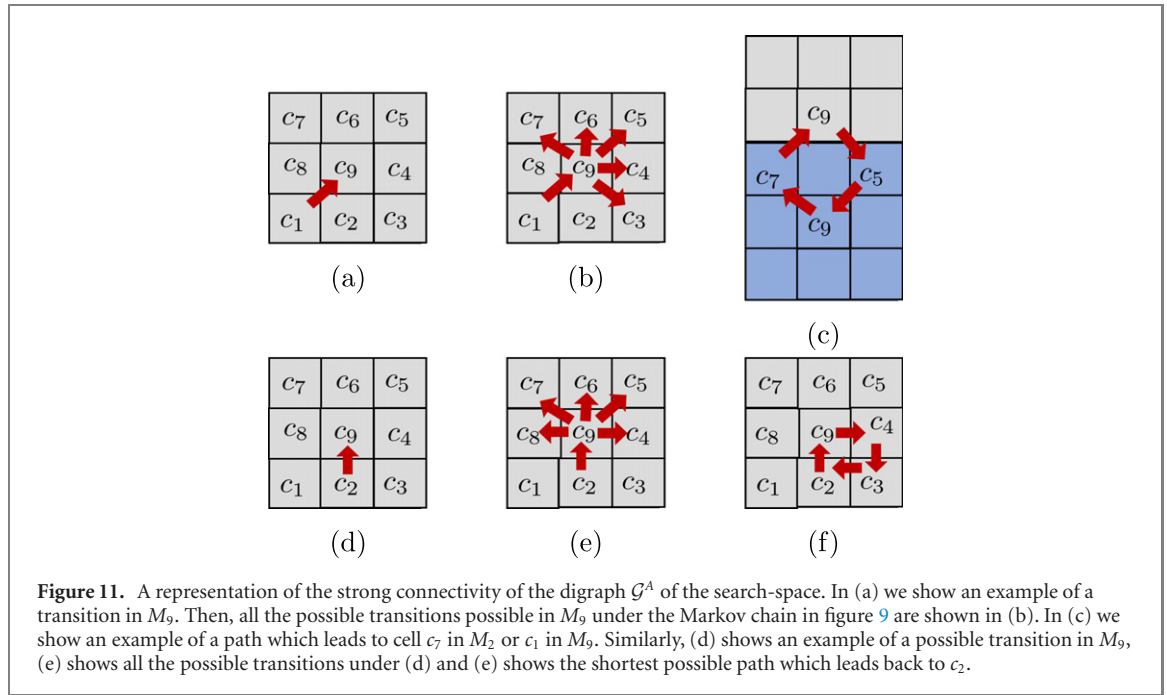
$$c_5 \xrightarrow{s_1} c_9 \xrightarrow{s_7} c_7. \quad (36)$$

The cell c_7 in M_2 is equivalent to cell c_1 in M_9 . Hence, there exists a path such that c_1 in M_9 is accessible from c_9 in M_9 . Likewise, due to the symmetry of the motif, similar analysis can be applied to show that after a transition from c_k to c_9 where $k \in \{3, 5, 7\}$, there always exists a path, such that, c_k is accessible from c_9 .

We can now extend a similar analysis to consider the transitions from c_l to c_9 where $l \in \{2, 4, 6, 8\}$. For $l = 2$, figure 11 shows a sequence of transitions which shows the existence of the following path,

$$c_2 \xrightarrow{s_6} c_9 \xrightarrow{s_4} c_4 \xrightarrow{s_2} c_3 \xrightarrow{s_8} c_2. \quad (37)$$

Note that the above path exists within the position motif and does not involve any of the overlapping neighboring motifs. Similar to c_2 , we can show that there always exists a path such that c_l is accessible from c_9 where $l \in \{4, 6, 8\}$. Hence, using the same analysis



for any two cells c_i and c_j in the position-space, we can always show that there exists a path such that if c_i is accessible from c_j , then c_j is also accessible from cell c_i . This proves the strong connectivity property of the digraph $\mathcal{G}^A$ of the position-space, hence, completing the proof. $\square$

Lemma 5.2. *Under the Markov chain of the moving $3 \times 3 \times 3$ motif in 3D, the digraph $\mathcal{G}^A$ of the 3D search-space is strongly connected.*

Proof. The proof of this lemma follows the same procedure of the proof of lemma 5.1 for the 2D case. Similar to figure 10, we can construct overlapping motifs of the 3D position-space motif of figure 7 to cover a sub-volume. We then show that a union of star-graphs exists under the Markov chains of the position-space motifs in 3D. We can then show that for any two cells, c_i and c_j , if c_i is accessible from c_j , then c_j is also accessible from c_i . Hence, this proves that the resulting digraph of the entire search-space is strongly connected. $\square$

Theorem 5.1. *Any cell in the 2D or 3D discretized search-space can be reached from any other cell with a non-zero probability.*

Proof. Let $\mathcal{M}^A$ be the Markov chain associated with the digraph $\mathcal{G}^A$ of the entire search-space. In lemmas 5.1 and 5.2 we proved that the digraph $\mathcal{G}^A$ of the 2D and 3D search-spaces are strongly connected. This implies that the probability of reaching any state of $\mathcal{M}^P$ from any other state strictly larger than zero. In other words, if an agent starts from any cell in the search-space, it is able to reach any other cell with a non-zero probability. $\square$

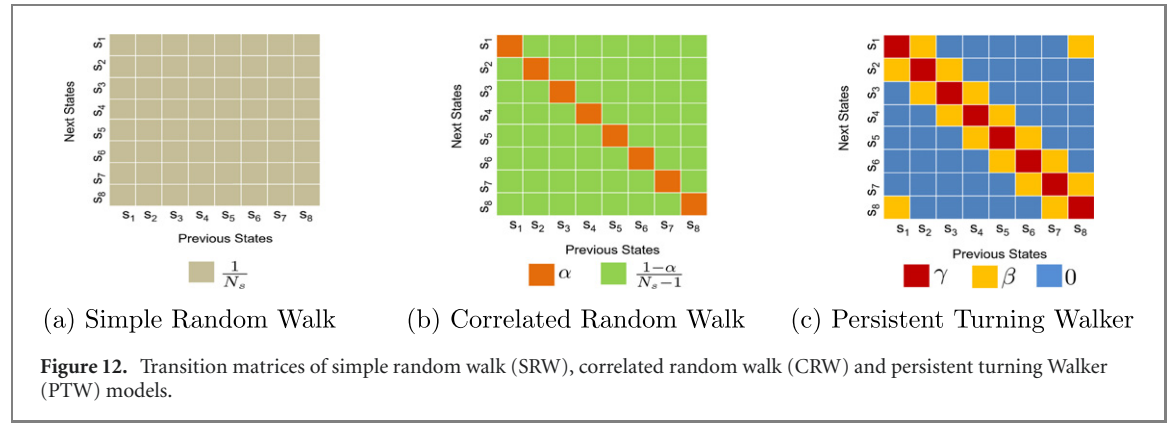
Remark 5.1. When the agent is at a cell adjacent to the boundary of the entire search-space, then we need to account for the possibility of colliding with the walls. This depends on the curvature of the boundary, the size of the robot, as well as the discretization size of the position-space motif. Hence, to avoid collision we need the discretization to be small enough such that the maximum possible in the position space enables the agent to switch direction at least parallel to the wall.

Remark 5.2. The digraph constructed from the Markov chain of the moving motif in (figure 9) shows the least connected graph which can be used to ensure coverage of the entire search-space. Any value of σ which results in a digraph having additional connected nodes than shown in figure 9 can be used to justify coverage in the same manner as described in this section.

6. Comparison with other strategies

In this section, we compare the area coverage of the PTW model with two other random walk strategies, the SRW and the CRW. The transition matrices of these strategies are shown in Figure 12). It is also trivial to extend the coverage proof of the PTW model and show that if an agent uses either of these models, the probability of an agent to reach any grid cell in the search-space is greater than zero. Hence, starting from any cell, the agent is able to reach any other cell in the search-space using either of the three strategies.

We now compare the performance of the SRW and CRW with the PTW. Let X be a discrete random process such that at time i , $X_i = 1$ if the robot visits an *unexplored* cell at time i , and $X_i = 0$ if the robot



visits an *explored* cell at time i . Let the discretized time step of a trajectory be denoted by h , where h is a constant. Then, given a trajectory of length k , i.e. $t = 0, h, 2h, \dots, kh$, the area covered during this trajectory is defined by

$$A_k = \sum_{i=0}^k X_i, \quad (38)$$

where each cell has a unit area. Note that the area definition (38) implicitly implies that the cells $c_{i-1}, c_{i+1} \in \mathcal{N}_{c,i}$, i.e. the cells c_{i-1}, c_i and cells c_i, c_{i+1} are adjacent to one another.

The expected number of cells or expected area covered till time t can be given as

$$E[A_k] = E\left[\sum_{i=0}^k X_i\right] = \sum_{i=0}^k E[X_i]. \quad (39)$$

We consider the following assumption,

Assumption 6.1.

$$\mathbb{P}(X_0, X_1, \dots, X_k) = \mathbb{P}(X_0)\mathbb{P}(X_1) \dots \mathbb{P}(X_k). \quad (40)$$

In any of the strategies (SRW, CRW, or PTW), the agent has no means of knowing whether the cell it is visiting at time t has been covered prior to time t . This is because the agent neither has a localization service nor keeps a record of the cells it has visited or yet to be visited. Hence, the aforementioned assumption can be justified since the probability of covering an already explored cell and the probability of covering an unexplored cell is independent of one another.

In view of assumption 6.1, the expectation (39) reduces to

$$E[A_k] = \sum_{i=0}^k (0 \times \mathbb{P}(X_i = 0) + 1 \times \mathbb{P}(X_i = 1)) \quad (41)$$

$$= \sum_{i=0}^k \mathbb{P}(X_i = 1). \quad (42)$$

We say that an agent exhibits backtracking under a given algorithm at a time instance k if the agent moves to the same grid cell which was visited at time instant

$k-1$. Let $\mathbb{P}_{BT}$ denote the backtracking probability of a search algorithm. Consider a neighborhood of the agent's current position, then if the only cell that has been visited by the agent, other than the cell that the agent is occupying, is the cell that the agent comes from, then for each neighboring cell, we have that

$$\mathbb{P}(X_i = 1) = 1 - \mathbb{P}_{BT}.$$

If there are other cell that has been previously visited by the agent, then

$$\mathbb{P}(X_i = 1) < 1 - \mathbb{P}_{BT}.$$

There for a particular algorithm, we can compute an upper bound for the area coverage as

$$E[A_k] \leq U_k = k(1 - \mathbb{P}_{BT}).$$

Theorem 6.1. Consider that at time instant k , U_k^{PTW} , U_k^{CRW} and U_k^{SRW} represent the upper bounds for the expected area covered by the PTW, CRW and SRW algorithms, respectively. Then for all values of k , the following hold:

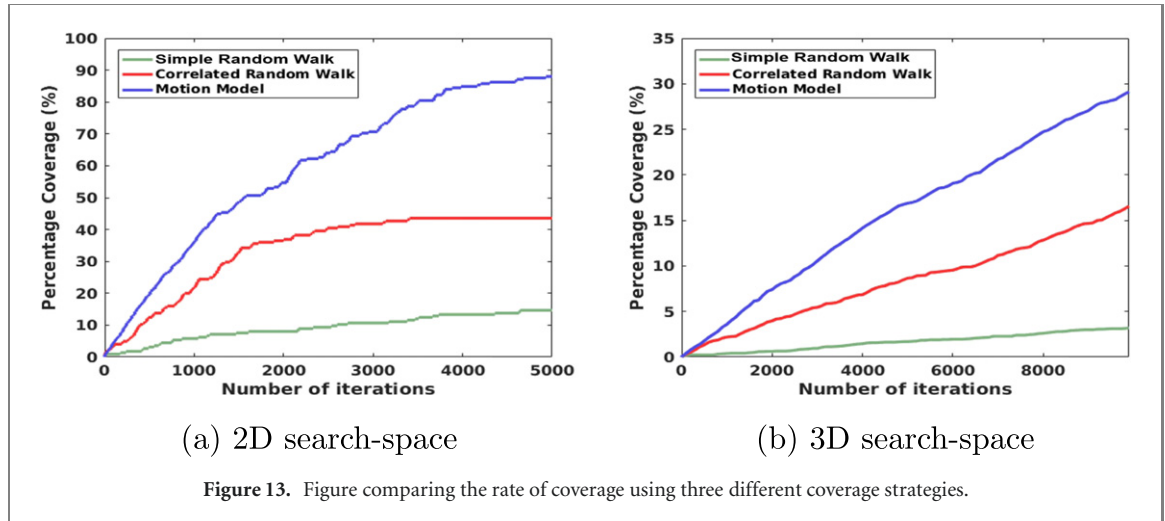
$$U_k^{PTW} \geq U_k^{CRW}, \quad (43)$$

and

$$U_k^{PTW} \geq U_k^{SRW}. \quad (44)$$

Proof. Given the definition of the upper bound, the conclusion is true because $\mathbb{P}_{BT}^{PTW} \leq \mathbb{P}_{BT}^{CRW}$ and $\mathbb{P}_{BT}^{PTW} \leq \mathbb{P}_{BT}^{SRW}$. $\square$

Remark 6.1. Unfortunately, we cannot claim that the expected area covered by an agent using the PTW model will always be greater than or equal to the area covered by the SRW or CRW models. However, since the upper bound for the area covered by PTW is greater than the upper bound for the area covered by the SRW and CRW, we can compare the search performance through numerical simulations using statistical tests. This will be performed in the next section.



7. Simulation results

In this section, we show simulation results for the proposed PTW model and compare the area coverage with the SRW and CRW models.

7.1. Simulation results

In order to visualize the area covered by an agent, the 3D PTW model was simulated for $k = 10\,000$ iterations. The area covered, A was calculated as follows:

$$A = \frac{N_X}{N_T} A_{ss}, \quad (45)$$

where N_X denotes the number of blocks (or grid cells) which have been covered by an agent, N_T is the total number of blocks in the search-space, and A_{ss} is the area of the search-space. Likewise, the rate of area coverage, R_A can be given as follows,

$$R_A = \frac{A}{K}, \quad (46)$$

where A is the area covered by an agent over K iterations.

Figure 13 shows the percentage of area covered by an agent with respect to time using each of the aforementioned coverage strategies. In the 2D case, a rectangular workspace of 1.5×1.5 was considered while in the 3D case, the workspace was extended to $1.5 \times 1.5 \times 1.5$. The space was divided into a grid where each grid cell was 0.1×0.1 (2D) and $0.1 \times 0.1 \times 0.1$ (3D). The simulation was done with the following parameters: $\sigma = 30$, $k_w = 0.4$, $\xi = 0.024$, $v = 0.8$, $\alpha = 0.9$ and $h = 0.01$. Note that the step size of the agent vh is 0.008 which is much less than the size of the grid cell 0.1. This implies that each iteration does not ensure that the agent will move to a different grid cell compared to the current one, at the next time step. However, this does not affect our results since we are interested in comparing the rate of coverage between different coverage strategies, and having a constant step size in each of these models still enables us to perform a fair comparison with one another. Since

$\alpha \approx 0.9$ for the PTW model (see figure 8(a)), we used the same value for the CRW for a fair comparison. Also, the position and heading variables at iteration $k = 0$ were kept constant across all three models in an experiment.

Figure 13 shows that the SRW model covers the area of the search-space at a much slower rate compared to the other two coverage strategies, while the PTW model covers more area in less number of iterations. As shown in figure 13(a), after 5000 iterations, an agent using SRW is able to cover approximately 15% of the search-space while the CRW covers close to 43%. The PTW model performs exceedingly well compared to both of the other strategies since it is able to cover almost 90% of the search-space in the same amount of time. We can see a similar trend for the 3D case (see figure 13(b)), where the rate of coverage is again highest for the proposed PTW model. The figure shows the results for the initial 10 000 iterations. It is straightforward to visualize that as the number of iterations becomes very large, it will ultimately enable the agent to cover the entire search-space.

Figure 14 shows sample trajectories of the agent when it uses each of the three coverage strategies in 3D. The starting position of the agent is denoted by the black dot in each graph and is kept constant in each coverage strategy to enable a fair comparison. It can be seen that the agent covers a significantly larger part of the search-space using the PTW model compared to the SRW and CRW models. This agrees with the results in figure 13 which shows a similar trend.

Mann Whitney test [42] is a non-parametric statistical test that is used to compare the difference in means of two arbitrary distributions. Since the area covered until time t follows an ordinal scale and can have any arbitrary distribution, it is natural to consider the Mann Whitney Wilcoxon test for our purpose. The parameter of interest in our case is the mean of the distribution which represents the covered area. If the mean area covered by strategy 1 and

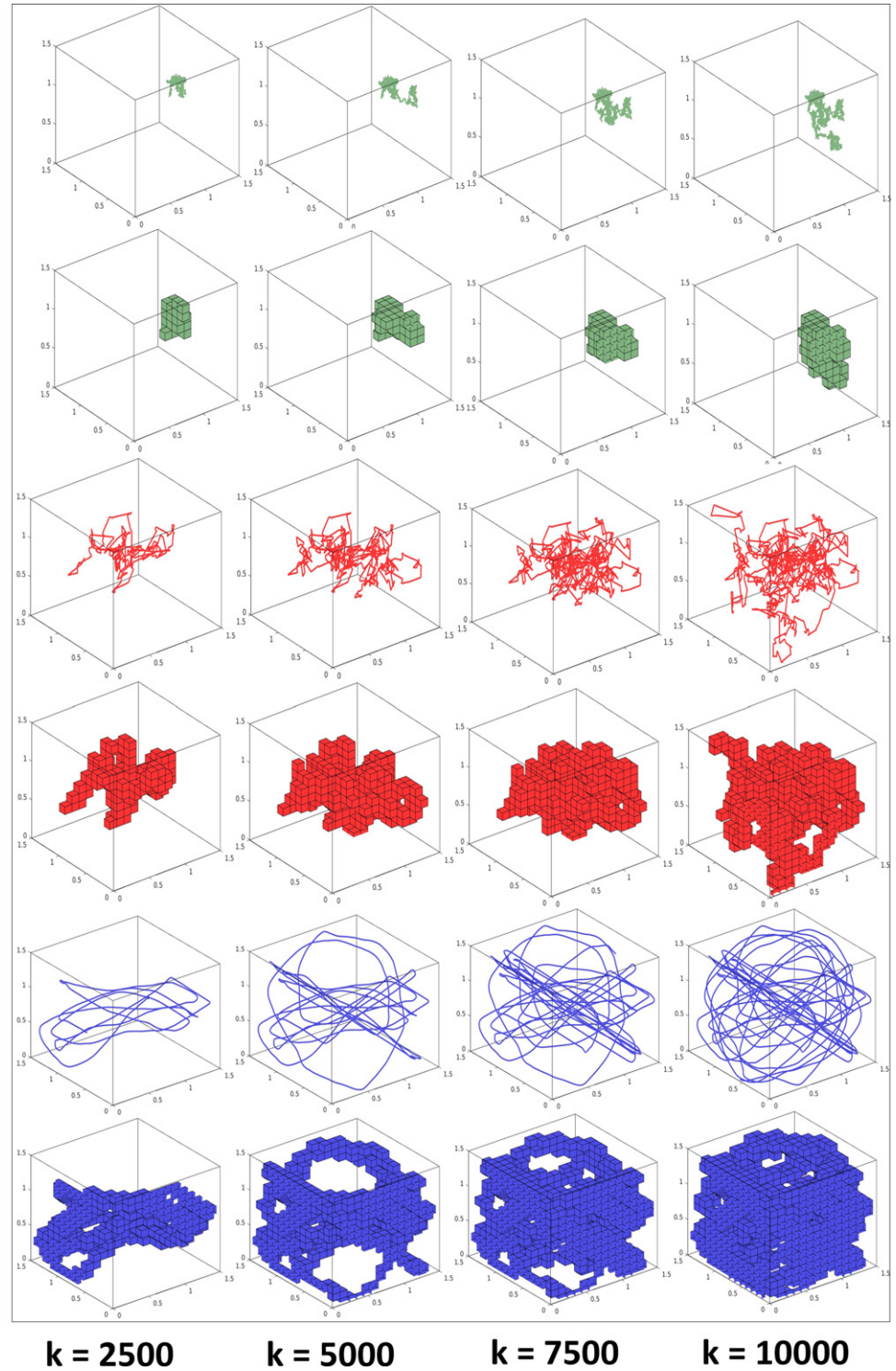
Simple
Random WalkCorrelated
Random WalkPersistent
Turning Walker

Figure 14. Figure comparing the area covered by an agent using three different coverage strategies in a 3D search-space.

strategy 2 are denoted by μ_1 and μ_2 , then we can state the null hypothesis as follows,

$$H_0 : \mu_1 = \mu_2. \quad (47)$$

We are interested in comparing the areas covered, we can state the one-sided alternative hypothesis as follows,

$$H_1 = \mu_1 > \mu_2. \quad (48)$$

We used a sample set containing Monte Carlo simulations for each of the strategies, the PTW, CRW,

and SRW. The sample-set contained the number of cells covered by each of the algorithms at the end of 10000 iterations (for 3D). For all three algorithms, the initial position and initial heading angle were kept the same so that the starting conditions affect the area coverage. We used a significance level, $\alpha = 0.05$ for all our results. The mean area covered by the 3D PTW model was significantly higher than the 3D CRW model ($Z = 7.7, p = 5.4 \times 10^{-15}$) and the 3D SRW model ($Z = 12.2, p = 1.3 \times 10^{-34}$). The results

Table 1. Results of the Wicoxon test in 3D.

Algorithm 1	Algorithm 2	Z-value	p-value	Result
PTW	CRW	7.7	5.40×10^{-15}	$\mu_1 > \mu_2$
PTW	SRW	12.2	1.30×10^{-34}	$\mu_1 > \mu_2$
CRW	SRW	12.2	1.3×10^{-34}	$\mu_1 > \mu_2$

are summarized in table 1 and can be given as follows,

$$\mathbb{E}(A_{PTW}) > \mathbb{E}(A_{CRW}) > \mathbb{E}(A_{RW}). \quad (49)$$

Hence, the mean area covered by the PTW model is greater than the mean area covered by the CRW model and the SRW models. All the aforementioned results in 3D can also be visualized in 2D and are given in appendix B.

7.2. Discussion

The 2D and 3D simulations along with the Mann Whitney statistical test results suggest that the PTW model achieves a faster rate of area coverage compared to the SRW and CRW strategies. These results are not surprising as the probability to move backwards in the PTW model is negligible while in the other two models, it is greater than zero. Hence, it is expected that the rate of area coverage for the PTW model will be faster than the other two models. Nevertheless, these experimental results are important to justify the faster rate of coverage exhibited by the PTW model. Additionally, these findings complement the theoretical result in theorem 6.1 which does not prove that the expected area covered by the PTW model is always greater than the expected area covered by the other two strategies.

It is also important to note here that the rate of coverage, R_A , using either of the aforementioned strategies will be much less as compared to other deterministic search methods, such as the lawnmower strategy [43]. The lawnmower strategy would require a lesser number of iterations to cover the entire grid. This is because these deterministic strategies rely on a localization service to keep track of the cells which the agent has visited and still has to visit. Based on this information, the agent moves in an optimal manner to ensure coverage of the entire search-space in less amount of time. This is not the case for the PTW model, which allows coverage of the entire search-space without any information about the agent's location. The lawnmower strategy also requires the agent to know the obstacle map of the search-space beforehand, like the shape of the search-space or the presence of any obstacles, and generally works well in a rectangular search-space. This is not the case in the PTW model where the search-space can be of any arbitrary shape and the agent does not need to know any information about its dynamics. Moreover, since the PTW incorporates a dynamic collision avoidance term, it allows the agent to avoid unanticipated obstacles (static or time-varying) while still being able to cover the search-space.

8. Conclusions and future work

To conclude, in this paper we proposed using the motion of fish as a coverage algorithm (in 2D and 3D) in GPS-denied environments. Compared to existing random search strategies, the proposed algorithm is able to cover the search-space at a faster rate. This is shown analytically by finding out the theoretical bounds on the backtracking probabilities of each model in addition to the simulation results. In future work, we aim to extend this model for multiple agent cases, which can again be inspired by the social interaction model of a school of fish.

Acknowledgments

The research work is supported by ONR Grants N00014-19-1-2556 and N00014-19-1-2266; AFOSR Grant FA9550-19-1-0283; NSF Grants CNS-1828678, S&AS-1849228 and GCR-1934836; NRL Grants N00173-17-1-G001 and N00173-19-P-1412 and NOAA Grant NA16NOS0120028.

Data availability statement

No new data were created or analysed in this study.

Appendix A. Analyzing the values of $\epsilon_1, \epsilon_2, \epsilon_3$

In this section, we provide an overview of the process that was followed to find the values of $\epsilon_1, \epsilon_2, \epsilon_3$ for the 2D transition matrix shown in figure 8(a). The same process can be used to find the values of the parameters in the 3D case as well.

As shown in section 4.4, ϵ_1 represents the largest possible backtracking probability associated with any given state s_i . Let $\mathbb{P}(s_i|s_j)$ denote the probability of transitioning into state s_i , given the previous state is s_j . Then, for a transition matrix generated using $\sigma = 515$, we can use 29 to find ϵ_1 as follows,

$$\epsilon_1 = \max \{ \mathbb{P}(s_1|s_5), \mathbb{P}(s_2|s_6), \mathbb{P}(s_3|s_7), \mathbb{P}(s_4|s_8), \mathbb{P}(s_5|s_1), \mathbb{P}(s_6|s_2), \mathbb{P}(s_7|s_3), \mathbb{P}(s_8|s_4) \}. \quad (A.1)$$

It can be seen in figure 8(a), that $\epsilon_1 = 0$ for $\sigma = 515$. Likewise, ϵ_2 is the minimum possible probability to stay in the same state. Again, we can use the definition in property (b) in section 4.4, to find the value of ϵ_2

Table B1. Wicoxon test results in 2D.

Algorithm 1	Algorithm 2	Z-value	p-value	Result
PTW	CRW	12.2	9.60×10^{-35}	$\mu_1 > \mu_2$
PTW	SRW	12.2	8.80×10^{-35}	$\mu_1 > \mu_2$
CRW	SRW	12.2	1.1×10^{-34}	$\mu_1 > \mu_2$

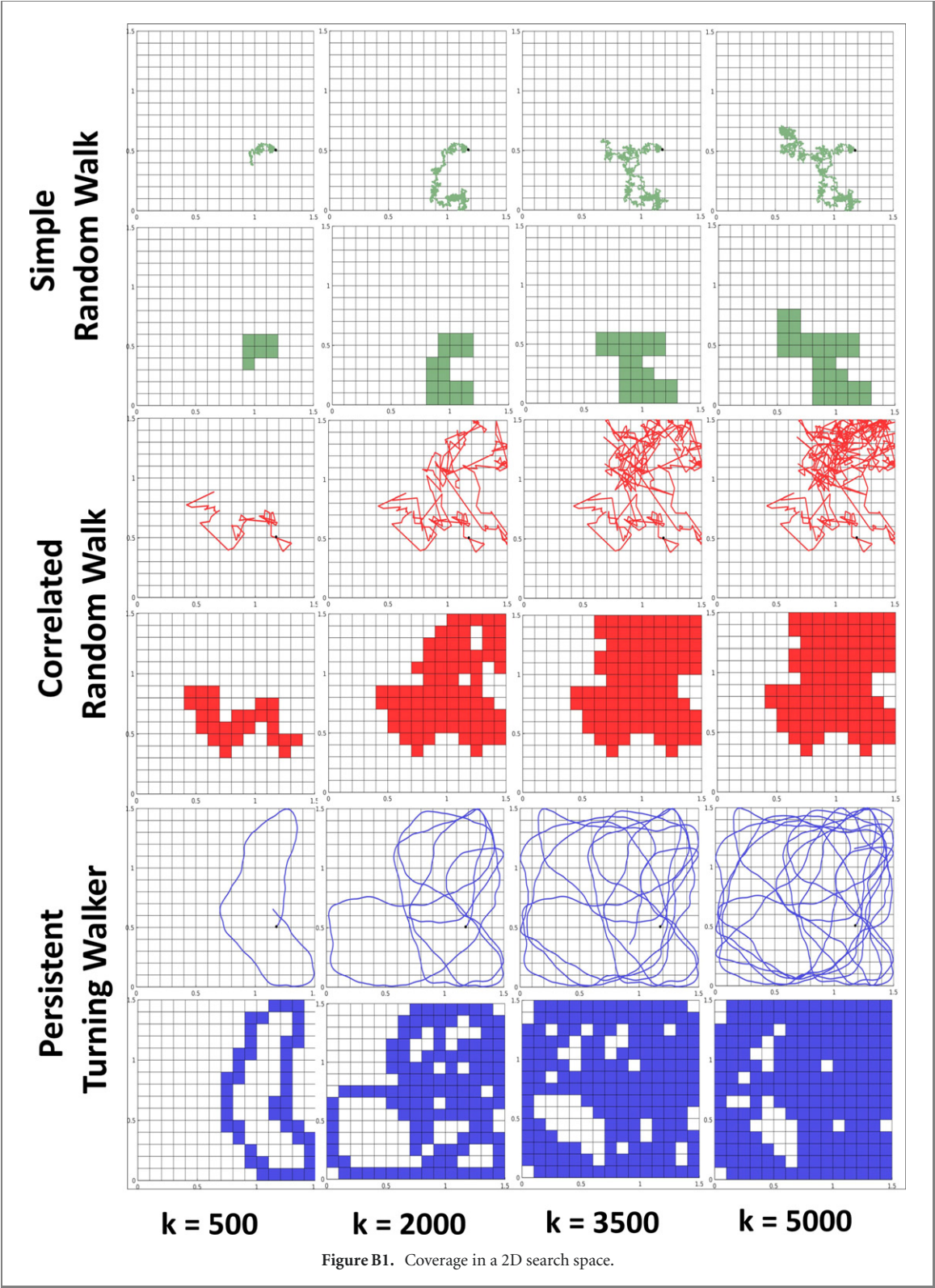
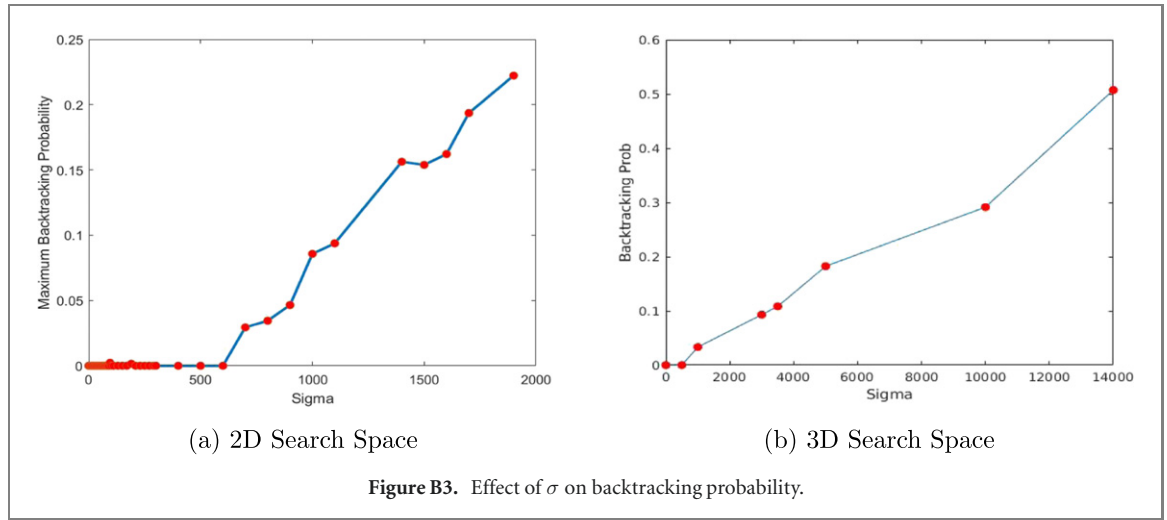
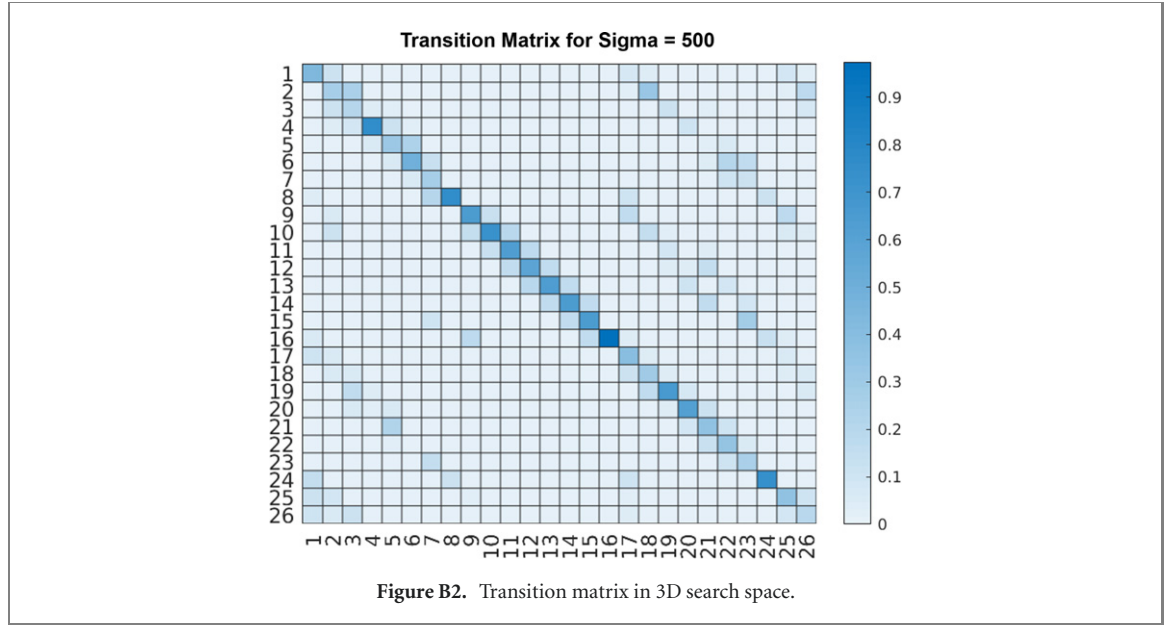


Figure B1. Coverage in a 2D search space.



as follows,

$$\epsilon_2 = \min \{ \mathbb{P}(s_1|s_1), \mathbb{P}(s_2|s_2), \mathbb{P}(s_3|s_3), \mathbb{P}(s_4|s_4), \mathbb{P}(s_5|s_5), \mathbb{P}(s_6|s_6), \mathbb{P}(s_7|s_7), \mathbb{P}(s_8|s_8) \} \quad (\text{A.2})$$

ϵ_2 corresponds to the minimum value which occurs on the diagonal of the transition matrix. For $\sigma = 515$, this corresponds to $\epsilon_2 = 0.38$. Lastly, for any given state s_i , ϵ_3 corresponds to the minimum probability to transition into states which are adjacent to the s_i 's neighboring states. Using property (c) in section 4.4, ϵ_3 can be given as follows,

$$\epsilon_3(s_1) = \min \{ \mathbb{P}(s_1|s_8), \mathbb{P}(s_1|s_7), \mathbb{P}(s_1|s_3), \mathbb{P}(s_1|s_2) \} \quad (\text{A.3})$$

$$\epsilon_3(s_2) = \min \{ \mathbb{P}(s_2|s_1), \mathbb{P}(s_2|s_3), \mathbb{P}(s_2|s_4), \mathbb{P}(s_2|s_8) \} \quad (\text{A.4})$$

$$\epsilon_3(s_3) = \min \{ \mathbb{P}(s_3|s_1), \mathbb{P}(s_3|s_2), \mathbb{P}(s_3|s_4), \mathbb{P}(s_3|s_5) \} \quad (\text{A.5})$$

$$\epsilon_3(s_4) = \min \{ \mathbb{P}(s_4|s_2), \mathbb{P}(s_4|s_3), \mathbb{P}(s_4|s_5), \mathbb{P}(s_4|s_6) \} \quad (\text{A.6})$$

$$\epsilon_3(s_5) = \min \{ \mathbb{P}(s_5|s_3), \mathbb{P}(s_5|s_4), \mathbb{P}(s_5|s_6), \mathbb{P}(s_5|s_7) \} \quad (\text{A.7})$$

$$\epsilon_3(s_6) = \min \{ \mathbb{P}(s_6|s_5), \mathbb{P}(s_6|s_4), \mathbb{P}(s_6|s_7), \mathbb{P}(s_6|s_8) \} \quad (\text{A.8})$$

$$\epsilon_3(s_7) = \min \{ \mathbb{P}(s_7|s_5), \mathbb{P}(s_7|s_6), \mathbb{P}(s_7|s_8), \mathbb{P}(s_7|s_1) \} \quad (\text{A.9})$$

$$\epsilon_3(s_8) = \min \{ \mathbb{P}(s_8|s_7), \mathbb{P}(s_8|s_6), \mathbb{P}(s_8|s_2), \mathbb{P}(s_8|s_1) \}. \quad (\text{A.10})$$

Then ϵ_3 can be given by,

$$\epsilon_3 = \min \{ \epsilon_3(s_1), \epsilon_3(s_2), \epsilon_3(s_3), \epsilon_3(s_4), \epsilon_3(s_5), \epsilon_3(s_6), \epsilon_3(s_7), \epsilon_3(s_8) \}. \quad (\text{A.11})$$

For the transition matrix given in figure 8(a), it can be seen that $\epsilon_3 = 0.02$.

Appendix B. Supplemental results

See table 2 and figures B1–B3.

ORCID iDs

Ayesha Khan  <https://orcid.org/0000-0003-4662-7493>

Fumin Zhang  <https://orcid.org/0000-0003-0053-4224>

References

- [1] Venkatesan S 2016 Auv for search & rescue at sea-an innovative approach 2016 *IEEE/OES Autonomous Underwater Vehicles (AUV)* (Piscataway, NJ: IEEE) pp 1–9
- [2] Niroui F, Zhang K, Kashino Z and Nejat G 2019 Deep reinforcement learning robot for search and rescue applications: exploration in unknown cluttered environments *IEEE Robot. Autom. Lett.* **4** 610–7
- [3] Matthews P and Greenspan S 2020 Robots in teams *Automation and Collaborative Robotics* (Berlin: Springer) pp 109–40
- [4] Jia Q, Xu H, Feng X, Gu H and Gao L 2019 Research on cooperative area search of multiple underwater robots based on the prediction of initial target information *Ocean Eng.* **172** 660–70
- [5] Andrade F A A, Hovenburg A, de Lima L N d, Rodin C D, Johansen T A, Storvold R, Correia C A M and Haddad D B 2019 Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control *Sensors* **19** 4067
- [6] Qi J, Song D, Shang H, Wang N, Hua C, Wu C, Qi X and Han J 2016 Search and rescue rotary-wing uav and its application to the Lushan ms 7.0 earthquake *J. Field Robot.* **33** 290–321
- [7] Sampedro C, Rodriguez-Ramos A, Bavl H, Carrio A, de la Puente P and Campoy P 2019 A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques *J. Intell. Robot. Syst.* **95** 601–27
- [8] Batalin M A and Sukhatme G S 2003 Efficient exploration without localization *Proc. ICRA'03. IEEE Int. Conf. Robotics and Automation* vol 2 (Piscataway, NJ: IEEE) pp 2714–9
- [9] Karapetyan N, Benson K, McKinney C, Taslakian P and Rekleitis I 2017 Efficient multi-robot coverage of a known environment 2017 *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (Piscataway, NJ: IEEE) pp 1846–52
- [10] Karapetyan N, Moulton J, Lewis J S, Li A Q, O'Kane J M and Rekleitis I 2018 Multi-robot dubins coverage with autonomous surface vehicles *IEEE Int. Conf. Robotics and Automation (ICRA)* (Piscataway, NJ: IEEE) pp 2373–9
- [11] Giuggioli L, Arye I, Heiblum Robles A and Kaminka G A 2018 From ants to birds: a novel bio-inspired approach to online area coverage *Distributed Autonomous Robotic Systems* (Berlin: Springer) pp 31–43
- [12] da Rosa R, Aurelio Wehrmeister M, Brito T, Lima J L and Pereira A I P N 2020 Honeycomb map: a bioinspired topological map for indoor search and rescue unmanned aerial vehicles *Sensors* **20** 907
- [13] Izumi S, Azuma S-I and Sugie T 2020 Multi-robot control inspired by bacterial chemotaxis: coverage and rendezvous via networking of chemotaxis controllers *IEEE Access* **8** 124172–84
- [14] Luo C, Jan G E, Chu Z, Li X and Aceves-Fernandez M 2018 Biologically inspired intelligence with applications on robot navigation *Artificial Intelligence-Emerging Trends and Applications* (Rijeka: InTech)
- [15] Bartumeus F, Da Luz M G E, Viswanathan G M and Catalan J 2005 Animal search strategies: a quantitative random-walk analysis *Ecology* **86** 3078–87
- [16] Pasternak Z, Bartumeus F and Grasso F W 2009 Lévy-taxis: a novel search strategy for finding odor plumes in turbulent flow-dominated environments *J. Phys. A: Math. Theor.* **42** 434010
- [17] Nurzaman S G, Matsumoto Y, Nakamura Y, Koizumi S and Ishiguro H 2009 Biologically inspired adaptive mobile robot search with and without gradient sensing *IROS 2009 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (Piscataway, NJ: IEEE) pp 142–7
- [18] Nurzaman S G, Matsumoto Y, Nakamura Y, Koizumi S and Ishiguro H 2009 Yuragi-based adaptive searching behavior in mobile robot: from bacterial chemotaxis to Levy walk *ROBIO 2008. IEEE Int. Conf. Robotics and Biomimetics* (Piscataway, NJ: IEEE) pp 806–11
- [19] Puljiz D, Varga M and Bogdan S 2012 Stochastic search strategies in 2D using agents with limited perception *IFAC Proc. Vol.* **45** 650–4
- [20] Rañó I and Santos J A 2017 A biologically inspired controller to solve the coverage problem in robotics *Bioinspiration Biomimetics* **12** 035002
- [21] Bentz W and Panagou D 2016 An energy-aware redistribution method for multi-agent dynamic coverage networks *IEEE 55th Conf. Decision and Control (CDC)* (Piscataway, NJ: IEEE) pp 2644–51
- [22] Heng L, Gotovos A, Krause A and Pollefeys M 2015 Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments 2015 *IEEE Int. Conf. Robotics and Automation (ICRA)* (Piscataway, NJ: IEEE) pp 1071–8
- [23] Siligardi L, Panerati J, Kaufmann M, Minelli M, Ghedini C, Beltrame G and Sabattini L 2019 Robust area coverage with connectivity maintenance 2019 *Int. Conf. Robotics and Automation (ICRA)* (Piscataway, NJ: IEEE) pp 2202–8
- [24] Zhong J, Cheng H, He L and Ouyang F 2019 Decentralized full coverage of unknown areas by multiple robots with limited visibility sensing *IEEE Robot. Autom. Lett.* **4** 338–45
- [25] Teruel E, Aragues R and López-Nicolás G 2019 A distributed robot swarm control for dynamic region coverage *Robot. Auton. Syst.* **119** 51–63
- [26] Galceran E and Carreras M 2013 A survey on coverage path planning for robotics *Robot. Auton. Syst.* **61** 1258–76
- [27] Gilhuly B and Smith S L 2019 Robotic coverage for continuous mapping ahead of a moving vehicle *IEEE 58th Conf. Decision and Control (CDC)* (Piscataway, NJ: IEEE) pp 8224–9
- [28] Lee T-K, Baek S-H, Oh S-Y and Choi Y-H 2010 Complete coverage algorithm based on linked smooth spiral paths for mobile robots 2010 *11th Int. Conf. Control Automation Robotics & Vision* (Piscataway, NJ: IEEE) pp 609–14
- [29] Fish F E 2020 Advantages of aquatic animals as models for bio-inspired drones over present AUV technology *Bioinspiration Biomimetics* **15** 025001
- [30] Gautrais J, Jost C, Soria M, Campo A, Motsch S, Fournier R, Blanco S and Theraulaz G 2009 Analyzing fish movement as a persistent turning walker *J. Math. Biol.* **58** 429–45
- [31] Gautrais J, Ginelli F, Fournier R, Blanco S, Soria M, Chaté H and Theraulaz G 2012 Deciphering interactions in moving animal groups *PLoS Comput. Biol.* **8** e1002678
- [32] Degond P and Motsch S 2008 Large scale dynamics of the persistent turning walker model of fish behavior *J. Stat. Phys.* **131** 989–1021
- [33] Khan A, Mishra V and Zhang F 2016 Bio inspired source seeking: a hybrid speeding up and slowing down algorithm 2016 *IEEE 55th Conf. Decision and Control (CDC)* (Piscataway, NJ: IEEE) pp 4334–9
- [34] Grewal Jasleen K, Krzywinski M and Altman N 2019 Markov models-Markov chains *Nature Methods* **16** 663–64

- [35] Hong L and Xu J 1999 Crises and chaotic transients studied by the generalized cell mapping digraph method *Phys. Lett. A* **262** 361–75
- [36] Yue X, Xu W, Zhang Y and Du L 2018 Analysis of global properties for dynamical systems by a modified digraph cell mapping method *Chaos Solitons Fractals* **111** 206–12
- [37] Clarke E M Jr, Grumberg O, Kroening D, Peled D and Veith H 2018 *Model checking* (Cambridge, MA: MIT Press)
- [38] Gerth R, Peled D, Vardi M Y and Wolper P 1995 Simple on-the-fly automatic verification of linear temporal logic *Int. Conf. Protocol Specification, Testing and Verification* (Berlin: Springer) pp 3–18
- [39] Renshaw E and Henderson R 1981 The correlated random walk *J. Appl. Probab.* **18** 403–14
- [40] Sun J and Hsu C t 1990 The generalized cell mapping method in nonlinear random vibration based upon short-time Gaussian approximation
- [41] Anderson T W and Goodman L A 1957 Statistical inference about Markov chains *Ann. Math. Stat.* **28** 89–110
- [42] Wilcoxon F, Katti S and Wilcox R A 1963 *Critical Values and Probability Levels for the Wilcoxon Rank Sum Test and the Wilcoxon Signed Rank Test* vol 1 (Pearl River, NY: American Cyanamid)
- [43] Hubenko A, Fonoberov V A, Mathew G and Mezic I 2011 Multiscale adaptive search *IEEE Trans. Syst. Man Cybern. B* **41** 1076–87