2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Bio Inspired Source Seeking: a Hybrid Speeding Up and Slowing Down Algorithm*

Ayesha Khan[1], Vivek Mishra[1] and Fumin Zhang[1]

***Abstract*— A novel bio-inspired strategy, the Hybrid Speeding Up Slowing Down (Hybrid SUSD) strategy, is introduced to achieve distributed control of a multi-agent system for the localization of multiple sources in a search space. Hybrid SUSD switches between bio-inspired exploration algorithms and exploitation algorithms. The exploration algorithms provide coverage of the workspace with non-zero probability. The exploitation algorithms leverage the SUSD strategy for source seeking without explicit gradient estimation. Conditions for switching between exploration and exploitation are developed based on measurements taken by an agent and the number of neighbors an agent may have. Given a confined search space, the convergence of the hybrid SUSD to locate a source is rigorously justified. Simulation results confirm that the strategy allows each agent to converge to one of the source locations. The Hybrid SUSD may be used as a distributed optimization algorithm that is able to find all minima of a function over a confined search space.**

I. INTRODUCTION

Mobile sensing agents can be deployed to measure an unknown field around a source. These measurements can represent temperature around a heat source, chemical concentration near a chemical plume etc. When the motion of agents is directed towards the local minima or maxima of a field, we refer to this as source seeking. Various approaches have been used in literature which devise source seeking strategies for both single and multi-agent systems [1][2].

Most source seeking algorithms assume that the agents have access to the gradient of a field [3]. Others rely on estimating the gradient either by taking measurements from multiple sensors or using a collaborative strategy [4]. Either way, these approaches assume that an agent has a reliable measurement of the field. Moreover, most of these strategies are designed assuming there is only a single source located in the field and are either applicable for multi-agent or single agent systems but not both [1][4]. This motivates us to develop a strategy for seeking multiple sources in a non-convex field, enabling every agent to reach one of the sources using an individual or collaborative approach.

We take inspiration from the collective motion in animal groups to devise such a strategy. One of the reasons being that animals are able to find sources in the presence of uncertainties in the environment and without any localization service. For instance, the algorithm by Wu et. al [5][6] takes inspiration from fish school behavior where they proposed a multi-agent source seeking algorithm that does not require explicit gradient estimation of the scalar field. However, this algorithm requires an agent to have measurements at all times and cannot locate multiple sources.

Furthermore, an efficient strategy to locate multiple sources requires a framework which allows exploration in the absence of reliable measurements and exploitation in the presence of reliable measurements. Apart from the various optimal coverage patterns [7], common exploration approaches also involve random walks [8]. Motion models of most animals are generally derived from simple random walk processes, for example, Theraulaz et. al in his recent work characterizes fish trajectories through a Persistent Turning Walker model (PTW) [9][10]. PTW is different from Persistent Random Walk (PRW) since it incorporates the correlation between subsequent angular velocities instead of the correlation between subsequent headings. Since these models do not require any localization service, they can be deployed to design exploration strategies where constant localization of the agent is difficult.

In this paper, we devise a novel bio-inspired source seeking strategy, which implements distributed control of a multi-agent system to localize multiple sources. Agents use an exploration strategy based on [9] to search the work space until a reliable measurement of a scalar field is encountered. At this stage, the agent adopts an exploitation strategy depending on whether it has neighbors or not. The allocation of neighbors is based on a simple distance metric. If an agent has neighbors, it adapts the SUSD strategy which enables the team to seek a source without explicit gradient estimation [5]. We show that by using Hybrid SUSD, a group of agents is always able to explore sources at multiple sites by using a collaborative or individual approach while remaining confined inside the region of interest. Moreover, this algorithm can be used for non-convex and discontinuous scalar fields to localize multiple sources using a distributed approach. We also show the convergence of the algorithm and verify the same using simulation techniques.

II. PROBLEM FORMULATION

Let there be a group of N agents trying to localize K sources in a two-dimensional search space Ω, such that Ω is defined as follows:

$$\Omega = \{\mathbf{r} : -\mathbf{c} \leq (\mathbf{r} - \mathbf{r}_{wc}) \leq \mathbf{c}, \mathbf{c} > 0\} \tag{1}$$

where $\mathbf{r}$ is a position vector and $\mathbf{r}_{wc}$ are the center coordinates of the search space. $\mathbf{c}$ represents any positive constant

*The research work is supported by ONR grants N00014-10-10712 (YIP) and N00014-14-1-0635; and NSF grants OCE-1032285, IIS-1319874, and CMMI-1436284.

[1]Ayesha Khan, [1]Vivek Mishra and [1]Fumin Zhang are with the College of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA - 30332 {ayeshakhan,vmishra9,fumin}@gatech.edu

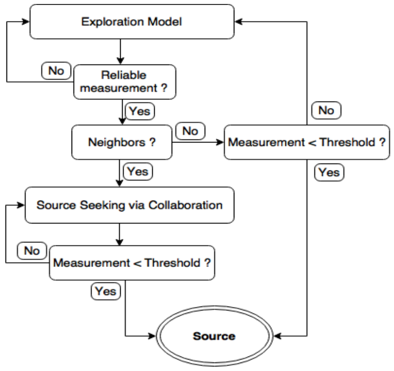


Fig. 1: Block digram showing Hybrid SUSD strategy

which bounds the search space.

Let each source $k \in \{1, ..., K\}$ be surrounded by a scalar field $z_k(\mathbf{r}), \forall \mathbf{r} \in \Omega$ denoting a measurement of light, chemical concentration or temperature. The following assumptions holds for $z_k(\mathbf{r})$:

Assumption 1: 1) *The field $z_k(\mathbf{r})$ is time-invariant.*

2) *$z_k(\mathbf{r})$ is bounded by a minimum value z_{min} and a maximum value z_{max} such that $z_{min} \leq z(\mathbf{r}) \leq z_{max}$ and $z_{max}, z_{min} > 0$,*
3) *Each source k is surrounded by a closed and connected region Ω_k such that $z_k(\mathbf{r})$ is considered unreliable for all $\mathbf{r} \notin \Omega_k$.*
4) *The minima of the field $z_k(\mathbf{r})$ is at the location of the source k and is denoted by $\mathbf{r}_{s,k}$.*
5) *The sources are separable, that is, for a source k and a source l, where $k \neq l$, the following condition holds:*

$$\Omega_k \cap \Omega_l = \phi \tag{2}$$

Given the above assumptions, we aim to design a source seeking algorithm which enables each agent $i \in \{1, ..., N\}$, to use a collaborative or individual approach in order to reach one of the sources present in the scalar field.

III. HYBRID SUSD STRATEGY

Hybrid SUSD is a bio-inspired algorithm which can be used by a multi-agent system for the localization of one or more sources in a given scalar field. It consists of an exploration strategy which enables an agent to explore the search space whether it has neighbors or not, until it encounters reliable measurements. At this stage, the agent switches to a multi-agent source seeking strategy only if it has neighbors, otherwise it remains in the exploration mode. The algorithm terminates when all the agents are able to reach one of the sources in the scalar field. A block diagram of the algorithm is shown in Fig.1.

A. Neighborhood Selection

In order to define the neighborhood of an agent i, we make use of a distance threshold d^{th}. If we define the neighborhood of an agent i by N_i, where $N_i \subset \{1, 2, ...N\}/\{i\}$, then an agent $j \in N_i$ if $d_{ij} \leq d^{th}$, where $d^{th} > 0$.

B. Exploration using Steering Control

To establish the behavior of agents in the absence of any reliable measurement, we take inspiration from the fish motion model by Theraulaz et. al [9][10]. This model is characterized by a constant swimming speed v and a correlation between angular velocities, w at consecutive instances of time.

Taking $\mathbf{r}_i$ as the position and ϕ_i as the heading of fish i, the following dynamical system represents the fish motion,

$$d\mathbf{r}_i = v \begin{bmatrix} cos\phi_i \\ sin\phi_i \end{bmatrix} dt \tag{3}$$

$$d\phi_i = w_i(t)dt \tag{4}$$

$$dw_i(t) = -v \left[\frac{1}{\xi}(w_i(t) - w_i^*(t))dt - \sigma dW \right] \tag{5}$$

where, σdW is a Weiner process of variance σ^2 and ξ is a constant. The target angular w^* velocity given by,

$$w_i^*(t) = \hat{k}_W \frac{sgn(\phi_{iW})}{\tau_{iW}} + \frac{1}{|N_i|} \sum_{j \in N_i} (k_p d_{ij} sin\theta_{ij} + \hat{k}_v v sin\phi_{ij}) \tag{6}$$

where, ϕ_{iW} denotes the angle between the heading of a fish i and the point of impact with respect to the normal of the wall; τ_{iW} is the time it would take for the fish i to hit the wall; d_{ij} is the distance between fish i and fish j; θ_{ij} is the bearing angle of fish j relative to the velocity of fish i; ϕ_{ij} denotes the relative heading of fish j compared to fish i; and $\hat{k}_W, k_p$ and $\hat{k}_v$ are constants. Lastly, N_i denotes the neighborhood of i as described in the previous subsection.

This steering model can be deployed as an exploration strategy for a group of N agents. An agent continues to explore the search space until it gets a reliable measurement of the source in which case it switches its strategy to that of source seeking depending on whether it has neighbors or not. To identify neighbors we use the neighborhood selection criterion mentioned in Section III-A .

C. Exploitation using SUSD strategy

For any source k, an agent i will switch to the exploitation strategy if the following two conditions are satisfied,
1) $\mathbf{r}_i \in \Omega_k$
2) $\mathbf{r}_j \in \Omega_k$ such that $j \in N_i$

1) Multi-agent strategy: Inspired by fish behavior, previous works [5][6] develop a source-seeking algorithm for a group of agents with no explicit gradient estimation of a scalar field. The model is termed as SUSD. The algorithm chooses a baseline for a group of agents and decomposes the velocity of each agent into two parts; the first part, $\mathbf{v}_i^{\perp}$ is chosen to be proportional to the measurements of the scalar field while the second part $\mathbf{v}_i^{\|}$ is designed to keep the inter-agent distance constant at all times. If $\mathbf{q}$ is a baseline used to achieve the decomposition of $\mathbf{v}$ and θ^q is the angle between a stated inertial frame and $\mathbf{q}$, then the velocity $\mathbf{v}_i$ of an agent i is given by the following equations:

$$\mathbf{v}_i^{\perp} = (kz(\mathbf{r}_i) + C) \begin{bmatrix} -sin\theta^q \\ cos\theta^q \end{bmatrix} \tag{7}$$

$$\mathbf{v}_i^{\|} = k_p \sum_{j \in N_i} ((\mathbf{r}_i - \mathbf{r}_j) \cdot \mathbf{q} - a_{ij}^0) \tag{8}$$

where, a_{ij}^0 represents the desired inter-agent distance between agent i and j while k, k_p and C are constants. An agent i looks for its closest agents along direction $\mathbf{q}$ and N_i denotes those agents.

Using (7-8) enables a team of agents to seek a source without explicit estimation of a gradient field. We incorporate this model in hybrid SUSD to enable source seeking for multi-agent systems. Moreover, using the neighborhood selection rule given in III-A gives the possibility of multiple teams to be formed at the same time to seek a single or multiple sources. Hence, this method allows seeking multiple sources in non-convex scalar fields.

2) Single agent: If $\mathbf{r}_i \in \Omega_k$ but there is no other agent j such that $j \in N_i$, then agent i is unable to estimate the gradient of the scalar field by itself. Hence, it keeps exploring the search space using the motion model (3-5) until it finds a neighbor or the stopping condition is met.

D. Stopping Condition

We define a scalar value $\delta > 0$ such that if for an agent i, $z(\mathbf{r}_i) \leq \delta$, we can assume that the agent i has reached the origin of the source and $v_i = 0$. Therefore, the algorithm terminates when $z(\mathbf{r}_i) \leq \delta$ for all i where $i = 1, 2.....N$.

Algorithm 1: Hybrid SUSD Strategy

Data: $z(\mathbf{r}_i(t)), \phi_i(t), \mathbf{r}_i(t)$
Result: $\phi_i(t+1), \mathbf{r}_i(t+1)$
Initialization $\phi_i(t_0), \mathbf{r}_i(t_0), w_i(t_0)$
while $z(\boldsymbol{r}_i(t)) \geq \delta$ **do**
  for $i = 1 : N$ **do**
    if $\boldsymbol{r}_i(t) \in \Omega$ **then**
      if $(\exists j \in N_i) \cap (\boldsymbol{r}_j(t) \in \Omega)$ **then**
        $v_i^{\perp}(t) = kz(\mathbf{r}_i(t)) + C$
        $v_i^{\parallel}(t) = kp \sum_{j \in N_i} ((\mathbf{r}_i(t) - \mathbf{r}_j(t)).\mathbf{q} - a_{i,j}^0)$
        $\mathbf{r}_i(t) = \mathbf{r}_i(t-1) + \begin{bmatrix} v_i^{\parallel}(t)\cos\left(\theta_i^q(t)\right) \\ v_i^{\perp}(t)\sin\left(\theta_i^q(t)\right) \end{bmatrix}$
        $\phi_i(t) =$
        $\arctan\left(v_i^{\perp}(t)\sin\theta_i^q(t) / v_i^{\parallel}(t)\cos\theta_i^q(t)\right)$
      else
        $\phi_i(t) = \phi_i(t-1) + w_i(t-1)dt$
        $w_i^*(t) = \hat{k}_W \frac{sgn(\phi_{iW})}{\tau_{iW}} +$
        $\frac{1}{N_i} \sum_{j \in V_i} (k_p d_{ij} sin\theta_{ij} + \hat{k}_v v sin\phi_{ij})$
        $dw_i(t) = -v[\frac{dt}{\xi}(w_i(t) - w_i^*(t)) - \sigma dW$
        $\mathbf{r}_i(t) = \mathbf{r}_i(t-1) + v \begin{bmatrix} \cos\left(\phi_i(t)\right) \\ \sin\left(\phi_i(t)\right) \end{bmatrix} dt$
      end
    else
      $\phi_i(t) = \phi_i(t-1) + w_i(t-1)dt$
      $w_i^*(t) = \hat{k}_W \frac{sgn(\phi_{iW})}{\tau_{iW}} +$
      $\frac{1}{N_i} \sum_{j \in V_i} (k_p d_{ij} sin\theta_{ij} + \hat{k}_v v sin\phi_{ij})$
      $dw_i(t) = -v[\frac{dt}{\xi}(w_i(t) - w_i^*(t)) - \sigma dW$
      $\mathbf{r}_i(t) = \mathbf{r}_i(t-1) + v \begin{bmatrix} \cos\left(\phi_i(t)\right) \\ \sin\left(\phi_i(t)\right) \end{bmatrix} dt$
    end
  end
end

IV. TERMINATION OF HYBRID SUSD

Given an agent i such that $\mathbf{r}_i \notin \Omega_k$, it can be proved that using motion model (3-5), agent i reaches a location such that $\mathbf{r}_i \in \Omega_k$ or $z(\mathbf{r}_i) \leq \delta$ with a non-zero probability. If $\mathbf{r}_i \in \Omega_k$, then it can be shown that using the SUSD strategy, an agent reaches the source such that $z(\mathbf{r}_i) \leq \delta$. Thus, given any initial position, an agent i will always be able to reach one of the sources with a non-zero probability. This denotes the termination criterion for the Hybrid SUSD algorithm.

A. Markov Chain Representation

We allow an agent to move in the search space using the motion model (3-5). For simplicity we consider a rectangular search space, but the results can also be extended for a circular search space.

1) Discretization of Search Space: We discretize the search space into an $n \times n$ grid such that there are a total of $N = n^2$ cells in the search space where $n > 2$ (see Fig. 2a). We divide the grid cells into three groups, namely, corner cells (cells 1, 2, 3 and 4 in Fig. 2a), border cells (cells 5, 6, 7 and 8 in Fig. 2a) and middle cells (cell 9 in Fig. 2a). An $n \times n$ grid will always have 4 corner, $(4n-8)$ border and $(n-2)^2$ middle cells for $n \geq 2$. Moreover, let N_{ci} be defined as the neighborhood of cell c_i such that N_{ci} contains all cells adjacent to c_i. For example in Fig. 2a, $N_{c7} = \{c_4, c_8, c_9, c_5, c_2\}$.

2) Discretized Search Model: If the size of each grid cell is $\triangle n \times \triangle n$, then for any $\triangle n$ such that $\triangle n < vdt$, the motion model (3-5) after discretization satisfies the following assumptions for any cell c_i:

Assumption 2: $P(\mathbf{r}_{t+dt} \in c_i | \mathbf{r}_t \in c_i) = 0$.

Assumption 3: For any grid cell $c_j \in N_{ci}$, $P(\mathbf{r}_{t+dt} \in c_j | \mathbf{r}_t \in c_i, \mathbf{r}_{t-dt} \in c_j) = 0$.

Assumption 4: For any two grid cells c_j, c_k where $c_j, c_k \in N_{ci}$ and $c_j \neq c_k$, $P(\mathbf{r}_{t+dt} \in c_k | \mathbf{r}_t \in c_i, \mathbf{r}_{t-dt} \in c_j) > 0$.

We used Monte Carlo simulations in order to analyze the probability distribution generated by the motion model (3-5). The initial position of an agent i, was drawn from a uniform distribution over the entire search space and can be denoted by $\mathbf{r}_{i0}$. The heading of the agent i, ϕ_{i0} was also initialized from a uniform distribution $[0, 2\pi]$. This was done for N agents. The trajectory of each agent i was integrated over dt by using the following:

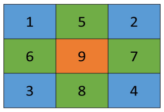
$$d\mathbf{r}_i = v \begin{bmatrix} cos\phi_{i0} \\ sin\phi_{i0} \end{bmatrix} dt \tag{9}$$

$$\mathbf{r}_i = d\mathbf{r}_i + \mathbf{r}_{i0} \tag{10}$$

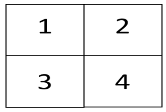
$$\mathbf{r}_{i0} = \mathbf{r}_i \tag{11}$$

The rest of the parameters such as w_i and ϕ_i were updated simultaneously according to the motion model (3-5). The simulations were performed several times and it was observed that the Assumptions (2-4) satisfy the model (3-5).

Note that the motion model (3-5) is different from PRW. In PRW, there is a positive probability of the agent to move to any of its neighboring cells, N_{ci}. On the other hand, an agent moving according to the motion model (3-5) has an



(a) Grid cells



(b) Kernel

Fig. 2: Search Space discretization

$$
\begin{array}{cc}
 & \begin{array}{cccccccccccc} 1_2 & 1_3 & 1_4 & 2_1 & 2_3 & 2_4 & 3_1 & 3_2 & 3_4 & 4_1 & 4_2 & 4_3 \end{array} \\
\begin{array}{c} 1_2 \\ 1_3 \\ 1_4 \\ 2_1 \\ 2_3 \\ 2_4 \\ 3_1 \\ 3_2 \\ 3_4 \\ 4_1 \\ 4_2 \\ 4_3 \end{array} &
\begin{bmatrix}
0 & 0 & 0 & 0 & p_9 & p_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{15} & p_{17} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{21} & p_{23} \\
0 & p_3 & p_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & p_{13} & 0 & p_{18} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{19} & 0 & p_{24} \\
p_1 & 0 & p_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & p_7 & 0 & p_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{20} & p_{22} & 0 \\
p_2 & p_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & p_8 & p_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & p_{14} & p_{16} & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

Fig. 3: State Transition Matrix of the motif, A. $1 < p_i < 0$ where $i = \{1, 2...24\}$. Each row represents the next state while each column represents the present state. For e.g. p_2 is the probability of going from state 1_2 to state 4_1.

additional constraint that the agent cannot immediately go back to the cell it came from. This is because, as seen in the simulations, the angular velocity $w_i(t)$ at any time is never large enough to allow a change in the heading angle equal to $\pm\pi$.

3) Markov Chain Representation: Based on the probability distribution of neighboring cells, we can represent the discretized motion model (3-5) by a finite, discrete Markov Chain $\mathcal{M}$ [11]. The states of $\mathcal{M}$ are defined using the above mentioned probabilities. We know that the movement of an agent to any cell c_i at $t + dt$ depends in which cells the agent was present at t and $t - dt$. Using the definition of N_{ci} as mentioned above, each grid cell c_i is associated with the number of states equal to the number of elements in N_{ci}. This means that each corner cell is associated with 3 states in the Markov chain $\mathcal{M}$. For example, in Fig. 2a, the corner cell c_1 is associated with three states namely, 1_5, 1_6 and 1_9. 1_5 represents the state that the agent is currently present in c_1, and it came to c_1 from c_5. Similarly, 1_6 represents the state that the agent is in c_1 and it came to c_1 from c_6. In the same way all the border cells are associated with 5 states and all middle cells have 8 states. Using this state model, we can construct the Markov chain $\mathcal{M}$ given by,

$$X_{k+1} = PX_k \tag{12}$$

where $X_k = [p_1, p_2, ...p_{N_T}]^T$ contains the probabilities of staying in states $[x_1, x_2, ...x_{N_T}]$ where x_i represents the states 1_6, 1_5, 2_4 etc. and P is the state transition matrix. We can perform one step analysis [11] of $\mathcal{M}$ to find the hitting probability of a given state from any initial state. The hitting probability of a state x_i in $\mathcal{M}$ is defined as the probability of reaching state x_i from any other state x_j and can be given as:

$$H_i^P = (I - Q_i^P)^{-1} R_i^P \tag{13}$$

where I and Q_i^P are both square matrices of size $(N_T - 1) \times (N_T - 1)$. I is an identity matrix while Q_i^p is formed by removing the i^{th} row and i^{th} column from the transition matrix P. R_i^P is the i^{th} column of P without the i^{th} element and therefore has $(N_T - 1)$ elements. Each j^{th} element of H_i represents the probability of reaching the state x_i given the initial state x_j.

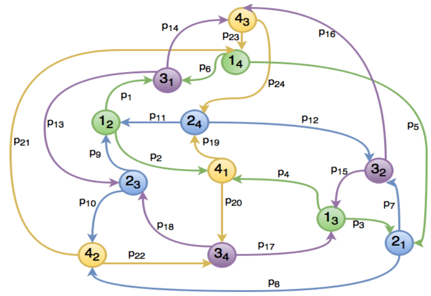


Fig. 4: The digraph $\mathcal{G}^A$ of the Markov chain $\mathcal{M}^A$

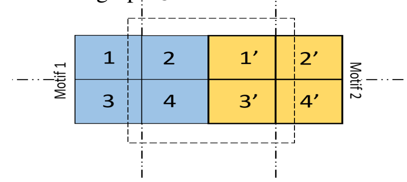


Fig. 5: Neighboring motifs

Let the associated digraph of the Markov chain $\mathcal{M}$ be denoted by G^P, where G^P has a set of nodes N_G where $N_G = 1, 2, ..., n_g$ and a set of edges E_G. Then, each node corresponds to a state of $\mathcal{M}$, and G^P contains an edge $(i, j) \in E_G$ if and only if $p_{ij} > 0$ [12]. G^P is said to be strongly connected if for any nodes u and v where $u, v \in N_G$, there exists a directed path from u to v and a directed path from v to u.

Lemma 4.1: The hitting probability of any state x_i from any other state x_j is greater than zero (i.e. $H_i^P > 0 \; \forall i$) if and only if the graph $\mathcal{G}^P$ is strongly connected.

Proof: From Section 17.2.2 in [12] we know that the directed graph $\mathcal{G}^P$ is strongly connected if and only if its associated Markov chain $\mathcal{M}$ is irreducible. Also, we know that the Markov chain $\mathcal{M}$ is irreducible, if and only if the hitting probability of any state x_j from any other state x_i is greater than zero. ■

4) Coverage Probability: We define a search space motif of $n = 2 \times 2$ cells which can be used to build the entire search space (Fig. 2b). Each motif is associated with a finite Markov chain $\mathcal{M}^A$ where $\mathcal{M}^A$ can be be given as:

$$X_{k+1} = AX_k \tag{14}$$

A is the state transition matrix associated with the Markov chain $\mathcal{M}^A$ and is given in Fig. 3. Let G^A be the associated digraph of $\mathcal{M}^A$ as shown in Fig. 4.

Lemma 4.2: The directed graph $\mathcal{G}^A$ derived from the Markov chain $\mathcal{M}^A$ of a motif is a strongly connected graph.

Proof: The hitting probability of any state j of a motif can be given as:

$$H_j^A = (I - Q_j^A)^{-1} R_j^A \tag{15}$$

Since all the elements of Q_j^A are less than 1 therefore we can expand $(I - Q_j^A)^{-1}$ using Neumann series and (15) can

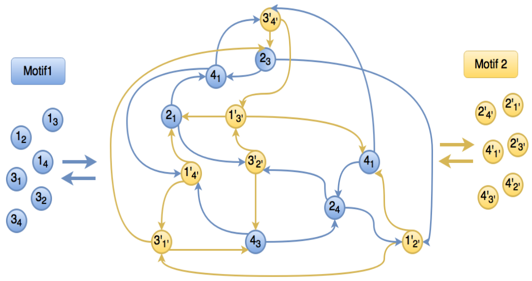


Fig. 6: The digraph formed from two neighboring motifs in Fig. 5

be written as:

$$H_j^A = R_j^A + Q_j^A R_j^A + \left(Q_j^A\right)^2 R_j^A + \left(Q_j^A\right)^3 R_j^A + \ldots\ldots \quad (16)$$

Since all the elements of Q_j^A and R_j^A are greater than or equal to zero $\forall j$, therefore we have:

$$H_j^A \geq R_j^A + Q_j^A R_j^A + \left(Q_j^A\right)^2 R_j^A + \left(Q_j^A\right)^3 R_j^A \quad (17)$$

Using transition matrix A (see Fig. 3) we calculated H_j^A using (17) and it was found that each element of H_j^A is greater than zero $\forall j$. Thus, using Lemma 4.1, we can say that the graph associated with a motif $\mathcal{G}^A$ is a strongly connected graph. ■

Theorem 4.1: If an agent starts from a cell in the discretized search space, then it can reach any other cell in the search space with a non-zero probability where Assumptions (2-4) are satisfied.

Proof: Let the search space contain a total of $N = 2n \times 2n$ cells and the Markov chain associated with the motion model be $\mathcal{G}^P$. The search space can be formed by a total of n^2 motifs by stacking the motifs next to each other as shown in Fig. 5. Let each motif k_i be associated with a Markov chain $\mathcal{M}_i$. From Lemma 4.2, each Markov chain $\mathcal{M}_i$ can be represented by a strongly connected graph $\mathcal{G}_i$.

From Fig. 5, we can see that two neighboring motifs contains a motif between themselves denoted by the dotted area. Fig. 6 shows the associated digraph of the motif formed between the two motifs. Again using Lemma 4.2 we can say that there is always a path to go from one motif to any of its neighboring motif.

Thus, the directed graph $\mathcal{G}^P$ formed by the Markov chain $\mathcal{M}^P$ associated with the whole search space can be represented by the union of strongly connected sub-graphs $\mathcal{G}_i$ (of each motif), such that each sub-graph has a path to and from its neighboring sub-graph. Hence, $\mathcal{G}^P$ is a strongly connected graph. Now using Lemma 4.1, we can say that the probability of reaching any state of $\mathcal{M}^P$ from any other state is strictly larger than zero. ■

B. Source Seeking Algorithms

Using SUSD, a group of N agents will always converge to the source location given they have a measurement of the scalar field at all times. This has been proved by Wu et. al in [5][6]. If v_c denotes the group velocity for N agents, then it can be shown that the direction of v_c for any group of agents aligns anti-parallel to the direction of the gradient field. This was proved in [5] by showing that i.e.$\frac{\mathbf{v}_c}{||\mathbf{v}_c||} \cdot \frac{\nabla z(\mathbf{r}_c)}{||\nabla z(\mathbf{r}_c)||}$ converges to -1, where $\mathbf{r}_c$ denotes the position of the group center. Therefore, using SUSD, a group of agents will always converge towards a source in a scalar field.

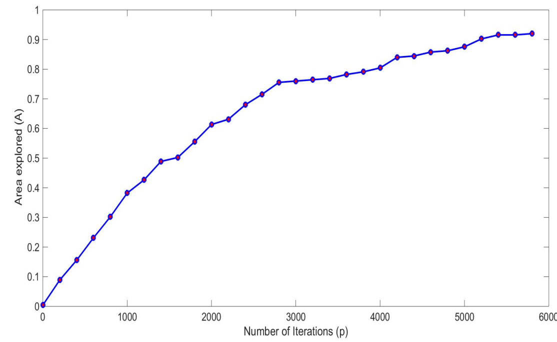


Fig. 7: Ratio of area covered by an agent with time, for the data in Fig. 8. The average coverage rate is almost 3.5% per iteration.

V. RESULTS

We simulated the steering model in the absence of any source for a single agent. This was done in order to visualize the area covered by an agent over 6000 iterations. Fig. 7 shows the percentage of area covered by an agent with respect to time, while Fig. 8 shows a visual representation of the area coverage in the search space. A rectangular workspace of 1.5×1.5 was considered and the space was divided into a grid where each grid cell was 0.1×0.1. The simulation was done for the following parameters: $\sigma = 20, k_w = 0.4, \xi = 0.024, v = 0.8$ and $\triangle t = 0.01$.

The area explored by an agent can then be given as:

$$A = \frac{N_X}{N_T} A_{ss} \quad (18)$$

where N_X denotes the number of blocks which have been explored 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.

As can be seen in Fig. 7 and Fig. 8, the agent covers a significant part of the grid while remaining confined inside the search space. At $p = 6000$ iterations, the agent has visited 209 grid cells out of 225 cells implying a coverage of 92.89%. This validates the claim that an agent is likely to cover all grid cells in a given search space with a non-zero probability. The rate at which the agent covers the grid, however, is much less as compared to other deterministic search methods, such as the lawnmower strategy [13]. Lawnmower strategy would require approximately $p = 225$ iterations to cover the entire grid. However, such deterministic strategies rely on a localization service to keep track of the cells which the agent has visited and has to still visit. This is not the case in the motion model (3-5), which allows reaching one of the sources without any knowledge of the agent's location. Moreover, the lawnmower strategy requires the search space to be rectangular. This is not the case in the motion model (3-5) where the search space can be circular as well.

We also simulated the Hybrid SUSD strategy for a group of $N = 10$ agents. This was done for K sources where $K \in \{1, 2, 4, 6, 8\}$. The case of $K = 2$ is shown in Fig. 9. A search space of 200×100 was taken and the sources were placed at the x,y coordinates (50, 50) and (150, 50). The

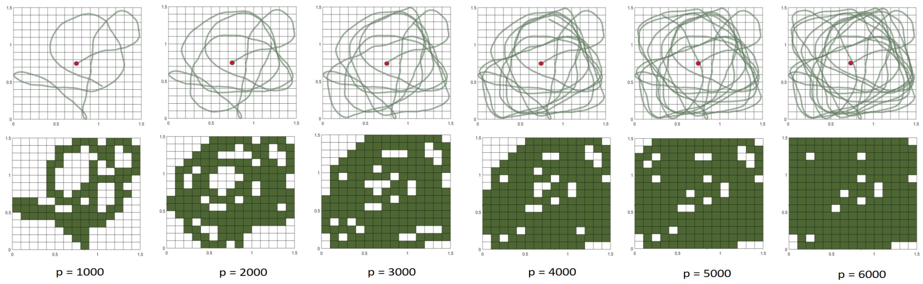


Fig. 8: Area covered by a an agent over p iterations. The agent's initial position is denoted by the red dot (0.75 0.75).

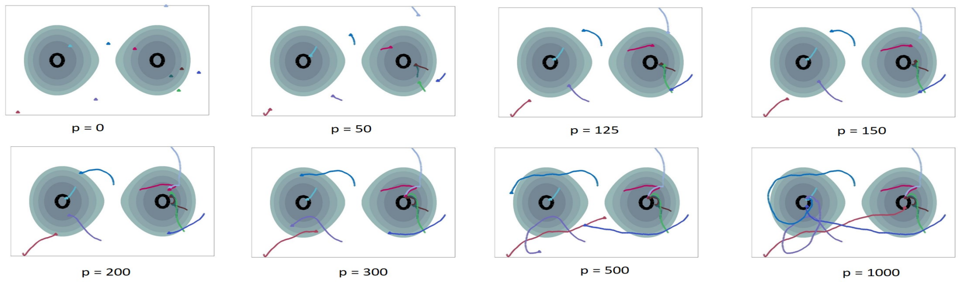


Fig. 9: Trajectories using Hybrid SUSD for $N = 10$ agents. White region denotes an area where agents do not get reliable measurements while the gray region is an area where agents get reliable measurements. The source is denoted by the black circle within gray region.

initial positions of the agents were drawn from a uniform distribution over the entire search space. For 1500 iterations, the following parameters were used for the motion model: $\sigma = 10, k_w = 0.94, \xi = 0.024$, $\triangle t = 0.01$, $d^{th} = 10$ and $v = 20$. Fig. 9 shows the trajectories of the agents using the Hybrid SUSD strategy. Of $N = 10$ agents, 8 of the agents switch to SUSD while the rest explore the workspace using the exploration model until they reach the source.

A large number of trials were carried out and the following observations were made. As the number of sources K is increased, there is a lower likelihood of all the sources to be localized for a give number of agents N. However, the likelihood of localization also depends on the initial positions of the N agents. Hence, we initialize the agents uniformly in the search space and also assign the initial heading direction uniformly from the interval $[0, 2\pi]$. We noticed that if $K = 2$, then if $N \geq 4$, the agents are able to find all the sources. For $K = 4$, the least number of agents to seek all the sources came out to be $N = 6$. We propose that there might be a relation between K and smallest N that can be used to seek all the sources. This might be an important basis to solve multi-modal optimization problems. Another important observation is that if the area of the region Ω_k around the sources is decreased, the average amount of time the agents remain in the exploration mode increases. This is an important implication for localization of sources which are spread far apart in the search space or which do not provide measurements at all times, for example chemical plumes.

VI. CONCLUSIONS AND FUTURE WORK

We proposed a bio-inspired strategy, the Hybrid SUSD which can achieve distributed control of a multi-agent system for the localization of multiple sources in a search space. Our simulation results suggest that there might be a relation between the number of agents N and the number of sources K that can be localized in a scalar field. This can form a basis to solve distributed non-convex optimization problems in the future.

REFERENCES

[1] M. Ye and G. Hu, "A distributed extremum seeking scheme for networked optimization," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 4928–4933.
[2] A. O. Vweza, K. T. Chong, and D. J. Lee, "Gradient-free numerical optimization-based extremum seeking control for multiagent systems," *International Journal of Control, Automation and Systems*, vol. 13, no. 4, pp. 877–886, 2015.
[3] Y. Liu and K. M. Passino, "Stable social foraging swarms in a noisy environment," *Automatic Control, IEEE Transactions on*, vol. 49, no. 1, pp. 30–44, 2004.
[4] E. Bıyık and M. Arcak, "Gradient climbing in formation via extremum seeking and passivity-based coordination rules," *Asian Journal of Control*, vol. 10, no. 2, pp. 201–211, 2008.
[5] W. Wu, I. D. Couzin, and F. Zhang, "Bio-inspired source seeking with no explicit gradient estimation," 2012.
[6] W. Wu and F. Zhang, "A speeding-up and slowing-down strategy for distributed source seeking with robustness analysis1."
[7] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 77–98, 2001.
[8] J. R. Howse, R. A. Jones, A. J. Ryan, T. Gough, R. Vafabakhsh, and R. Golestanian, "Self-motile colloidal particles: from directed propulsion to random walk," *Physical review letters*, vol. 99, no. 4, p. 048102, 2007.
[9] J. Gautrais, F. Ginelli, R. Fournier, S. Blanco, M. Soria, H. Chaté, and G. Theraulaz, "Deciphering interactions in moving animal groups," *PLoS Comput Biol*, vol. 8, no. 9, p. e1002678, 2012.
[10] J. Gautrais, C. Jost, M. Soria, A. Campo, S. Motsch, R. Fournier, S. Blanco, and G. Theraulaz, "Analyzing fish movement as a persistent turning walker," *Journal of mathematical biology*, vol. 58, no. 3, pp. 429–445, 2009.
[11] N. Privault, *Understanding Markov chains: examples and applications*. Springer Science & Business Media, 2013.
[12] J. Jarvis and D. R. Shier, "Graph-theoretic analysis of finite markov chains," *Applied mathematical modeling: a multidisciplinary approach*, 1999.
[13] A. Hubenko, V. A. Fonoberov, G. Mathew, and I. Mezić, "Multiscale adaptive search," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, no. 4, pp. 1076–1087, 2011.