

A Derivative-free Distributed Optimization Algorithm with Applications in Multi-Agent Target Tracking

Said Al-Abri, Tony X. Lin, Robert S. Nelson, and Fumin Zhang

Abstract—Multi-agent tracking of a moving target can be modeled as a distributed optimization problem of a time-varying objective function that has an optimum at the ideal sensing states of the agents. The inputs to the objective function are some observed parameters of the target which are obtained from onboard sensory information. In particular, due to recent progress in learning-based vision techniques, these observed parameters may be of large dimensions which may limit communication capabilities, especially for large groups of coordinating agents. In cases where the analytical form of the objective function is unknown or the gradient of the objective function is difficult to estimate, which may occur due to a fast moving target or the high dimensionality of the observation parameters, gradient-based solutions may be inapplicable or computationally prohibitive to apply. In this paper, we propose a derivative-free distributed optimization algorithm based on distributed active perception for multi-agent target tracking. Our proposed method can optimize objective functions without knowledge of the gradient and does not require communication. We derive the information dynamics for general dimensions which are used to analyze the tracking convergence. Simulations and experimental results are provided.

I. INTRODUCTION

In surveillance and path planning, multi-agent target tracking is an important problem in designing how a swarm of agents should move in order to maximize some information of a target [1], [2], [3], [4], [5], [6]. In many cases, these problems can be formulated as a distributed optimization problem in which the optimum exists at the ideal sensing states of each agent. For example, the authors of [7], [8] both suggest information-based approaches to finding ideal information improvements in an active sensing framework. In particular, [7] proposes a decentralized approach to solving multi-agent target-tracking problems, much like the objective of this paper.

However, the approach proposed in [7] depends on the gradient of the distributed objective function, leading to possible difficulties if the gradient is ill-defined or difficult to compute. In fact, many methods in the literature rely on computing the gradient of the distributed objective function in order to coordinate the multi-agent target tracking [9], [10], [11]. While able to achieve good cooperative tracking performance, these methods may no longer be applicable when the sensing modality does not provide the necessary information to compute the gradient. For example, the gradient of an objective function based on the distance of each agent to the target requires the direction to the target, which may

not be available in beacon-like ranging sensors (e.g. UWB radios, acoustic beacons). In addition, even when a sensing modality provides the necessary information to evaluate the gradient, finding the gradient may be extremely difficult, for example, in cases where the sensing information is extremely high dimensional (e.g. vision methods). The recent success of learning in vision-based perception, in particular, motivates a need to develop novel control methodologies that can take advantage of these new sensing capabilities.

As a result, we propose the use of our previous source-seeking strategy, called Speeding-Up and Slowing-Down (SUSD) [12], [13], to solve multi-agent target-tracking problems that can be formulated as distributed optimization problems. The SUSD method does not require an expression for the gradient and only relies on evaluations of the objective function in order to minimize the objective function. The application of our approach in this paper, therefore, yields the following main contributions: **i)** designing of function mappings that handle high-dimensional visual perceptual information, **ii)** derivation of the optimization dynamics of SUSD for general dimension and under complete graphs, **iii)** convergence analysis of the formation control law while perturbed by SUSD, **iv)** convergence analysis of the SUSD method for distributed optimization problems, and **v)** validation for two case studies in simulation and experiments using the Robotarium [14]. A video of the simulations and experiments is provided at https://youtu.be/cL_z8rHUsdM.

Our proposed framework would allow for the utilization of recent learning-based high-dimensional vision techniques that incur high communication costs. Learning-based vision methods, such as those proposed in [15], [16], [17], [18], [19], extract relevant features and information that can be used to encode a scalar function mapping. The proposed scalar function mappings, for example, then eliminate the need to communicate high-dimensional vision data among the agents. The distributed derivative-free SUSD optimization algorithm, therefore, provides a promising solution to challenging optimization problems of unknown, complex, and time-varying functions.

The structure of the paper is as follows: Section II introduces the distributed optimization problem with varying perception methodologies, and in Section III we describe the active perception framework of SUSD used to solve the multi-target tracking problem. Section IV presents the dynamics of the agents employing the SUSD control law for multi-agent target tracking. Section V provides a convergence analysis of our proposed method and Section VI

Said Al-Abri, Tony X. Lin, Robert S. Nelson, and Fumin Zhang are at School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, E-mail: {saidalabri, tlin339, rnelson71, fumin}@gatech.edu.

validates our approach in simulation and experiments. We conclude with comments on our approach and future work in Section VII.

II. PROBLEM FORMULATION

Consider a swarm of M agents. Each agent is located at a position $\mathbf{r}_i \in \mathbb{R}^d$, where $i = 1, \dots, M$. The interactions among the agents are described by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{1, \dots, M\}$ is the set of all nodes, and $\mathcal{E} = \{(i, j) | i, j \in \mathcal{V}\}$ is the set of all edges. Let $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$ be the set of neighboring agents, and let $\bar{\mathcal{N}}_i = \{i\} \cup \mathcal{N}_i$. A graph is complete if each agent knows information about all other agents, and is incomplete if otherwise.

Assumption 2.1: The graph is undirected and complete for all time.

Assumption 2.2: Each agent i can locally measure the relative positions $(\mathbf{r}_j - \mathbf{r}_i)$ for all $j \in \mathcal{N}_i$.

Suppose that the swarm is required to track a moving target. The position of the target is denoted by $\mathbf{r}_0 \in \mathbb{R}^d$ and its velocity is given by $\dot{\mathbf{r}}_0 = \mathbf{u}_0$.

Assumption 2.3: The speed of the target is unknown but is bounded such that $\|\dot{\mathbf{r}}_0\| = \|\mathbf{u}_0\| \leq \bar{s}$.

Furthermore, suppose that each agent observes information about the target denoted by $\mathcal{I}_i(t) = \mathcal{I}(\mathbf{r}_i(t), \mathbf{r}_0(t))$. The dimension $\mathcal{M}$ of the space of $\mathcal{I}$ might be large such as that of a stream of images. It is desired, due to communication constraints, to not communicate the information $\mathcal{I}$ among the agents. Let the velocity of each agent be described by

$$\dot{\mathbf{r}}_i = \mathbf{u}_i, \quad i = 1, \dots, M, \quad (1)$$

where $\mathbf{u}_i$ is a control input to be designed based only on the local available information.

The two main problems we consider in this paper are: (i) design a perception mapping that transforms the local information $\mathcal{I}_i$ into a scalar function, and (ii) design a distributed active perception control law $\mathbf{u}_i$ such that the swarm collectively tracks the target. Integrating a distributed high-dimensional perception mapping with a task-oriented distributed control law is challenging especially when the agents have constraints on the communication bandwidth.

III. THE PROPOSED FRAMEWORK

We propose the following framework for target tracking described by the following procedure.

- 1) **Perception:** design an objective mapping $z : \mathcal{M} \rightarrow \mathbb{R}$ such that $z_i(t) = z(\mathcal{I}(\mathbf{r}_i(t), \mathbf{r}_0(t)))$ maps the local information $\mathcal{I}_i$ to a scalar value. This function has to satisfy the following requirement.

Assumption 3.1: The function z is required to be smooth with a unique minimum.

- 2) **Optimization and Control:** design a distributed control law $\mathbf{u}_i$ such that the agents collectively solve the optimization problem

$$\min_{\mathbf{r}_1, \dots, \mathbf{r}_M} \frac{1}{M} \sum_{i=1}^M z(\mathcal{I}(\mathbf{r}_i(t), \mathbf{r}_0(t))). \quad (2)$$

The integration of the above two steps leads to a distributed active perception algorithm where agents move in a way that increases information about the target. It is challenging to solve the optimization problem (2) as in general the gradient of $z(\mathcal{I}(\mathbf{r}_i(t), \mathbf{r}_0(t)))$ is hard to be known. We overcome this challenge in this paper by leveraging the derivative-free SUSD optimization algorithm in [12], [13].

A. Mapping of Target Information

In this section, we present one information mapping that incorporates the distance to the target, and another one that incorporates vision information.

1) *The Distance-Based Function:* If distance to the target is used as an information, then this information is maximized by getting closer to the target. Consider the mapping

$$z_i(\mathcal{I}(\mathbf{r}_i(t), \mathbf{r}_0(t))) = \|\mathbf{r}_i - \mathbf{r}_0\|^2, \quad i = 1, 2, \dots, M, \quad (3)$$

where $\mathbf{r}_0$ is the target location. Each agent can compute (3) by using the measured relative position of the target. Alternatively, agents can measure the distance (3) directly using for example infrared or ultrasonic sensors.

2) *The Image-Based Function:* Next, consider function mapping to handle high-dimensional visual perception information. In the literature, many neural network object detectors have been proposed in order to identify what set of pixels, usually described by a rectangle known as a “bounding box”, contain a target. In general, these object detectors can be described as a mapping $g : \mathcal{I} \rightarrow \mathcal{B}$ where $\mathcal{I}$ is high-dimensional image space and $\mathcal{B} \subset \mathbb{R}^{2 \times 4}$ describes the four pixel positions of the bounding box corners. In this work, we consider a simulated 2D example, where agents directly access the bounding line points in an image, i.e. $\mathcal{B} \subset \mathbb{R}^2$ in order to describe the two-pixel positions between which contains the target. We assume the target is represented by a circle with radius R and the pixel positions are equivalent to having access to two relative angles, called b_1 and b_2 , that describe the angle of the agent to each edge of the target. Fig. 1 demonstrates the 2D bounding line perception function g . We can then produce a scalar function

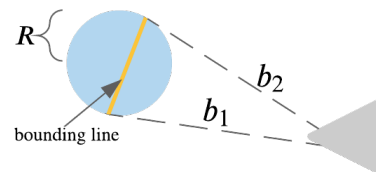


Fig. 1: Visual explanation of the bounding line perception function g .

that satisfies Assumption 3.1 by providing a desired angle width b_{des} . We can then choose z as

$$z(\mathbf{b}) = [b_{\text{des}} - (b_1 - b_2)]^2. \quad (4)$$

The full evaluation of the scalar function, given a position $\mathbf{r}$, a heading direction θ (which in this work is the SUSD direction as detailed in [12], [13]), and an image sampling function $h : \mathbb{R}^2 \times [-\pi, \pi] \rightarrow \mathcal{I}$ that represents an image generated by a camera is $z(g(h(\mathbf{r}, \theta)))$ which is smooth,

and has a unique minimum at $(\mathbf{b}_1 - \mathbf{b}_2) = \mathbf{b}_{\text{des}}$ as required by Assumption 3.1.

B. The Derivative-Free SUSD Optimization Algorithm

From the locally observed relative positions, each agent computes the covariance matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$ defined by

$$\mathbf{C} = \sum_{k=1}^M (\mathbf{r}_k - \mathbf{r}_c)(\mathbf{r}_k - \mathbf{r}_c)^\top, \quad (5)$$

where $\mathbf{r}_c = \frac{1}{M} \sum_{k=1}^M \mathbf{r}_k$ is the center of the swarm. Let $\{\mathbf{v}_1, \dots, \mathbf{v}_d\}$ be the eigenvectors of the covariance matrix (5) associated with the eigenvalues $\{\lambda_1, \dots, \lambda_d\}$, ordered from the smallest eigenvalue λ_1 to the largest eigenvalue λ_d . Observe that the eigenvectors of $\mathbf{C}$ produces the principal components of the spatial distribution of the agents.

Let the velocity of each agent be described by (1). Then, each agent modulates its motion by

$$\mathbf{u}_i = k_1 z_i \mathbf{v}_1 + k_2 \sum_{j \in \mathcal{N}_i} (\|\mathbf{r}_j - \mathbf{r}_i\|^2 - d_{ij}^2)(\mathbf{r}_j - \mathbf{r}_i), \quad (6)$$

where k_1 and k_2 are tuning constants. The first term in (6) enables each agent to speed up or slow down, hence the name (SUSD), along the motion direction $\mathbf{v}_{1,i}$ based on the function evaluation $z_i = z(\mathcal{I}(\mathbf{r}_i(t), \mathbf{r}_0(t)))$. We call the motion direction $\mathbf{v}_1$ the SUSD direction which is the eigenvector of the covariance matrix (5) associated with the smallest eigenvalue λ_1 . The second term in (6) enforces the i th agent to maintain a desired distance specified by d_{ij} with its j^{th} neighbor. Pseudo code of the SUSD algorithm is shown in **Algorithm 1**.

Algorithm 1 The Distributed SUSD Optimization

```

1: Input: number of agents  $M$ , initial positions  $\mathbf{r}_i(0)$ ,
   desired distance  $d_{ij}$ , discretization step  $\eta$ , gains  $k_1$  and
    $k_2$ , and total iterations  $K$ .
2: for Iterations  $k = 1, \dots, K$ , do
3:   for  $i = 1, \dots, M$ , do
4:     compute  $\mathbf{v}_1(k) = \text{PCA}\{\mathbf{r}_l(k)\}_{l=1}^M$ .
5:     compute  $z_i(\mathbf{r}_i(k))$ .
6:     compute  $w_{ij}(k) = (\|\mathbf{r}_j(k) - \mathbf{r}_i(k)\|^2 - d_{ij}^2)$ .
7:     compute  $\mathbf{v}_{i,f}(k) = \sum_{j \in \mathcal{N}_i} w_{ij}(k)(\mathbf{r}_j(k) - \mathbf{r}_i(k))$ 
8:     update  $\mathbf{r}_i(k+1) = \mathbf{r}_i(k) + \eta(k_1 z_i(\mathbf{r}_i(k))\mathbf{v}_1(k) + k_2 \mathbf{v}_{i,f}(k))$ .
9:   end for
10: end for
```

Remark 1: The sign of the PCA eigenvector $\mathbf{v}_1(k)$ in step 4 in Algorithm 1 is chosen such that $\langle \mathbf{v}_1(k), \mathbf{v}_1(k-1) \rangle \geq 0$.

IV. THE OPTIMIZATION DYNAMICS OF HIGH-DIMENSION DISTRIBUTED ACTIVE PERCEPTION

In order to analyze the relationship between the SUSD direction $\mathbf{v}_1$ and the function gradient $\nabla z(\mathbf{r})$, we need to study the time-evolution of the SUSD direction $\mathbf{v}_1$ under the agents velocity (1) and the control law (6). We call the time derivatives of the eigenvectors of the covariance matrix (5) the optimization dynamics.

In this section, we first study the optimization dynamics under any control law, and then we obtain the closed-loop dynamics under the motion (1) and control law (6). We obtain several results for general dimensions and for incomplete and complete graphs. To increase the readability of the paper, we move all proofs of this section to the Appendix in Section VIII.

Theorem 4.1: Under the motion system (1) with a general control $\mathbf{u}_i$, the open-loop optimization dynamics are

$$\dot{\mathbf{v}}_a = \sum_{b \neq a} \kappa_{a,b} \mathbf{v}_b, \quad a = 1, \dots, d, \quad (7)$$

where

$$\kappa_{a,b} = \frac{1}{\lambda_a - \lambda_b} \mathbf{v}_a^\top \sum_{k=1}^M \left[(\mathbf{r}_k - \mathbf{r}_c) \mathbf{u}_k^\top + \mathbf{u}_k (\mathbf{r}_k - \mathbf{r}_c)^\top \right] \mathbf{v}_b. \quad (8)$$

See proof in Section VIII.

Theorem 4.2: Under the motion system (1) and the control input (6), the closed-loop optimization dynamics are

$$\dot{\mathbf{v}}_a = \sum_{b \neq a} \kappa_{a,b} \mathbf{v}_b, \quad a = 1, \dots, d, \quad (9)$$

where

$$\begin{aligned} \kappa_{a,b} = & \frac{k_1}{\lambda_a - \lambda_b} \mathbf{v}_a^\top \sum_{k=1}^M z_k \left[(\mathbf{r}_k - \mathbf{r}_c) \mathbf{v}_1^\top + \mathbf{v}_1 (\mathbf{r}_k - \mathbf{r}_c)^\top \right] \mathbf{v}_b \\ & + \frac{k_2}{\lambda_a - \lambda_b} \sum_{k=1}^M \left[(\mathbf{r}_k - \mathbf{r}_c) \mathbf{s}_k^\top + \mathbf{s}_k (\mathbf{r}_k - \mathbf{r}_c)^\top \right] \mathbf{v}_b. \end{aligned} \quad (10)$$

See proof in Section VIII.

Using Taylor approximation, we write

$$z_k - z_c = \langle \mathbf{r}_k - \mathbf{r}_c, \nabla z_c \rangle + \nu_k, \quad (11)$$

where $\nabla z_c = \nabla z(\mathbf{r}_c)$ is the gradient at the center of the swarm, and ν_k is the remaining higher-order components of the field approximation.

Assumption 4.1: At each instant of time, the function z is approximately linear around the vicinity of the center $\mathbf{r}_c$ and thus $\nu_k \approx 0$.

This assumption implies that z can be viewed as a piecewise linear function. This is realistic only when the desired inter-agent distances d_{ij} in (6) are small enough.

Theorem 4.3: Consider a complete graph. Assume Assumption 4.1 holds. Then, under the motion system (1) and the control input (6), the dynamics of the SUSD direction are

$$\dot{\mathbf{v}}_1 = \left(\sum_{b \neq 1} \frac{\mathbf{v}_b \mathbf{v}_b^\top}{\lambda_1 - \lambda_b} \right) (k_1 \mathbf{C} \nabla z_c + k_2 S \mathbf{v}_1), \quad (12)$$

where

$$S = \sum_{k=1}^M \left[(\mathbf{r}_k - \mathbf{r}_c) \mathbf{s}_k^\top + \mathbf{s}_k (\mathbf{r}_k - \mathbf{r}_c)^\top \right]. \quad (13)$$

See proof in Section VIII.

V. CONVERGENCE ANALYSIS

Here, we first study the convergence of the formation control law. Then, we study the convergence of the SUSD direction to the negative direction of the optimized function gradient. Finally, we study the target tracking convergence.

A. Convergence Analysis of the Formation

Combining the motion system (1) along with the control law (6), we get

$$\dot{\mathbf{r}}_i = k_1 z_i \mathbf{v}_1 + k_2 \sum_{j \in \mathcal{N}_i} w_{ij} (\mathbf{r}_j - \mathbf{r}_i), \quad (14)$$

where $w_{ij} = w_{ji} = (\|\mathbf{r}_j - \mathbf{r}_i\|^2 - d_{ij}^2)$.

Let $\mathbf{r}_c = \frac{1}{M} \sum_{i=1}^M \mathbf{r}_i$ be the center of formation. Then, we obtain $\dot{\mathbf{r}}_c = \frac{1}{M} \sum_{i=1}^M z_i \mathbf{v}_1 + \frac{1}{M} \sum_{i=1}^M \sum_{j \in \mathcal{N}_i} w_{ij} (\mathbf{r}_j - \mathbf{r}_i)$. Since $w_{ij} = w_{ji}$, then $\frac{1}{M} \sum_{i=1}^M \sum_{j \in \mathcal{N}_i} w_{ij} (\mathbf{r}_j - \mathbf{r}_i) = 0$. This implies that

$$\dot{\mathbf{r}}_c = k_1 z_a \mathbf{v}_1, \quad (15)$$

where $z_a = \frac{1}{M} \sum_{i=1}^M z_i$ is the average function value. Hence, the center of the swarm is moving according to (15). We then view the SUSD component, i.e. $k_1 z_i \mathbf{v}_1$, in (14) as a perturbation to the formation. We first obtain $\dot{w}_{ij} = (\frac{\partial w_{ij}}{\partial \mathbf{r}_i})^\top \dot{\mathbf{r}}_i + (\frac{\partial w_{ij}}{\partial \mathbf{r}_j})^\top \dot{\mathbf{r}}_j = -2(\mathbf{r}_i - \mathbf{r}_j)^\top (\dot{\mathbf{r}}_i - \dot{\mathbf{r}}_j)$. Then, using (14), we derive

$$\begin{aligned} \dot{w}_{ij} = & -2k_1(z_i - z_j) \langle \mathbf{r}_i - \mathbf{r}_j, \mathbf{v}_1 \rangle \\ & - 2k_2 \left\langle \mathbf{r}_i - \mathbf{r}_j, \sum_{m=1}^m [w_{im}(\mathbf{r}_m - \mathbf{r}_i) - w_{jm}(\mathbf{r}_m - \mathbf{r}_j)] \right\rangle. \end{aligned} \quad (16)$$

Define $\mathbf{w} = [w_{12}, w_{23}, \dots, w_{(M-1)M}]^\top \in \mathbb{R}^{\frac{M(M-1)}{2}}$ to be a vector of all w_{ij} . Let $\mathbf{z} = [z_1, \dots, z_M]^\top \in \mathbb{R}^M$.

Theorem 5.1: Consider the system (14). When $k_1 = 0$, the origin $\mathbf{w} = \mathbf{0}$ of the unforced system of (16) is asymptotically stable. On the other hand, for $k_1 \neq 0$, if $\|\mathbf{z}\|$ is bounded from above, then the origin $\mathbf{w} = \mathbf{0}$ of the forced system of (16) is input-to-state stable.

Proof: Let $V_1 = \frac{1}{4} \sum_{(i,j) \in \mathcal{E}} w_{ij}^2 = \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M w_{ij}^2$ be a Lyapunov candidate function where $V_1 = 0$ if and only if $w_{ij} = 0$ for all i, j . Then, when $k_1 = 0$, we obtain

$$\dot{V}_1 = -k_2 \sum_{i=1}^M \sum_{j=1}^M w_{ij} (\mathbf{r}_j - \mathbf{r}_i)^\top (\mathbf{r}_j - \mathbf{r}_i) \leq 0. \quad (17)$$

Note that, since $\dot{\mathbf{r}}_c = 0$, then this implies that the center of the swarm is stationary, and hence it is impossible to have $\mathbf{r}_i = \mathbf{r}_j$ for all (i, j) unless all the agents start initially from the same position. Hence, $\dot{V}_1 = 0$ if and only if $w_{ij} = 0$ for all (i, j) , and by using LaSalle Invariance Principle we conclude that the origin of the unforced system is asymptotically stable. On the other hand, when $k_1 \neq 0$, we obtain

$$\dot{V}_1 \leq -(1-\epsilon)k_2 \sum_{i=1}^M \sum_{j=1}^M w_{ij} (\mathbf{r}_j - \mathbf{r}_i)^\top (\mathbf{r}_j - \mathbf{r}_i), \forall \|\mathbf{w}\| > \rho(\|\mathbf{z}\|),$$

where $\rho(\|\mathbf{z}\|) = \sqrt{M} \frac{k_1}{2\epsilon k_2} \|\mathbf{z}\|$ is a class $\mathcal{K}$ function as $\|\mathbf{z}\|$ is assumed to be bounded from above. Let $\alpha_1(\|\mathbf{w}\|) = \alpha_2(\|\mathbf{w}\|) = \frac{1}{2} \|\mathbf{w}\|^2$ be class $\mathcal{K}_\infty$ functions such that $\alpha_1(\|\mathbf{w}\|) \leq V_1(\mathbf{w}) \leq \alpha_2(\|\mathbf{w}\|)$. Then, according to **Theorem 4.19** in [20], system (16) is input-to-state stable. ■

B. Convergence Analysis of SUSD

Let $\mathbf{N} = \frac{\nabla z_c}{\|\nabla z_c\|}$ be a unit-length vector along the gradient direction at the center of the swarm. Define $\theta = 1 + \langle \mathbf{N}, \mathbf{v}_1 \rangle$.

Then, we obtain $\dot{\theta} = \langle \dot{\mathbf{N}}, \mathbf{v}_1 \rangle + \langle \dot{\mathbf{N}}, \mathbf{v}_1 \rangle$. Using (12), we derive

$$\dot{\theta} = -k_1 \|\nabla z_c\| \sum_{b \neq 1} \frac{\lambda_b}{\lambda_b - \lambda_1} \langle \mathbf{N}, \mathbf{v}_b \rangle^2 + \delta_1, \quad (18)$$

where $\delta_1 = \langle \dot{\mathbf{N}}, \mathbf{v}_1 \rangle + k_2 \sum_{b \neq 1} \frac{\langle \mathbf{N}, \mathbf{v}_b \rangle \langle \mathbf{v}_b, S \mathbf{v}_1 \rangle}{\lambda_1 - \lambda_b}$.

Theorem 5.2: Consider (18). Assume that $\|\nabla z_c\| \geq \mu$ where $\mu > 0$ is a constant. Then when $\delta_1 = 0$, the equilibrium $\theta = 0$ is asymptotically stable and thus whenever $\theta(0) \in [0, 1)$, then $\theta(t) \rightarrow 0$ as $t \rightarrow \infty$. Furthermore, when $|\delta_1| < \epsilon k_1 \|\nabla z_c\|$ for some $\epsilon \in (0, 1)$, then (18) is locally input-to-state stable.

Proof: Let $D_2 = [0, 1)$ be a domain of interest. Let $V_2 : D \rightarrow \mathbb{R}$ defined by $V_2 = \frac{\theta}{1-\theta}$ be a Lyapunov candidate function where $V_2 = 0$ if and only if $\theta = 0$. Additionally, $V_2 \rightarrow \infty$ whenever $\theta \rightarrow 1$. Then, when $\delta_1 = 0$, we obtain

$$\dot{V}_2 \leq -k_1 \|\nabla z_c\| \frac{\theta(2-\theta)}{(1-\theta)^2} \leq 0, \quad (19)$$

where, since $\|\nabla z_c\| \geq \mu$, then $\dot{V}_2 = 0$ if and only if $\theta = 0$. Therefore, $\theta = 0$ is asymptotically stable. Additionally, since $\dot{V}_2 \rightarrow -\infty$ as $\theta \rightarrow 1$, then the domain $D_2 = [0, 1)$ is forward invariant. When $\delta_1 \neq 0$,

$$\dot{V}_2 \leq -(1-\epsilon)k_1 \|\nabla z_c\| \frac{\theta(2-\theta)}{(1-\theta)^2}, \quad \forall |\theta| \geq \rho_2(|\delta_1|), \quad (20)$$

where $\rho_2(|\delta_1|) = 1 - \sqrt{1 - \frac{|\delta_1|}{k_1 \|\nabla z_c\|}}$ is a class $\mathcal{K}$ function. Since it is assumed that $|\delta_1| < \epsilon k_1 \|\nabla z_c\|$, then the set $\theta \in [\rho_1(|\delta_1|), 1)$ is non-empty. Let $\alpha_3(|\theta|) = \alpha_4(|\theta|) = \frac{|\theta|}{1-|\theta|}$ be class $\mathcal{K}$ functions such that $\alpha_3(|\theta|) \leq V_2(\theta) \leq \alpha_4(|\theta|)$. Then, using **Definition 3.3** of local input-to-state stability in [21], and according to **Theorem 4.19** in [20], system (18) is locally input-to-state stable. ■

C. Convergence of the Tracking

Let $z_c = z(\mathbf{r}_c, \mathbf{r}_0)$. we derive $\dot{z}_c = (\frac{\partial z_c}{\partial \mathbf{r}_c})^\top \dot{\mathbf{r}}_c + (\frac{\partial z_c}{\partial \mathbf{r}_0})^\top \dot{\mathbf{r}}_0$. Then, using (15) and the definition $\frac{\partial z_c}{\partial \mathbf{r}_c} = \nabla z_c$ and $\frac{\partial z_c}{\partial \mathbf{r}_0} = \nabla z_0$, we obtain

$$\dot{z}_c = k_1 z_c \|\nabla z_c\| (\theta - 1) + \delta_2, \quad (21)$$

where $\theta = 1 + \langle \mathbf{N}, \mathbf{v}_1 \rangle$, and $\delta_2 = \langle \nabla z_0, \dot{\mathbf{r}}_0 \rangle$. Note that, since we ignore the higher-order terms in (11), then we substituted for z_a in (15) the value $z_a = z_c + \sum_{k=1}^M \nu_k = z_c$.

Theorem 5.3: Consider (21). Assume that $\|\nabla z_c\| \geq \mu$ where $\mu > 0$ is a constant. Additionally, assume that $\theta < 1$. Then when $\delta_2 = 0$, the equilibrium $z_c = 0$ is asymptotically stable and thus $z_c(t) \rightarrow 0$ as $t \rightarrow \infty$. Furthermore, when $|\delta_2| \leq \bar{s} \|\nabla z_0\|$, then (21) is locally input-to-state stable.

Proof: Let $D_3 = [0, \infty)$ be a domain of interest. Let $V_3 : D \rightarrow \mathbb{R}$ defined by $V_3 = \frac{1}{2} z_c^2$ be a Lyapunov candidate function where $V_3 = 0$ if and only if $z_c = 0$. Then, when $\delta_2 = 0$, we obtain

$$\dot{V}_3 \leq -\epsilon k_1 \|\nabla z_c\| z_c^2, \quad (22)$$

where, since $\|\nabla z_c\| \geq \mu$, then $\dot{V}_3 = 0$ if and only if $z_c = 0$. Then the equilibrium $z_c = 0$ is asymptotically stable. On the other hand, when $\delta_2 \neq 0$, then

$$\dot{V}_3 \leq -\epsilon(1-\epsilon)k_1 \|\nabla z_c\| z_c^2, \quad \forall |z_c| \geq \rho_3(|\delta_2|), \quad (23)$$

where $\rho_3(|\delta_2|) = \sqrt{\frac{|\delta_2|}{\epsilon^2 k_1 \|\nabla z_c\|}}$ is a class $\mathcal{K}$ function. Since it is assumed that $|\delta_2| \leq \bar{s} \|\nabla z_0\|$, then $\rho_3(|\delta_2|) \leq \sqrt{\frac{\bar{s}}{\epsilon^2 k_1}}$ is finite. Let $\alpha_5(|z_c|) = \alpha_6(|z_c|) = \frac{1}{2}|z_c|^2$ be class $\mathcal{K}_\infty$ functions such that $\alpha_5(|z_c|) \leq V_3(z_c) \leq \alpha_6(|z_c|)$. Then, by **Definition 3.3** of local input-to-state stability in [21], and according to **Theorem 4.19** in [20], system (21) is locally input-to-state stable. $\blacksquare$

VI. SIMULATIONS AND EXPERIMENTAL RESULTS

In this section, we validate our approach by studying three case studies: **i)** a 3D simulation using the distance-based objective function, **ii)** a 2D simulation using the image-based cost function, and **iii)** a 2D experiment using the Robotarium and the distance-based objective function. Videos of the simulations and experiments are provided at https://youtu.be/cL_z8rHUsdM.

A. 3D Simulations

In this simulation experiment, we direct the target to follow a path given by $\mathbf{r}_0(t) = [\cos t \quad \sin t \quad 2t - 3]^\top$. We use a swarm of 6 agents where each agent evaluates z_i using (3). In (6), we set $k_1 = k_2 = 1$, and $d_{ij} = 1.5$ for all i, j . As shown in Fig. 2, the agents are able to follow the target along its trajectory while keeping the desired formation.

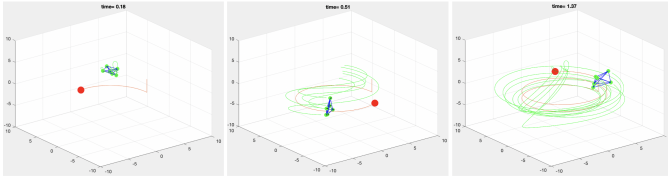


Fig. 2: A swarm of 6 agents tracking a moving target.

B. Vision-based Tracking Simulations

In this experiment, we apply the image mapping scalar function in (4). We use a simulated target with $R = 0.5$ that follows the trajectory $\mathbf{r}_0(t) = c_0 [\cos(c_1 t) \quad \sin(c_1 t)]^\top$, where $c_0 = 6$ and $c_1 = 0.006$. In addition, we consider a second scenario where the scalar function is allowed to be negative, i.e. (4) is modified to $z(\mathbf{b}) = [b_{\text{des}} - (b_1 - b_2)]$, and the target follows a trajectory $\mathbf{r}_0(t) = c_0 [\sin(c_2 t) \quad \sin(c_3 t)]^\top$, where $c_3 = 0.009$ and $c_4 = 0.006$. Note that in the second trajectory, the agents can now back-up to turn and improve their view of the target. Fig. 3 demonstrates the SUSD approach using the two image mapping scalar functions.

C. Robotarium Simulations and Experiments

In this experiment, we implement our algorithm in mobile robots at the Georgia Tech Robotarium. One of the robots is used to represent the target which is designed to be moving in a lemniscate shape according to $\mathbf{r}_0(t) = \alpha [\cos t \quad \cos t \sin t]^\top$, where $\alpha = 1$ determines the radius of curvature. We then use three additional robots for implementing the SUSD Algorithm. Each robot evaluates the distance-based target information function according to $z(\mathbf{r}_i) = 2.5(\|\mathbf{r}_i - \mathbf{r}_0\|^2) + 0.2$, where we added 0.2 so that the

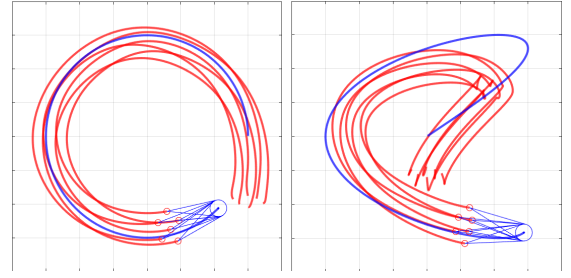


Fig. 3: The image-based tracking case where blue rays indicate $\mathbf{b}$ for each agent.

robots have a minimum nonzero speed. We set the desired inter-agent distances in (6) to be $d_{12} = d_{21} = 0.55$, and $d_{13} = d_{31} = d_{23} = d_{32} = 0.5$. We set the tuning parameters in (6) to be $k_1 = 0.4$ and $k_2 = 0.01$. As seen from Fig. 4, the swarm is able to continuously track the target throughout its path while keeping a consistent formation.

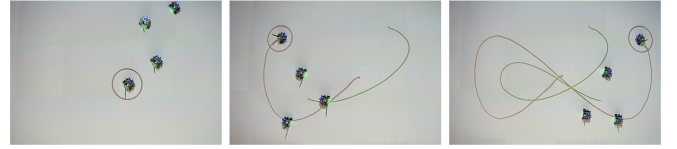


Fig. 4: A swarm of 3 robots tracking a target robot outlined by a circle. The trails show the trajectory of the target and the center of the swarm.

VII. CONCLUSION

In this work, we have presented a new application of our previous decentralized source-seeking strategy called SUSD for use in solving distributed optimization problems. Our application is supported by our analysis in the derivation and convergence of the optimization dynamics along with case studies in using the approach to solve a multi-agent target tracking problem. In future work, we will consider the use of this method for handling extremely high dimensional optimization problems, such as those found in the optimization of deep neural networks.

ACKNOWLEDGEMENTS

S. Al-Abri, T. Lin, and F. Zhang were supported by ONR grants N00014-19-1-2556 and N00014-19-1-2266; NSF grants OCE-1559475, CNS-1828678, and S&AS-1849228; NRL grants N00173-17-1-G001 and N00173-19-P-1412 ; and NOAA grant NA16NOS0120028.

VIII. PROOFS OF THE OPTIMIZATION DYNAMICS

Proof: [Proof of Theorem 4.1] Taking the time-derivative of (5), we obtain

$$\dot{C} = \sum_{k=1}^M [(\dot{\mathbf{r}}_k - \dot{\mathbf{r}}_c)(\mathbf{r}_k - \mathbf{r}_c)^\top + (\mathbf{r}_k - \mathbf{r}_c)(\dot{\mathbf{r}}_k - \dot{\mathbf{r}}_c)^\top]. \quad (24)$$

However, $\dot{\mathbf{r}}_c = \frac{1}{M} \sum_{k=1}^M \dot{\mathbf{r}}_k = \frac{1}{M} \sum_{k=1}^M \mathbf{u}_k$. This implies that

$$\dot{\mathbf{r}}_k - \dot{\mathbf{r}}_c = \frac{1}{M} \sum_{l=1}^M (\mathbf{u}_k - \mathbf{u}_l). \quad (25)$$

Substituting (25) into (24), and using the fact that $\sum_{k=1}^M \mathbf{u}_c(\mathbf{r}_k - \mathbf{r}_c)^\top = \mathbf{u}_c \sum_{k=1}^M (\mathbf{r}_k - \mathbf{r}_c)^\top = 0$, we obtain

$$\dot{\mathbf{C}} = \sum_{k=1}^M \left[\mathbf{u}_k(\mathbf{r}_k - \mathbf{r}_c)^\top + (\mathbf{r}_k - \mathbf{r}_c) \mathbf{u}_k^\top \right]. \quad (26)$$

Moreover, by definition, for any eigenvectors $\mathbf{v}_a$ and $\mathbf{v}_b$ of the covariance $\mathbf{C}$, we have $\mathbf{C}\mathbf{v}_a = \lambda_a \mathbf{v}_a$ and $\mathbf{C}\mathbf{v}_b = \lambda_b \mathbf{v}_b$. Taking the derivative, we obtain $\dot{\mathbf{C}}\mathbf{v}_a + \mathbf{C}\dot{\mathbf{v}}_a = \dot{\lambda}_a \mathbf{v}_a + \lambda_a \dot{\mathbf{v}}_a$. Taking the inner product on both sides, we obtain

$$\langle \mathbf{v}_b, \dot{\mathbf{C}}\mathbf{v}_a \rangle + \langle \mathbf{v}_b, \mathbf{C}\dot{\mathbf{v}}_a \rangle = \dot{\lambda}_a \langle \mathbf{v}_b, \mathbf{v}_a \rangle + \lambda_a \langle \mathbf{v}_b, \dot{\mathbf{v}}_a \rangle. \quad (27)$$

Since $\mathbf{C}$ is symmetric, then $\dot{\mathbf{C}}$ is also symmetric. This implies that $\langle \mathbf{v}_b, \mathbf{C}\dot{\mathbf{v}}_a \rangle = \langle \mathbf{C}\mathbf{v}_b, \dot{\mathbf{v}}_a \rangle = \lambda_b \langle \mathbf{v}_b, \dot{\mathbf{v}}_a \rangle$. Using this along with the fact that $\langle \mathbf{v}_b, \mathbf{v}_a \rangle = \langle \mathbf{v}_a, \mathbf{v}_b \rangle = 0$, we obtain from (27)

$$\langle \mathbf{v}_b, \dot{\mathbf{v}}_a \rangle = \frac{1}{\lambda_a - \lambda_b} \langle \mathbf{v}_b, \dot{\mathbf{C}}\mathbf{v}_a \rangle. \quad (28)$$

Since $\mathbf{C}$ is symmetric, one can always find a complete set of orthogonal eigenvectors $\{\mathbf{v}_1, \dots, \mathbf{v}_d\}$. Therefore, for all $a = 1, \dots, d$, we may write $\dot{\mathbf{v}}_a = \sum_{b \neq a} \langle \mathbf{v}_b, \dot{\mathbf{v}}_a \rangle \mathbf{v}_b$. This implies that the dynamics of the a -th eigenvector are

$$\dot{\mathbf{v}}_a = \sum_{b \neq a} \frac{1}{\lambda_a - \lambda_b} \langle \mathbf{v}_b, \dot{\mathbf{C}}\mathbf{v}_a \rangle \mathbf{v}_b. \quad (29)$$

Substituting (26) into (29), and using (8) for $\kappa_{a,b}$, we obtain the desired result (7). ■

Proof: [Proof of Theorem 4.2] Let $w_{ij} = (\|\mathbf{r}_j - \mathbf{r}_i\|^2 - d_{ij}^2)$, where $w_{ij} = w_{ji}$. Let $\mathbf{s}_i = \sum_{j \in \mathcal{N}_i} w_{ij}(\mathbf{r}_j - \mathbf{r}_i)$. Then, we can write (6) as:

$$\mathbf{u}_i = k_1 z_i \mathbf{v}_1 + k_2 \mathbf{s}_i. \quad (30)$$

Substituting (30) into (8) yields the desired result (10). ■

Proof: [Proof of Theorem 4.3] For $a = 1$, using (9) and (10), we obtain

$$\dot{\mathbf{v}}_1 = \sum_{b \neq 1} \kappa_{1,b} \mathbf{v}_b, \quad (31)$$

where

$$\begin{aligned} \kappa_{1,b} = & \frac{k_1}{\lambda_1 - \lambda_b} \sum_{k=1}^M (z_k - z_c) \langle \mathbf{r}_k - \mathbf{r}_c, \mathbf{v}_b \rangle \\ & + \frac{k_2}{\lambda_1 - \lambda_b} \mathbf{v}_1^\top \sum_{k=1}^M \left[(\mathbf{r}_k - \mathbf{r}_c) \mathbf{s}_k^\top + \mathbf{s}_k (\mathbf{r}_k - \mathbf{r}_c)^\top \right] \mathbf{v}_b. \end{aligned} \quad (32)$$

Let S be as defined in (13). Then, we may write (31) and (32) as

$$\dot{\mathbf{v}}_1 = \left(\sum_{b \neq 1} \frac{\mathbf{v}_b \mathbf{v}_b^\top}{\lambda_1 - \lambda_b} \right) \left(k_1 \sum_{k=1}^M (z_k - z_c) (\mathbf{r}_k - \mathbf{r}_c) + k_2 S \mathbf{v}_1 \right). \quad (33)$$

Using (11) along with Assumption 4.1, we obtain

$$\sum_{k=1}^M (z_k - z_c) (\mathbf{r}_k - \mathbf{r}_c) = \mathbf{C} \nabla z_c, \quad (34)$$

Finally, substituting (34) into (33) yields the desired result (12). ■

REFERENCES

- [1] Huma Ghafoor and Youngtae Noh. An overview of next-generation underwater target detection and tracking: An integrated underwater architecture. *IEEE Access*, 7:98841–98853, 2019.
- [2] Xiaoyu Cai and Marcio de Queiroz. Formation maneuvering and target interception for multi-agent systems via rigid graphs. *Asian journal of control*, 17(4):1174–1186, 2015.
- [3] Lebin Liu, Chaomin Luo, and Furoo Shen. Multi-agent formation control with target tracking and navigation. In *2017 IEEE International Conference on Information and Automation (ICIA)*, pages 98–103. IEEE, 2017.
- [4] Robert K Lee, Christopher A Kitts, Michael A Neumann, and Robert T McDonald. 3-d adaptive navigation: Multirobot formation control for seeking and tracking of a moving source. In *2020 IEEE Aerospace Conference*, pages 1–12. IEEE, 2020.
- [5] Junchong Ma, Huimin Lu, Junhao Xiao, Zhiwen Zeng, and Zhiqiang Zheng. Multi-robot target encirclement control with collision avoidance via deep reinforcement learning. *Journal of Intelligent & Robotic Systems*, pages 1–16, 2019.
- [6] Zheping Yan, Xiangling Liu, Jiajia Zhou, and Di Wu. Coordinated target tracking strategy for multiple unmanned underwater vehicles with time delays. *IEEE Access*, 6:10348–10357, 2018.
- [7] Timothy H Chung, Vijay Gupta, Joel W Burdick, and Richard M Murray. On a decentralized active sensing strategy using mobile sensor platforms in a network. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 2, pages 1914–1919. IEEE, 2004.
- [8] Allison Ryan and J Karl Hedrick. Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems*, 58(5):574–584, 2010.
- [9] Zongyao Wang and Dongbing Gu. Cooperative target tracking control of multiple robots. *IEEE Transactions on Industrial Electronics*, 59(8):3232–3240, 2011.
- [10] Lili Ma and Naira Hovakimyan. Cooperative target tracking in balanced circular formation: Multiple uavs tracking a ground vehicle. In *2013 American Control Conference*, pages 5386–5391. IEEE, 2013.
- [11] Lara Briñón-Arranz, Alexandre Seuret, and António Pascoal. Circular formation control for cooperative target tracking with limited information. *Journal of the Franklin Institute*, 356(4):1771–1788, 2019.
- [12] Said Al-Abri, Wencen Wu, and Fumin Zhang. A gradient-free three-dimensional source seeking strategy with robustness analysis. *IEEE Transactions on Automatic Control*, 64(8):3439–3446, 2018.
- [13] Said Al-Abri, Tony X Lin, Molei Tao, and Fumin Zhang. A derivative-free optimization method with application to functions with exploding and vanishing gradients. *IEEE Control Systems Letters*, 5(2):587–592, 2020.
- [14] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt. The Robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1699–1706, May 2017.
- [15] Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 528–537, 2018.
- [16] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2345–2353, 2018.
- [17] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.
- [18] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *UK Workshop on Computational Intelligence*, pages 191–202. Springer, 2018.
- [19] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.
- [20] Hassan K Khalil. *Nonlinear systems*. Prentice-Hall, 2002.
- [21] SN Dashkovskiy, Denis V Efimov, and Eduardo D Sontag. Input to state stability and allied system properties. *Automation and Remote Control*, 72(8):1579, 2011.