A Swarm Robotic Approach to Inspection of 2.5D Surfaces in Orbit

Bahar Haghighat[1†], Julia Ebert[1], Johannes Boghaert[1], Ariel Ekblaw[2], and Radhika Nagpal[1]

[1]John A. Paulson School of Engineering and Applied Science, Harvard University, Boston, Massachusetts
(bahar@seas.harvard.edu, ebert@g.harvard.edu, jboghaert@seas.harvard.edu, nagpal@g.harvard.edu)
[2]Media Lab Space Exploration Initiative, Massachusetts Institute of Technology, Cambridge, Massachusetts
(aekblaw@mit.edu)

Abstract: Robotic inspection offers a robust, scalable, and flexible alternative to deploying fixed sensor networks or human inspectors. While prior work has mostly focused on single robot inspections, this work studies the deployment of a swarm of inspecting robots on a simplified surface of an in-orbit infrastructure. The robots look for points of mechanical failure and inspect the surface by assessing propagating vibration signals. In particular, they measure the magnitude of acceleration they sense at each location on the surface. Our choice for sensing and analyzing vibration signals is supported by the established position of vibration analysis methods in industrial infrastructure health assessment. We perform simulation studies in Webots, a physics based robotic simulator, and present a distributed inspection algorithm based on bio-inspired particle swarm optimization and evolutionary algorithm niching techniques to collectively localize an *a priori* unknown number of mechanical failure points. To perform the vibration analysis and obtain realistic acceleration data, we use the ANSYS multi-physics simulation software and model mechanical failure points as vibration sources on the surface. We deploy a robot swarm comprising eight robots of 10-cm size that use a bio-inspired inchworming locomotion pattern. The swarm is deployed on 2.5D (that is curved 2D) cylindrical surfaces with and without obstacles to investigate the robustness of the algorithm in environments with varying geometric complexity. We study three performance metrics: (1) proximity of the localized sources to their ground truth locations, (2) time to localize each source, and (3) time to finish the inspection task given an 80% surface coverage threshold. Our results show that the robots accurately localize all the failure sources and reach the coverage threshold required to complete the inspection. This work demonstrates the viability of deploying robot swarms for inspection of potentially complex 3D environments.

Keywords: Swarm robotics, robotic inspection, vibration sensing, target search, niching, particle swarm optimization (PSO)

1. INTRODUCTION

Many industries, such as agriculture, bridge maintenance, and wind turbine maintenance are actively investing in robotic inspection solutions [1, 2]. Robotic inspection aims to reduce the risk, cost, and possible service downtime associated with inspection of infrastructure by supporting human inspection through reducing the need to deploy humans. Deploying robots becomes particularly useful when inspection must be carried out in dangerous conditions or over extended periods of time. Human inspectors can then be deployed only after the inspector robots have identified and localized existing issues that require further action.

Robotic inspection can be highly beneficial for supporting long-term deployments of space infrastructure [3]. Across a long deployment time, wear and tear on the deployed structures becomes non-negligible. It is then valuable to be able to identify and mend damages before they become a source of major structural failure. The operation of the International Space Station (ISS) is an example scenario. The ISS has now been in operation in orbit for over two decades. As the structure ages, failures are arising [4]. In the near future, this could also apply to the Lunar Gateway space station envisioned by NASA in the context of the Artemis program. The program, launched in 2017, involves deployment of a Base Camp on the Moon and the Lunar Gateway space station in lunar orbit to support long-term science and human exploration, by the year 2025. Regular inspection can help extend the lifetime of such long-term infrastructure deployments.

Vibration sensing and analysis methods are well studied for evaluating structural integrity [5]. The underlying theoretical methods are based on the vibration response or modal analysis of an *a priori* known structure [6, 7]. Given the nondestructive and noninvasive nature of vibration-based inspection systems, they can be safely applied to sensitive or damaged structures [8]. Depending on the vibration signal profile, the exact signal processing and failure identification method may differ [9]. Standard monitoring approaches typically rely on a large set of static sensors with fixed sampling rates [10]. Companies such as Gecko Robotics and Waygate Technologies offer solutions for vibration-based inspection of structures by deploying single unit robots that move over the infrastructure to cover the inspection area [11, 12].

While an automated inspection task may be performed by a single mobile robotic unit (deployed post-construction) or by a fixed network of sensors (deployed pre- or post-construction), there are multiple benefits in employing a swarm of robots. Swarms are known for being resilient to failure of the individual comprising units. Compared to fixed sensor networks – which are typical for many environmental monitoring applications because of their ease of deployment – swarms can provide more dynamic and flexible coverage performances in surveying and inspection applications [13]. To achieve a low-cost swarm deployment, minimizing the complexity and cost of the individual robotic units in the swarm is critical. This drive for simplicity in individual robotic units has been the motivation behind employing

† Bahar Haghighat is the presenter of this paper.

simplistic bio-inspired robot designs and algorithms.

The inspection task that we plan to undertake can be formulated as a generalization of the source localization task, which is a well-studied problem [14-16]. Source localization typically comprises three components: (i) finding a cue, (ii) tracing the cue and localizing a source, and (iii) confirming a localized source. The inspection task can then be formulated as a repetition of source localization until a termination condition is met. Thus, it comprises two main search behavioral components: a *local search* behavior to localize a new source in the search space and a *global search* behavior to maximize exploration and coverage of the search space.

For the *global search* behavioral component, coverage can be maximized through (i) a random exploration or (ii) a structured exploration of the search space. Lévy flights and Brownian motion random walks can provide a random exploration of the search space [17-19]. For the structured exploration approaches, it is shown that the problem of optimizing the coverage using a robot swarm for an *a priori* known environment is NP-complete [20]. The basic lawnmower problem in an unobstructed environment and the traveling salesman problem are also shown to be NP-hard [21, 22]. These problem types mean that there is no guaranteed method to determine the optimal solution for covering the search space, however, near optimal solutions are possible [20].

For the *local search* behavioral component, a taxonomy of source localization techniques is presented in [23]. Three main categories of search methods can be identified: (i) reactive methods, (ii) heuristic cognitive methods, and (iii) probabilistic cognitive methods. Reactive search methods, such as gradient-based and bug algorithms, guide the search by relying solely on the latest observations available to the robots. These methods are typically simple and require little memory and computational resources [24-26], but have been shown to perform poorly in complex scenarios [14, 16]. Cognitive methods, on the other hand, combine incoming observations with previously gathered information in order to guide the search [23]. Heuristic cognitive search methods see the source localization problem as an optimization problem. The objective function that needs to be optimized can then be, for instance, the gas concentration detected by the robots [23]. Heuristic methods lend themselves well to multi-robot search scenarios [23]; by design, their mathematical optimization counterparts typically deploy multiple agents or candidate solutions that move in the search space and are evaluated over time. The most known bio-inspired examples of such optimization techniques are the Particle Swarm Optimization (PSO) [27] and the Cuckoo Search (CS) methods [28]. PSO-based multi-robot search implementations have been studied in [29, 30]. Probabilistic cognitive search methods use probabilistic inference to derive the distribution of the cue in the search space, maximized at the location of the source, based on the gathered information [23, 26]. This, however, requires the knowledge of a dispersion model for the given cue and environment [23]. The inference methods are often based on the Bayesian inference framework, examples of which are Hidden Markov Models [31] and Particle Filters (PFs) [32]. Another notable example is infotaxis, which is based on an entropy-reduction principle [33]. Probabilistic cognitive search methods are applicable only as long as their underlying model assumptions hold and accurate cue dispersion models are available. In odor source localization problems where gas dispersion models are available, these search methods are widely studied. However, in failure source localization using vibration cues, they remain less applicable.

We believe that small-scale low-cost vibration sensing robot swarms hold a great potential for a variety of structural integrity inspection tasks. In this work, we use a bio-inspired inchworming robot model and present a swarm inspection algorithm based on the Particle Swarm Optimization (PSO) search method – a well-studied and widely applied technique in multi-robot search scenarios – employed on the swarm. In particular, our work puts forward four main contributions.

- **Localizing an *a priori* unknown number of failure sources:** Multi-source localization is an active topic of research [34, 35]. Inspection tasks have the additional challenge that the number of failure sources is *a priori* unknown. To cope with this problem and define a task termination condition, we employ a heuristic strategy as well as a coverage certainty grid map.
- **Relying on vibration signals:** Techniques for vibration sensing are well-studied for structural inspection applications [5]. However, compared to other methods such as odor sensing, vibration sensing remains under-addressed in autonomous inspection swarms. In this work, we use a realistically modeled vibration signal (using ANSYS software) that is applied to simplified spacecraft surface sections.
- **Traversing curved surfaces:** 2.5D environments are largely unexplored in source localization and target search scenarios in the literature. Curved surfaces pose challenges for robot locomotion and can alter the cue propagation through the structure, depending on the geometrical complexity of the environment. We also consider obstacles representing spacecraft features in our modeling, which pose additional challenges to robot locomotion and cue propagation.
- **Using bio-inspired inchworm robot concept:** In simulation, soft-bodied robots are typically more challenging to model and control than their rigid-bodied counterparts, but physical soft-bodied robots are generally simpler to control compared to rigid-bodied robots. This is due to the large mechanical coupling of the degrees of freedom in their soft structure. Multiple soft-bodied inchworming robot designs have been developed in the literature [36, 37]. In this work, we consider a simulated bio-inspired inchworm-like soft-bodied robot model. The swarm algorithm that we develop is then deployed on a swarm of size $N = 8$ inchworming robots. We specifically adapt our algorithm for execution on the inchworming robot model, but without loss of generality; our proposed swarm inspection algorithm can be deployed on any robot capable of traversing the target surface.

2. PROBLEM STATEMENT

The inspection task that we undertake in this paper can be formally defined as the repeated localization of any multitude

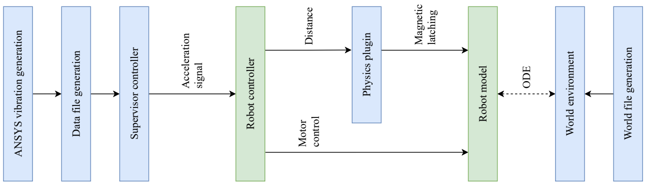

Fig. 1: The pipeline structure of our simulation framework. Vibration signals are modeled offline in ANSYS and used to generate signals sensed in the Webots robot simulation.

of failure sources on a 2.5D surface (a 2D curved surface) in orbit, using a swarm of robots that rely on sensing the vibration signal as a cue, until a termination condition is met.

A *failure source* is defined as a feature that disturbs the normal functioning of a system. Detecting a failure source requires knowledge of the functional state of the system. We expect that failure sources such as cracks and fissures on the surface can result in creation of specific vibration signal profiles that are detectable in the presence of the endemic vibration energy of the system. In our model of the failure sources, we further simplify the points of mechanical failure as sources of induced vibration. We model the vibration source as an external force applied to the surface following a sinusoidal pattern at a frequency of $1\mathrm{Hz}$ and amplitude of $1\mathrm{N}$, which falls within the mid-frequency range of the vibratory regime of the ISS [38]. The amplitude is chosen such that the resulting acceleration values are within the ISS acceleration spectrum ranging from below a micro-g to 10 milli-g [38].

The *cue* is the acceleration signal measured during the search. For spacecraft surface inspection, we hypothesize that the cue can be an endemic or induced vibration signal.

3. SIMULATION FRAMEWORK

The simulation framework serves as the virtual environment in which we deploy and study our inspecting robot swarm. Two main software components are used: the ANSYS software, which we use for creating realistic vibration signals propagating on a surface that models a floating structure in orbit, and the Webots robotic simulator [39], which we use for simulating the locomotion of our robots, their sensors, actuators, and the surface that they traverse. The overall pipeline of our simulation framework is shown in Fig.1.

Within Webots, we created three main components: (i) a bio-inspired robot model that comprises a multi-segment piece-wise approximation of a flexible-bodied robot of size $20\times100\times7\mathrm{mm}$, which uses an inchworming locomotion pattern, (ii) target surfaces that the robots move on and inspect, and (iii) a supervisor controller code that is responsible for passing on the vibration data to the robots as they move on the surface, depending on their location on the surface at each simulation step. The supervisor controller emulates a black box that contains an acceleration sensor along with a processing unit that returns to the robots the maximum observed acceleration amplitude at each location on the surface. The robot model is the most complex of the three components. For the sake of brevity, and since this is not the focus of the

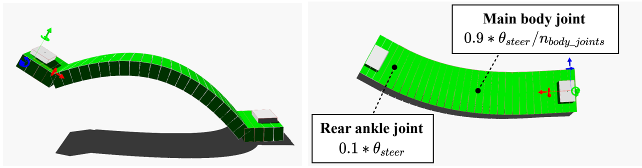

Fig. 2: Bio-inspired inchworming robot model in Webots.

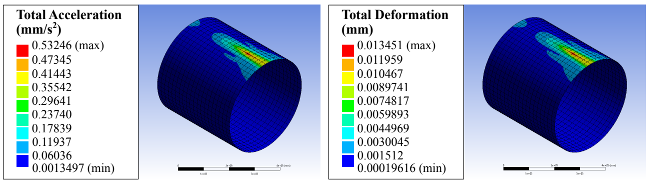

Fig. 3: Surface models with elastic support boundary conditions in ANSYS.

work here, we explain two main features of the robot model: (i) the inchworming locomotion pattern and (ii) the sensing and communication capabilities of the simulated robot.

The robot model, as shown in Fig. 2, is capable of moving on flat and curved surfaces. The simulated robot has two connectors on its feet that approximate switchable magnetic connectors. The main robot body is a simulated flexible structure that bends along its main section to allow moving forward and steering to left and right, and along the two ankle joints to allow lifting the main body and adjusting the position of the front foot relative to the next landing surface. The inchworming locomotion pattern allows the robot to achieve a step size of about half its body size, roughly $5\mathrm{cm}$. The robots are assumed to have knowledge of the map of the environment as well as their location using a global positioning sensor, in particular the location of the walls and obstacles. A loss-free global communication channel is assumed in between the robots. The robots use communication to share their locations on the map, which is then used to perform collision avoidance with other robots.

Within ANSYS, we use the Transient Analysis to subject the surface model to a transient sinusoidal load case of $1\mathrm{N}$ at $1\mathrm{Hz}$ representing the vibration source. To represent the placement of the surface in orbit, we use an elastic support boundary condition that involves the notion of foundation stiffness expressed in N/mm^3. This is typically used to model soil supported or submersed structures. We empirically set the foundation stiffness parameter to $0.0001\mathrm{N}/\mathrm{mm}^3$ by running a series of simulations and qualitatively evaluating the results in discussion with a human expert. The deformation amplitude for the applied load case is $0.013\mathrm{mm}$. Fig. 3 shows the surface models. In order to simplify the data processing and export, we create text files that approximate the acceleration signal observed in ANSYS with 2D Gaussian distributions. This data is then retrieved by the supervisor controller code in Webots and passed on to the robots depending on their location on the surface at each simulation step, as shown in the block diagram structure in Fig. 1.

Algorithm 1 Inspection Algorithm Overview

run Lévy Random Walk (`RW`) ▷ Initialize
while $coverage < 80\%$ **do**
  if $robot$ in collision **then** ▷ Collision avoidance
    run Collision Avoidance (`CA`)
  else if cue picked up **then** ▷ Local search
    run Particle Swarm Optimization (`PSO`)
    if cue is a $source$ **then** ▷ Source confirmation
      declare $source$
      run Directed Walk (`DW`) ▷ Re-initialization
      return to Lévy Random Walk (`RW`)
    end if
  else
    run Lévy Random Walk (`RW`) ▷ Global search
  end if
  update $coverage$ ▷ Update coverage
end while

4. PROPOSED ALGORITHM

The inspection task, as we formally defined in Section 2, can be interpreted as a generalization of the source localization problem, which is well studied in the literature [14-16, 40]. The source localization problem can be divided into three main sub-problems: (i) finding a cue, (ii) tracing the cue and localizing the source, and (iii) confirming the source location. Our formal definition of the multi-source inspection task involves a repetition of these steps as well as a termination condition based on a predefined coverage threshold.

Knowing that source confirmation and reaching the coverage threshold require local and global search behaviors, respectively, we define two goals for our inspection algorithm: (i) finding all the sources present in the search space as fast as possible, and (ii) reaching the global coverage threshold required to complete the inspection task as fast as possible. In the following, we describe our choice for the local search, i.e. the behavioral component that results in localizing a source in the environment, and the global search, i.e. the behavioral component that results in exploring the environment and reaching the coverage threshold termination condition.

The overall algorithm structure is shown in Algorithm 1. There are four main control states in the algorithm, which we cover briefly in this paragraph and detail further below. In the absence of any prior sensing of a cue, the robots start in the Random Walk (RW) state, performing an unbiased Lévy random walk around the environment until they sense a cue. Upon sensing a cue, the robot will start performing a biased random walk in the Particle Swarm Optimization (PSO) state, and moving towards the source. Once a robot is finished localizing a source, it starts in the Directed Walk (DW) state and moves to an unexplored area in the environment. The robot will start in the Collision Avoidance (CA) state at any point in time if it is closer than a threshold distance to a static obstacle in the environment or to another moving robot.

We use a multi-modal variation of the PSO algorithm combined with a niche formation behavior as our heuristic local search component. The PSO velocity update is as follows. For each individual i and dimension j we have:

$$\begin{aligned} v_{i,j}^{t} = \omega * v_{i,j}^{t-1} &+ c_1 * \mathrm{rnd}()^t \times \left(p_{\mathrm{best}_{i,j}} - \mathrm{pos}_{i,j}^{t-1}\right) \\ &+ c_2 * \mathrm{rnd}()^t \times \left(g_{\mathrm{best}_{i,j}} - \mathrm{pos}_{i,j}^{t-1}\right) \end{aligned} \quad (1)$$

where p_{best} and g_{best} are the positions of the best values observed by the individual and group, respectively. The inertia term ω and c_1, c_2 are weights to balance exploration and exploitation in order to find the optimal value of the function. We use the parameters $\omega = 0.15$, $c_1 = 0.35$, $c_2 = 0.5$. To account for the constraints of the specific robot model, we introduce restrictions on the velocity (determined by the gait cycle of roughly 3s) and the steering angle (0.8 radians) that is required by the algorithm for the robot to execute. The steering angle is passed on to the robot controller.

Niche formation is part of the local search behavior and allows for confirming an identified source location. In particular, once a robot is in the vicinity of a source and starts the PSO state, it forms a two-robot niche by recruiting its nearest neighbor. The recruited robot then starts moving towards the location of the identified source. Once it perceives a cue, it performs a biased PSO walk towards the source.

After localizing a source, a robot engages in a directed walk behavior, moving towards unexplored parts of the environment. This is achieved by using a sliding window approach to identify the least covered areas and then performing a roulette wheel sampling where the likelihood of selecting a less covered goal position increases quadratically.

Collision avoidance is done based on the Artificial Potential Field (APF) method using (i) a map of the environment in which the boundaries of the arena and the obstacles are marked and (ii) by communicating with other robots and obtaining their location on the map. Each obstacle then contributes a repulsion term to update the velocity of the robot. For a repulsive term i in dimension j we have:

$$r_{i,j} = c_{\mathrm{weight}} \times \left[\frac{1}{d_i} - \frac{1}{\mathrm{threshold}_i}\right] \times \left[\frac{\mathrm{pos}_j - \mathrm{point}_{i,j}}{d_i^3}\right] \quad (2)$$

where d_i is the distance from the robot to obstacle i, pos_j is the robot's position in dimension j and $\mathrm{point}_{i,j}$ is the closest point on obstacle i in dimension j. The threshold is the distance to the obstacle below which the robot will engage in collision avoidance. Beyond this range, no collision avoidance is performed with the obstacle. Threshold and weight values can be different depending on the type of obstacle.

We require a coverage guarantee to ensure that, given enough time, all the sources that are present in the search space will be found. We deploy a Lévy random walk for the global search behavioral component. The Lévy random walk assigns a random orientation (angle) to the robot and a random step length (magnitude), following a Lévy distribution. This exploratory random walk guarantees full coverage of the search space asymptotically. We terminate the inspection earlier, based on a predefined coverage threshold value that is calculated based on a coverage map.

All robots have access to a shared coverage map, which is used to compute the covered area and check the coverage threshold condition. This map is represented as a grid-based map of $10 \times 10\mathrm{cm}$ cells. As the robots move through the environment sensing the value of the vibration signal and confirm

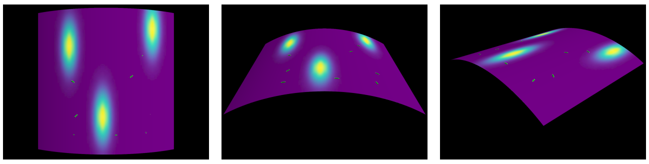

Fig. 4: `Webots` simulation environment for Scenario I.

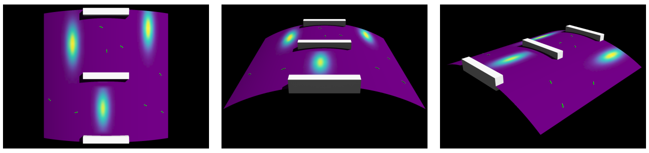

Fig. 5: `Webots` simulation environment for Scenario II.

new source locations, they update this shared map. The map is updated using a simplified sensor model; the sensor model is a two dimensional Gaussian Probability Density Function (PDF). To update the coverage map based on a single robot observation, the sensor model PDF is superimposed on the coverage map, centered around the reporting robot, and the maximum certainty value is updated on the map.

5. SIMULATION EXPERIMENTS

This section describes our experimental objectives and the two simulation setups used in our simulation experiments.

5.1. Experimental Objectives

We consider three main aspects of the swarm performance when analyzing the dynamics of the robot swarm while conducting the inspection task and to evaluate the swarm performance. These aspects are different from, but related to the performance metrics that we describe in the following paragraph. In particular, in each scenario we would like to see that the swarm succeeds (i) in localizing all the sources (localization success), (ii) in reaching the coverage threshold for terminating the inspection (termination success), and (iii) that all of the robots in the swarm manage to maneuver around in the search space, without getting lost or stuck, following the cue on the surfaces to the source locations while avoiding obstacles (maneuverability success).

To quantify the swarm performance on these three aspects, we consider the following performance metrics, by taking inspiration from similar metrics used in the fields of source localization and target search [35, 41]. In each scenario, we quantify (i) the source localization accuracy or the proximity of a confirmed source location to the ground truth location of the source, (ii) the time to find each source present in the search space, and (iii) the time to reach the coverage threshold termination criterion. To gain more insight into the dynamics of the inspecting swarm, we will look at the time the robots spend in each of the four main control states of Collision Avoidance (CA), Particle Swarm Optimization (PSO), Random Walk (RW), and Directed Walk (DW).

5.2. Experimental Scenarios

The real-world inspection task that underlies our developments is a complex one. Within the scope of this work, we simplify the problem to two simulation studies, described below. In each case, we deploy a swarm of size $N = 8$.

Scenario I comprises a curved 2.5D cylindrical surface with projected flat dimensions of $4 \times 4\mathrm{m}$. The ANSYS simulations involves a full cylindrical surface of 2mm thickness, 4m radius, and 6m axial length. The sources of vibration are at locations $(2\mathrm{m}, 3\mathrm{m})$, $(1\mathrm{m}, 1\mathrm{m})$, and $(3.5\mathrm{m}, 0.5\mathrm{m})$ on the projected surface reference frame. The entire surface is subject to a foundation stiffness of $1 \times 10^{-4} \frac{\mathrm{N}}{\mathrm{mm}^3}$, and the mesh is sized uniformly with nodes of $100 \times 100\mathrm{mm}$. At the location of the vibration source, we apply a load case with a sinusoidal of amplitude 1N and frequency 1Hz. The peak amplitude at steady state at each location, i.e. after roughly 9.75s, is then considered for constructing the 2D Gaussian signal used in Webots (see Section 3). The ANSYS simulations revealed that the vibration propagation on a cylindrical surface is strongly biased along its length.

Scenario II is an extension of Scenario I; we further increase the geometrical complexity of the search space by introducing three rectangular obstacles representing features such as ridges or add-on sections on the surface. The obstacles are expected to affect both the propagation of the vibration signal on the surface and the movement of the robots.

6. RESULTS

In this Section we present and discuss the simulation results obtained from 10 trials of each experimental scenario detailed in Section 5. The random seed for each robot is fixed but the starting positions are randomized in each run. By running a number of simulation experiments with different swarm sizes and observing the effect of the robot density in the environment on the inspection performance, we chose the $N = 8$ swarm size. For the sake of brevity, those studies are not discussed here. Looking at the overall performance of our algorithm between the two experimental scenarios, we note how the swarm performance changes as the level of complexity in the search space increases.

The results are shown in Fig. 6. Three main metrics are considered, namely the source localization accuracy; the time to localize each source and reach the coverage threshold; and the time spent in each of the four main control states. The complexity of the search space increases from Scenario I to Scenario II. This increase in complexity clearly affects the inspection completion time as defined by the time the 80% coverage threshold is reached and the time before each source is discovered (Fig. 6b,e). This can also be noted by comparing the time spent in the RW control state among the two scenarios (Fig. 6c,f). We can explain the variation in source localization accuracy (Fig. 6a,d) by considering three main factors. First, the more time the robots spend in PSO control state versus the CA control state, the higher the chances will be that they achieve a better localization of the source. Second, the parameters of the PSO algorithm determine the way robots take advantage of their own and also other robots' measurements of the cue to find their way to the source. These parameters are not optimized at the moment and we can hypothesize that they may depend on the over-

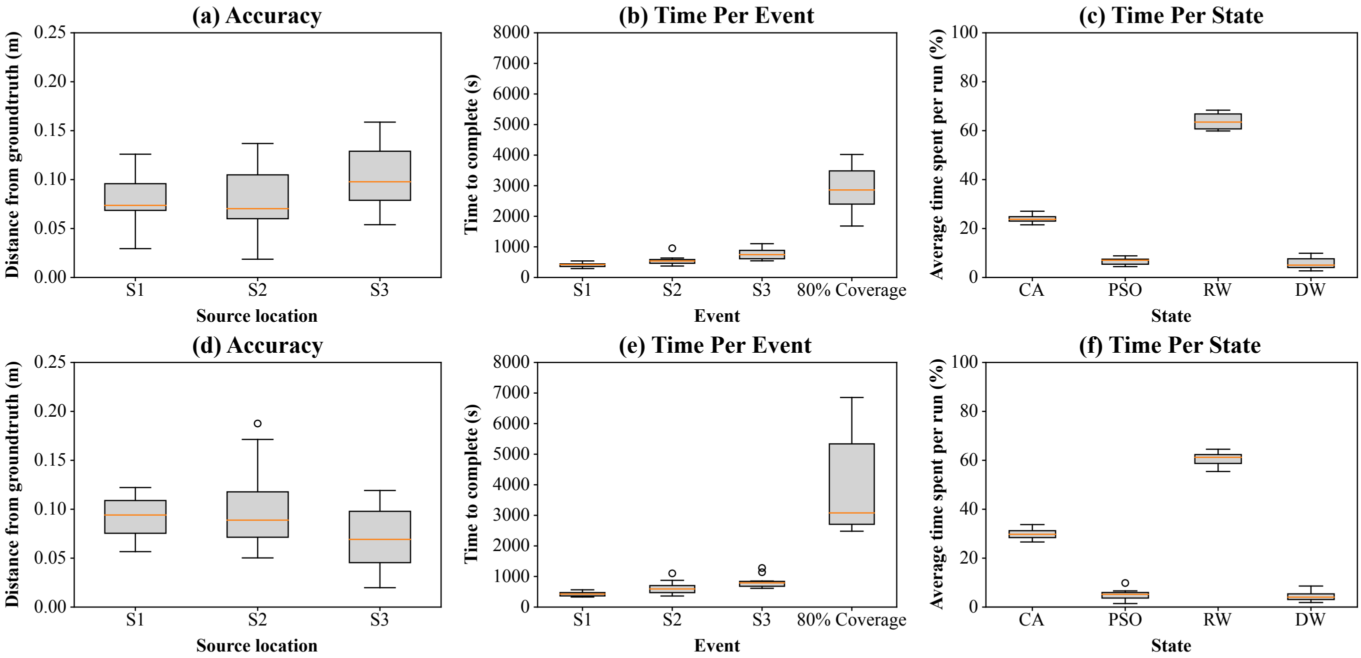

Fig. 6: Performance results for Scenario I (top row) and for Scenario II (bottom row).

all geometry that defines the search space. Lastly, the interplay between the step size of the surveying robots, the shape and spread of the cue, and the placement of the source in the arena play a significant role in how accurately a source can be localized. We hypothesize that by optimizing the PSO parameters and the step size parameter for a given search space, source localization accuracy can be enhanced.

7. CONCLUSION

In this work, we presented a swarm robotic inspection approach for localizing multiple sources of mechanical failure on a 2.5D surface in orbit by sensing vibration signals. The particular real world scenario that underlies our developments is the need for inspection of the outer surface of long-term deployed spacecrafts. The goal of the inspection is to identify mechanical failures such as fissures and cracks resulting from natural wear and tear of the structures.

Our work here focused on development and presentation of a modeling and algorithmic framework for the surface inspection task using a swarm of robots. We deployed our proposed algorithm on a simulated swarm of vibration sensing surface-crawling robots that use an inchworming gait for locomotion. Within the simulated world where the robots perform the inspection task, we modeled the points of mechanical failure on the surface under inspection as sources of vibration. The robots then used the signal that propagates through the surface as a cue for localizing the sources of vibration. We simulated realistic vibration signal propagation in ANSYS, then simplified data transfer by fitting 2D Gaussian functions to the simulation results. In the Webots robotic simulator, we investigated the performance of the swarm within two experimental scenarios comprising three sources on a 2.5D cylindrical surface and three sources on a 2.5D cylindrical surface with additional obstacles on the surface. Our algorithm succeeded at finding all the sources present in the search space in each of the two experimental scenarios. Additionally, we showed that we are able to reach a predefined coverage threshold as a termination criterion for the inspection task. Our results provide evidence supporting the viability of robot swarms for inspection of 2.5D surfaces based on sensing vibration cues on the surface.

Future work will involve extending our modeling and algorithmic framework in several ways. First, we plan to develop a fully automated simulation pipeline to facilitate randomized studies of a variety of environments. In particular, we plan to automate the currently manual process of simulating the vibration signal within ANSYS and transferring the corresponding data to a file format that is accessible by the simulated robots within Webots. Second, we plan to implement realistic constraints in the communication range and bandwidth used by the simulated robots within Webots. Third, given a specific search environment, we plan to leverage the automated simulation framework developed in this work to perform a parameter optimization in order to find the set of parameters (i.e., number of robots, local search behavior parameters, global search behavior parameters, obstacle avoidance behavior parameters, etc.) that result in the best desired performance metrics.

Our hope is that this work inspires further studies of robot swarms as a robust, scalable, and low-cost solution for a wider variety of structural inspection applications.

ACKNOWLEDGEMENT

We thank Dr. Harald Wild at ETH Zürich for his help with the ANSYS simulations. This was supported by a Swiss National Science Foundation fellowship P400P2_191116 (B. Haghighat) and a National Aeronautics and Space Administration (NASA) grant 80NSSC21K0353 (A. Ekblaw).

REFERENCES

[1] C. Carbone, O. Garibaldi, and Z. Kurt, "Swarm Robotics as a Solution to Crops Inspection for Precision Agriculture," *KnE Engineering*, vol. 3, no. 1, p. 552, Feb. 2018.

[2] D. Carrillo-Zapata, E. Milner, J. Hird, G. Tzoumas, P. J. Vardanega, M. Sooriyabandara, M. Giuliani, A. F. T. Winfield, and S. Hauert, "Mutual Shaping in Swarm Robotics: User Studies in Fire and Rescue, Storage Organization, and Bridge Inspection," *Frontiers in Robotics and AI*, vol. 7, p. 53, Apr. 2020.

[3] J. J. Aloor, S. Sajeev, and A. Shakya, "Space Robotics versus Humans in Space," 2020.

[4] "Russian cosmonauts find new cracks in ISS module," *Reuters*, Aug. 2021. [Online]. Available: https://www.reuters.com/lifestyle/science/russian-cosmonauts-find-new-cracks-iss-module-2021-08-30/

[5] J. Park, "Special Feature Vibration-Based Structural Health Monitoring," *Applied Sciences*, vol. 10, no. 15, p. 5139, Jul. 2020.

[6] S. Doebling, C. Farrar, M. Prime, and D. Shevitz, "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review," Tech. Rep. LA–13070-MS, 249299, May 1996.

[7] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, "A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications," *Mechanical Systems and Signal Processing*, vol. 147, p. 107077, Jan. 2021.

[8] G. Boscato, L. Z. Fragonara, A. Cecchi, E. Reccia, and D. Baraldi, "Structural Health Monitoring through Vibration-Based Approaches," *Shock and Vibration*, vol. 2019, pp. 1–5, Feb. 2019.

[9] Wei Fan and Pizhong Qiao, "Vibration-based Damage Identification Methods: A Review and Comparative Study," *Structural Health Monitoring*, vol. 10, no. 1, pp. 83–111, Jan. 2011.

[10] V. Ganesan, T. Das, N. Rahnavard, and J. L. Kauffman, "Vibration-based monitoring and diagnostics using compressive sensing," *Journal of Sound and Vibration*, vol. 394, pp. 612–630, Apr. 2017.

[11] "Gecko robotics ndt inspection services," *GeckoRobotics*, Dec. 2021. [Online]. Available: https://www.geckorobotics.com/ndt-inspection-services

[12] "Advanced inspection robot for remote inspection (bike)," *WaygateTechnologies*, Dec. 2021. [Online]. Available: https://inspection-robotics.com/bike/

[13] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, and A. Ijspeert, "Environmental monitoring using autonomous vehicles: a survey of recent searching techniques," *Current Opinion in Biotechnology*, vol. 45, pp. 76–84, Jun. 2017.

[14] G. Kowadlo and R. A. Russell, "Robot Odor Localization: A Taxonomy and Survey," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, Aug. 2008.

[15] A. Lilienthal, A. Loutfi, and T. Duckett, "Airborne Chemical Sensing with Mobile Robots," *Sensors*, vol. 6, no. 11, pp. 1616–1678, Nov. 2006.

[16] X.-x. Chen and J. Huang, "Odor source localization algorithms on mobile robots: A review and future outlook," *Robotics and Autonomous Systems*, vol. 112, pp. 123–136, Feb. 2019.

[17] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches," *Nature*, vol. 401, no. 6756, pp. 911–914, Oct. 1999. [Online]. Available: http://www.nature.com/articles/44831

[18] V. V. Palyulin, A. V. Chechkin, and R. Metzler, "Levy flights do not always optimize random blind search for sparse targets," *Proceedings of the National Academy of Sciences*, vol. 111, no. 8, pp. 2931–2936, Feb. 2014.

[19] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "A Swarm Robotic Exploration Strategy Based on an Improved Random Walk Method," *Journal of Robotics*, vol. 2019, pp. 1–9, Mar. 2019. [Online]. Available: https://www.hindawi.com/journals/jr/2019/6914212/

[20] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, "Efficient Multi-Robot Coverage of a Known Environment," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1846–1852, Sep. 2017, arXiv: 1808.02541. [Online]. Available: http://arxiv.org/abs/1808.02541

[21] E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," *Discrete Applied Mathematics*, vol. 55, no. 3, pp. 197–218, Dec. 1994.

[22] E. M. Arkin, S. P. Fekete, and J. S. Mitchell, "Approximation algorithms for lawn mowing and milling: A preliminary version of this paper was entitled The lawnmower problem and appears in the Proc. 5th Canad. Conf. Comput. Geom., Waterloo, Canada, 1993, pp. 461-466." *Computational Geometry*, vol. 17, no. 1-2, pp. 25–50, Oct. 2000.

[23] T. Jing, Q.-H. Meng, and H. Ishida, "Recent Progress and Trend of Robot Odor Source Localization," *IEEJ Transactions on Electrical and Electronic Engineering*, p. tee.23364, May 2021.

[24] R. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, "A comparison of reactive robot chemotaxis algorithms," *Robotics and Autonomous Systems*, vol. 45, no. 2, pp. 83–97, Nov. 2003.

[25] E. Persson and D. A. Anisi, "A Comparative study of robotic gas source localization algorithms in industrial environments," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 899–904, Jan. 2011.

[26] N. Voges, A. Chaffiol, P. Lucas, and D. Martinez, "Reactive Searching and Infotaxis in Odor Source Localization," *PLoS Computational Biology*, vol. 10, no. 10, p. e1003861, Oct. 2014.

[27] R. Eberhart and J. Kennedy, "Particle swarm optimiza-

tion," in *Proceedings of the IEEE international conference on neural networks*, vol. 4, 1995, pp. 1942–1948.

[28] X.-S. Yang and Suash Deb, "Cuckoo Search via Levy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. Coimbatore, India: IEEE, 2009, pp. 210–214.

[29] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," in *2007 IEEE Swarm Intelligence Symposium*. Honolulu, HI, USA: IEEE, Apr. 2007, pp. 332–339.

[30] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A PSO-based Mobile Sensor Network for Odor Source Localization in Dynamic Environment: Theory, Simulation and Measurement," in *2006 IEEE International Conference on Evolutionary Computation*. Vancouver, BC, Canada: IEEE, 2006, pp. 1036–1043.

[31] S. Pang and J. Farrell, "Chemical Plume Source Localization," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 5, pp. 1068–1080, Oct. 2006.

[32] J.-G. Li, Q.-H. Meng, F. Li, M. Zeng, and D. Popescu, "Mobile robot based odor source localization via particle filter," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. Shanghai, China: IEEE, Dec. 2009, pp. 2984–2989.

[33] M. Vergassola, E. Villermaux, and B. I. Shraiman, "Infotaxis as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, pp. 406–409, Jan. 2007.

[34] Q. Tang, F. Yu, Z. Xu, and P. Eberhard, "Swarm Robots Search for Multiple Targets," *IEEE Access*, pp. 1–1, 2020.

[35] J. Zhang, D. Gong, and Y. Zhang, "A niching PSO-based multi-robot cooperation method for localizing odor sources," *Neurocomputing*, vol. 123, pp. 308–317, Jan. 2014.

[36] M. Duduta, D. R. Clarke, and R. J. Wood, "A high speed soft robot based on dielectric elastomer actuators," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4346–4351.

[37] W. Wang, J.-Y. Lee, H. Rodrigue, S.-H. Song, W.-S. Chu, and S.-H. Ahn, "Locomotion of inchworm-inspired robot made of smart soft composite (ssc)," *Bioinspiration & biomimetics*, vol. 9, no. 4, p. 046006, 2014.

[38] K. McPherson, K. Hrovat, E. Kelly, and J. Keller, "ISS Researcher's Guide: Acceleration Environment," National Aeronautics and Space Administration, Tech. Rep.

[39] O. Michel, "WebotsTM: Professional Mobile Robot Simulation," *arXiv:cs/0412052*, Dec. 2004, arXiv: cs/0412052.

[40] A. Hayes, A. Martinoli, and R. Goodman, "Swarm robotic odor localization," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 2. Maui, HI, USA: IEEE, 2001, pp. 1073–1078.

[41] U. Jain, R. Tiwari, and W. W. Godfrey, "Multiple odor source localization using diverse-PSO and group-based strategies in an unknown environment," *Journal of Computational Science*, vol. 34, pp. 33–47, May 2019.