An Approach Based on Particle Swarm Optimization for Inspection of Spacecraft Hulls by a Swarm of Miniaturized Robots

Bahar Haghighat[1,2], Johannes Boghaert[1,3], Zev Minsky-Primus[1], Julia Ebert[1], Fanghzheng Liu[4], Martin Nisser[4], Ariel Ekblaw[5], and Radhika Nagpal[1,2]

[1] Harvard University, Boston, MA, USA
[2] Princeton University, Princeton, NJ, USA
[3] Swiss Federal Institute of Technology in Zürich (ETHZ), Zürich, Switzerland
[4] Massachusetts Institute of Technology (MIT), Cambridge, MA, USA
[5] MIT Media Lab Space Exploration Initiative, Cambridge, MA, USA
bahar.haghighat@princeton.edu

Abstract. The remoteness and hazards that are inherent to the operating environments of space infrastructures promote their need for automated robotic inspection. In particular, micrometeoroid and orbital debris impact and structural fatigue are common sources of damage to spacecraft hulls. Vibration sensing has been used to detect structural damage in spacecraft hulls as well as in structural health monitoring practices in industry by deploying static sensors. In this paper, we propose using a swarm of miniaturized vibration-sensing mobile robots realizing a network of mobile sensors. We present a distributed inspection algorithm based on the bio-inspired particle swarm optimization and evolutionary algorithm niching techniques to deliver the task of enumeration and localization of an *a priori* unknown number of vibration sources on a simplified 2.5D spacecraft surface. Our algorithm is deployed on a swarm of simulated cm-scale wheeled robots. These are guided in their inspection task by sensing vibrations arising from failure points on the surface which are detected by on-board accelerometers. We study three performance metrics: (1) proximity of the localized sources to the ground truth locations, (2) time to localize each source, and (3) time to finish the inspection task given a 75% inspection coverage threshold. We find that our swarm is able to successfully localize the present sources accurately and complete the predefined inspection coverage threshold.

1 Introduction

Many industries, such as agriculture, bridge and wind turbine maintenance, and space exploration are actively investing in robotic inspection [8–10, 24]. The overarching goal is to reduce the risk, cost, and service downtime by supporting human inspection. Deploying robots becomes particularly useful when inspection must be carried out in dangerous conditions or over extended periods of time. In particular, long-term space infrastructure deployments will benefit from robotic

inspection [3]. Across a long deployment time, damages caused by structural fatigue and micrometeoroid and orbital debris (MMOD) become non-negligible [16]. Identifying and mending such damages before they become a source of major structural failure is critical. As an example, the International Space Station (ISS) has now been in operation for over two decades. As the structure ages, failures arise [16, 1]. In the near future, this could also apply to the Lunar Gateway space station and the lunar surface Base Camp of NASA's Artemis program. Regular inspection is instrumental to extending lifetime of such deployments.

Vibration sensing and analysis methods are widely used in structural health monitoring applications [30, 33]. In aerospace applications, in particular, the accelerometer-based wing leading edge impact detection system (WLEIDS) was set up and flown on all shuttle flights after the 2003 fatal accident of the Columbia space shuttle. Currently, more than 80 accelerometers are in operation on the ISS for structural dynamics monitoring [33]. The underlying theoretical methods for vibration analysis are based on the vibration response or modal analysis of an *a priori* known structure [13, 6]. The signal processing and failure identification methods depend on the specific target systems [38]. Standard practices typically involve deployment of a large set of static sensors with fixed sampling rates [15].

An automated inspection task may be performed by using a network of static sensors (deployed pre- or post-construction) or a single mobile robot (deployed post-construction). There are, however, multiple benefits in using a swarm of mobile robots. Swarms are known for their resilience to failure of individual units. Compared to fixed sensor networks, used in many environmental monitoring applications, robot swarms provide dynamic and flexible coverage performances [7]. Minimizing the complexity and cost of the individual robotic units is required for achieving low-cost swarm operations. This drive for simplicity has been the motivation behind employing bio-inspired algorithms and miniaturized robots.

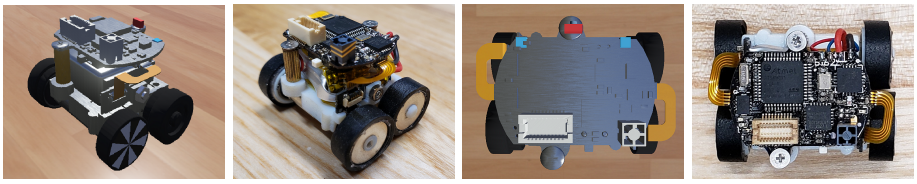
An automated inspection task can be formulated based on the well-studied source localization task [21, 23, 11], that involves three components: (i) finding a cue, (ii) tracing the cue to a source location, and (iii) confirming a localized source. We formulate our inspection task as a repetition of a source localization task until a termination condition is reached. This requires two high-level search behaviors: 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. In what follows, we briefly review the literature for both search behaviors.

Global search methods aim to maximize coverage through (i) a random or (ii) a systematic exploration of the search space. Lévy flights and Brownian motion random walks explore a search space randomly [36, 27, 28]. The basic lawnmower problem in an unobstructed environment and the traveling salesman problem are examples of systematic exploration methods. Being NP-hard [5, 4], there is no guaranteed way to determine the optimal solution to these problems in order to cover the search space, however, near optimal solutions are possible [20]. ***Local search*** methods aim to localize a source [19]. Three main categories of these search methods can be identified: (i) reactive, (ii) heuristic cognitive, 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 made by the robots. These methods are typically simple and require little memory and computational resources [34, 31, 37], but have been shown to perform poorly in complex search scenarios [11, 21]. Cognitive methods combine incoming observations with previously gathered information to guide the search [19]. ***Heuristic cognitive*** search methods see the source localization problem as an optimization problem. The objective function to be optimized can, in the case of an odor source localization problem for instance, be the gas concentration sensed by the robots [19]. Heuristic methods typically lend themselves well to multi-robot search scenarios [19]; by design, their mathematical optimization counterparts deploy multiple agents as candidate solutions that explore the search space. The most known bio-inspired example of heuristic optimization methods are the Particle Swarm Optimization (PSO) [14] and the Cuckoo Search (CS) [39]. PSO-based multi-robot search has been studied in [32, 18]. ***Probabilistic cognitive*** search methods use probabilistic inference to derive the distribution of the cue in the search space [19, 37]. This derivation requires a known dispersion model for any given cue and environment [19] and is often based on the Bayesian inference framework, such as Hidden Markov Models [29] and Particle Filters (PFs) [22]. Another example in this category is infotaxis, which uses an entropy-reduction principle [35]. Probabilistic cognitive search methods are applicable only as long as their underlying model assumptions hold and accurate cue dispersion models are available; for this reason, these methods remain less applicable to localizing failure sources using vibration cues.

We believe that small-scale vibration-sensing robot swarms have a great potential for a variety of structural health monitoring tasks. In this work, we contribute towards realizing such potential by presenting a simulation and algorithmic framework that enables a simulated swarm of miniaturized robots to inspect simplified spacecraft hull surface models. To the best of our knowledge, our work is the first to propose and demonstrate the utility of vibration-sensing surface-inspecting robot swarms. We plan to conduct and present real robot experiments in future works. In this work, we contribute and combine three main elements:

- **Localizing an *a priori* unknown number of failure sources:** Unlike source localization, in inspection tasks the number of failure sources is *a priori* unknown. We address this by employing a PSO-based heuristic local search as well as a coverage maximizing Lévy random walk global search.
- **Using vibration sensing for localizing the failure sources:** Compared to odor sensing paradigms, vibration sensing remains under-addressed in autonomous inspecting robot swarms. We employ a realistically modeled vibration signal (ANSYS software) on simplified spacecraft surface sections.
- **Performing inspection traversing curved surfaces:** Despite their omnipresence, 2.5D environments (i.e., curved 2D surfaces) remain largely unexplored in the target search literature. Curved surfaces pose challenges for robot locomotion and can alter vibration cue propagation through the structure. We study simplified 2.5D surface sections and consider obstacles representing simple surface features typical of spacecraft hulls in our modeling.



(a) Simulated robot (b) Real robot (c) Simulated robot (d) Real robot

Fig. 1: We use a realistic model of the Rovable robot in our simulation experiments. The real Rovable robot (b,d) and its simulation model created in `Webots` (a,c) have similar physical properties. For scale, each wheel is 12mm in diameter.

2 Problem Statement

We formally define the inspection task that we set out to undertake as the repeated localization of any multitude of *failure sources* on a 2.5D (a 2D curved) surface in orbit, using a swarm of robots that sense the vibration signal as a *cue*, until a termination condition based on the overall surface coverage level is met.

A *failure source* is then 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 hypothesize that failure sources such as cracks and fissures on the surface result in creation of specific vibration signal profiles that are detectable in the presence of endemic or induced vibration energy [2]. In our modeling of the failure sources, we further simplify the points of mechanical failure as sources of induced vibration applying force to the surface following a sinusoidal pattern at a frequency of 1Hz, which falls within the mid-frequency range of the vibratory regime of the ISS [25]. The amplitude of the sinusoidal load, set to 1N, is chosen such that the resulting acceleration values are within the ISS acceleration spectrum ranging from below a micro-g to 10 milli-g [25]. The *cue* is then the acceleration signal that is sensed during the inspection task.

3 Simulation Framework

Our 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 shell structure in orbit, and the Webots robotic simulator [26], which we use for simulating the operation of our robots.

Within Webots, we have three main components: (i) a realistic robot model of a 3-cm sized 4-wheeled robot with magnetic wheels, (ii) ferromagnetic target surfaces that the robots traverse to inspect, and (iii) a supervisor controller script that passes on the vibration data to the robots, emulating the function of a black box that contains an acceleration sensor and a processing unit that returns the maximum observed acceleration amplitude. The robot model shown in Fig. 1 is based on the Rovable robot. Originally designed as a mobile wearable

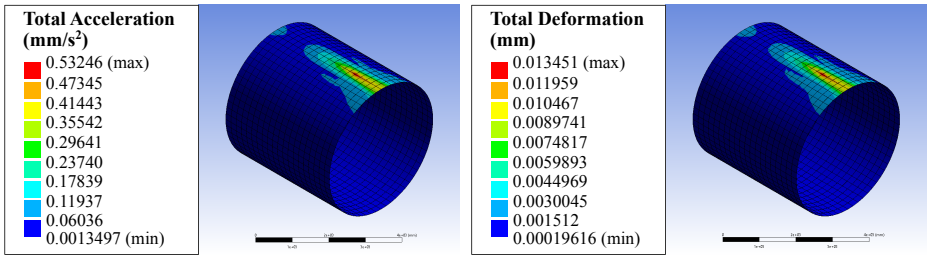


Fig. 2: We use a cylindrical surface model of 4m diameter, 4m length, and 2mm thickness within ANSYS software as a simplified spacecraft hull model and empirically tune the elastic support boundary condition parameter to 10^{-4}N/mm^3 and the load case amplitude to 1N to mimic the vibration regime of the ISS [25]. Vibration propagation is strongly biased along the cylindrical surface's length.

robot, Rovables can sense acceleration using their on-board IMUs [12]. Rovables are capable of wireless communication and low-power localization using their wheel encoders and on-board IMUs for inertial-based navigation. The robots are able to carry loads of 1.5N and can adhere to ferromagnetic surfaces using their magnetic pincher-wheels. A newer version of the robot is equipped with a vertically-mounted linear actuator for inducing vibrations, which could potentially be utilized as an alternative to using endemic vibrations energy. All basic operations of Rovables have been tested in zero gravity conditions in multiple real-life parabolic flights. In this paper, we study the zero gravity conditions in simulation. Within Webots, the Rovable proto file captures the physical properties of the real robot, such as mass, center of mass, and surface contact properties.

The simulated Rovables are additionally assumed to have knowledge of the map of the environment as well as their own locations on the map using a global positioning sensor. The location of the obstacles on the surface are thus known to the robots. This is a realistic assumption because spacecraft hulls are routinely modeled in extensive detail. A loss-free infinite-range communication channel is also assumed in between the robots. The robots share their locations on the map and use this information to perform collision avoidance with one another.

Within ANSYS, we use the Transient Analysis to subject the surface model to a sinusoidal load case of 1N at 1Hz representing a vibration source. To represent the placement of the surface model 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 0.0001N/mm^3 by running a series of simulations and evaluating the results in discussion with a human expert. The resulting deformation amplitude for the applied load case is 0.013mm. Fig. 2 shows the surface model used for the empirical calibrations. In order to reduce the computational cost of the data processing and export pipeline, we create data files that approximate the time-dependent acceleration data from ANSYS with 2D Gaussian distributions that represent the amplitude of the acceleration data on the surface. This data is then retrieved by the super-

Algorithm 1 Inspection Algorithm Overview

```
1: run Lévy Random Walk (RW)                                  ▷ Initialize
2: while coverage < 75% do
3:     if robot in collision then                             ▷ Collision avoidance
4:         run Collision Avoidance (CA)
5:     else if cue picked up then                             ▷ Local search
6:         run Particle Swarm Optimization (PSO)
7:         if cue is a source then                            ▷ Source confirmation
8:             declare source
9:             run Directed Walk (DW)                         ▷ Re-initialization
10:            return to Lévy Random Walk (RW)
11:        end if
12:    else
13:        run Lévy Random Walk (RW)                          ▷ Global search
14:    end if
15:    update coverage                                        ▷ Update coverage
16: end while
```

visor controller script in our Webots simulation and is passed to the simulated robots according to their location on the surface at each simulation step.

4 Proposed Algorithm

The overall structure of our inspection algorithm is shown in Algorithm 1. We use a multi-modal variation of the PSO algorithm that takes advantage of a niche formation behavior to allow parallel search for multiple sources as our local search strategy combined with a random walk approach as our global search strategy. Formation of niches happens simultaneously as the robots switch from global to local search upon sensing a cue. We do not consider merging of the niches, if robots from two niches come close they repel each other. There are four main control states in the algorithm, which we explain briefly in this paragraph and in more detail in the following ones. 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 while simultaneously forming a niche by recruiting a second robot for a second opinion on the source location. 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 and the niche is dismantled. The robot will execute the Collision Avoidance (CA) state at any point in time if it is closer than a threshold distance to a static obstacle or moving robot in the environment.

For each particle i and dimension j, the PSO velocity update rule is as follows:

$$v_{ij}^{t} = \omega * v_{ij}^{t-1} + c_1 * \mathrm{rnd}()^t \times \left(p_{\mathrm{best}_{ij}} - \mathrm{x}_{ij}^{t-1}\right) + c_2 * \mathrm{rnd}()^t \times \left(g_{\mathrm{best}_{ij}} - \mathrm{x}_{ij}^{t-1}\right) \quad (1)$$

where p_{best} and g_{best} are respectively the positions of the best values observed by the individual i and the corresponding niche. The inertia term $\omega = 0.15$, $c_1 =$

0.35, and $c_2 = 0.5$ are weights that balance exploration and exploitation in the search space. The niche formation behavior is part of the local search behavior and allows for confirming an identified source location. Here, we consider niches of size 2. In particular, once a robot is in the vicinity of a source and starts the PSO state, it engages in niche formation by recruiting its nearest neighbor within a maximum range of 1m. The recruited robot then starts in PSO state, moving towards the identified source location.

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 and inversely with surface coverage level there. Upon localization, a source is marked on the coverage map as a circular obstacle region with a radius determined by the range a cue was first perceived from by an approaching robot, deterring the robots from the cue of a discovered source.

Collision avoidance is performed using the artificial potential field (APF) method based on (i) a map of the environment in which the boundaries of the arena and the obstacles are known and (ii) by communicating with other robots to obtain their location on the map. Each obstacle contributes a repulsive term to update a robot's velocity. The repulsive term i in dimension j for robot r is:

$$v^r_{i,j} = w_i \times \left(\frac{1}{d^r_i} - \frac{1}{\theta_i}\right) \times \left(\frac{\mathrm{x}_j - \mathrm{p}_{i,j}}{(d^r_i)^3}\right) \tag{2}$$

where d_i is the distance from the robot to obstacle i, x_j is the robot's position in dimension j and $\mathrm{p}_{i,j}$ is the closest point on obstacle i in dimension j. The threshold θ_i is the distance to the obstacle i below which the robot will engage in collision avoidance. The threshold and weight values depend on the obstacle. There are three obstacle types: (i) static, which includes the arena boundaries and the obstacles ($w = 0.075m$, $\theta = 5 \times 10^{-4}$), (ii) dynamic, which includes a moving robot ($w = 0.12m$, $\theta = 3 \times 10^{-4}$), and (iii) niche, which includes a robot that is part of a niche ($w = 0.75m$, $\theta = 5 \times 10^{-4}$).

Given enough time, we would like all the sources that are present in the search space be successfully localized. We employ a Lévy random walk for the global search behavior. 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 based on a predefined coverage threshold value and using a coverage map that is shared between the robots.

The shared coverage map is used to compute the covered area and check the coverage threshold termination condition. This map is represented as a grid-based map of 10×10cm cells. As the robots move across the surface, sense the value of the vibration signal, and confirm new source locations, they update the shared coverage map using their internal sensor model. We use a simplified sensor model that is a two dimensional Gaussian distribution of $\mathcal{N}(\mu = 0m, \sigma_x = \sigma_z = 0.1m)$. To update the coverage map based on a single robot's observation, the sensor model distribution centered around the location of the reporting robot is

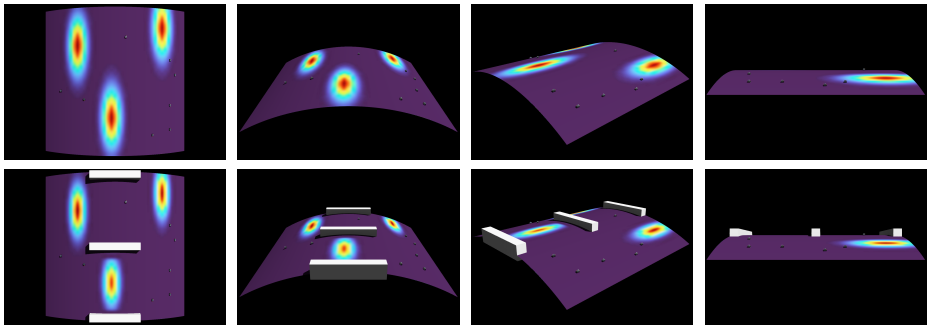


Fig. 3: We study two surface models in our simulation experiments in `Webots`, one with no obstacles in Scenario I (top row) and one with three cuboid obstacles in Scenario II (bottom row). The three vibration sources and the cue spread are visualized. The acceleration cue spread is affected by the presence of obstacles.

superimposed on the coverage map by comparing the coverage value in the map and the coverage value from the sensor model at each point. The coverage value in the map is then replaced by the maximum of these two values. The same update rule is applied for merging coverage information coming from multiple robots to update the shared coverage map.

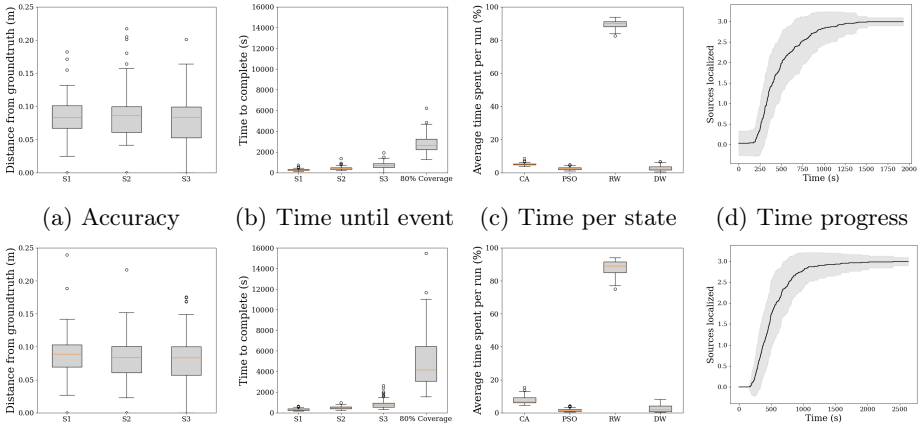
5 Simulation Experiments

We perform our simulations on the Amazon Web Services (AWS) cloud platform using Webots Docker image. Each simulation instance is launched on a 4-core CPU with 8GB of RAM and takes roughly 3 hours to finish.

5.1 Experimental Objectives

Our desired objective for a given inspection experiment is threefold. We would like 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, sensing the cue to the source locations while avoiding obstacles (maneuverability success).

To quantify the swarm performance on these aspects, we take inspiration from metrics used in the fields of source localization and target search and consider three performance metrics [17, 40]. In each scenario, we quantify (i) the source localization accuracy, that is the proximity of a confirmed source location to its ground truth location, (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 insight into the control dynamics of the inspecting swarm, we look at the time the robots spend in each of the control states described in Section 4.



(a) Accuracy (b) Time until event (c) Time per state (d) Time progress

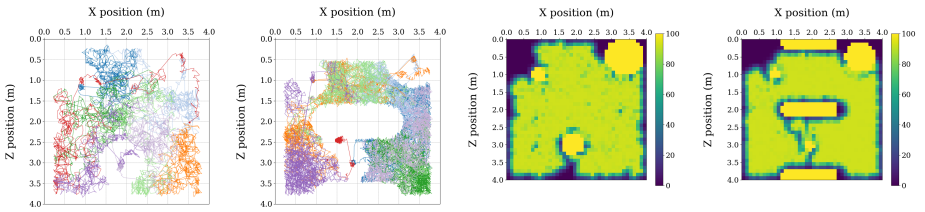
(e) Accuracy (f) Time until event (g) Time per state (h) Time progress

Fig. 4: Presence of obstacles on the surface under inspection negatively impacts the temporal performance of the swarm but has minimal impact on the source localization accuracy. Three main performance metrics are studied for Scenario I (top row) and Scenario II (bottom row): the localization accuracy (a,e), the time elapsed before the discovery of a source and before reaching the %75 coverage threshold (b,f), and the time spent by the robots in each of the four main control states (c,g). The plots show results for 100 simulation experiments per scenario.

5.2 Experimental Scenarios

The real-world inspection problem that underlies our research is a complicated undertaking. Within the scope of this work, we study two simplified problems. We consider two experimental scenarios. In each scenario, we deploy a swarm of size $N = 8$ robots to inspect the surfaces for sources of vibration.

Scenario I comprises a 2.5D curved cylindrical surface with projected flat dimensions of 4×4m. The ANSYS simulations involve a full cylindrical surface of 2mm thickness, 4m radius, and 6m axial length. The surface section is a quarter of the full cylinder with the arena edges 1m away from the cylinder edges. The sources of vibration are at locations ($x = 2\text{m}, z = 3\text{m}$), ($x = 1\text{m}, z = 1\text{m}$), and ($x = 3.5\text{m}, z = 0.5\text{m}$) on the projected surface reference frame, with the origin at the top-left corner. The entire surface is subject to a foundation stiffness of $10^{-4} \frac{\text{N}}{\text{mm}^3}$, and the mesh is sized uniformly with cells of 10×10cm. At the location of each vibration source, we apply a sinusoidal load case with an amplitude of 1N and frequency of 1Hz. The peak amplitude at steady state at each mesh node, i.e. after roughly 9.75s, is then used for constructing the 2D Gaussian signal used in Webots (see Section 3). For Scenario I, we use $\mathcal{N}(\mu = 0m,\, \sigma_x = 0.15m, \sigma_z = 0.45m,)$ scaled by 0.7708 for all three sources. For Scenario II we use $\mathcal{N}(\mu = 0m,\, \sigma_x = 0.1m, \sigma_z = 0.25m)$ scaled by 0.6511 at location ($x = 2\text{m}, z = 3\text{m}$) and



(a) Scenario I (b) Scenario II (c) Scenario I (d) Scenario II

Fig. 5: Presence of obstacles on the surface under inspection affects the robots trajectories and the overall coverage maps, as expected. Trajectory and coverage plots for two example runs of Scenario I (a,c) and Scenario II (b,d) are compared. Upon localization, a source is marked on the coverage map as a region with a radius determined by the range a cue was first perceived from by an approaching robot. Because the spread of the cue is larger along the z axis, the size of the marked regions differs depending on the direction of the first approaching robot.

$\mathcal{N}(\mu = 0m, \sigma_x = 0.15m, \sigma_z = 0.45m)$ scaled by 0.7334 and 0.7359 at locations $(x = 1\text{m}, z = 1\text{m})$ and $(x = 3.5\text{m}, z = 0.5\text{m})$, respectively.

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

6 Results

Simulation results are obtained from 100 trials of the two experimental scenarios described 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.

The performance results are shown in Fig. 4. Three main metrics are considered, namely the source localization accuracy; the time to localize each source and to 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 defined as the time the 75% coverage threshold is reached, and the time each source is discovered (Fig. 4b,f). This is also visible in the time progress of discovering sources as shown in Fig. 4d,h. The solid line shows the average and the shaded area shows the one standard deviation over 100 runs. The same effect can be noted by comparing the time spent in the RW control state among the two scenarios (Fig. 4c,g). We can explain the variation in source localization

accuracy (Fig. 4a,e) by considering three main factors. First, the more time the robots spend in the 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 how the 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 hypothesize that they may depend on the overall geometry that defines the search space. Lastly, the interplay between the shape and spread of the cue and the placement of the sources in the arena plays a significant role in how accurately a source can be localized. We hypothesize that by optimizing the PSO parameters and the random walk step size parameter for a given search space, source localization accuracy can be enhanced. Figure 5 shows the trajectory and the coverage map for an example run of the algorithm in each scenario. The effect of obstacles on the motion of the robots and their coverage behavior is visible in these figures.

7 Conclusion

We developed a simulation and algorithmic framework to study the performance of a swarm of vibration sensing miniaturized wheeled robots that inspect simplified surface section models of spacecraft hulls for failure sources. We modeled the points of mechanical failure as sources of vibration. The robots use the vibration signal propagating through the surface as a cue for localizing sources of vibration. We simulated realistic vibration signal propagation using the ANSYS software, then simplified data transfer by fitting 2D Gaussian functions to the simulation results. Using Webots robotic simulator, we studied the performance of the swarm within two experimental scenarios comprising three sources on a 2.5D cylindrical surface with and without additional obstacles on the surface. Our results provide supporting evidence for the viability of robot swarms for surface inspection tasks based on sensing vibration cues on the surface.

Future work will involve leveraging and extending our modeling and algorithmic framework for studying scenarios of higher complexity in multiple 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 process of simulating the vibration signal from ANSYS such that the data is directly 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 simulation framework developed in this work to perform a parameter optimization in order to find the set of parameters that result in improved performance metrics.

Our hope is that this work supports and inspires studies of vibration-sensing robot swarms as a flexible solution for structural surface inspection applications.

Acknowledgements. We thank Dr. Harald Wild at ETH Zürich for his help with the ANSYS simulations. This work was supported by a Swiss National Sci-

ence Foundation fellowship award P400P2_191116, an Office of Naval Research grant N00014-20-1-2320, and a National Aeronautics and Space Administration (NASA) grant 80NSSC21K0353.

References

1. Russian cosmonauts find new cracks in ISS module. Reuters (Aug 2021), https://www.reuters.com/lifestyle/science/russian-cosmonauts-find-new-cracks-iss-module-2021-08-30/
2. Abu-Mahfouz, I., Banerjee, A.: Crack detection and identification using vibration signals and fuzzy clustering. Procedia Computer Science **114**, 266–274 (2017)
3. Aloor, J.J., Sajeev, S., Shakya, A.: Space Robotics versus Humans in Space (2020)
4. Arkin, E.M., Fekete, S.P., Mitchell, J.S.: 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 **17**(1-2), 25–50 (Oct 2000). https://doi.org/10.1016/S0925-7721(00)00015-8
5. Arkin, E.M., Hassin, R.: Approximation algorithms for the geometric covering salesman problem. Discrete Applied Mathematics **55**(3), 197–218 (Dec 1994). https://doi.org/10.1016/0166-218X(94)90008-6
6. Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., Inman, D.J.: 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 **147**, 107077 (Jan 2021). https://doi.org/10.1016/j.ymssp.2020.107077
7. Bayat, B., Crasta, N., Crespi, A., Pascoal, A.M., Ijspeert, A.: Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. Current Opinion in Biotechnology **45**, 76–84 (Jun 2017). https://doi.org/10.1016/j.copbio.2017.01.009
8. Bualat, M., Edwards, L., Fong, T., Broxton, M., Flueckiger, L., Lee, S.Y., Park, E., To, V., Utz, H., Verma, V., et al.: Autonomous robotic inspection for lunar surface operations. In: Field and Service Robotics. pp. 169–178. Springer (2008)
9. Carbone, C., Garibaldi, O., Kurt, Z.: Swarm Robotics as a Solution to Crops Inspection for Precision Agriculture. KnE Engineering **3**(1), 552 (Feb 2018). https://doi.org/10.18502/keg.v3i1.1459
10. Carrillo-Zapata, D., Milner, E., Hird, J., Tzoumas, G., Vardanega, P.J., Sooriyabandara, M., Giuliani, M., Winfield, A.F.T., Hauert, S.: Mutual Shaping in Swarm Robotics: User Studies in Fire and Rescue, Storage Organization, and Bridge Inspection. Frontiers in Robotics and AI **7**, 53 (Apr 2020)
11. Chen, X.x., Huang, J.: Odor source localization algorithms on mobile robots: A review and future outlook. Robotics and Autonomous Systems **112**, 123–136 (Feb 2019). https://doi.org/10.1016/j.robot.2018.11.014
12. Dementyev, A., Kao, H.L.C., Choi, I., Ajilo, D., Xu, M., Paradiso, J.A., Schmandt, C., Follmer, S.: Rovables: Miniature On-Body Robots as Mobile Wearables. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology. pp. 111–120. ACM, Tokyo Japan (Oct 2016). https://doi.org/10.1145/2984511.2984531, https://dl.acm.org/doi/10.1145/2984511.2984531

13. Doebling, S., Farrar, C., Prime, M., Shevitz, D.: 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). https://doi.org/10.2172/249299
14. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. vol. 4, pp. 1942–1948 (1995)
15. Ganesan, V., Das, T., Rahnavard, N., Kauffman, J.L.: Vibration-based monitoring and diagnostics using compressive sensing. Journal of Sound and Vibration **394**, 612–630 (Apr 2017). https://doi.org/10.1016/j.jsv.2017.02.002
16. Hyde, J.L., Christiansen, E.L., Lear, D.M.: Observations of mmod impact damage to the iss. In: International Orbital Debris Conference. No. JSC-E-DAA-TN75127 (2019)
17. Jain, U., Tiwari, R., Godfrey, W.W.: Multiple odor source localization using diverse-PSO and group-based strategies in an unknown environment. Journal of Computational Science **34**, 33–47 (May 2019). https://doi.org/10.1016/j.jocs.2019.04.008
18. Jatmiko, W., Sekiyama, K., Fukuda, T.: 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. pp. 1036–1043. IEEE, Vancouver, BC, Canada (2006). https://doi.org/10.1109/CEC.2006.1688423
19. Jing, T., Meng, Q.H., Ishida, H.: Recent Progress and Trend of Robot Odor Source Localization. IEEJ Transactions on Electrical and Electronic Engineering p. tee.23364 (May 2021). https://doi.org/10.1002/tee.23364
20. Karapetyan, N., Benson, K., McKinney, C., Taslakian, P., Rekleitis, I.: Efficient Multi-Robot Coverage of a Known Environment. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 1846–1852 (Sep 2017). https://doi.org/10.1109/IROS.2017.8206000, http://arxiv.org/abs/1808.02541, arXiv: 1808.02541
21. Kowadlo, G., Russell, R.A.: Robot Odor Localization: A Taxonomy and Survey. The International Journal of Robotics Research **27**(8), 869–894 (Aug 2008). https://doi.org/10.1177/0278364908095118
22. Li, J.G., Meng, Q.H., Li, F., Zeng, M., Popescu, D.: 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. pp. 2984–2989. IEEE, Shanghai, China (Dec 2009). https://doi.org/10.1109/CDC.2009.5400388
23. Lilienthal, A., Loutfi, A., Duckett, T.: Airborne Chemical Sensing with Mobile Robots. Sensors **6**(11), 1616–1678 (Nov 2006). https://doi.org/10.3390/s6111616
24. Liu, Y., Hajj, M., Bao, Y.: Review of robot-based damage assessment for offshore wind turbines. Renewable and Sustainable Energy Reviews **158**, 112187 (2022)
25. McPherson, K., Hrovat, K., Kelly, E., Keller, J.: ISS Researcher's Guide: Acceleration Environment. Tech. rep., National Aeronautics and Space Administration
26. Michel, O.: WebotsTM: Professional Mobile Robot Simulation. arXiv:cs/0412052 (Dec 2004), arXiv: cs/0412052
27. Palyulin, V.V., Chechkin, A.V., Metzler, R.: Levy flights do not always optimize random blind search for sparse targets. Proceedings of the National Academy of Sciences **111**(8), 2931–2936 (Feb 2014). https://doi.org/10.1073/pnas.1320424111
28. Pang, B., Song, Y., Zhang, C., Wang, H., Yang, R.: A Swarm Robotic Exploration Strategy Based on an Improved Random Walk Method. Journal

of Robotics **2019**, 1–9 (Mar 2019). https://doi.org/10.1155/2019/6914212, https://www.hindawi.com/journals/jr/2019/6914212/
29. Pang, S., Farrell, J.: Chemical Plume Source Localization. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) **36**(5), 1068–1080 (Oct 2006). https://doi.org/10.1109/TSMCB.2006.874689
30. Park, J.: Special Feature Vibration-Based Structural Health Monitoring. Applied Sciences **10**(15), 5139 (Jul 2020). https://doi.org/10.3390/app10155139
31. Persson, E., Anisi, D.A.: A Comparative study of robotic gas source localization algorithms in industrial environments. IFAC Proceedings Volumes **44**(1), 899–904 (Jan 2011). https://doi.org/10.3182/20110828-6-IT-1002.01532
32. Pugh, J., Martinoli, A.: Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In: 2007 IEEE Swarm Intelligence Symposium. pp. 332–339. IEEE, Honolulu, HI, USA (Apr 2007). https://doi.org/10.1109/SIS.2007.367956
33. Richards, W.L., Madaras, E.I., Prosser, W.H., Studor, G.: Nasa applications of structural health monitoring technology. In: International Workshop on Structural Health Monitoring. No. DFRC-E-DAA-TN11102 (2013)
34. Russell, R., Bab-Hadiashar, A., Shepherd, R.L., Wallace, G.G.: A comparison of reactive robot chemotaxis algorithms. Robotics and Autonomous Systems **45**(2), 83–97 (Nov 2003). https://doi.org/10.1016/S0921-8890(03)00120-9
35. Vergassola, M., Villermaux, E., Shraiman, B.I.: Infotaxis as a strategy for searching without gradients. Nature **445**(7126), 406–409 (Jan 2007). https://doi.org/10.1038/nature05464
36. Viswanathan, G.M., Buldyrev, S.V., Havlin, S., da Luz, M.G.E., Raposo, E.P., Stanley, H.E.: Optimizing the success of random searches. Nature **401**(6756), 911–914 (Oct 1999). https://doi.org/10.1038/44831, http://www.nature.com/articles/44831
37. Voges, N., Chaffiol, A., Lucas, P., Martinez, D.: Reactive Searching and Infotaxis in Odor Source Localization. PLoS Computational Biology **10**(10), e1003861 (Oct 2014). https://doi.org/10.1371/journal.pcbi.1003861
38. Wei Fan, Pizhong Qiao: Vibration-based Damage Identification Methods: A Review and Comparative Study. Structural Health Monitoring **10**(1), 83–111 (Jan 2011). https://doi.org/10.1177/1475921710365419
39. Yang, X.S., Suash Deb: Cuckoo Search via Levy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). pp. 210–214. IEEE, Coimbatore, India (2009). https://doi.org/10.1109/NABIC.2009.5393690
40. Zhang, J., Gong, D., Zhang, Y.: A niching PSO-based multi-robot cooperation method for localizing odor sources. Neurocomputing **123**, 308–317 (Jan 2014). https://doi.org/10.1016/j.neucom.2013.07.025