

Article

Source Localisation Using Wavefield Correlation-Enhanced Particle Swarm Optimisation

George Rossides ^{1,2} , Alan Hunter ³ and Benjamin Metcalfe ^{2,*} ¹ Marine Robotics Innovation Centre, Cyprus Marine and Maritime Institute, Larnaca 6023, Cyprus; george.rossides@cmmi.blue or g.rossides@bath.ac.uk² Department of Electronic & Electrical Engineering, University of Bath, Bath BA2 7AY, UK³ Department of Mechanical Engineering, University of Bath, Bath BA2 7AY, UK; a.j.hunter@bath.ac.uk

* Correspondence: b.w.metcalfe@bath.ac.uk

Abstract: Particle swarm optimisation (PSO) is a swarm intelligence algorithm used for controlling robotic swarms in applications such as source localisation. However, conventional PSO algorithms consider only the intensity of the received signal. Wavefield signals, such as propagating underwater acoustic waves, permit the measurement of higher order statistics that can be used to provide additional information about the location of the source and thus improve overall swarm performance. Wavefield correlation techniques that make use of such information are already used in multi-element hydrophone array systems for the localisation of underwater marine sources. Additionally, the simplest model of a multi-element array (a two-element array) is characterised by operational simplicity and low-cost, which matches the ethos of robotic swarms. Thus, in this paper, three novel approaches are introduced that enable PSO to consider the higher order statistics available in wavefield measurements. In simulations, they are shown to outperform the standard intensity-based PSO in terms of robustness to low signal-to-noise ratio (SNR) and convergence speed. The best performing approach, cross-correlation bearing PSO (XB-PSO), is capable of converging to the source from as low as -5 dB initial SNR. The original PSO algorithm only manages to converge at 10 dB and at this SNR, XB-PSO converges 4 times faster.

Keywords: particle swarm optimisation; source localisation; marine swarm robotics; wavefield correlation



Citation: Rossides, G.; Hunter, A.; Metcalfe, B. Source Localisation Using Wavefield Correlation-Enhanced Particle Swarm Optimisation. *Robotics* **2022**, *11*, 52. <https://doi.org/10.3390/robotics11020052>

Academic Editors: Jacopo Aguzzi, Giacomo Picardi, Damianos Chatzievangelou, Simone Marini, Sascha Flögel, Sergio Stefanni, Peter Weiss and Daniel Mihai Toma

Received: 20 March 2022

Accepted: 11 April 2022

Published: 18 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Swarm robotics is the discipline of cooperative robotics, concerned with developing emergent collaborative behaviour in large numbers of robots. Robotic swarms are characterised by flexibility, scalability and robustness [1], as well as simplicity and low-cost, making them well suited for use in large, unexplored environments where semi-disposable systems are needed. For this reason, they have been applied to marine robotic applications including environmental monitoring [2], source detection of underwater oil spills [3] and exploration [4].

Robotic swarms are typically controlled using *swarm intelligence algorithms* (SIAs) [5]. One of the most common SIAs is particle swarm optimisation (PSO), which was inspired by the collective motion of birds and fish [6] and operates by assigning a fitness or cost to different locations. The robots are then drawn towards those locations. There have been numerous studies on the characteristics of PSO, leading to variants that address different limitations, depending on the application. Other SIAs include glowworm swarm optimisation (GSO) [7], which assigns a fitness value on each robot of the swarm, instead of locations in space; ant colony optimisation (ACO) [8], which mimics the foraging strategies employed by ants through the use of pheromones; and the firefly algorithm (FA), [9] which is similar to GSO, but each robot is attracted to all other robots and the strength of the attraction depends on the distance between robots.

PSO converts local signal intensity collected by sensors on each robot into location fitness (or cost). The robots communicate this information among the swarm and each then changes its motion based on both the local and collective knowledge of intensity, in order to converge to the source. PSO has been used in this way in applications such as olfaction source localisation [10–12]. Additionally, a variety of PSO variants have been proposed that enable PSO to control physical robotic swarms. Such techniques are not concerned with the way that the location of the source is identified and instead focus on controlling the motion of the robots of the swarm, by accommodating for underactuated motion, other physical motion constraints like maximum velocity and acceleration, communication constraints, and collision and obstacle avoidance. Robotic PSO (RPSO) is one such variant that incorporates obstacle avoidance into standard PSO [13]. Extended versions of RPSO also consider additional physical constraints of the robots [14,15]. Distributed PSO (dPSO) is another PSO variant that is instead concerned with minimising the information exchanged between the robots of the swarm [16]. Extensions of this algorithm therefore primarily focus on maintaining communication between the robots and possible ground stations or other communication points [17,18]. Finally, Micro-PSO (μ PSO) is a variant that was designed to operate with a smaller number of robots and which has been adapted for use in simultaneous localisation and mapping (SLAM) tasks [19]. All of the aforementioned PSO variants have been implemented using intensity-based fitness assignment. Nevertheless, alternative ways of fitness assignment can be also employed to enhance source localisation performance.

Acoustic sensing is the primary method for source localisation in marine applications due to the efficient propagation of underwater sound waves. Sounds emitted by targets (such as marine mammals or submersible vehicles) propagate through the water and may be sensed by transducers such as hydrophones (e.g., passive SONAR). Akin to olfactory sensing, the intensity of acoustic fields can be used directly in PSO to control robotic swarms for the purpose of source localisation. However, acoustic fields contain rich spatio-temporal information (and thus higher-order statistics) that can be extracted using wavefield correlation techniques [20,21]. The measurement of field intensity as a function of direction can be performed using a multi-sensor array; for example the cross-correlation between two sensors provides directional information that can assist in the localisation of acoustic sources [22]. Two sensors are the fewest needed to perform cross-correlation, and whilst more sensors can improve the direction finding performance this is counter to the ethos of swarm robotics.

Thus, this paper introduces three novel PSO fitness-assignment approaches that use the correlation of signal readings from two sensors fitted to each robot. The first approach assigns fitness to a location based on only the correlation of the readings from each sensor, whilst the second and third modifications estimate the direction towards the source, thus exploiting the properties of acoustic wavefields to improve overall performance.

Section 2 of this paper demonstrates the original PSO algorithm as applied to source localisation, wherein the source emits a continuous signal of constant average intensity. In Section 3, three new PSO fitness-assignment approaches are introduced, detailing the theory of operation and the expected advantages. Section 4 introduces a simulation environment that is used to assess the performance of the algorithms, and Section 5 presents the results of these simulations. Section 6 offers a discussion and proposes different scenarios in which the new algorithms could be used alongside future avenues.

2. Amplitude-Particle Swarm Optimisation (A-PSO)

Particle Swarm Optimisation Theory

In PSO, each robot is capable of measuring the fitness of its current location and of remembering the previous location with the highest fitness [6]. At each timestep k , each robot i communicates its own personal best location $\mathbf{y}^i[k]$ to the rest of the swarm, and the best among them is selected as the global best location $\mathbf{y}_g[k]$.

The spatial behaviour of each robot can be controlled using any PSO variant dedicated towards the control of physical robotic swarms [13–19]. For this paper, it is characterised by the following position and velocity update equations [15], which introduce maximum velocity limitation to the motion of the swarm,

$$\mathbf{x}^i[k+1] = \mathbf{x}^i[k] + \Delta t \mathbf{u}^i[k+1], \quad (1)$$

and

$$\mathbf{u}^i[k+1] = \omega \mathbf{u}^i[k] + c_1 \mathbf{r}_1 \circ \text{sgn}(\mathbf{y}^i[k] - \mathbf{x}^i[k]) + c_2 \mathbf{r}_2 \circ \text{sgn}(\mathbf{y}_g[k] - \mathbf{x}^i[k]). \quad (2)$$

The vectors $\mathbf{x}^i[k]$ and $\mathbf{u}^i[k]$ are the position and velocity of robot i at timestep k . The operator \circ is the Schur product. The parameter $\omega \in [0, 1)$ is called the inertia weight and it is used to bound the velocity of the robot [23]. The parameters $c_1 > 0$ and $c_2 > 0$ are the cognitive and social coefficients and control the tendency to move towards the personal best location or the global best location [23]. Each element j of the vectors \mathbf{r}_1 and \mathbf{r}_2 is drawn from the uniform distribution,

$$\mathbf{r}_1^j, \mathbf{r}_2^j \sim U(0, 1) \quad 1 \leq j \leq d,$$

where d is the number of dimensions of the environment. The parameter Δt is the PSO controller loop delay (i.e., the update rate is $1/\Delta t$). Delays caused by inter-robot communication and processing of sensor inputs will lead to larger values of Δt . The sgn function in (2) introduces a maximum velocity limit into the PSO controller [15], given by

$$V = \frac{\hat{c}\sqrt{d}}{1 - \omega}. \quad (3)$$

The parameter \hat{c} is the behaviour coefficient and is given by

$$\hat{c} = c_1 + c_2. \quad (4)$$

In PSO robotic applications, the fitness of a location is typically calculated using the average intensity of the observed signal. Let f_A be the fitness function that needs to be maximised, such that the fitness of location \mathbf{x} at time $t_k = k \Delta t$ is given by

$$f_A(t_k, \mathbf{x}) = \frac{1}{T} \int_{-T}^0 |s(t_k + t, \mathbf{x})|^2 dt, \quad (5)$$

where $s(t_k + t, \mathbf{x})$ is the observed signal at position \mathbf{x} and T is its duration. This type of PSO will be referred to as amplitude-PSO (A-PSO), due to the use of signal intensity (relating to signal amplitude) to assign fitness to a location.

Typically, in PSO algorithms, each robot remembers its personal best location until it discovers a new location of higher fitness. This permanence precludes the use of dynamic fitness functions (i.e., moving sources, mutable environments, noise, etc.) and is referred to as the *Outdated Memory Problem* [24]. To address this, a forgetting property can be introduced that forces the robots to select new personal best locations over time [25,26]. Fusing a typical forgetting function [27] with (5), the fitness $\hat{f}^i[k]$ of the personal best location $\mathbf{y}^i[k]$ of robot i at timestep k is given by

$$\hat{f}^i[k] = \begin{cases} f_A(t_k, \mathbf{x}^i[k]), & \text{for } f_A(t_k, \mathbf{x}^i[k]) > \hat{f}^i[k-1] \\ \hat{f}^i[k-1] \times e^{-a}, & \text{otherwise} \end{cases} \quad (6)$$

where a is a positive scalar. This describes the case when a new personal best location is found (i.e., $f_A(t_k, \mathbf{x}^i[k]) > \hat{f}^i[k-1]$), and when the old personal best location is maintained. The overall personal best and global best location selection process is described by Algorithm A1, where \mathbf{y}_g and \hat{f}_g are the global best location and its associated fitness (assumed to be communicated to all robots at each time step).

It is possible to identify the following two major weaknesses of A-PSO.

1. As A-PSO relies on the intensity of the received signal, its performance is dependent on the Signal-to-Noise Ratio (SNR). This in turn will limit the maximum range of the algorithm and its ability to follow the gradient of the fitness function.
2. Each robot can only calculate the fitness of its current position and therefore, the exploration capabilities of the swarm depend on its span—this is referred to as the *Diversity Loss Problem* [24]. When the swarm is concentrated in a small area, far away from the source, its exploration and convergence capabilities will be limited.

These weaknesses can be improved by including wavefield correlation into the A-PSO algorithm, and it is this idea that will be explored in the rest of this paper.

3. Wavefield Correlation PSO

Instead of measuring only the signal intensity with a single sensor it is possible to correlate the signals from a pair of sensors. In the presence of noise, the correlation of two signals can offer ways to address the negative effects of noise, theoretically achieving faster and over-time more consistent convergence towards the source. Additionally, the information collected from two sensors can be combined to estimate the direction to the source.

Cross-correlation is the mathematical process used to describe the similarity between two signals. Let $s_1(t)$ and $s_2(t)$ represent two different signals (i.e., hydrophone signals), given by

$$\begin{aligned} s_1(t) &= s(t) + n_1(t), \\ s_2(t) &= s(t + \tau_{lag}) + n_2(t), \end{aligned} \quad (7)$$

where $s(t)$ is the received signal, $n_1(t)$ and $n_2(t)$ represent uncorrelated noise and τ_{lag} is the time difference between the signal being received by each sensor, as illustrated by Figure 1a. The cross-correlation of the two signals is described by

$$R(\tau) = \int_{-\infty}^{\infty} s_1(t)s_2(t + \tau) dt, \quad (8)$$

where τ is the lag. In a robotic scenario, cross-correlation can be used to calculate the angle of arrival [28] of a signal at a pair of wavefield sensors (e.g., hydrophones). The value of τ that maximises R can be associated to τ_{lag} using

$$\tau_{lag} = \underset{\tau}{\operatorname{argmax}} \{R(\tau)\}, \quad (9)$$

thereby enabling identification of the delay between the time of arrivals of the two signals.

When the separation of the two sensors D and the propagation speed of the received signal c are known, τ_{lag} can be used to calculate the angle of arrival α of the signal and therefore the direction towards the source. Assuming that the sensors are placed at the front and back of the robot, the angle of arrival α is given by

$$\alpha = \cos^{-1} \left(\frac{\tau_{lag} c}{D} \right). \quad (10)$$

The angle of arrival of the signal is calculated with respect to the orientation of the robot. There always exist two candidate directions of arrival (directional rays) as shown in Figure 1b given by

$$\mathbf{r}_+[k] = \begin{bmatrix} \cos(\alpha + o[k]) \\ \sin(\alpha + o[k]) \end{bmatrix} \quad \mathbf{r}_-[k] = \begin{bmatrix} \cos(-\alpha + o[k]) \\ \sin(-\alpha + o[k]) \end{bmatrix}, \quad (11)$$

respectively, where $o[k]$ is the orientation of the robot at timestep k . In this way, when the robot is facing the source, $\mathbf{r}_+[k] \approx \mathbf{r}_-[k]$ and both directional rays point towards the same direction.

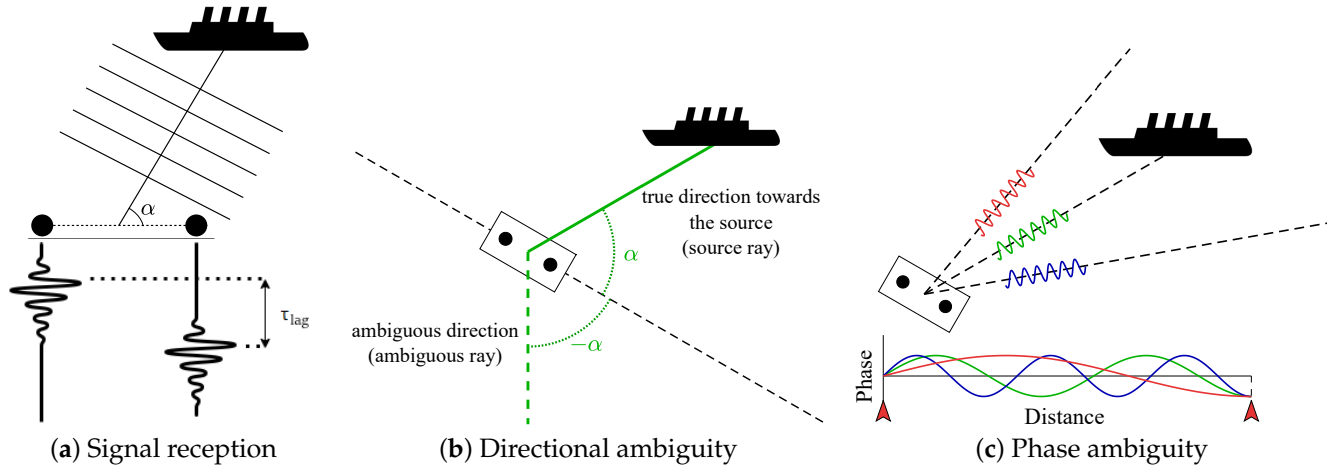


Figure 1. Two sensors (black circles) on a single robot receive an incoming signal emitted by an external source (ship), at an angle α . In (a), the signal is first received by the right-most sensor and then by the left-most one after a delay of τ_{lag} , which can be used to estimate the angle α in (b). Phase ambiguities are related to the sensor separation, as shown in (c) where the phase lines show how all three signals can result in the same phase difference for the two sensors.

Additional ambiguities can be introduced due to the sensor separation D . This is because τ_{lag} depends on the phase difference of the signals from the two sensors. When the received signals are narrowband, it is possible that they will have the same phase difference when arriving at the robot from different angles, as shown in Figure 1c. To avoid this, the sensor separation is limited to $D \leq \lambda_c/2$, where λ_c is the wavelength corresponding to the central frequency of the signal [28]. Furthermore, as the bandwidth of the signal increases, these types of ambiguities are generally reduced. This is because the signal consists of additional lower frequencies (higher wavelengths) that can be used in place of λ_c to resolve ambiguities.

The non-normalised correlation coefficient ρ is another metric provided by cross-correlation that shows the similarity between the two signals, given by

$$\rho = R(\tau_{lag}) = \max_{\tau} \{R(\tau)\}. \quad (12)$$

ρ can be used to enhance the SNR of a received signal [29,30]. The value of ρ depends on the intensity of $s(t)$ (as the intensity of $s(t)$ drops, e.g., due to attenuation, the value of ρ drops as well).

3.1. Cross-Correlation Particle Swarm Optimisation (X-PSO)

Since the non-normalised correlation coefficient ρ scales with the intensity of the received signal, it can be used in the same way that signal intensity is used in A-PSO. This can be achieved by replacing f_A with f_X given by,

$$f_X(t_k, \mathbf{x}) = \max_{\tau} \left\{ \int_{-T}^0 s_1(t_k + t, \mathbf{x}) s_2(t_k + t + \tau, \mathbf{x}) dt \right\}, \quad (13)$$

Therefore, the fitness $\hat{f}^i[k]$ of the personal best location $\mathbf{y}^i[k]$ of robot i at timestep k is given by (6) using $f_X(t_k, \mathbf{x}^i[k])$ instead of $f_A(t_k, \mathbf{x}^i[k])$. This approach is called cross-correlation PSO (X-PSO). The overall personal and global best selection process is described by Algorithm A2.

The advantage of X-PSO is that making use of the non-normalised correlation coefficient ρ of two different signals can reduce the effects of noise, boosting SNR.

3.2. Bearing Particle Swarm Optimisation (B-PSO)

It is possible to use the directional rays to estimate the location of the source in an approach called bearing-PSO (B-PSO). In A-PSO and X-PSO, fitness is assigned to the current location of the robot. However, in B-PSO, fitness is assigned to locations elsewhere indicated by directional ray intersections.

In theory, at each timestep each robot could consider all intersections with all other robots of the swarm to maximise information gain, but this would require communication with all other robots, greatly increasing communication delays. Instead, at each timestep, each robot i receives the directional rays of only one other random robot j and combines it with its own to calculate ray intersections [31]. Combining each ray of robot i with each ray of robot j , can result from zero to four intersections at any timestep. The aim of robot i is to select the intersection that more closely approximates the location of the source.

Assuming that the SNR is significantly greater than 0 dB then on average at least one directional ray from each robot will point towards the source. Consider a single robot rotating about its central axis; one ray will remain pointing towards the source whilst the other (the ambiguous $-\alpha$ ray in Figure 1b) will rotate throughout 360 degrees. Extending to a swarm of robots, the ambiguous (rotating) rays will likely lead to ray intersections that are close to the robots (distributed around the swarm), whilst the stable source rays are likely to lead to intersections that are distant (i.e., distributed around the source). Therefore, to increase the probability that an intersection of two unambiguous rays (i.e., those that point towards the source) is identified, robot i selects the *furthest* of its intersections from its current position.

A drawback of this process is that there is a risk that the selected intersections may be beyond the source or behind the swarm (especially for low SNR). Therefore, the most distal intersection (once selected) \mathbf{p}_s^i is assigned a fitness f_B based on its distance from robot i given by

$$f_B(\mathbf{x}^i, \mathbf{p}_s^i) = \frac{1}{|\mathbf{x}^i - \mathbf{p}_s^i|}. \quad (14)$$

where \mathbf{x}^i is the location of robot i . Thus, locations that are far away are progressively down weighted.

As with A-PSO and X-PSO, the selected intersection replaces the personal best location of robot i if its fitness is higher than the previous personal best location. Note that maximising the fitness function f_B does not necessarily imply convergence to the source. Thus, a forgetting function is used to ensure that the personal best location is updated as robot i moves towards the source. In contrast to A-PSO and X-PSO, the forgetting function is fundamental for the correct operation of B-PSO. Therefore, the fitness $\hat{f}^i[k]$ of the personal best location $\mathbf{y}^i[k]$ of robot i at timestep k is given by (6) using $f_B(\mathbf{x}^i[k], \mathbf{p}_s^i[k])$ instead of $f_A(t_k, \mathbf{x}^i[k])$.

Finally, to locate the source the centroid of the personal best locations (i.e., the centroid of the selected and weighted intersections) may be computed and used as the global best location \mathbf{y}_g ,

$$\mathbf{y}_g[k] = \frac{\sum_{i=1}^M \mathbf{y}^i[k]}{M}, \quad (15)$$

where M is the total number of robots. The overall process is described by Algorithm A3.

B-PSO selects personal best locations that are distant from the swarm, thus fostering an exploration behaviour that addresses Weakness 2 (i.e., its exploration capabilities are not limited by the span of the swarm). However, B-PSO may still be susceptible to distal intersections caused by ambiguous rays (especially if they are located behind the swarm), since they may not be immediately replaceable due to low fitness. Therefore, a more direct

way of excluding such intersections using correlation measurements is considered in the next approach.

3.3. Cross-Correlation-Bearing Particle Swarm Optimisation (XB-PSO)

In order to identify intersections caused by ambiguous rays, X-PSO and B-PSO are now combined to form a cross-correlation-bearing PSO (XB-PSO) approach.

In XB-PSO both correlation coefficient ρ and directional ray information of robot j are known by robot i , and it is assumed that ρ is large close to the source. Thus, each intersection \mathbf{p}^{ij} formed by directional rays of robots i and j can be validated (as pointing towards the source) using either of the following conditions

$$|\mathbf{x}^i[k] - \mathbf{p}^{ij}[k]| < |\mathbf{x}^j[k] - \mathbf{p}^{ij}[k]| \quad \text{and} \quad \rho^i[k] > \rho^j[k] \quad (16a)$$

$$|\mathbf{x}^i[k] - \mathbf{p}^{ij}[k]| > |\mathbf{x}^j[k] - \mathbf{p}^{ij}[k]| \quad \text{and} \quad \rho^i[k] < \rho^j[k], \quad (16b)$$

where $\rho^i[k]$ and $\rho^j[k]$ are the non-normalised correlation coefficients of robots i and j , respectively at timestep k . Intersections that do not satisfy either of these conditions are likely intersections of ambiguous rays and are therefore ignored.

From here, the process is identical to B-PSO, where the selected intersection \mathbf{p}_s^i is assigned a fitness using (14). This fitness is compared to the fitness \hat{f}^i of the personal best location and if it is higher, \mathbf{p}_s^i becomes the new personal best location. Furthermore, the selection of the global best location is performed in the same manner as B-PSO, as described by (15). The overall personal and global best location selection process for XB-PSO is described by Algorithm A4.

4. Simulated Environment

In order to evaluate the efficacy of the proposed fitness assignment methods a series of simulations were performed using MATLAB. The simulations were framed in a marine defense application in which the goal is to detect and localise uncrewed underwater vehicles (UUV) [32–35]. The simulations consist of a single static UUV (the source of an acoustic signal) that is being localised by a swarm of uncrewed surface vehicles (USV) in a moderate sea state. Defining this scenario enables the selection of several key parameters given in Table 1, and are chosen primarily for the purposes of illustration by example. Whilst this scenario constrains the motion of the USVs to 2D there is nothing that precludes the use of higher dimensions.

The simulated environment is a 2D infinite plane representing the water surface, and there is assumed to be no friction and no body inertia. Acoustic signals are modelled with constant propagation at 1500 m/s, consistent with acoustic propagation close to the water surface, below any significant thermocline and in non-polar regions [36]. Multipath effects, such as between the water surface and the ocean floor are neglected.

4.1. Robots

The simulated robots (USVs) are equipped with two hydrophones located at the bow and stern. The swarm is initialised at random positions inside a circular area of radius d_0 around the origin, which represents deployment from a stationary ship. The velocity of each robot is directly controlled by the PSO controller defined in (2). The robots are not forced to maintain a minimum separation from each other and collisions are ignored.

4.2. Source

A single source (a static UUV) is positioned at a distance d_s from the origin. Convergence to the source is achieved when the centre of mass of the swarm (CoM) reaches a distance d_c from the source at time t_c . The source emits a continuous acoustic signal of constant power spectral density (PSD_s in units of dB re 1 $\mu\text{Pa}^2/\text{Hz}$ at 1 m) over a bandwidth B . This relates to the noise generated by a UUV motor considering a narrow bandwidth B around the peak frequency f_c of the power spectral density of the motor signal. For elec-

trically propelled UUVs the typical power spectral density peaks at 120 dB re 1 $\mu\text{Pa}^2/\text{Hz}$, in the range f_c from several hundred Hz to a few kHz [37–39]. The simulated signal is therefore modelled using a filtered Gaussian process with central frequency f_c , bandwidth B and Q-factor given by $Q = f_c/B$. The source level intensity I_s (with units of Pa^2 at 1 m) of the simulated signal is given by

$$I_s = 10^{\frac{\text{PSD}_s}{10}} \times 10^{-12} \times B, \quad (17)$$

representing the area under the constant power spectral density curve between $f_c - B/2$ and $f_c + B/2$. Equation (17) can be found by first converting dB to a ratio, normalising for 1 μPa^2 and integrating over B , assuming constant power spectral density over this range. Only the Q-factor will be provided in the results to allow the generalisation of the results.

The source signal attenuates depending on the distance travelled, such that the intensity of the signal at distance r is given by

$$I = \frac{I_s}{r^2}. \quad (18)$$

Filtered additive white Gaussian noise (AWGN) is added to the signals with constant power spectral density over bandwidth B (PSD_n in units of dB re 1 $\mu\text{Pa}^2/\text{Hz}$) and average intensity N , approximating a moderate sea state [40]. The intensity N with units of Pa^2 is related to PSD_n in the same way as I_s is related to PSD_s using (17). Therefore, the SNR at distance r from the source is given by

$$\text{SNR} = 10\log_{10}\left(\frac{I_s}{N}\right) - 10\log_{10}(r^2) \quad (19)$$

The SNR of the signals at the origin (i.e., the location of the initial deployment of the swarm) SNR_0 is found by setting $r = d_s$ in (19). Varying the value of d_s thus varies SNR_0 and provides a mechanism for readily assessing convergence with different initial SNR.

4.3. Normalised Units

Normalised units are used to enable the generalisation of the results to multiple scenarios. Normalised time is described in *timesteps*, such that each timestep has size Δt seconds as defined in Section 2. Normalised distance is relative to a reference distance R , defined as the distance in metres at which $I = N$ (i.e., at $r = R$ m, $\text{SNR} = 0$ dB). Therefore, R is given by

$$R = \sqrt{\frac{I_s}{N}} \quad (20)$$

Varying Δt varies the controller loop delays (i.e., different communication rate between robots), while varying R enables the consideration of different signal intensities and noise levels.

Table 1. Parameter values to approximate a marine source localisation scenario.

Parameter	Value	Justification
Timestep (Δt)	1 s	A typical controller timestep size for robotic applications.
Noise PSD (PSD_n)	60 dB re 1 $\mu\text{Pa}^2/\text{Hz}$	Equivalent to a moderate sea state [40].
Source PSD (PSD_s)	120 dB re 1 $\mu\text{Pa}^2/\text{Hz}$ at 1 m	Equivalent to a typical uncrewed underwater vehicle [37,38].

Table 1. *Cont.*

Parameter	Value	Justification
Reference distance (R)	1000 m	Calculated using (20).
Source centre frequency (f_c)	1 kHz	The central frequency of a typical uncrewed underwater vehicle [37].
Maximum velocity (V)	2 m/s	Typical uncrewed surface vehicle maximum speed range is 1.5 m/s to 5 m/s [41].
Signal propagation speed (c)	1500 m/s	Speed of sound in water [36].
No. Robots (M)	10	Common swarm size in marine robotics [4].
Starting radius (d_0) and convergence radius (d_c)	50 m	Sufficient to accommodate 10 robots.
Forgetting function scaling parameter (a)	1	Frequent updating of personal best locations.

5. Results

Simulations were run to validate the proposed algorithms and to evaluate their performance (convergence capability) in a range of different scenarios. In all simulations the parameters given in Table 1 were used and each scenario was repeated for ($N_r = 100$) repeats (selected as the nearest multiple of 10 above which the statistical significance of the results did not change), to account for the effects of the random noise processes and the random vectors in (2).

This section will describe the convergence speed of the algorithms, their maximum range and their spatial behaviour.

5.1. Initial Signal-to-Noise Ratio

In order to characterise the convergence capabilities of the proposed algorithms the initial SNR (SNR_0) was varied from 10 to -5 dB. The ability of the swarms to achieve convergence more rapidly, and at lower SNR_0 , implies that they are more robust to low SNR and have a larger effective range. The Q-factor of the received signal was fixed at $Q = 1.5$ (i.e., one-octave bandwidth), the sensor separation for each robot was $D = \lambda_c/2$ and the time-bandwidth product of the cross-correlated signals was $\text{TBP} = 100$.

Figure 2 shows the median and 90% percentile ranges of the distance from the source to the CoM of the swarm as function of time for $\text{SNR}_0 = 10, 4, 0, -3$ and -5 dB. These results show that A-PSO and X-PSO can only converge when $\text{SNR}_0 \geq 10$ dB, with $t_c = 4500$ and $t_c = 3400$, respectively. The performance of A-PSO and X-PSO are comparable, although the slightly faster convergence of X-PSO does indicate a resilience to low SNR that improves Weakness 1 of A-PSO.

B-PSO and XB-PSO both show convergence at all SNR_0 values, with similar convergence behaviours. B-PSO has an average t_c of 1200, 2700 and 4400 and XB-PSO an average $t_c = 1100, 2600$ and 4400 for $\text{SNR}_0 = 10, 4$ and 0 dB, respectively. This result demonstrates that these algorithms are more robust to low SNR compared to both A-PSO and X-PSO, and thus have superior maximum range.

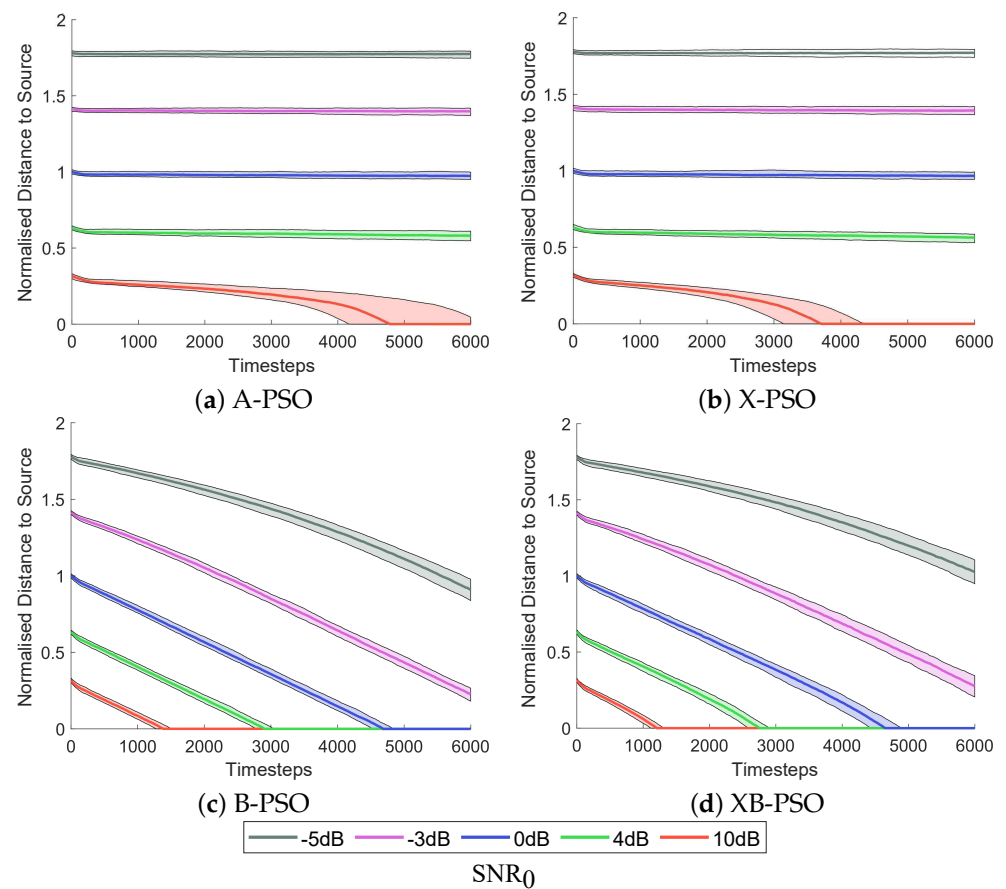


Figure 2. Distances of CoM to source at different initial SNR values (SNR_0). The solid lines represent the median distance from source over ($N_r = 100$ simulations) and the transparent areas the 90% percentile range.

Spatial Analysis

Since both B-PSO and XB-PSO employ personal best locations that distant from the swarm, it is expected that they will exhibit unique spatial motions compared to A-PSO and X-PSO that may explain their enhanced performance. In order to understand this behaviour the spatial routes followed by the swarms were analysed for an $SNR_0 = 10$ dB, as at this SNR_0 all algorithms achieved convergence. A graphical representation of the spatial routes is shown in Figure 3, where the contours show the probability that the CoM of a swarm will pass through that location. The routes of three randomly selected swarms are also shown by the red, green and blue solid lines, while the corresponding dashed lines represent the projections of these routes on the vertical plane (i.e., Timesteps vs. Normalised distance X).

The results show that the different ways of selecting personal best locations (i.e., at the current location of a robot versus at a point defined by the intersection of directional rays) lead to very different spatial behaviour. A-PSO and X-PSO follow direct but zig-zagging routes, whereas B-PSO and XB-PSO move in an almost straight line towards the source.

The contours on the spatial axes of Figure 3 show similar results. The robots controlled by A-PSO and X-PSO are spread out until they get close to the source (occupancy probabilities ($>50\%$) only appear at 0.25 normalised distance). On the other hand, B-PSO and XB-PSO exhibit less spreading and high occupancy probabilities appear at 0.1 normalised distance. These results provide some insight into the convergence performance seen in Figure 2. At 10 dB SNR_0 , B-PSO and XB-PSO are more resilient to noise than A-PSO and X-PSO and so address Weakness 2 of A-PSO (i.e., they exhibit superior exploration capabilities that are not limited by the span of the swarm).

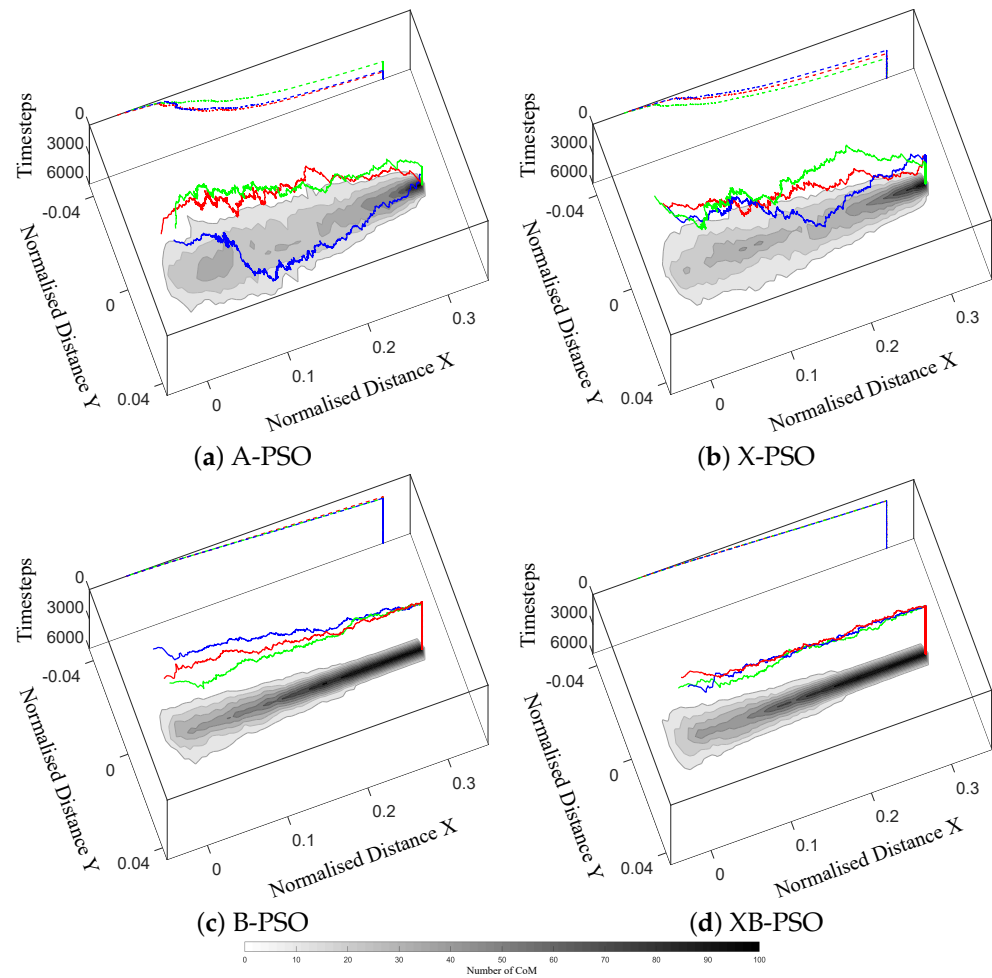


Figure 3. Graphical representation of the routes followed by the CoM of ($N_r = 100$) swarms for each method. The image at the bottom plane of each graph shows contours of the number of swarms (CoM) that passed through that location. The red, green and blue lines represent the routes followed by three randomly selected swarms with respect to time. The corresponding dashed lines represent the 2D projections of these routes on the vertical plane (Timesteps vs. Normalised distance X).

5.2. Q and Sensor Separation

The results of Section 5.1 are presented in normalised time and distance to enable their generalisation to different scenarios. Adjusting Δt allows the adaptation for different controller loop delays, while adjusting R enables adaptation for different values of PSD_n and PSD_s , (see (20)). Furthermore, the maximum velocity V of the robots can be expressed in normalised distance per timestep and is inversely proportional to convergence time t_c (i.e., $V \propto 1/t_c$). Therefore, as V varies for different scenarios the convergence time t_c will also vary inversely.

However, normalised time and distance cannot account for changes in Q (e.g., supertankers have much lower f_c than UAVs [42]) or the sensor separation D which can affect the convergence performance. Therefore, a sensitivity analysis was performed in which Q and sensor separation D were varied for $SNR_0 = 10$ dB.

Figure 4 shows the normalised median time to convergence for different values of Q and $\frac{D}{\lambda_c}$ computed over ($N_r = 100$) repeats. Figure 4 shows that the performance of B-PSO and XB-PSO decreases as Q increases and the signal approaches a sinusoid (narrowband), resulting in ambiguities in the lag output of the cross-correlation function, as discussed in Section 3, whereas A-PSO and X-PSO are insensitive. Despite the decrease in performance, XB-PSO still converges faster than B-PSO for all Q values.

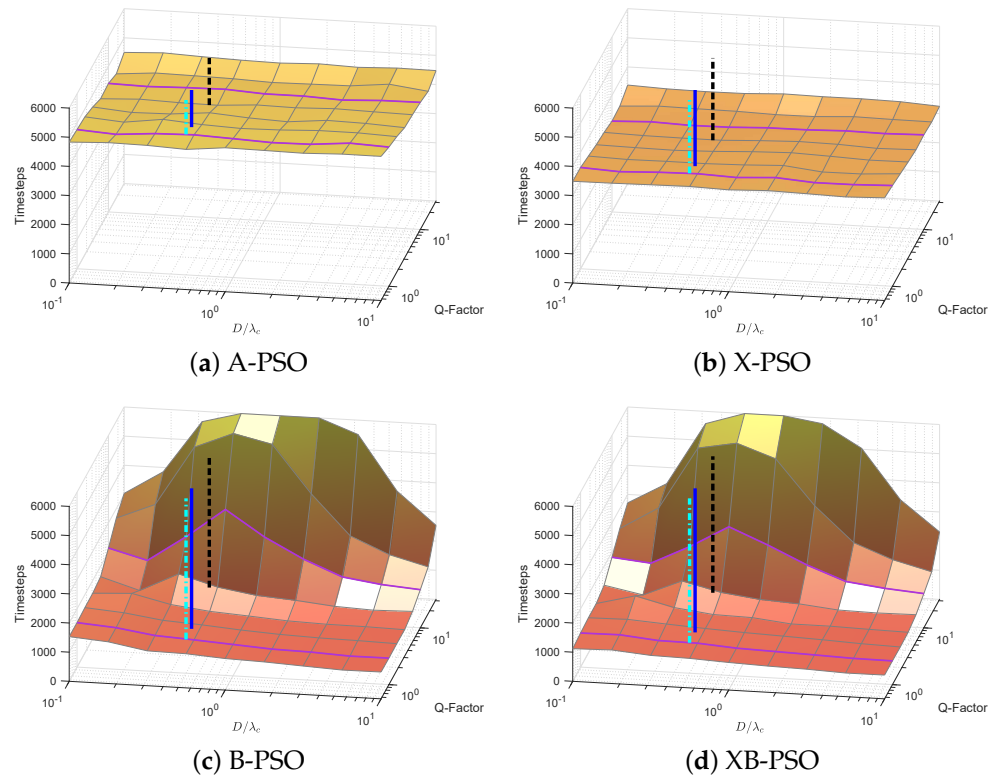


Figure 4. Normalised time needed for the CoM of a swarm to reach convergence (distance d_c from the source). Each point represents the median performance over ($N_r = 100$) swarms. The purple horizontal lines represent the locations where Q is 1 and 10. The vertical blue solid line represents the simulations of Figure 2. The vertical black dashed and cyan dashed-dotted lines represent the Q -factors of the sound generated by two electrically-propelled UUVs, REMUS-100 and Odyssey IIb respectively [37–39]. Note that although a D/λ_c axis is included for A-PSO, it does not affect the position of the sensor, since only one is used for that algorithm, located in the middle of the robot.

A-PSO and X-PSO are also insensitive to D/λ_c , whereas the performance of B-PSO and XB-PSO degrades as D/λ_c approaches 1 for large values of Q . This is because phase ambiguities are introduced as the sensor separation increases beyond $\lambda/2$. However, as the separation approaches 10λ , the performance of B-PSO and XB-PSO improves again because, even though more ambiguities are introduced, they are more evenly distributed around the robot. Therefore, as the robots change orientation the ambiguities are averaged out by the correct predictions of others.

6. Discussion

This paper has introduced three novel methods for location fitness assignment in PSO for the control of robotic swarms that extend and adapt intensity-based fitness (A-PSO) for applications of source localisation in scenarios that include wavefields. The results presented in Section 5 demonstrate that the novel algorithms (X-PSO, B-PSO and XB-PSO) address or improve the weaknesses of A-PSO. B-PSO and XB-PSO select personal best locations that are distant from the swarm, which enables them to converge with initial SNR as low as -5 dB. The convergence speed of the algorithms is affected by narrowband signals and sensor separations close to the wavelength of the received signal. XB-PSO converges faster than B-PSO, likely due to directional ray intersections that occur in the opposite direction from the source which cannot be effectively screened without also considering the signal intensity. However, the difference between the performance of the two methods is small.

For most applications, B-PSO and XB-PSO both offer a good choice of fitness assignment methods, especially when the sensor separation can be selected or broadband source signals are available. On the other hand, for specific applications where narrowband signals are used and the sensor separation is limited, X-PSO provides a more robust alternative. As examples, the results include the use cases of the REMUS-100 and Odyssey IIb, two common electrically-propelled UUVs. It can be seen that the fast convergence and low SNR requirements of B-PSO and XB-PSO make them ideal choices for the detection of relatively silent vehicles that emit low Q-factor signals.

The results make several key assumptions: that the emitted signal is continuous and of constant power spectral density over all frequencies, the robots operate in 2D environment with no friction and that the robots have unbounded rotational speed. Further studies are required to fully understand the implications of these assumptions.

Possible applications of the methods include any form of wavefield source localisation problem, including acoustic source localisation and radar localisation. Additionally, B-PSO and XB-PSO produce global best locations that are approximations of the source location, so can be used for source interaction (e.g., target entrapment and tracking). This capability is a major advancement to A-PSO that will enable its use in a wider range of source localisation scenarios.

7. Conclusions

This paper has presented three novel methods for location fitness assignment in PSO for the control of robotic swarms that consider the higher order statistics available in wavefield measurements. In simulations, they have been shown to outperform the original intensity-based PSO in terms of robustness to low signal-to-noise ratio (SNR) and convergence speed. The best performing method, cross-correlation bearing PSO (XB-PSO), is capable of converging to the source from as low as -5 dB initial SNR. The standard intensity-based PSO only manages to converge at 10 dB and at this SNR, XB-PSO converges 4 times faster

Additionally, B-PSO and XB-PSO may allow distant interaction with the source, a capability that is not offered by A-PSO or other standard fitness assignment methods. The algorithms have been demonstrated in simulation with an example scenario of an uncrewed underwater vehicle localisation. These novel algorithms significantly improve on the performance of A-PSO, and they may serve as inspiration for a new family of swarm robotic control algorithms that sense coherent wavefields.

Author Contributions: Conceptualization, G.R., A.H. and B.M.; methodology, G.R., A.H. and B.M.; software, G.R.; validation, G.R.; formal analysis, G.R. and A.H.; investigation, G.R.; resources, B.M.; writing—original draft preparation, G.R.; writing—review and editing, B.M. and A.H.; visualization, G.R.; supervision, B.M. and A.H.; project administration, B.M.; funding acquisition, B.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by both the UK Natural Environment Research Council (NERC) and Engineering and Physical Sciences Research Council (EPSRC) grant number NE/N012070/1. This research was partially funded by CMMI Cyprus Marine and Maritime Institute. CMMI was established by the CMMI/MaRITeC-X project as a “Center of Excellence in Marine and Maritime Research, Innovation and Technology Development” and has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 857586 and matching funding from the Government of the Republic of Cyprus.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Algorithms

Algorithm A1: Personal and global best location selection for A-PSO.

```

1 foreach robot do
2   robot.x  $\leftarrow$  robot.GetCurrentPosition();
3   s[]  $\leftarrow$  robot.ReadSignal(); // Get sensor reading
4    $f_A \leftarrow \text{mean}(\text{abs}(s[])^2)$ ; // Assign fitness
5   robot.UpdatePersonalBestLocation( $f_A$ );
6 end
7 // Calculate global best location
8  $\hat{f}_g \leftarrow 0$ ; // Reset global best location fitness
9 foreach robot do
10  if robot. $\hat{f} > \hat{f}_g$  then // Update global best location
11     $\hat{f}_g \leftarrow \text{robot}.\hat{f}$ ;
12     $y_g \leftarrow \text{robot}.\mathbf{y}$ ;
13  end
14 end

15 Function UpdatePersonalBestLocation(self, fitness) :
16  if fitness  $>$  self. $\hat{f}$  then // Update personal best location
17    self. $\hat{f} \leftarrow \text{fitness}$ ;
18    self. $\mathbf{y} \leftarrow \text{self}.\mathbf{x}$ ;
19  else // Apply forgetting function
20    self. $\hat{f} = \text{self}.\hat{f} \times e^{-a}$ ;
21  end
22 end

```

Algorithm A2: Personal and global best location selection for X-PSO.

```

1 foreach robot do
2   robot.CalculateCorrelation();
3    $f_X \leftarrow \text{robot}.\rho$ ; // Assign fitness to current location
4   robot.UpdatePersonalBestLocation( $f_X$ ); // As in Algorithm A1
5 end
6 // Calculate global best location
7  $\hat{f}_g \leftarrow 0$ ; // Reset global best location fitness
8 foreach robot do
9   if robot. $\hat{f} > \hat{f}_g$  then // Update global best location
10     $\hat{f}_g \leftarrow \text{robot}.\hat{f}$ ;
11     $y_g \leftarrow \text{robot}.\mathbf{y}$ ;
12  end
13 end

14 Function CalculateCorrelation(self) :
15  self. $\mathbf{x} \leftarrow \text{self}.\text{GetCurrentPosition}()$ ;
16   $s_1[], s_2[] \leftarrow \text{self}.\text{ReadSignals}()$ ; // Get sensor readings
17   $R[] \leftarrow \text{xcorr}(s_1[], s_2[])$ ; // Cross-correlate signals
18   $I_{lag} \leftarrow \text{argmax}(R[])$ ;
19  self. $\rho \leftarrow R[I_{lag}]$ ; // Calculate correlation coefficient
20 end

```

Algorithm A3: Personal and global best location selection for B-PSO.

```

1 foreach robot do
2   | robot.CalculateCorrelation();
3 end
4 foreach robot do // Select personal best locations
5   | other  $\leftarrow$  robot.SelectRandomOtherRobot(); // Uniformly random selection
6   | p[] = []; // Initialise empty arrays of intersections
7   | // There can be up to 4 intersections
8   | foreach ray  $\in$  robot.r do
9     |   | foreach otherRay  $\in$  other.r do
10      |   |   | p[].append(CalculateRayIntersections(robot.x, ray, other.x, otherRay));
11      |   |   end
12     |   end
13   | d[] = |robot.x − p[]|; // Distances from robot to intersections
14   | ps  $\leftarrow$  p[argmax(d[])]; // Select furthest intersection
15   | fB  $\leftarrow$  1/max(d[]); // Assign fitness to selected intersection
16   | robot.UpdatePersonalBestLocation(fB, ps);
17 end
18 // Calculate global best location
19 yg  $\leftarrow$  [0,0]; // Reset global best location
20 foreach robot do // Sum personal best locations
21   | yg  $\leftarrow$  yg+robot.y;
22 end
23 yg  $\leftarrow$  yg/M; // Calculate centroid

24 Function CalculateCorrelation(self) :
25   | self.x  $\leftarrow$  self.GetCurrentPosition();
26   | o  $\leftarrow$  self.GetCurrentOrientation();
27   | s1[], s2[]  $\leftarrow$  self.ReadSignals(); // Get sensor readings
28   | [R[],  $\tau$ []]  $\leftarrow$  xcorr(s1[], s2[]); // Cross-correlate signals
29   | Ilag  $\leftarrow$  argmax(R[]);
30   | self. $\rho$   $\leftarrow$  R[Ilag]; // Calculate correlation coefficient
31   |  $\tau_{lag}$   $\leftarrow$   $\tau$ [Ilag]; // Calculate lag
32   |  $\alpha$   $\leftarrow$  acos( $\tau_{lag}$  * c/D);
33   | self.r[1]  $\leftarrow$  [cos( $\alpha$  + o); sin( $\alpha$  + o)]; // Calculate directional rays
34   | self.r[2]  $\leftarrow$  [cos(− $\alpha$  + o); sin(− $\alpha$  + o)];
35 end

36 Function UpdatePersonalBestLocation(self, fitness, ps) :
37   | if fitness > self. $\hat{f}$  then // Update personal best location
38     |   | self. $\hat{f}$   $\leftarrow$  fitness;
39     |   | self.y  $\leftarrow$  ps;
40   | else // Apply forgetting function
41     |   | self. $\hat{f}$  = self. $\hat{f}$  × e−a;
42   | end
43 end

```

Algorithm A4: Personal and global best location selection for XB-PSO.

```

1 foreach robot do
2   | robot.CalculateCorrelation(); // As in Algorithm A3
3 end
4 foreach robot do // Select personal best locations
5   | other ← robot.SelectRandomOtherRobot(); // Uniformly random selection
6   | p[] = []; // Initialise empty arrays of intersections
7   | // There can be up to 4 intersections
8   | foreach ray ∈ robot.r do
9     |   | foreach otherRay ∈ other.r do
10      |   |   | p[].append(CalculateRayIntersections(robot.x, ray, other.x, otherRay));
11      |   | end
12   | end
13   | d[] = |robot.x − p[]|; // Distances from robot to intersections
14   | other_d[] = |other.x − p[]|; // Dist from other robot to intersections
15   | for q ← length(p[]) to 0 do // Remove non-valid intersections
16     |   | if robot.ρ > other.ρ and d[q] > other_d[q] then p[].remove(q);
17     |   | else if robot.ρ < other.ρ and d[q] < other_d[q] then p[].remove(q);
18   | end
19   | d[] = |robot.x − p[]|; // Distances from robot to intersections
20   | p_s ← p[argmax(d[])]; // Select furthest intersection
21   | f_B ← 1/max(d[]); // Assign fitness to selected intersection
22   | robot.UpdatePersonalBestLocation(f_B, p_s); // As in Algorithm 3
23 end

24 // Calculate global best location
25 y_g ← [0,0]; // Reset global best location
26 foreach robot do // Sum personal best locations
27   | y_g ← y_g + robot.y;
28 end
29 y_g ← y_g / M; // Calculate centroid

```

References

1. Nedjah, N.; Junior, L.S. Review of methodologies and tasks in swarm robotics towards standardization. *Swarm Evol. Comput.* **2019**, *50*, 100565. [\[CrossRef\]](#)
2. Lončar, I.; Babić, A.; Arbanas, B.; Vasiljević, G.; Petrović, T.; Bogdan, S.; Mišković, N. A Heterogeneous Robotic Swarm for Long-Term Monitoring of Marine Environments. *Appl. Sci.* **2019**, *9*, 1388. [\[CrossRef\]](#)
3. Gupta, R.; Bayal, R.K. Source Detection of Oil Spill using Modified Glowworm Swarm optimization. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 14–16 October 2020; pp. 1–6. [\[CrossRef\]](#)
4. Griffiths Sánchez, N.D.; Vargas, P.A.; Couceiro, M.S. A Darwinian Swarm Robotics Strategy Applied to Underwater Exploration. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6. [\[CrossRef\]](#)
5. Senanayake, M.; Senthoooran, I.; Barca, J.C.; Chung, H.; Kamruzzaman, J.; Murshed, M. Search and tracking algorithms for swarms of robots: A survey. *Robot. Auton. Syst.* **2016**, *75*, 422–434. [\[CrossRef\]](#)
6. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Nanjing, China, 10–13 December 1995; Volume 4, pp. 1942–1948. [\[CrossRef\]](#)
7. Krishnanand, K.N.; Ghose, D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intell.* **2009**, *3*, 87–124. [\[CrossRef\]](#)
8. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [\[CrossRef\]](#)
9. Yang, X.S. Firefly Algorithm, Stochastic Test Functions and Design Optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [\[CrossRef\]](#)
10. Marques, L.; Nunes, U.; de Almeida, A.T. Particle swarm-based olfactory guided search. *Auton. Robot.* **2006**, *20*, 277–287. [\[CrossRef\]](#)

11. Meng, Q.H.; Yang, W.X.; Wang, Y.; Zeng, M. Collective Odor Source Estimation and Search in Time-Variant Airflow Environments Using Mobile Robots. *Sensors* **2011**, *11*, 10415–10443. [\[CrossRef\]](#)
12. Feng, Q.; Zhang, C.; Lu, J.; Cai, H.; Chen, Z.; Yang, Y.; Li, F.; Li, X. Source localization in dynamic indoor environments with natural ventilation: An experimental study of a particle swarm optimization-based multi-robot olfaction method. *Build. Environ.* **2019**, *161*, 106228. [\[CrossRef\]](#)
13. Couceiro, M.S.; Rocha, R.P.; Ferreira, N.M.F. A PSO multi-robot exploration approach over unreliable MANETs. *Adv. Robot.* **2013**, *27*, 1221–1234. [\[CrossRef\]](#)
14. Yang, J.; Wang, X.; Bauer, P. Extended PSO Based Collaborative Searching for Robotic Swarms with Practical Constraints. *IEEE Access* **2019**, *7*, 76328–76341. [\[CrossRef\]](#)
15. Rossides, G.; Metcalfe, B.; Hunter, A. Particle Swarm Optimization—An Adaptation for the Control of Robotic Swarms. *Robotics* **2021**, *10*, 58. [\[CrossRef\]](#)
16. Hereford, J.M.; Siebold, M.; Nichols, S. Using the Particle Swarm Optimization Algorithm for Robotic Search Applications. In Proceedings of the 2007 IEEE Swarm Intelligence Symposium, Honolulu, HI, USA, 1–5 April 2007; pp. 53–59. [\[CrossRef\]](#)
17. Perreault, L.; Wittie, M.P.; Sheppard, J. Communication-aware distributed PSO for dynamic robotic search. In Proceedings of the 2014 IEEE Symposium on Swarm Intelligence, Orlando, FL, USA, 9–12 December 2014; pp. 1–8. [\[CrossRef\]](#)
18. Du, Y. A Novel Approach for Swarm Robotic Target Searches Based on the DPSO Algorithm. *IEEE Access* **2020**, *8*, 226484–226505. [\[CrossRef\]](#)
19. Bakhale, M.; Hemalatha, V.; Dhanalakshmi, S.; Kumar, R.; Siddharth Jain, M. A Dynamic Inertial Weight Strategy in Micro PSO for Swarm Robots. *Wirel. Pers. Commun.* **2020**, *110*, 573–592. [\[CrossRef\]](#)
20. Poursheikhali, S.; Zamiri-Jafarian, H. TDOA based target localization in inhomogenous underwater wireless sensor network. In Proceedings of the 2015 5th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 29 October 2015; pp. 1–6. [\[CrossRef\]](#)
21. Zhang, B.; Hu, Y.; Wang, H.; Zhuang, Z. Underwater Source Localization Using TDOA and FDOA Measurements with Unknown Propagation Speed and Sensor Parameter Errors. *IEEE Access* **2018**, *6*, 36645–36661. [\[CrossRef\]](#)
22. Li, P.; Zhang, X.; Zhang, W. Direction of Arrival Estimation Using Two Hydrophones: Frequency Diversity Technique for Passive Sonar. *Sensors* **2019**, *19*, 2001. [\[CrossRef\]](#)
23. Cleghorn, C.W.; Engelbrecht, A.P. Particle swarm stability: A theoretical extension using the non-stagnate distribution assumption. *Swarm Intell.* **2018**, *12*, 1–22. [\[CrossRef\]](#)
24. Blackwell, T. Particle Swarm Optimization in Dynamic Environments. In *Evolutionary Computation in Dynamic and Uncertain Environments*; Yang, S., Ong, Y.S., Jin, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 29–49. [\[CrossRef\]](#)
25. Carlisle, A.; Dozier, G. Adapting Particle Swarm Optimization to Dynamic Environments. In Proceedings of the International Conference on Artificial Intelligence, Acapulco, Mexico, 11–14 April 2000.
26. Fernandez-Marquez, J.L.; Arcos, J.L. An Evaporation Mechanism for Dynamic and Noisy Multimodal Optimization. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO '09), Montreal, QC, Canada, 8–12 July 2009; Association for Computing Machinery: New York, NY, USA, 2009; pp. 17–24. [\[CrossRef\]](#)
27. White, K.G. Forgetting functions. *Anim. Learn. Behav.* **2001**, *29*, 193–207. [\[CrossRef\]](#)
28. Munoz, D.; Bouchereau, F.; Vargas, C.; Enriquez, R. CHAPTER 2—Signal Parameter Estimation for the Localization Problem. In *Position Location Techniques and Applications*; Munoz, D., Bouchereau, F., Vargas, C., Enriquez, R., Eds.; Academic Press: Oxford, UK, 2009; pp. 23–65. [\[CrossRef\]](#)
29. Zhang, J.; Cao, L.; Zhang, Z.; Xu, J. Research on the signal separation method based on multi-sensor cross-correlation fusion algorithm. In Proceedings of the 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), Beijing, China, 17–19 August 2017; pp. 561–564. [\[CrossRef\]](#)
30. Mikhael, M.R.M.; Alink, M.S.O.; Kokkeler, A.B.J. Using Multiple Chains in Cross-Correlation Receivers to Improve Sensitivity. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–7. [\[CrossRef\]](#)
31. Schneider, P.J.; Eberly, D.H. CHAPTER 7-INTERSECTION IN 2D. In *Geometric Tools for Computer Graphics*; The Morgan Kaufmann Series in Computer Graphics; Schneider, P.J., Eberly, D.H., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 2003; pp. 241–284. [\[CrossRef\]](#)
32. Railey, K. Demonstration of Passive Acoustic Detection and Tracking of Unmanned Underwater Vehicles. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2018.
33. Railey, K.; DiBiasi, D.; Schmidt, H. An acoustic remote sensing method for high-precision propeller rotation and speed estimation of unmanned underwater vehicles. *J. Acoust. Soc. Am.* **2020**, *148*, 3942–3950. [\[CrossRef\]](#)
34. Railey, K.; DiBiasi, D.; Schmidt, H. Passive acoustic detection and tracking of an unmanned underwater vehicle from motor noise. *J. Acoust. Soc. Am.* **2021**, *149*, A34–A35. [\[CrossRef\]](#)
35. Open Cooperation for European maritime awareNess. Ocean2020 Mediterranean Sea Trials. 2021. Available online: <https://ocean2020.eu/sea-trials/> (accessed on 20 August 2021).
36. Kuperman, W.A. Acoustics, Deep Ocean. In *Encyclopedia of Ocean Sciences*; Steele, J.H., Ed.; Academic Press: Oxford, UK, 2001; pp. 61–72. [\[CrossRef\]](#)

37. Gebbie, J.; Siderius, M.; Allen, J.S.R. Aspect-dependent radiated noise analysis of an underway autonomous underwater vehicle. *J. Acoust. Soc. Am.* **2012**, *132*, EL351–EL357. [[CrossRef](#)] [[PubMed](#)]
38. Zimmerman, R.; D'Spain, G.L.; Chadwell, C.D. Decreasing the radiated acoustic and vibration noise of a mid-size AUV. *IEEE J. Ocean. Eng.* **2005**, *30*, 179–187. [[CrossRef](#)]
39. Holmes, J.D.; Carey, W.M.; Lynch, J.F. An overview of unmanned underwater vehicle noise in the low to mid frequencies bands. *Proc. Meet. Acoust.* **2010**, *9*, 65007. [[CrossRef](#)]
40. Xerandy, X.; Znati, T.; K, L. Cost-Effective, Cognitive Undersea Network for Timely and Reliable Near-Field Tsunami Warning. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 224–233. [[CrossRef](#)]
41. Verfuss, U.K.; Aniceto, A.S.; Harris, D.V.; Gillespie, D.; Fielding, S.; Jiménez, G.; Johnston, P.; Sinclair, R.R.; Sivertsen, A.; Solbø, S.A.; et al. A review of unmanned vehicles for the detection and monitoring of marine fauna. *Mar. Pollut. Bull.* **2019**, *140*, 17–29. [[CrossRef](#)] [[PubMed](#)]
42. Hildebrand, J. Impacts of Anthropogenic Sound on Cetaceans. In *Marine Mammal Research: Conservation Beyond Crisis*; Reynolds, J.E., Perrin, W.F., Reeves, R.R., Montgomery, S., Eds.; Johns Hopkins University Press: Baltimore, MD, USA, 2005.