Impact of initialization of a modified particle swarm optimization on cooperative source searching

Mad Helmi Ab. Majid[1], Mohd Rizal Arshad[2], Mohd Faid Yahya[3], Abu Bakar Ibrahim[1]
[1]Department of Software Engineering and Smart Technology, Faculty of Computer and Meta-Technology, Universiti Pendidikan Sultan Idris, Perak, Malaysia
[2]School of Electrical and Electronic Engineering, Universiti Sains Malaysia (Engineering Campus), Penang, Malaysia
[3]Department of Mechatronics Engineering, Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

Article Info

Article history:

Received May 11, 2023
Revised Jul 29, 2023
Accepted Aug 1, 2023

Keywords:

Cooperative searching
Initialization
Parameter optimization
Particle swarm optimization
Source searching
Swarm robotics

ABSTRACT

Swarm robotic is well known for its flexibility, scalability and robustness that make it suitable for solving many real-world problems. Source searching which is characterized by complex operation due to the spatial characteristic of the source intensity distribution, uncertain searching environments and rigid searching constraints is an example of application where swarm robotics can be applied. Particle swarm optimization (PSO) is one of the famous algorithms have been used for source searching where its effectiveness depends on several factors. Improper parameter selection may lead to a premature convergence and thus robots will fail (i.e., low success rate) to locate the source within the given searching constraints. Additionally, target overshooting and improper initialization strategies may lead to a nonoptimal (i.e., take longer time to converge) target searching. In this study, a modified PSO and three different initializations strategies (i.e., random, equidistant and centralized) were proposed. The findings shown that the proposed PSO model successfully reduce the target overshooting by choosing optimal PSO parameters and has better convergence rate and success rate compared to the benchmark algorithms. Additionally, the findings also indicate that the random initialization give better searching success compared to equidistant and centralize initialization.

Corresponding Author:

Mad Helmi Ab. Majid
Department of Software Engineering and Smart Technology, Faculty of Computer and Meta-Technology, Universiti Pendidikan Sultan Idris
35900 Tanjong Malim, Perak, Malaysia
Email: madhelmi@meta.upsi.edu.my

1. INTRODUCTION

Target searching is an important task but very challenging to solve in real world scenarios. Target searching is usually associated with complex and time-consuming processes despite strict constraints to be met. For example, the searching mission for black box of Malaysian Airlines MH370 in the Indian Ocean took huge efforts and involved a complex and challenging search environment. In this case, the boundary of the searching environment based on global positioning system (GPS) coordinates was set before the searching process took place. In addition to complex operations and huge challenges, the rescuers have a very limited time to search for the black box before the battery of the black box dies out. This real scenario demonstrates the importance of target searching and the need for a better target searching strategy and solution in dealing with real world searching problems [1]. Other examples of target searching include search and rescue of earthquake victims, demining operation, radioactive leakage source detection, environmental monitoring and surveillance [2]–[4].

The primary objective of target searching is to locate the source to its proximity location within a specified boundary (e.g., search space dimension) and operational constraints (e.g., time, accuracy) [5]. Autonomous robots have been researched for the purpose of target searching tasks for a very long time. However, searching using a single autonomous robotic platform is difficult to achieve optimal searching result. Alternatively, cooperative searching using a group of robots has been proven to be more efficient and able to give optimal searching results compared to a single robot searching [6]. From this perspective, the concept of swarm robotics using relatively simple robots offered greater advantages for solving complex target searching as a result of its robustness, scalability and flexibility characteristics [7]. Additionally, incorporating intelligence algorithms allow swarm robots to make automatic decisions intelligently with minimal interruption from humans while ensuring optimal accuracy of the search output [8].

There are different types of algorithms have been developed for swarm robotic target search such as based on swarm intelligent (SI) algorithm (e.g., particle swarm optimization (PSO) [9]–[11], Ant colony optimization (ACO) [12], bean optimization [13] and bacteria foraging optimization (BFO) [14]), behavior-based approaches (e.g., group explosion strategy [15], firework explosion inspired [16] and sweep cleaning [17] and stigmergy [18]), random walk (e.g., levy flight [19], Brownian motion [20]) dan hybrid strategy (e.g., PSO-BFO [21], PSO-fruit fly optimization algorithm (FOA) [22], triangle formation [23] and random walk or stochastic [24]). Among these types of swarm robotic target searching algorithms, SI based algorithms have gain the greatest attention from the researchers because of: i) cooperative nature of the algorithm itself ii) intelligence decision making capability, iii) high convergence rate, and iv) meet swarm robot characteristics (i.e., scalable, flexible and robust).

PSO is the most common SI based algorithm for swarm robotic target searching. Initially PSO was implemented for target searching in its original form but many modifications have been proposed to improve the original PSO performance in target searching tasks. The implementation in its original form includes for example by Hereford *et al.* [25] and Ab Aziz *et al.* [26]. On the other hand, the modifications of PSO were performed for the following purposes: inclusion of obstacles avoidance capability [27], avoid trapping into local optima or premature convergence [28], [29], incorporate multi target searching [30], [31], integrate mobile target searching capability [32]. However, to the best of the author’s knowledge, the analysis of target overshooting and the impact of different initialization strategies have not been deeply studied. In this paper, we proposed a modified PSO algorithm to minimize target overshooting in addition to fast convergence and avoiding premature convergence. Additionally, we also evaluate and compare the performance of the modified PSO with three different initialization strategies. The rest of this paper is organized as follows: Section 2 for research methodology, Section 3 for results and discussion and Section 4 for conclusion and recommendations of future work.

2. RESEARCH METHOD

In this research, a modified PSO is proposed to improve PSO convergence speed, avoid robots stuck in a local optimum or premature convergence and minimize target overshooting. Additionally, three initialization strategies were designed for the proposed modified PSO. To validate the performance of the modified PSO and evaluate the effect of different initialization strategies, a series of simulations using MATLAB was performed. For the testing purpose, we implemented the algorithm on a swarm of autonomous surface vehicles (ASV) which was represented by a complete kinematic and dynamic mathematical model and a complete control system.

2.1. Modified PSO

Firstly, one of the criteria for good searching is a fast convergence. The faster the convergence, the faster the location of the source can be determined and thus, meet the timing constraint of the searching task. From the PSO perspective, fast convergence can be achieved through fast information sharing and update among the robots in the swarm. Typically, this problem is associated with waiting periods using a synchronous update. In this study, asynchronous updates will be used to improve the convergence speed. Secondly, due to the nature of the PSO, robots may get stuck at local optima due to premature convergence. Premature convergence occurs because of rapid decreasing of velocity as the robots rapidly approach or converge to a local optima position where the updated velocity approaches zero and cause robots to lose their exploration capability (i.e., decrease of swarm diversity) before finally stop moving. Once robots are trapped into local optima, they have no capability to escape and are permanently stuck and fail to reach global convergence. As a result, an inaccurate solution is obtained which affects the accuracy of the source’s position estimation. Thirdly, target overshooting will result in inaccurate approximation of the target’s location. Target overshooting primarily happened due to large velocity assigned to the robots as they approached the optimal position. Consequently, more control effort is necessary to accommodate the back-and-forth movements especially for a robot with limited turning capability before robots can reach a stable convergence. Typically, this problem is caused by excessive velocity magnitude due to non-optimal or fixed value of inertia weight and inappropriate selection of velocity limit, V_{max}. To resolve these problems, asynchronous dynamically adjustable particle swarm optimization (ADAPSO) is proposed in this study. The velocity, v and position, p update equations of the proposed ADAPSO are given by (1) and (2):

$$v_i(k_i+1) = \omega_i(k_i)v_i(k_i) + c_{1i}(k_i)r_{1,i}\big(pBest_i(k_i) - p_i(k_i)\big) \\ + c_{2i}(k_i)r_{2,i}\big(gBest(k_i) - p_i(k_i)\big) \quad (1)$$

$$p_i(k_i+1) = p_i(k_i) + v_i(k_i+1) + q_i(k+1) \quad (2)$$

where ω is the inertia weight, c is the acceleration coefficient, r is the random number between 0 and 1, *pBest* is the personal best position, *gBest* is the global best position, k is the iteration number and i is the robot's index. The vector q is used to adjust position of the robot such that communication between robots is maintained. In this formulation, the velocity update is set as $\|v\| \le V_{max}$ to avoid explosion. The inertia weight is defined as (3).

$$\omega_i(k_i) = \omega_{ini} - \alpha\big(h_i(k_i) + g_i(k_i)\big) + \beta s(k_i) \quad (3)$$

where the evolutionary speed factor, h, and aggregation degree, s (defined according to [33]) and the convergence speed factor, g are defined as:

$$h_i(k_i) = 1 - \left[\frac{min\big(f(pBest(k_i-1)), f(pBest(k_i))\big)}{max\big(f(pBest(k_i-1)), f(pBest(k_i))\big)}\right] \quad (4)$$

$$s_i(k_i) = \frac{min\big(f_{kBest}(k_i), \bar{f}(k_i)\big)}{max\big(f_{kBest}(k_i), \bar{f}(k_i)\big)} \quad (5)$$

$$g_i(k_i) = 1 - \left[\frac{min\big(f(gBest(k_i-1)), f(gBest(k_i))\big)}{max\big(f(gBest(k_i-1)), f(gBest(k_i))\big)}\right] \quad (6)$$

where f is the fitness function of the robot expressed as measured source signal intensity. If the corresponding robot is the best robot (i.e., robot with highest fitness value or highest source intensity measurement), the second and the third terms of (1) become zero or close to zero due to the fact that robot current position is close to *pBest* and *gBest* positions. As a result, the exploration capability of that robot temporarily (i.e., as long as it is the best robot) dies out. For this reason, the inertia weight adjustment equation is further modified as (7).

$$\omega_i(k_i) = \begin{cases} \omega_i(k_i) + \frac{\omega_{ini}}{k_i} & \text{if } i \text{isthebestrobot} \\ \omega_i(k_i) & \text{otherwise} \end{cases} \quad (7)$$

Small cognitive component (i.e., the term with *pBest*) and large social component (i.e., the term with *gBest*) promotes search exploration space at the beginning and in the later stage, when cognitive component becomes larger and social component becomes smaller, the possibility of convergence to global optima is improved. The acceleration coefficients are defined as (8) and (9):

$$c_{1,i}(k) = c_{ini} + \frac{1}{2}cos\left(\frac{t}{tT_{max}}()\right) \quad (8)$$

$$c_{2,i}(k) = c_{ini} - \frac{1}{2}cos\left(\frac{t}{tT_{max}}()\right) \quad (9)$$

where c_{ini} is the initial acceleration coefficient, t is current time, t_T is the total time taken to complete the previous operation (i.e., sum of time taken for searching) process and t_{max} is the maximum searching time allowed (i.e., limited by robot power resource).

2.2. Initialization strategies

In this study, three different initialization strategies were proposed relevant to the source searching task which are known as random initialization, equidistant initialization and centralized initializations as illustrated in Figure 1. Firstly, for a random initialization, robots are randomly distributed in a search space such that their initial position can be expressed as (10):

$$P_i(0) = \begin{bmatrix} (x_{max} - x_{min})rand + x_{min} \\ (y_{max} - y_{min})rand + y_{min} \end{bmatrix}, \forall i \in N \tag{10}$$

where x_{min} and y_{min} are the minimum initialization positions and x_{max} and y_{max} are the maximum initialization positions. The *rand* is the random number between 0 and 1 and i is the index of the individual robot. Random initialization is a realistic initialization method in source searching because the proposed algorithm involves stochastic elements and robots can perform source searching when there is no initial guest of the source location. Secondly, for an equidistance initialization strategy, robots are dispersed at an equal angle and radius surrounding the possible source location, r_{rs} such that

$$p_i(0) = r_{rs} \begin{bmatrix} cos(2\pi i/N) \\ sin(2\pi i/N) \end{bmatrix}, \quad \forall i \in N \tag{11}$$

where N is the total number of robots used in the searching process. In a real implementation, this method is not practically feasible because of difficulty in setting the exact robot arrangement at desired angle but it can be useful for evaluating and benchmarking performance of the tested algorithm.

Thirdly, in a centralized deployment, robots are deployed from the same site (i.e., typically a known reference coordinate) of the search space. This initialization method is typically useful for evaluating performance of the algorithm when high similarity in terms of measured source signal intensity among the robots exists. Mathematically, a centralize initialization is defined by (12).

$$p_i(0) = \begin{cases} p_0, & \text{if} \quad i = 1 \\ p_0 + 2r_{ini} \begin{bmatrix} cos\left(\frac{2\pi}{N-1} i\right) \\ sin\left(\frac{2\pi}{N-1} i\right) \end{bmatrix}, & \text{if} \quad i > 1 \end{cases} \tag{12}$$

where $p_{0,i}$ is the initial position of the robot i, p_0 is the initial position of deployment of the first robot and r_{ini} is the initialization radius. In case none of the robots detect the source once they are deployed, each robot will move randomly in the search space.

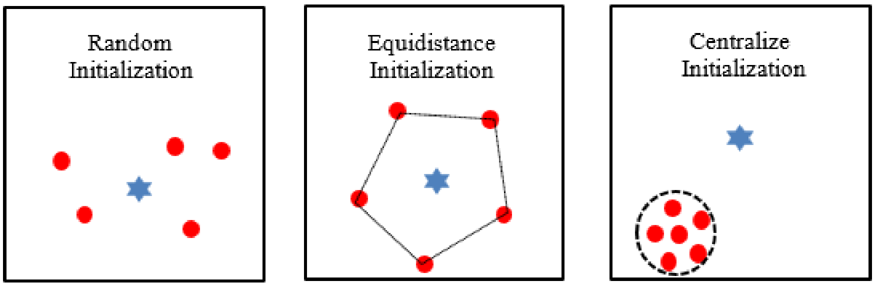

Figure 1. Different strategies of initializations

2.3. Termination criteria

In swarm robotics source searching tasks, robot physical limit and accuracy of the estimated source position must be considered in order to set proper termination conditions. To obtain accurate estimation, only converged robots are considered for the purpose of estimating the source position based on average value. Notice that since r_{con} is not physically measurable because the source's position is unknown and needs to be estimated, the number of converged robots should be decided based on the intensity difference between the converged robots. Let the final intensity measured by robot i is $I_{i,f}(t)$, the position of the converge robots can be defined as:

$$p_i = \left\{p_1, \dots, p_{N_s} : \left(\frac{|I_{i,f}(p_i) - I_{f,max}||}{I_{f,max} \times 100} < \Lambda\right), \forall i \neq j, i, j = 1, \dots, N_s \right\} \tag{13}$$

where $\Lambda \in [0,100]$ is the threshold of percentage intensity difference and $I_{f,max}$ is the maximum intensity or fitness of the *gBest* robot. Any robot in the swarm is classified as a converged robot if its percentage intensity difference is less than certain percentage. A small value of Λ provides a better approximation of the source's position compared to a large value of Λ. Once terminated, the source position can be estimated from the average position of the converge robots or taken to be equal to the final *gBest* position. In this study, the following termination conditions are used to terminate the searching process:

– Maximum operating time is reached, $t > t_{max}$, or
– At least 3 robots successfully converge where fitness difference among the converged robots should be less than $\Lambda \leq 5\%$, and,
– No improvement of *gBest* value is observed for the last 10 iterations

2.4. Simulation setup

For the purpose of evaluating the performance of the proposed PSO and initialization strategy, a series of simulations were conducted. In this simulation, robots are a swarm of Autonomous Surface Vehicles (ASVs) described by a complete kinematics and dynamics model with a proper speed and heading controllers. To simulate performance of the proposed source searching strategy using ASV swarming platform, a swarm robotic simulator was developed by using MATLAB^{TM} as shown in Figure 2. This simulator integrates the localization algorithm, control laws and ASV model as illustrated in Figure 3. The outputs from the proposed source searching algorithm were used to generate the desired waypoint and thus, the desired robot path. From the desired path, the reference control signals were computed which were then fed into speed and heading controllers. The output of the controller was fed into an ASV model where the states of the ASV can be determined. The controller keeps tracking the path until the robot reaches the desired waypoint within an acceptable radius. Once the robot reaches the desired waypoint, a new waypoint is generated and the process is repeated. This is a continuous loop process where it runs until the desired task is completed (i.e., robot successfully reaches convergence) or the desired termination conditions are satisfied or terminated by the user. The source used in this simulation is an underwater acoustic source represented by a mathematical model representing a signal intensity decaying model.

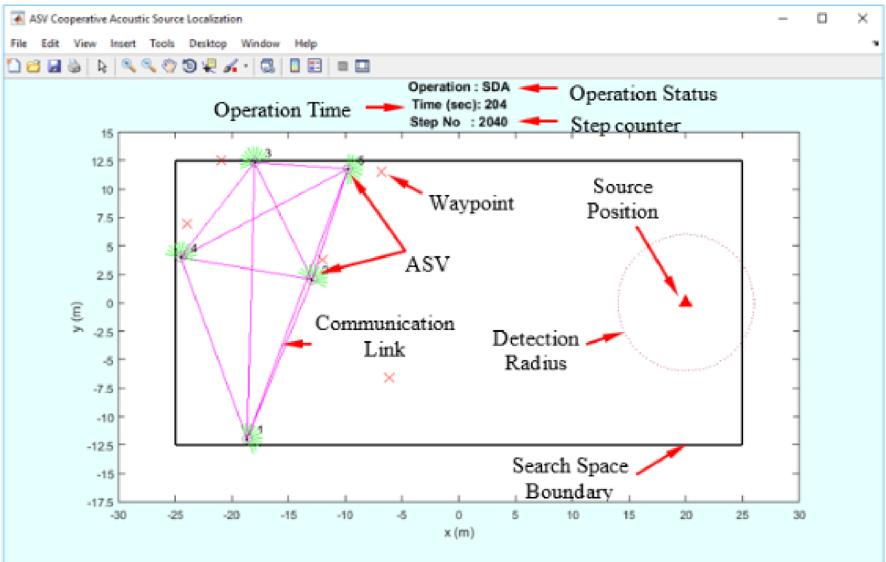

Figure 2. Swarm robot simulator

For the simulation purposes, the acoustic source is represented by an intensity decaying model based on a cylindrical spreading model given by (14).

$$P_a = 2\pi r h_s I \tag{14}$$

where P_a is the power of the source, r is the radius from the source (i.e., distance from ASV to the source), h_s is depth of the source (assumed to be constant) and I is the intensity of the source. By the energy conservation law, the intensity of the acoustic source can be derived as (15):

$$P_a = 2\pi r_0 h_s I_0 = 2\pi r h_s I \quad (15)$$

and solving for I gives:

$$I = I_0\left(\frac{r_0}{r}\right) + w \quad (16)$$

where $r=d_{rs}$ is the distance between source position and current robot position, w is the added parameter representing white Gaussian noise generated using MATLAB function *wgn()*, r_0 is the reference radius and I_0 is the corresponding acoustic source level. Note that this approximation model is a simplified model where it does not consider refraction, reflection and propagation effect of the sound wave. White Gaussian noise is considered since it closely represents the noise model in actual environment measured by the sensor. The illustration of the source intensity with and without noise effect is shown in Figure 4(a) and 4(b), respectively.

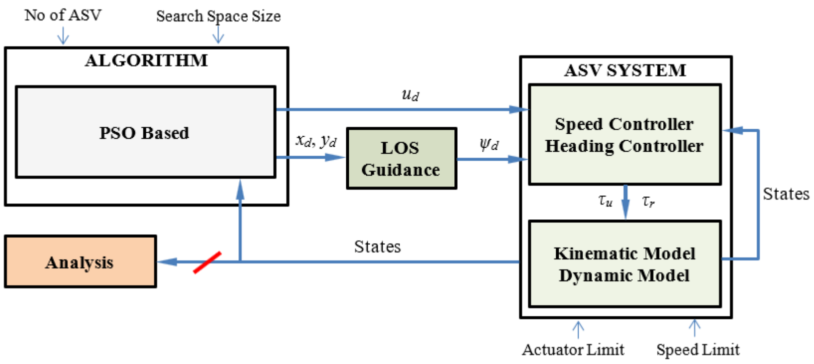

Figure 3. Swarm robotic simulator architecture

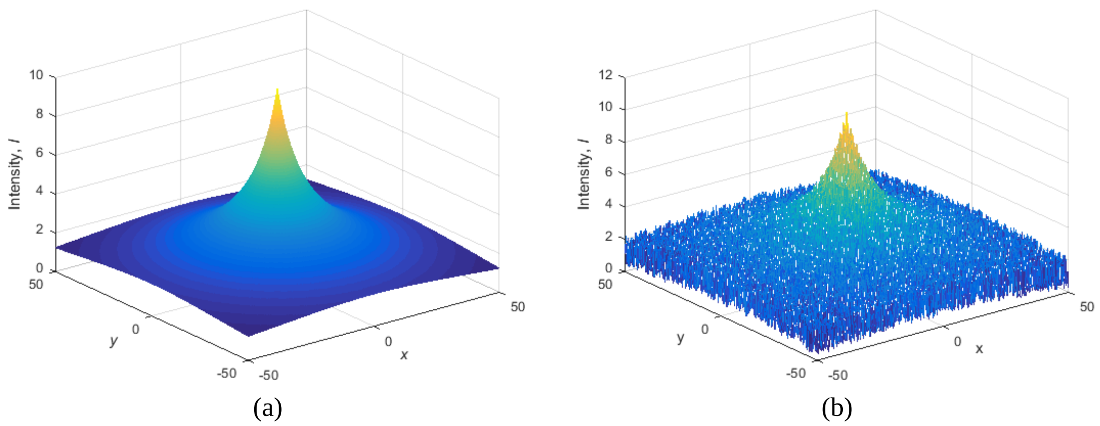

Figure 4. Acoustic source (a) without noise (b) with noise

3. RESULTS AND DISCUSSION

Based on the proposed algorithm and methodology, a series of simulations were performed in order to evaluate the fulfilment of the research objectives. The first objective of this research as previously stated is to evaluate performance of the proposed PSO for source searching in terms of target overshooting and impact of parameters on searching capability. The second objective is to evaluate the impact of different initialization strategies on source searching performance.

3.1. Modified PSO performance

The ADAPSO velocity limit, V_{max}, should be examined as one of the ADAPSO parameters since it has a direct impact on the efficiency of the source searching. In this study, the value of V_{max} is independent of the maximum robot velocity since the position update of the ADAPSO represents the desired waypoint instead of the robot position update step. To ensure quick convergence and reduce target overshooting (i.e., oscillation around the source), which necessitates additional control efforts, it is essential to find the optimal value V_{max}. A choice of a small V_{max} value may result in a slow convergence (indicated by actual searching time), as shown in Figure 5, and may cause the robot to become trapped in local optima because robots lose its capability to explore the searched area. On the other hand, a choice of a large V_{max} may result in large target overshooting or a continuous oscillation around the source, which is ineffective in terms of the desired control, as shown in Figure 6. Figure 7 depicts the potential ideal convergence, which is quick with nearly no oscillation convergence. Notice that equidistant initialization was used in this analysis to obtain consistent initialization position for different repetition of the simulations.

A simulation was executed with various V_{max} values to ascertain the best value, and the time it takes to attain convergence for various initialization types (such as centralize, equidistance, and random) was noted. Figure 7 displayed the amount of time required for various values of V_{max} to converge for various types of initializations in a search space with a size of 25x50 m^2. The best value for V_{max}, which may be used with various initialization procedures is 1.0 as shown in Figure 8. As a result, for the remaining simulation experiments, this value is set as the velocity limit for ADAPSO velocity update.

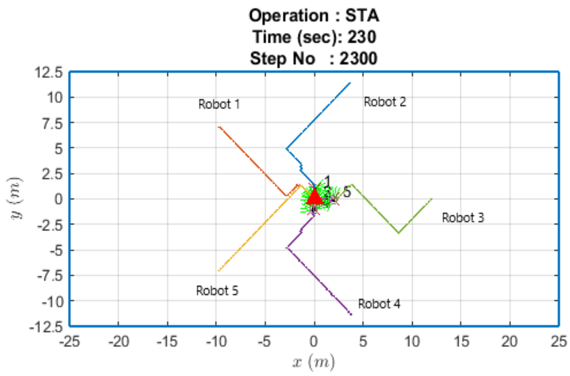

Figure 5. Slow convergence of the proposed PSO for small $V_{max} = 0.1$

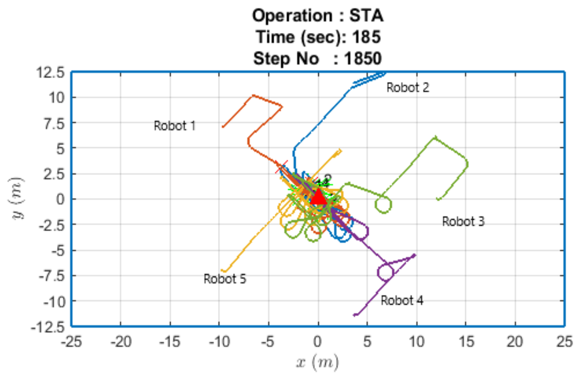

Figure 6. Oscillatory convergence of the proposed PSO for large $V_{max} = 5$

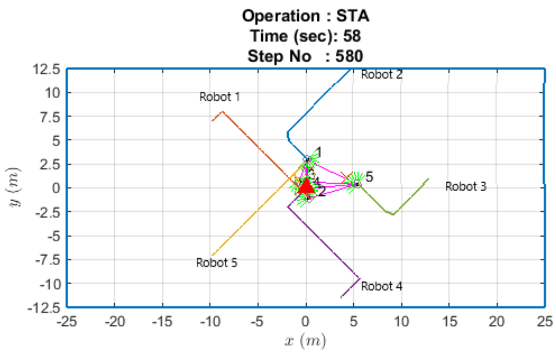

Figure 7. Optimal convergence of the proposed PSO when $V_{max} = 1$

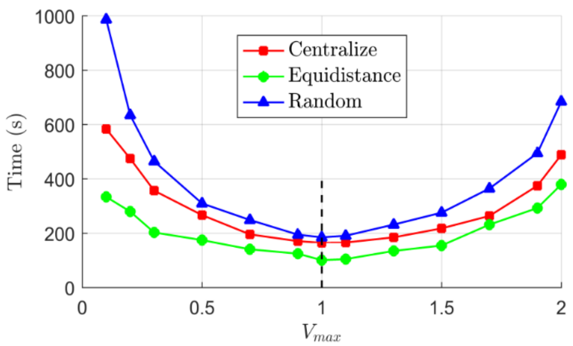

Figure 8. Optimal value of V_{max} for the proposed PSO

3.2. Impact of ADAPSO parameters

In this section, the effect of various ADAPSO parameters on source searching performance will be examined. Depending on where the source signal is possibly detected, source searching can start from a variety of initialization configurations. For instance, if the source signal is discovered only after a few iterations of random movement, the robots must execute source searching from the random position initialization. Similarly, if the source signal is discovered only after the robots have been deployed into the search space. Therefore, by taking into account various initialization strategies, a general overview of the ADAPSO performance in terms of fitness differences between the robots (i.e., robot similarity) was provided. Thus, the robustness of the proposed algorithm can be assessed taking into account a variety of initial source signal strengths by evaluating various initialization strategies of the ADAPSO.

For various starting procedures, the effect of various ADAPSO parameters (see (1) through (9)) on the searching performance is depicted by histograms as in Figure 9. When its effect on the source searching performance was assessed, the appropriate parameter was relaxed (i.e., set to zero) for this reason. Despite the fact that robots were deployed using various initialization techniques, it is clear from the figure that each parameter has a favorable effect on the overall ADAPSO performance. Even when the distance between the source position and first deployment site is similar, it takes more time to obtain convergence since the initial degree of similarity among the robots in a centralized initialization strategy is high. This is due to the fact that velocity update is minimal and the cognitive and social components of the ADAPSO become small when *gBest* and *pBest* are near to each other. Because robots were initialized at equal radial distances from the source, the equidistance initialization strategy also has the fastest convergence speed. The equally distributed beginning pattern quickens the robots' convergence rate even though their initial fitness is equivalent. On the other hand, random initialization has a medium rate of convergence since the robots' similarity depends on their initial random positions. The pace of convergence increases with the proximity of the robots to the source and with their random distribution. For the source searching task, however, only random and centralized initializations are feasible and can easily be employed.

Figure 10 illustrates how ADAPSO parameters affect the search results in terms of swarm best fitness, or *gBest*. The outcome demonstrated the beneficial effects of several ADAPSO parameters in terms of convergence speed and accuracy. As shown in Figure 10, deactivating one of the parameters decreases the ADAPSO's effectiveness. The combination of those parameters in the suggested ADAPSO algorithm, it is evidence from this finding, aids the robots in achieving a faster convergence speed and greater accuracy.

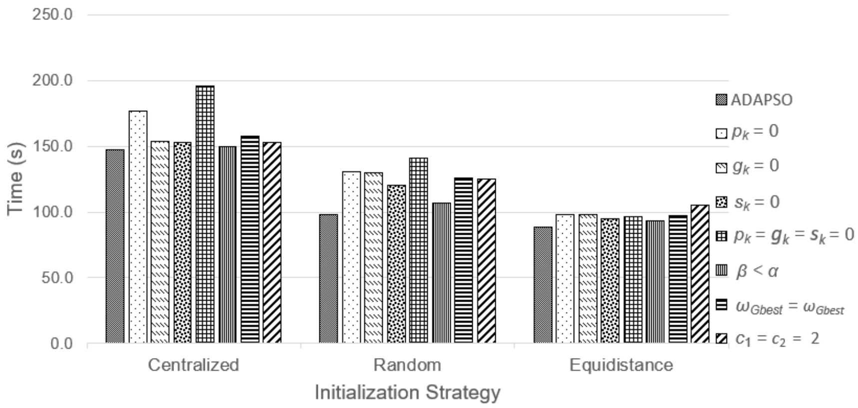

Figure 9. Impact of parameters on the overall ADAPSO performance

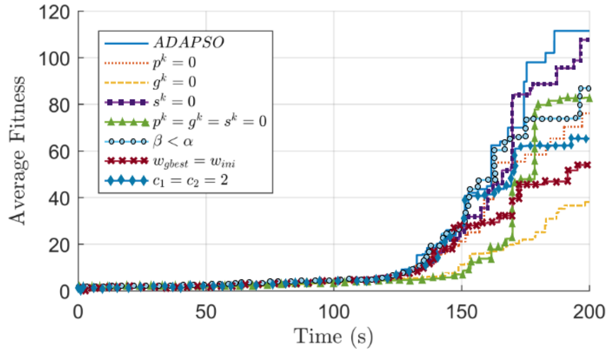

Figure 10. The impact of parameters on the overall ADAPSO performance in term of average final fitness for t_{max} = 200 s

3.3. Impact of initialization

Similar to the preceding part, three initialization strategies: random, equidistant, and centralized or initialization from the same region were taken into consideration to study the impact of initialization strategy. The goal of this section is to evaluate the proposed algorithm's performance and robustness to other benchmark methods under various initial robot configurations and robot counts. Benchmarking of ADAPSO's performance is done using the basic PSO algorithms listed in Table 1. This simulation takes into account a search space of 25×50 m^2. Figure 11 displays the simulations' outcomes. Figure 12 displays the success rate for various initialization techniques and for various robot counts. The dashed line indicates no convergence is observed.

According to the findings, the ADAPSO algorithm performs better than other algorithms in each of the three different initialization setups. Due to their poor ability to escape local optima, particularly when a small number of robots were taken into account, especially in a random and centralized initialization, conventional PSO algorithms such as IWPSO and CFPSO for small numbers of robots fail to achieve convergence compared to other algorithms. Because of its weak capacity to escape local optima caused by constant inertia weight, the IWPSO completely fails in the centralized initialization. Although the CFPSO can typically reach convergence, its rate of convergence is slow, especially when there are many robots that share characteristics, as was seen in the centralized initialization. However, as shown in Figure 12, the CFPSO has a greater propensity to trap into local optima when there are fewer robots present. It should be noted that IWPSO and CFPSO are non-adaptive PSOs, which is the fundamental cause of the two algorithms' inability to escape local optima when there is a significant degree of robot similarity. An example of a complete illustration of a source searching using ADAPSO is shown in Figure 13. In this figure, the robot traces during a complete source searching process when a source is positioned at $\boldsymbol{p_s} = (20, 0)$ in an obstacle's free environment. In this example, robots were deployed close to each other using a centralized initialization. Robots cooperatively search the target based on detected source signal and exchange the information within the swarm to reach convergence.

Table 1. Selected benchmark PSO algorithms

Reference	Algorithm	Remarks
This study	ADAPSO	Adaptive
Zou *et al.* [34]	IWPSO	Non-adaptive
Yang *et al.* [33]	DAPSO	Adaptive
Zou *et al.* [34]	CFPSO	Non-adaptive
Hereford *et al.* [35]	PEPSO	Non-adaptive

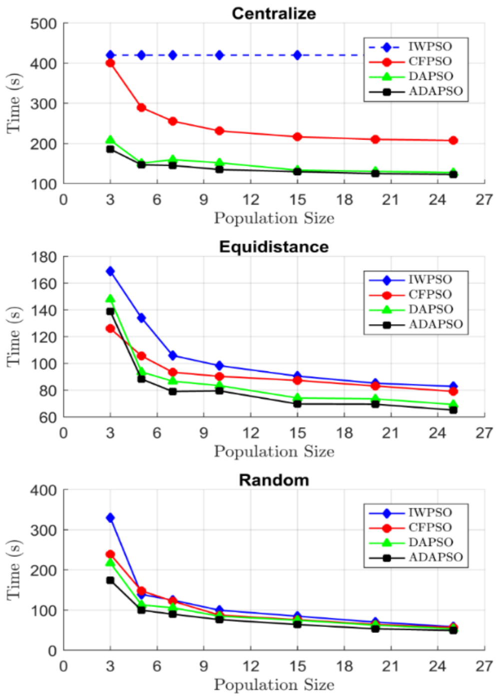

Figure 11. Performance of ADAPSO for different initialization methods in a search space of 25×50 m^2 using N=5

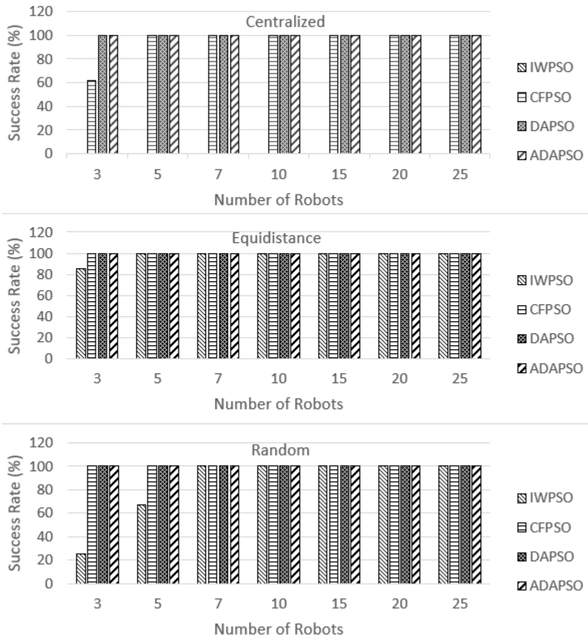

Figure 12. Success rate for different initialization configurations

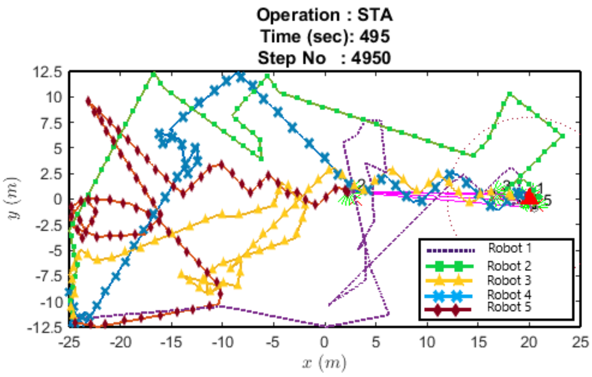

Figure 13. Robot traces during a complete source searching process when source located at p_s = (20, 0) in obstacles free environment.

4. CONCLUSION

In this study, source searching based on a modified PSO has been proposed and tested using a swarm of ASVs. The findings of the study showed that the proposed ADAPSO algorithm has successfully achieved better searching capability after the velocity limit is optimized by avoiding excessive oscillation. In addition, the impact of each parameter of ADAPSO has been evaluated and the result showed that each parameter has significant impact on the searching effectiveness. Moreover, it is also shown that each initialization strategy has a different impact on the searching operation. For future work, further improvement will be considered. In some searching scenarios multiple sources may exist within the same search space. In this situation, robots must be able to partition themselves into multiple sub swarms if all targets are the targets of interest or otherwise, they must be able to differentiate a correct target from the false targets. This process involves task distribution among the robots in the swarm may improve capability of the algorithm which may bring a step further towards real world implementation.

REFERENCES

[1] B. Septfons, A. Chehri, H. Chaibi, R. Saadane, and S. Tigani, “Swarm robotics: Moving from concept to application,” in *Smart Innovation, Systems and Technologies*, vol. 310, Springer Nature Singapore, 2022, pp. 179–189, doi: 10.1007/978-981-19-3455-1_14.

[2] M. H. A. Majid, M. R. Arshad, and R. M. Mokhtar, "Swarm robotics behaviors and tasks: A technical review," in *Studies in Systems, Decision and Control*, vol. 371, Springer International Publishing, 2022, pp. 99–167, doi: 10.1007/978-3-030-74540-0_5.

[3] P. G. Faria Dias, M. C. Silva, G. P. Rocha Filho, P. A. Vargas, L. P. Cota, and G. Pessin, "Swarm robotics: A perspective on the latest reviewed concepts and applications," *Sensors*, vol. 21, no. 6, pp. 1–30, Mar. 2021, doi: 10.3390/s21062062.

[4] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm and Evolutionary Computation*, vol. 50, p. 100565, Nov. 2019, doi: 10.1016/j.swevo.2019.100565.

[5] M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robotics and Autonomous Systems*, vol. 75, pp. 422–434, Jan. 2016, doi: 10.1016/j.robot.2015.08.010.

[6] Z. H. Ismail and M. G. M. Hamami, "Systematic literature review of swarm robotics strategies applied to target search problem with environment constraints," *Applied Sciences (Switzerland)*, vol. 11, no. 5, p. 2383, Mar. 2021, doi: 10.3390/app11052383.

[7] C. Calderón-Arce, J. C. Brenes-Torres, and R. Solis-Ortega, "Swarm robotics: Simulators, platforms and applications review," *Computation*, vol. 10, no. 6, p. 80, May 2022, doi: 10.3390/computation10060080.

[8] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI*, vol. 7, Apr. 2020, doi: 10.3389/frobt.2020.00036.

[9] M. Dadgar, S. Jafari, and A. Hamzeh, "A PSO-based multi-robot cooperation method for target searching in unknown environments," *Neurocomputing*, vol. 177, pp. 62–74, 2016, doi: 10.1016/j.neucom.2015.11.007.

[10] H. L. Kwa, G. Tokic, R. Bouffanais, and D. K. P. Yue, "Heterogeneous swarms for maritime dynamic target search and tracking," Oct. 2020, doi: 10.1109/IEEECONF38699.2020.9389145.

[11] W. Fu, J. Qin, W. X. Zheng, Y. Chen, and Y. Kang, "Resilient cooperative source seeking of double-integrator multi-robot systems under deception attacks," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4218–4227, May 2021, doi: 10.1109/TIE.2020.2987270.

[12] Z. Husain, D. Ruta, F. Sare, Y. Al-Hammadi, and A. F. Isakovic, "Inverted ant colony optimization for search and rescue in an unknown maze-like indoor environment," in *GECCO 2018 Companion - Proceedings of the 2018 Genetic and Evolutionary Computation Conference Companion*, Jul. 2018, pp. 89–90, doi: 10.1145/3205651.3205738.

[13] X. Zhang and M. Ali, "A bean optimization-based cooperation method for target searching by swarm UAVs in unknown environments," *IEEE Access*, vol. 8, pp. 43850–43862, 2020, doi: 10.1109/ACCESS.2020.2977499.

[14] B. Yang, Y. Ding, and K. Hao, "Target searching and trapping for swarm robots with modified bacterial foraging optimization algorithm," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, Jun. 2015, vol. 2015-March, no. March, pp. 1348–1353, doi: 10.1109/WCICA.2014.7052915.

[15] Z. Zheng and Y. Tan, "Group explosion strategy for searching multiple targets using swarm robotic," in *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, Jun. 2013, pp. 821–828, doi: 10.1109/CEC.2013.6557653.

[16] Y. Tan, "Group explosion strategy for multiple targets search in swarm robotics," in *Fireworks Algorithm*, Springer Berlin Heidelberg, 2015, pp. 285–299, doi: 10.1007/978-3-662-46353-6_17.

[17] C. K. Y. Fermin *et al.*, "Development and implementation of swarm sweep cleaning protocol for quadrotor unmanned aerial vehicle (QUAV) swarm," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, Nov. 2017, vol. 2017-December, pp. 1988–1991, doi: 10.1109/TENCON.2017.8228186.

[18] Q. Tang, F. Yu, Y. Zhang, L. Ding, and P. Eberhard, "A stigmergy based search method for swarm robots," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10386 LNCS, Springer International Publishing, 2017, pp. 199–209, doi: 10.1007/978-3-319-61833-3_21.

[19] Y. Katada, S. Hasegawa, K. Yamashita, N. Okazaki, and K. Ohkura, "Swarm crawler robots using Lévy flight for targets exploration in large environments," *Robotics*, vol. 11, no. 4, p. 76, Jul. 2022, doi: 10.3390/robotics11040076.

[20] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "A swarm robotic exploration strategy based on an improved random walk method," *Journal of Robotics*, vol. 2019, pp. 1–9, Mar. 2019, doi: 10.1155/2019/6914212.

[21] Q. Zhang, X. Wu, and X. Qi, "Target searching for multiple robots using hybrid particle swarm and bacterial foraging optimization," *IOP Conference Series: Earth and Environmental Science*, vol. 440, no. 4, p. 42063, Feb. 2020, doi: 10.1088/1755-1315/440/4/042063.

[22] H. Tang, W. Sun, H. Yu, A. Lin, M. Xue, and Y. Song, "A novel hybrid algorithm based on PSO and FOA for target searching in unknown environments," *Applied Intelligence*, vol. 49, no. 7, pp. 2603–2622, Jan. 2019, doi: 10.1007/s10489-018-1390-0.

[23] J. Li and Y. Tan, "Triangle formation based multiple targets search using a swarm of robots," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9713 LNCS, Springer International Publishing, 2016, pp. 544–552, doi: 10.1007/978-3-319-41009-8_59.

[24] B. Pang, Y. Song, C. Zhang, and R. Yang, "Effect of random walk methods on searching efficiency in swarm robots for area exploration," *Applied Intelligence*, vol. 51, no. 7, pp. 5189–5199, Jan. 2021, doi: 10.1007/s10489-020-02060-0.

[25] J. M. Hereford, M. Siebold, and S. Nichols, "Using the particle swarm optimization algorithm for robotic search applications," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium, SIS 2007*, Apr. 2007, pp. 53–59, doi: 10.1109/SIS.2007.368026.

[26] N. A. A. Aziz and Z. Ibrahim, "Asynchronous particle swarm optimization for swarm robotics," *Procedia Engineering*, vol. 41, pp. 951–957, 2012, doi: 10.1016/j.proeng.2012.07.268.

[27] Q. Tang, L. Ding, F. Yu, Y. Zhang, Y. Li, and H. Tu, "Swarm robots search for multiple targets based on an improved grouping strategy," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1943–1950, Nov. 2018, doi: 10.1109/TCBB.2017.2682161.

[28] C. H. Yuen and K. T. Woo, "A survivability enhanced swarm robotic searching system using multi-objective particle swarm optimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10386 LNCS, Springer International Publishing, 2017, pp. 167–175, doi: 10.1007/978-3-319-61833-3_18.

[29] M. Dadgar, M. S. Couceiro, and A. Hamzeh, "RDPSO diversity enhancement based on repulsion between similar ions for robotic target searching," in *19th CSI International Symposium on Artificial Intelligence and Signal Processing, AISP 2017*, Oct. 2018, vol. 2018-January, pp. 275–280, doi: 10.1109/AISP.2017.8324096.

[30] V. Garg, A. Shukla, and R. Tiwari, "AERPSO — An adaptive exploration robotic PSO based cooperative algorithm for multiple target searching," *Expert Systems with Applications*, vol. 209, Art. no. 118245, Dec. 2022, doi: 10.1016/j.eswa.2022.118245.

[31] J. Yang, R. Xiong, X. Xiang, and Y. Shi, "Exploration enhanced RPSO for collaborative multitarget searching of robotic swarms," *Complexity*, vol. 2020, pp. 1–12, Nov. 2020, doi: 10.1155/2020/8863526.

[32] M. D. Phung and Q. P. Ha, "Motion-encoded particle swarm optimization for moving target search using UAVs," *Applied Soft Computing*, vol. 97, Art. no. 106705, Dec. 2020, doi: 10.1016/j.asoc.2020.106705.

[33] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1205–1213, Jun. 2007, doi: 10.1016/j.amc.2006.12.045.
[34] R. Zou, V. Kalivarapu, E. Winer, J. Oliver, and S. Bhattacharya, "Particle swarm optimization-based source seeking," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 865–875, Jul. 2015, doi: 10.1109/TASE.2015.2441746.
[35] J. M. and M. A., "Bio-inspired search strategies for robot swarms," in *Swarm Robotics from Biology to Robotics*, InTech, 2010, doi: 10.5772/8600.

BIOGRAPHIES OF AUTHORS

Mad Helmi Ab. Majid received the B.Eng. degree in mechatronics engineering and the M.S. mechatronics engineering from International Islamic University Malaysia (IIUM), Malaysia, in 2009 and 2012 respectively. He holds a Ph.D. degree in robotics from Universiti Science Malaysia, Penang, in 2019. Currently, he is a senior lecturer at the Department of Software Engineering and Smart Technology, Faculty of Computer and Meta-Technology, Universiti Pendidikan Sultan Idris, Malaysia. His research interests include robotics, artificial intelligence, internet of things, smart systems, mechatronics system design, swarm intelligence, biomimetics, optimization, cooperative robotics, educational robotics, and STEM education. He can be contacted at email: madhelmi@meta.upsi.edu.my.

Mohd Rizal Arshad graduated from the University of Liverpool (UK), in 1994 with a B.Eng. in the field of medical electronics and instrumentation. He then pursued his MSc. in electronic control engineering at the University of Salford (UK), graduating in Dec. 1995. Following from this, in early 1996, he continued his study with a PhD degree in electronic engineering back at the University of Liverpool (UK) with specialization in robotic vision system. After completing his PhD training, i.e., January 1999, he started working at the Universiti Sains Malaysia (USM), Malaysia as a full-time academic. He has supervised many postgraduate students at the MSc. and PhD. levels. He has also published actively in local and international publications. He is currently a full professor in robotics at the School of Electrical and Electronic Engineering, USM. He is currently the president of the Malaysian Society for Automatic Control Engineers (MACE) and past-chair of the Oceanic Engineering Society (OES) Malaysia Chapter. In early 2017, he was awarded with the professional engineer (P.Eng) status by the Board of Engineer, Malaysia (BEM). Prof Rizal is well known as the pioneer of underwater system technology research efforts in Malaysia. His research projects are mainly in the area of underwater robotic platform development, new sensing device and mechanisms, and intelligent control algorithms. He has strong international research networks with researchers from countries such as United Kingdom, Singapore, India, USA and France. He can be contacted at email: eerizal@usm.my.

Mohd Faid Yahya graduated from the International Islamic University Malaysia in 2009 with a Bachelor degree in mechatronics engineering. In 2013, he completed a MSc. degree in mechatronics engineering at the same university. He then completed his PhD. degree at the University of Science, Malaysia with a specialization in robotics in 2019. His interest includes vision-based robot control, visual serving, control system, machine vision, and artificial intelligence. He is currently a senior lecturer under Mechatronics Engineering Department at Faculty of Electrical Engineering in Universiti Teknikal Malaysia Melaka. He can be contacted at email: faid@utem.edu.my.

Abu Bakar Ibrahim is an associate professor in the Computing Department, Department of Software Engineering and Smart Technolgy, Faculty of Computer and Meta-Technology at Sultan Idris Education University, Malaysia. He is received a B.Sc. in electrical engineering and a Master's Degree from Universiti Teknologi Malaysia (UTM) in 1998 and 2000, respectively. He received a Ph.D. in electronic engineering (communication) from Universiti Teknikal Malaysia Melaka (UTeM) in the years 2013. He had professional teaching experience in the development of low noise amplifier (LNA), radio frequency communication system, instructional technology, engineering mathematics, wireless communication and engineering education. He has developed confidence and interest in researching and teaching areas to enhance creative innovation in engineering, science and technology. He can be contacted at email: abubakar.ibrahim@meta.upsi.edu.my.