Covenant Journal of Informatics & Communication Technology. Vol. 9 No. 1, June, 2021
ISSN: ***print*** **2354 – 3566** ***electronics*** **2354 – 3507 DOI:**

An Open Access Journal Available Online

Obstacle Avoidance Scheme Based Elite Opposition Bat Algorithm for Unmanned Ground Vehicles

Zaharuddeen Haruna, Muhammed Bashir Mu'azu, Yusuf Abubakar Sha'aban, Emmanuel Adewale Adedokun

Department of Computer Engineering, Faculty of Engineering, ABU Zaria, Nigeria.
elzet2007@gmail.com, muazumb1@yahoo.com, shaabanyusuf@gmail.com, adewaleadedokun@yahoo.com

Received: 30.03.2021 Accepted: 31.05.2021

Date of Publication: June, 2021

Abstract— Unmanned Ground Vehicles (UGVs) are intelligent vehicles that operate in an obstacle environment without an onboard human operator but can be controlled autonomously using an obstacle avoidance system or by a human operator from a remote location. In this research, an obstacle avoidance scheme-based elite opposition bat algorithm (EOBA) for UGVs was developed. The obstacle avoidance system comprises a simulation map, a perception system for obstacle detection, and the implementation of EOBA for generating an optimal collision-free path that led the UGV to the goal location. Three distance thresholds of 0.1 m, 0.2 m, and 0.3 m was used in the obstacle detection stage to determine the optimal distance threshold for obstacle avoidance. The performance of the obstacle avoidance scheme was compared with that of bat algorithm (BA) and particle swarm optimization (PSO) techniques. The simulation results show that the distance threshold of 0.3 m is the optimal threshold for obstacle avoidance provided that the size of the obstacle does not exceed the size of the UGV. The EOBA based scheme when compared with BA and PSO schemes obtained an average percentage reduction of 21.82% in terms of path length and 60% in terms of time taken to reach the target destination. The uniqueness of this approach is that the UGV avoid collision with an obstacle at a distance of 0.3 m from nearby obstacles as against taking three steps backward before avoiding obstacle.

Keywords/Index Terms—Unmanned Ground Vehicle, Elite Opposition Bat Algorithm, Distance Threshold, Path Length, Time Taken

1. Introduction

Unmanned Ground Vehicles (UGVs) have been used to replace humans for various military and civilian applications especially in the areas that are mostly dangerous and hazardous to human beings. UGVs has applications such as environmental monitoring, reconnaissance, search and rescue operations, bomb detection, cleaning, data acquisition, patrol, transportation, surveillance, packet delivery, etc. (Anupama *et al.*, 2014; Jabbarpour *et al.*, 2017). The operation of UGVs is divided into two classes: autonomous which is control without an onboard human operator and teleoperated which is control via a communication link from a remote position by a human operator. For the UGVs to be able to execute tasks in autonomous or teleoperated mode, path planning is necessary.

The path planning of UGV is the generation of an optimal collision-free motion path for UGV to move from its source location to destination in an environment populated with obstacles (static or dynamic). The optimality of the path is calculated based on minimum energy consumption, cost, shortest distance, and shortest time. The shortest distance and shortest time are the most frequently used path planning criteria (Abbas & Ali, 2014; Hossain & Ferdous, 2015). The path planning of UGV is categorized into global or off-line path planning and local or on-line path planning. Off-line path planning of UGV is implemented when information about the environment is fully known in advance. As such, UGV only computes the optimal path once at the beginning of its motion and then follows the generated path to the destination. On the other hand, on-line path planning is implemented in an environment where information about the environment is not completely known in advance. As such, UGV extracts these information using sensors from the environment as it traverses to the destination (Raja & Pugazhenthi, 2012). The path planning problem of UGV belongs to a larger class of problems relating to routing and scheduling which is known as nondeterministic polynomial complete problem (Hosseinzadeh & Izadkhah, 2010). The path planning problem has been regarded as an optimization problem since the advent of artificial intelligence (Negnevitsky, 2005).

Optimization refers to the process of computing the best value(s) of parameters of a problem using available scarce resources to determine the minimum or maximum cost of an objective function under some defined constraints (Audee *et al.*, 2019; Corne *et al.*, 1999; Horst *et al.*, 2000; Saremi *et al.*, 2017). The formulation of an optimization problem is presented generally in the following sequence (Saremi *et al.*, 2017): identification of the problem's parameters, recognition of the constraints that are subjected to the parameters, formulation of the cost function, selection of suitable optimizer to solve the problem. The cost function must describe the problem mathematically in correct order and express clearly the correlation between the parameters of the problem (Civicioglu & Besdok, 2013). In this research, an optimization problem is formulated using distance as a function of energy consumption of the UGV. The cost function is optimized so as to determine a collision-free motion path for the UGV.

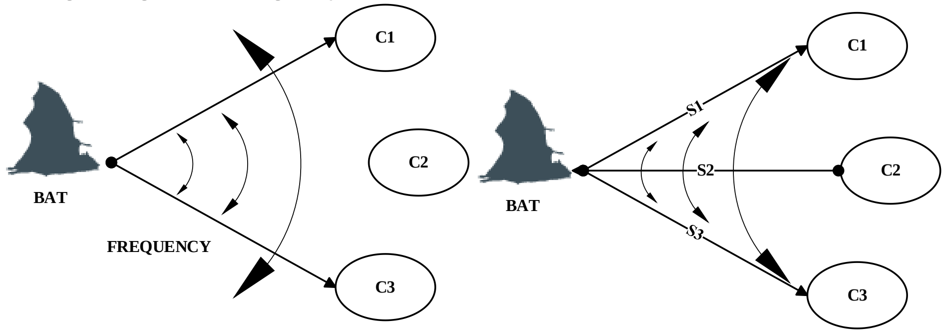
Different techniques have been presented in the literature for path planning of UGV. These techniques are categorized into classical approaches (road map, cell decomposition, visibility graph and artificial potential field)

(Lv & Feng, 2017), heuristic techniques (A* algorithm (Duchoň *et al.*, 2014), D* algorithm (Al-Mutib *et al.*, 2011), hill climbing algorithm (Hoffmann, 2000) and genetic algorithm (Lamini *et al.*, 2018; Sahu & Mishra, 2017)) and metaheuristic techniques (bat algorithm (Gigras & Vasishth, 2015), cuckoo search algorithm (Mohanty & Parhi, 2013; Mohanty & Parhi, 2016), particle swarm algorithm (Gigras & Vasishth, 2015), ant colony algorithm (Gigras & Gupta, 2012; Jabbarpour *et al.*, 2017; Neydorf *et al.*, 2018), firefly algorithm (Brand & Yu, 2013; Chen *et al.*, 2017), artificial immune system (Deepak & Parhi, 2013; Eslami *et al.*, 2012), bacteria foraging optimization (Hossain & Ferdous, 2015), artificial bee colony (Abbas & Ali, 2014; Contreras-Cruz *et al.*, 2015; Li *et al.*, 2018) and plant growth algorithm (Zhou *et al.*, 2017)). This research focuses on metaheuristic search algorithm for path planning of UGV.

Meta-heuristic search algorithms are global optimization techniques used for solving real-life and complex optimization problems (Annicchiarico *et al.*, 2005; Gandomi *et al.*, 2013). These techniques mimic the natural phenomena or social behavior by using stochasticity and iterations to obtain better solutions for optimization problems (Talbi, 2009). The algorithms also use exploration (also known as diversification) and exploitation (also known as intensification) to obtain better search performance. Exploitation generally seek for the best solution around the current generated solutions, while exploration process uses randomization to enables the algorithm explores the solution search space efficiently (Yang, 2010). Striking balance between these two phenomenon will result in obtaining a global optimum solution of an optimization problem (Saremi *et al.*, 2017). The bat algorithm which is developed by mimicking the echolocation features of bats is an example of meta-heuristic search algorithm.

The bat is an intelligent animal with an extraordinary navigation capability. The bat while searching for food, navigate to a desired target from its current location by emitting an ultrasonic sound pulse and listen for the reflected sound that bounces back from nearby objects. The variation in the magnitude of the echo (reflected sound) as received by each of the bat's ear is used to localize the position of the objects. The time delay between the emission of the sound pulses and the reception of the echo is used to determine its distance to an object. It uses the compass principle to obtain the target direction so as to beacon directly to the goal. Bats use the Doppler effect and structure of the echoes to respectively determine the speed and shape of an object along their moving paths. These allow them to employ acoustic landmarks (trees and rocks) for foraging and navigation (Geva-Sagiv *et al.*, 2015). In this research, the navigation principle of bats will be mimicked to design an obstacle avoidance scheme that will guide the UGV to traverse to the target location in an environment with obstacles. Figure 1 presents obstacle sensing using the echolocation feature of bats.

As shown in Figure 1, bats emit a very loud sound pulse at a certain frequency (Voigt *et al.*, 2017). C_1, C_2 and C_3 are the surrounding objects in the environments that serve as obstacles. When an obstacle is sensed with the aid of echolocation as shown in Figure 1(a), the bats make decisions based on the acquired information from the environment to avoid colliding with the obstacles. This information is in form of distance (S1, S2, and S3) between bat and obstacles which is computed based on the principle of echo as shown in Figure 1(b). If the distance is within a threshold value, the bat avoids the obstacle by generating an obstacle free path. Otherwise, the bat continues to move until the threshold is reached. This gives it the ability in guiding a moving object like UGV to reach its destination in the presence of obstacles. The organization, structure, presentation and discussion of results in this paper were improved using Misra (2021).



a. Obstacle Sensing b. Distance Computation

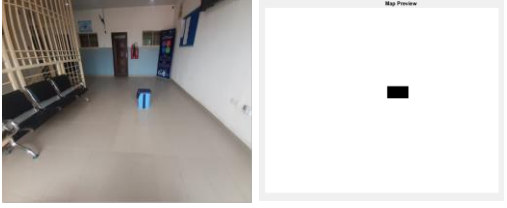
Figure 1: Obstacles Sensing through Echolocation

2. Methodology

The obstacle avoidance system consists of a simulation map, a sensing system for determining the external state of the UGV, and implementation of EOBA based obstacles avoidance scheme.

2.1 Development of Simulation Map

The simulation map is developed using a simulation map generator in the MATLAB App tab which accepts a captured image (obstacleImage.jpg) of the track of the UGV as one of the inputs. The two other inputs are the dimension of the UGV track (2.5 x 4) *m* and the number of obstacles (as 1). Figure 2 presents the obstacle map of the UGV track.



a. Captured Image b. Generated Map in 2D

Figure 2: Obstacle Map

Figure 2(a) is the image of the reception of Computer Engineering Department captured using Samsung Galaxy A30 with the following properties: 4G RAM, the processor of 1.8GHz, the camera of 15.93MP, and image size of 4608x3456 pixels. The blue object in the image is a bin that is placed as an obstacle in the environment. The corresponding image generated by the simulation map generator app is presented in Figure 2(b) with the black rectangle as the obstacle.

2.2 Obstacle Detection System

The developed obstacle detection system is presented in Fig. 3, which consists of a single HC-SR04 ultrasonic sensor (US) for generating information about the state of the environment. The inputs to the obstacle detection system are distance threshold and US value, which is the distance of the UGV to the obstacle along its line of sight. Figure 3 presents the flowchart of the obstacle detection system.

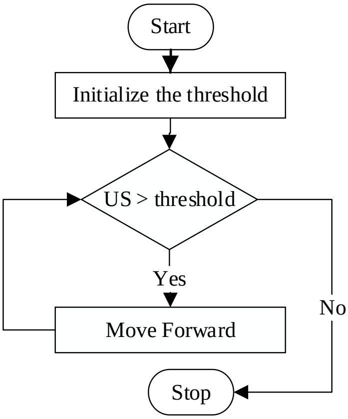


Figure 3: Flowchart of Obstacle Detection System

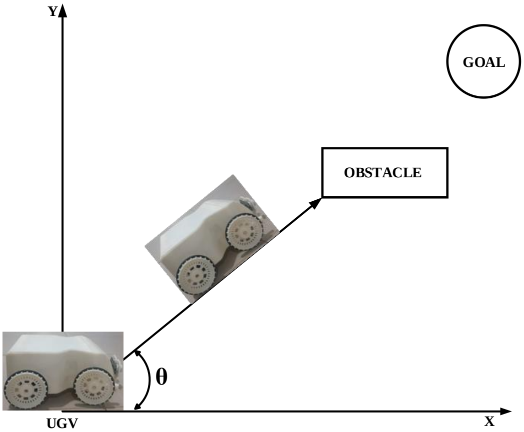


Figure 4: Conceptualization of the UGV Obstacle Detection

If the reading from the ultrasonic sensor is greater than the threshold, the UGV moves towards the target destination, else it halts the motion. The obstacle detection in Figure 3 is conceptualized in Figure 4:

Figure 4 presents the motion of the UGV towards the goal location in the presence of an obstacle along its motion path. The motion of the UGV was halted as the distance threshold is met. The Simulink model of the obstacle detection system is presented in Figure 5:

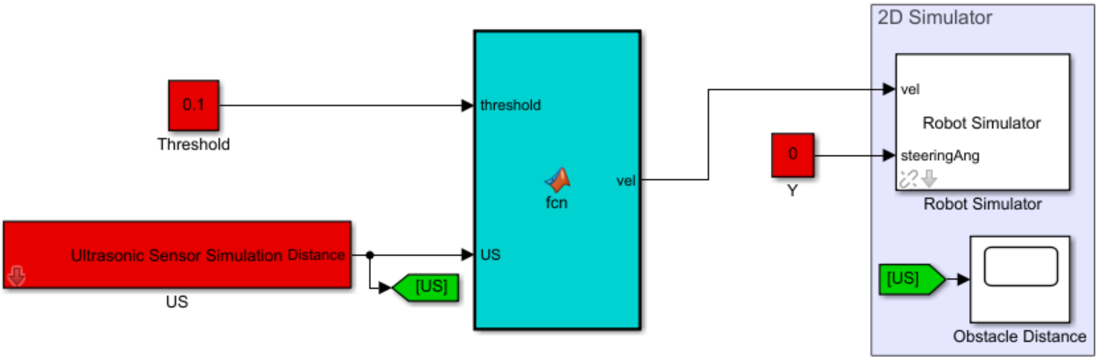


Figure 5: Simulink Model of Obstacle Detection System

The MATLAB US block in Figure 5 has a sensing range of 0.02 m to 2 m with a measuring accuracy of 0.002 m. A MATLAB function block is used for processing the data received from the US for comparison with the threshold and the output is displayed in a 2D robot simulator (which contains the generated obstacle map). The distance threshold of 0.1 m, 0.2 m, and 0.3 m were used to determine the optimal distance threshold for the UGV to avoid colliding with any obstacle along its line of sight.

2.3 Obstacle Avoidance Scheme

To avoid colliding with the static obstacle(s) in the environment, the elite opposition bat algorithm (EOBA) is adopted from Haruna *et al*. (2017). The procedure involves in the implementation of EOBA for generating an optimal coordinate are discussed as follows:

2.3.1 : Simulation Parameters of EOBA

The parameters of the EOBA were initialized with the values reported in Table 1:

Table 1. EOBA Simulation and Control Parameters

S/N	Simulation Parameters	Value
a.	Population Size	25
b.	Frequency Range	0-2
c.	Initial velocity of bats	0
d.	Pulse emission rate	0.5
e.	Loudness	0.25
f.	Maximum iteration	100
g.	Dimension of search space	2

2.3.2 : Algorithm Input

The inputs to the EOBA are the coordinates of the current state of the UGV and that of the target destination. The coordinate of the current state is the state of the UGV when the distance threshold is met. The EOBA will then generate optimal coordinate, that is, the coordinate with minimum distance and bending energy to the destination. This is mathematically presented as:

$$[F_{min}, best] = EOBA([x, y], [x_t, y_t]) \quad (1)$$

Equation (1) is a function that calls the EOBA to optimize the path of the UGV, where Fmin is the fitness value with the lowest distance and best is the optimal coordinate that generated Fmin.

2.3.3 : Coordinate Update

Based on the best value generated by the EOBA, the state of the UGV is updated as follows:

$$\begin{aligned} x_{new} &= x + best(1) \\ y_{new} &= y + best(2) \end{aligned} \quad (2)$$

Where x_{new} and y_{new} are respectively the coordinate of the new state along the x and y directions. The flowchart for the developed scheme is shown in Figure 6:

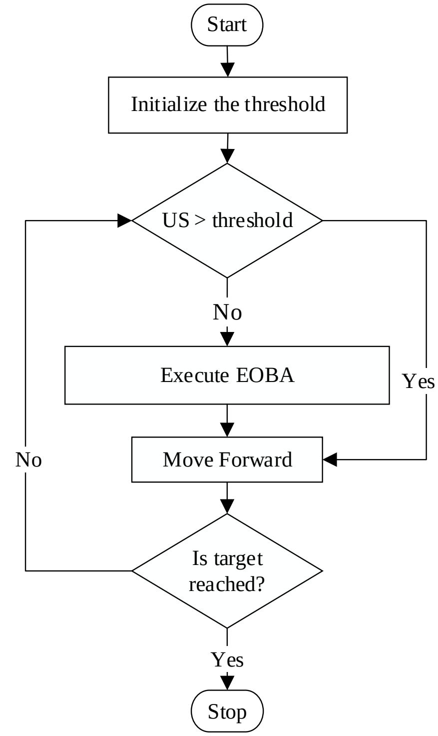


Figure 6: Flowchart of the EOBA based Obstacle Avoidance Scheme

Figure 6 shows the flowchart of the EOBA based obstacle avoidance scheme that generates the optimal coordinate that steers the UGV to avoid collision with obstacles. The system is initialized by defining the desired distance threshold for safe navigation of the UGV in the presence of an obstacle. The UGV check whether there is an obstacle along its path using the ultrasonic sensor (US) prior to the commencement of its motion. This is achieved by comparing the reading from the US with the desired distance threshold. If the reading from the US is less than or equal to the desired distance threshold, EOBA based obstacle avoidance scheme is implemented, else it moves towards the target destination. This is conceptualized in Figure 7:

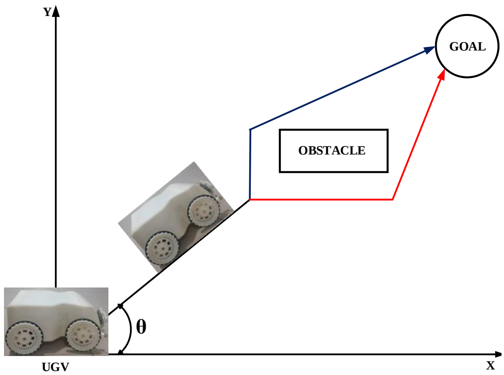


Figure 7: Framework of the UGV Obstacle Avoidance

Figure 7 shows the concept of the EOBA based obstacle avoidance scheme. It can be seen from the Figure that there are two possible paths (blue and red) the UGV can follow to reach the goal location. The scheme selects the optimal path (i.e., the shortest path to the goal location) for the UGV to reach the target. The developed Simulink model is presented in Figure 8:

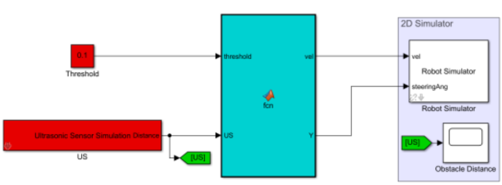


Figure 8: Simulink Model of EOBA based Obstacle Avoidance Scheme

Figure 8 is the Simulink model of the obstacle avoidance scheme using EOBA which output an optimal coordinate that is transformed to speed and steering angle to the 2D robot simulator.

3. Results and Discussions

The developed perception system consists of two Simulink models: obstacle detection system and EOBA based obstacle avoidance scheme.

3.1 Generated Simulation Map

The map presented in Figure 1 is the obstacle map developed using MATLAB Simulation map generator. The model of the UGV connected to a 2D simulator containing the generated map is shown in Figure 9:

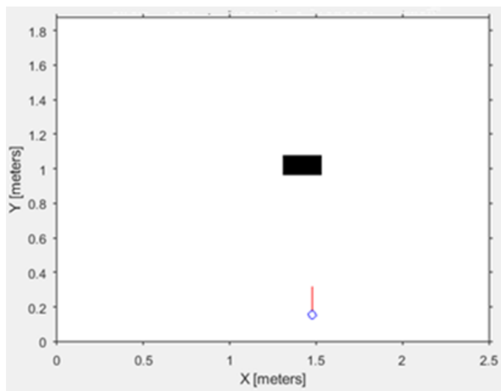
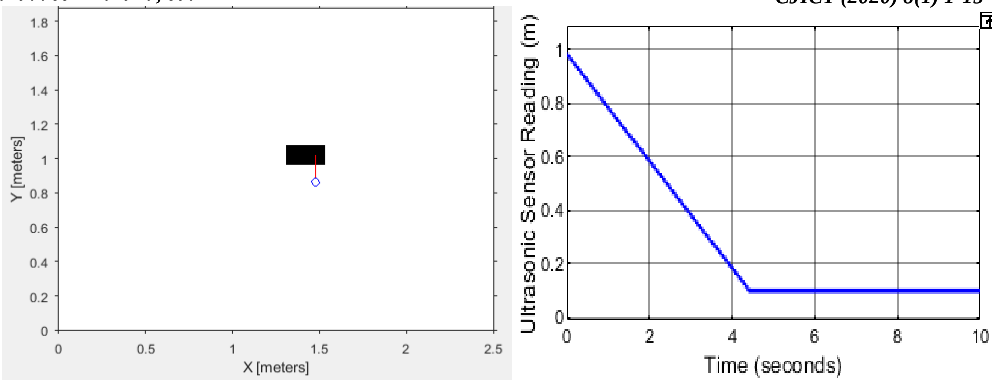


Figure 9. UGV based Obstacle Map

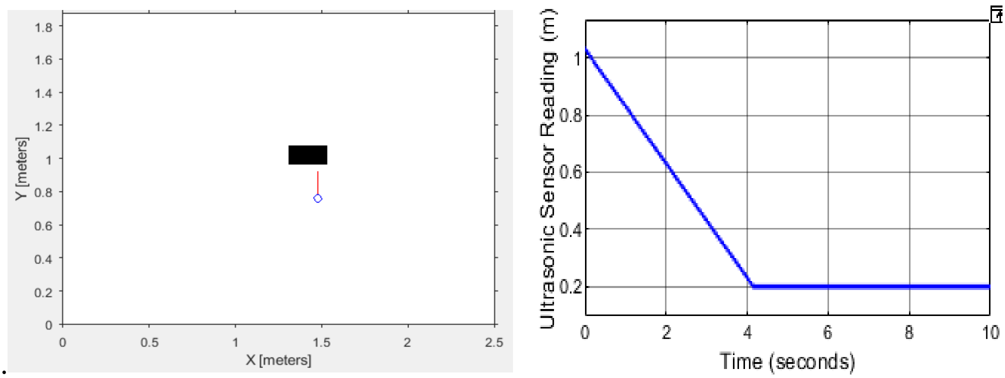
Figure 9 presents the generated obstacle map, where the blue circle and the red line represent the UGV and orientation of the UGV respectively. The black rectangular object is the obstacle in the environment.

3.2 Obstacle Detection System

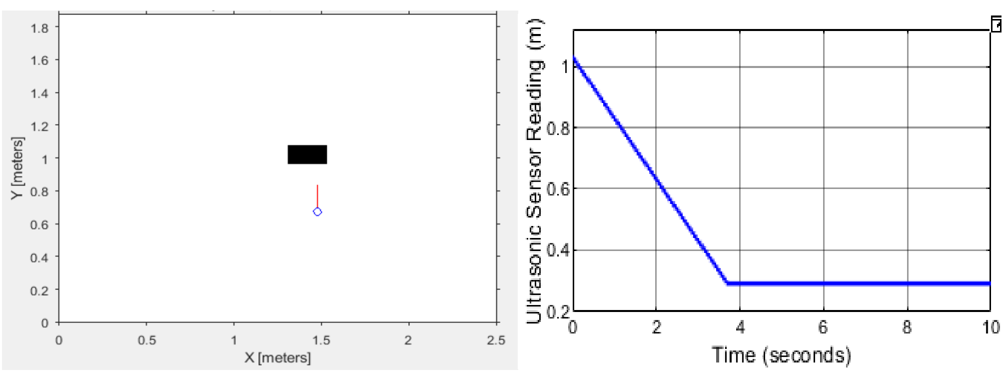
The Simulink model developed in Figure 4 is simulated when different distance thresholds of 0.1 m, 0.2 m, and 0.3 m were used. The result obtained is shown in Figure 10.



a. Distance threshold is set as 0.1 m



b. Distance threshold is set as 0.2 m



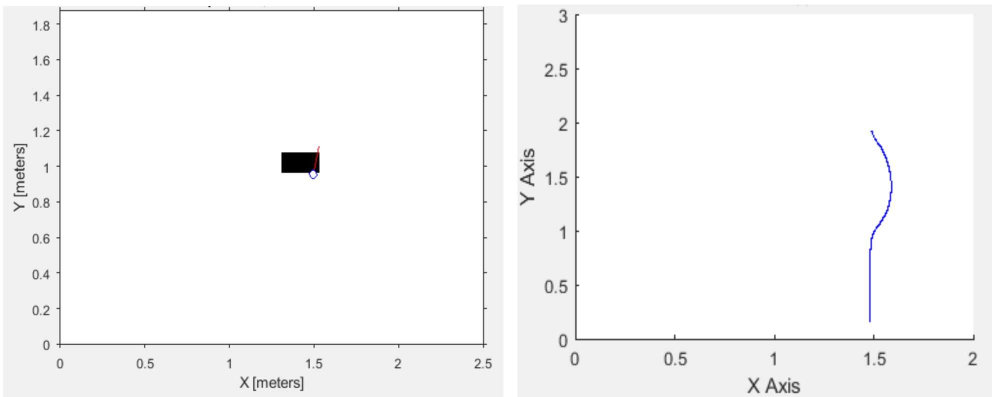
c. Distance threshold is set as 0.3 m

Fig. 10: Obstacle Detection using Ultrasonic Senor

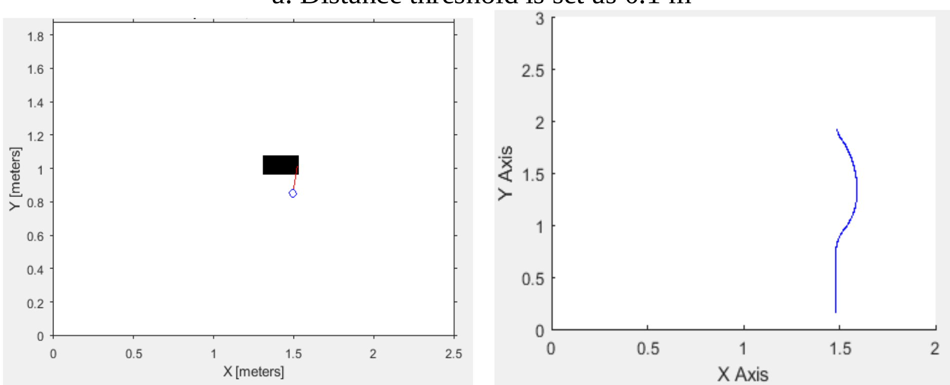
Figures 10(a), (b), and (c) are the obstacle detection results of the UGV with different distance threshold values of 0.1m, 0.2m and 0.3m respectively. The distance between the UGV and the obstacles can be clearly presented in the white background figures while the motion of the UGV is presently in the black background figures. From the figures, it is evident that UGV can effectively avoid collision with the obstacles when the distance threshold is 0.3m. This is due to the spacing between it and the obstacle and the size of the obstacle.

3.3 Obstacle Avoidance Scheme

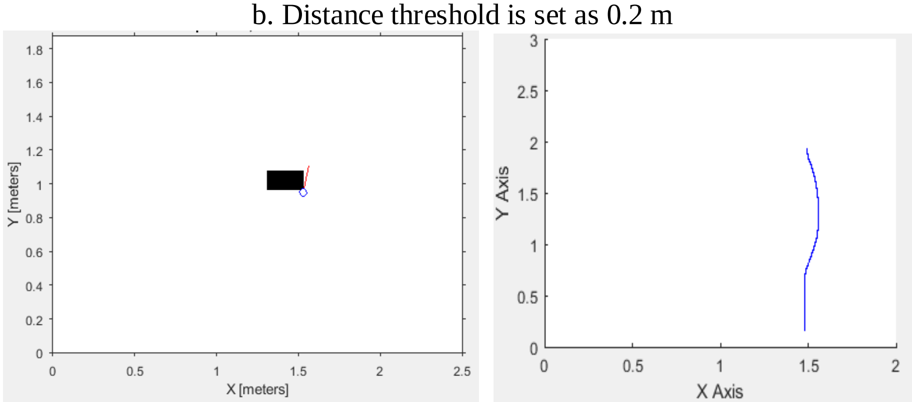
The Simulink model developed in Figure is simulated when different distance thresholds of 0.1 m, 0.2 m, and 0.3 m were used. The simulation result obtained is shown in Figure 11:



a. Distance threshold is set as 0.1 m



b. Distance threshold is set as 0.2 m



c. Distance threshold is set as 0.3 m

Figure 11: Implementation of EOBA based Obstacle Avoidance Scheme

Figure 11 is the motion of UGV from a source positioned at (1.4775, 0.15409, pi) to a goal positioned at (1.5,1.8). The motion is completely linear until the reading from the US is either 0.1 m, 0.2 m, or 0.3 m, when the EOBA based obstacle avoidance scheme was implemented as presented in Fig. 7 to steer the UGV from colliding with the obstacle as shown in Figure 10 a, b and c (second column). It can be seen from Figures 10(a and b) that the distance threshold of 0.1m and 0.2m are not optimal for obstacle avoidance as the vehicle collided with the obstacle as shown in the first column. It is evident that the UGV avoid been collided with the obstacle when the distance threshold is 0.3m. therefore, depending on the size of the UGV and the obstacle, a distance threshold of 0.3 m (30 cm) is safe for optimal obstacle avoidance.

3.4 Comparative Analysis

The performance of the developed obstacle avoidance scheme was evaluated by comparing with that of bat algorithm and particle swarm optimization methods reported in terms of path length and time taken to reach the destination.

Table 2. Performance Comparison in Navigating to Goal (5m,5m)

Obstacle Avoidance Scheme	Path Length (m)	Time (s)
PSO based scheme	3.49	7.06
BA based Scheme	2.74	4.78
EOBA based Scheme	2.40	2.28

Table 2 presents the results obtained when the EOBA based obstacle avoidance scheme was compared with standards methods, the simulation results obtained in terms of path length and time taken by the UGV to reach the destination shows that the developed scheme (EOBA based) is superior to the standard schemes. This also confirmed the superiority of the bat algorithm over the particle swarm algorithm as stated by Yang (2010). Figure 12 presents a clear comparison of the three obstacle avoidance schemes.

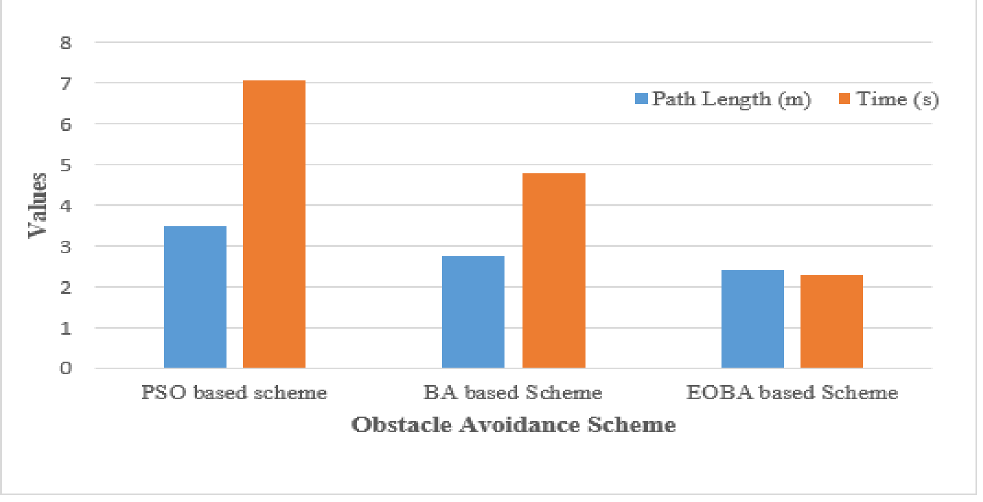


Figure 12. Performance Comparison of Obstacle Avoidance Scheme

4. Related Work

This section focused on the review of works that are based on the development of path planning approaches showing their basic principles, strengths and weaknesses which presented the need to develop an optimal path planning scheme.

Abbas and Ali (2014) presented a path planning scheme for an autonomous mobile robot based on a directed artificial bee colony (ABC) algorithm. In the work, an improved ABC called directed artificial bee colony (DABC) was proposed to solve the path planning problem by using the current direction of the best bee to guide the other bees towards the direction. A 10x10 simulation environment was then developed with the robot starting point as (0,0) and the robot target point as (10,10). The performance of the scheme was tested and compared with the standard ABC algorithm, bacteria colony, and genetic algorithm. The result shows that the proposed scheme is effective and obtains trajectories with satisfactory results. However, the path followed by the autonomous mobile robot is not always the optimal path as the robot had to encounter obstacles that increased the computational time and path length.

Hossain and Ferdous (2015) proposed a new path planning technique based on bacterial foraging optimization (BFO) to obtain a collision free path for a robot traversing towards the target point and avoiding obstacles using randomly distributed circular particles around the robot. The autonomous robot used an ultrasonic sensor to perceive the obstacle's location in the environment. The performance of the proposed technique was compared with the basic BFO and particle swarm optimization algorithm. The results show that the proposed technique provided a better optimal collision free path with minimum time taken in reaching the target destination. However, the robot at some points had to encounter obstacles before reaching the target which resulted in obtaining a non-optimal path.

Gigras and Vasishth (2015) showed the growth of path planning by comparing two algorithms of initial and recent phases, in which one is particle swarm optimization (PSO) algorithm, while the other is the bat algorithm (BA). The results show that the BA was more efficient than the PSO, as the number of iterations taken by BA was less than that of PSO, which also showed that the time taken by BA in finding an obstacle free path would be lesser than that of PSO. However, the path followed by the robot is not always optimal as the robot has to come in contact with the robot before avoiding it.

Roy *et al.* (2017) presented a path planning of a mobile robot using reinforcement learning and image processing. In their work, the image of the indoor environment was captured from the ceiling and processed using the image processing technique. Template matching of CV such as low pass filtering, smoothening, contour mapping and condensation algorithm was then used to detect and track the position of the mobile robot in the environment. Q-learning which is a model free reinforcement machine learning technique was used for the implementation of path planning in the indoor environment by planning an optimal path for the mobile robot from start location to goal location. The simulation result shows that Q-learning technique is effective for path planning of mobile robot moving from start location to goal state. However, the path followed by the robot is not always optimal as the robot at

times had to encounter obstacles before avoiding them.
Zhou *et al*. (2017) presented a novel path planning technique developed based on plant growth mechanism. In the work, a bio-inspired computing algorithm was developed that mimics the plant growth mechanism. The algorithm works based on four rules of plant growth which are: phototropism, negative geotropism, apical dominance, and branch. The performance of the developed algorithm was evaluated using two-dimensional path planning problems. Seed germ and lighting source were considered as the starting point and target point of the algorithm respectively, while the plant growth behavior was considered to be the same in each iteration. The simulation results when compared with A*, ABC and DE showed that the algorithm presents a good path planning ability with a suitable configuration parameter. However, the path followed by the robot is not always optimal due to the increase in path length and the vehicle will consume more energy due to its continuous oscillation.
Singh and Thongam (2018) presented fuzzy logic control system for navigation of mobile robot in an environment with static obstacles. The fuzzy logic control system is a Mandani fuzzy inference developed in MATLAB R2014a which accepts the angle between the orientation of the robot and robot rotation angle and the distance generated by an ultrasonic sensor between the robot and the obstacle. The output of the developed control system are the wheels angular velocities of the robot. The performance of the developed control system was evaluated using minimum bending energy, shortest path length and minimum time in avoiding obstacles using a simulation environment. The robot reaches the target destination without collision with any obstacle in the environment and energy consumption was reduced by the reduction in travelling time. However, the path followed by the robot is not optimal even in an environment without obstacle, which resulted in an increase in path length and time to reach the target destination.
Lamini *et al*. (2018) presented an autonomous mobile robot path planning using genetic algorithm (GA) approach. In their work, the GA was modified by improving its crossover operator and then used to generate optimal path in a static environment. This enables the algorithm to avoid premature convergence and generate feasible path with better fitness value. The performance of the GA based path planning approach was evaluated and compared with different GA based approaches in different simulation environment with grid maps sizes of (29x30), (13x30), (5x9), and (60x50) cells. The simulation result shows that the GA based path planning approach generated an optimal path with minimum number of turns and iteration. However, path followed by the robot is not optimal which resulted in paths with longer path length and time taken to reach the target.
Rostami *et al*. (2019) presented a mobile robot obstacles avoidance technique for generating an optimal path in a static environment. A modified artificial potential field algorithm was developed to address the drawback of the standard artificial potential algorithm so that the mobile robot will not be trapped in local minima. The developed obstacles avoidance technique based modified artificial potential technique was compared with existing path planning approaches in generating safe and collision-free optimal path. Circular obstacles with

different radius were modelled in the environment and the developed technique was effective without being trapped in local minima. However, the path followed by the robot is not always optimal as at times the robot has to be in contact with the obstacle before avoiding it. This increases the computational cost and energy consumption of the robot.

Ajeil *et al.* (2020) developed hybrid optimization path planning algorithm for a mobile robot in static and dynamic environments. The hybrid algorithm is a combination of PSO and modified frequency BA. A multi objective optimization problem was formulated based on shortest distance and path smoothness which was optimized by the developed path planning algorithm to generate an optimal collision free path. The developed path planning algorithm was compared with genetic, bee colony, chaotic bee swarm optimization and directed artificial bee colony-based path planning algorithms. The algorithm when tested under different scenarios showed that the developed path planning algorithm obtained the best path with minimum travelled distance. However, the mobile robot has to take three steps backwards when an obstacle is encountered, this resulted in an increase in time to reach the target destination.

5. Conclusion

In this research, an obstacle avoidance scheme has been developed based on the navigation concept of bats and its performance was evaluated by comparing with standard schemes, bat algorithm and particle swarm optimization-based schemes. The simulation results showed the effectiveness of all the schemes in generating a collision free path, but the EOBA based scheme generated a path with shorter path length and time to reach the destination. The developed scheme when compared with BA and PSO schemes obtained an average percentage reduction of 21.82% in terms of path length and 60% in terms of time taken to reach the target destination. The uniqueness of this approach is that the UGV avoid collision with obstacle at a distance of 0.3m from nearby obstacles as against taking three steps backwards before avoiding obstacle.

References

Abbas, N. H., & Ali, F. M. (2014). Path planning of an autonomous mobile robot using directed artificial bee colony algorithm. *International Journal of Computer Applications, 96*(11), 11-16.

Ajeil, F. H., Ibraheem, I. K., Sahib, M. A., & Humaidi, A. J. (2020). Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. *Applied Soft Computing, 89*, 106076.

Al-Mutib, K., AlSulaiman, M., Emaduddin, M., Ramdane, H., & Mattar, E. (2011). *D* lite based real-time multi-agent path planning in dynamic environments*. Paper presented at the 2011 Third International Conference on Computational Intelligence, Modelling & Simulation.

Annicchiarico, W., Periaux, J., Cerrolaza, M., & Winter, G. (2005). *Evolutionary algorithms and intelligent tools in engineering optimization*: Wit Pr/Computational Mechanics.

Anupama, J., Kavitha, A., Harsha, S., & Karthick, M. (2014). Design and development of autonomous ground vehicle for wild life monitoring 2 (5). In: May.

Audee, S. Y., Mu'azu, M. B., Man-Yahaya, S., Haruna, Z., Tijani, S. A., & Oyibo, P. (2019). Development of a Dynamic Cuckoo Search Algorithm. *Covenant Journal of Informatics and Communication Technology, 7*(2), 66-83.

Brand, M., & Yu, X.-H. (2013). *Autonomous robot path optimization using firefly algorithm.* Paper presented at the 2013 International Conference on Machine Learning and Cybernetics.

Chen, X., Zhou, M., Huang, J., & Luo, Z. (2017). *Global path planning using modified firefly algorithm.* Paper presented at the 2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS).

Civicioglu, P., & Besdok, E. (2013). A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review, 39*(4), 315-346.

Contreras-Cruz, M. A., Ayala-Ramirez, V., & Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing, 30*, 319-328.

Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., & Price, K. V. (1999). *New ideas in optimization*: McGraw-Hill Ltd., UK.

Deepak, B., & Parhi, D. (2013). Intelligent adaptive immune-based motion planner of a mobile robot in cluttered environment. *Intelligent Service Robotics, 6*(3), 155-162.

Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. (2014). Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering, 96*, 59-69.

Eslami, A., Asadi, S., Soleymani, G., & Azimirad, V. (2012). A real-time global optimal path planning for mobile robot in dynamic environment based on artificial immune approach. *GSTF Journal on Computing, 2*(1).

Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). *Metaheuristic applications in structures and infrastructures*: Newnes.

Geva-Sagiv, M., Las, L., Yovel, Y., & Ulanovsky, N. J. N. R. N. (2015). Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation. *16*(2), 94.

Gigras, Y., & Gupta, K. (2012). Ant colony based path planning algorithm for autonomous robotic vehicles. *International Journal of Artificial Intelliegence Applications, 3*(6), 31-38.

Gigras, Y., & Vasishth, O. (2015). *Comparison of BAT with PSO for Path Planning Problems.* Paper presented at the International Journal of Engineering Development and Research.

Haruna, Z., Mu'azu, M. B., Abubilal, K. A., & Tijani, S. A. (2017). *Development of a modified bat algorithm using elite opposition—Based learning.* Paper presented at the 2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON).

Hoffmann, J. (2000). *A heuristic for domain independent planning and its use in an enforced hill-climbing algorithm.*

Paper presented at the International Symposium on Methodologies for Intelligent Systems.

Horst, R., Pardalos, P. M., & Van Thoai, N. (2000). *Introduction to global optimization*: Springer Science & Business Media.

Hossain, M. A., & Ferdous, I. (2015). Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique. *Robotics and Autonomous Systems, 64*, 137-141.

Hosseinzadeh, A., & Izadkhah, H. (2010). Evolutionary approach for mobile robot path planning in complex environment. *International Journal of Computer Science Issues, 7*(4), 1-9.

Jabbarpour, M. R., Zarrabi, H., Jung, J. J., & Kim, P. (2017). A green ant-based method for path planning of unmanned ground vehicles. *IEEE access, 5*, 1820-1832.

Lamini, C., Benhlima, S., & Elbekri, A. (2018). Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Computer Science, 127*, 180-189.

Li, X., Huang, Y., Zhou, Y., & Zhu, X. (2018). *Robot Path Planning Using Improved Artificial Bee Colony Algorithm.* Paper presented at the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC).

Lv, T., & Feng, M. (2017). A smooth local path planning algorithm based on modified visibility graph. *Modern Physics Letters B, 31*(19-21), 1740091.

Misra, S. (2021). *A Step by Step Guide for Choosing Project Topics and Writing Research Papers in ICT Related Disciplines.* Paper presented at the Information and Communication Technology and Applications: Third International Conference, ICTA 2020, Minna, Nigeria, November 24–27, 2020, Revised Selected Papers 3.

Mohanty, P. K., & Parhi, D. R. (2013). *Cuckoo search algorithm for the mobile robot navigation.* Paper presented at the International Conference on Swarm, Evolutionary, and Memetic Computing.

Mohanty, P. K., & Parhi, D. R. (2016). Optimal path planning for a mobile robot using cuckoo search algorithm. *Journal of Experimental Theoretical Intelligence, 28*(1-2), 35-52.

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*: Pearson education.

Neydorf, R., Yarakhmedov, O., Polyakh, V., Chernogorov, I., & Vucinic, D. (2018). Robot path planning based on ant colony optimization algorithm for environments with obstacles. In *Improved Performance of Materials* (pp. 175-184): Springer.

Raja, P., & Pugazhenthi, S. (2012). Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences, 7*(9), 1314-1320.

Rostami, S. M. H., Sangaiah, A. K., Wang, J., & Liu, X. (2019). Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *EURASIP Journal on Wireless Communications Networking, 70*(1), 1-19.

Roy, N., Chattopadhay, R., Mukherjee, A., & Bhuiya, A. (2017). Implementation of

Image Processing and Reinforcement Learning in Path Planning of Mobile Robots. *International Journal of Engineering Science, 15211*.

Sahu, D., & Mishra, A. K. (2017). Mobile robot path planning by genetic algorithm with safety parameter. *International Journal of Engineering Science Computing, 7*(8), 14723-14727.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. *Advances in Engineering Software, 105*, 30-47.

Singh, N. H., & Thongam, K. (2018). Mobile robot navigation using fuzzy logic in static environments. *Procedia Computer Science, 125*, 11-17.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation* (Vol. 74): John Wiley & Sons.

Voigt, C. C., Frick, W. F., Holderied, M. W., Holland, R., Kerth, G., Mello, M. A., . . . Yovel, Y. (2017). Principles and patterns of bat movements: from aerodynamics to ecology. *The Quarterly Review of Biology, 92*(3), 267-287.

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74): Springer.

Zhou, Y., Wang, Y., Chen, X., Zhang, L., & Wu, K. (2017). A novel path planning algorithm based on plant growth mechanism. *Soft Computing, 21*(2), 435-445.