SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Infusing work-based experience through Project-Based Learning (POPBL) in Water Courses.

Norazlina Bateni[1,*], Azee Nurriha Juzi[1], Kuryati Kipli[2], Kasumawati Lias[2], Inawati Othman[1] Dayangku Salma Awang Ismail[1], Rosmina Ahmad Bustami[1], Nor Azalina Rosli[1]

[1] UNIMAS Water Center, Faculty of Engineering, Universiti Malaysia Sarawak, 94300, Sarawak

ARTICLE INFO

Article history:
Received 27 November 2026
Received in revised form 20 February 2026
Accepted 15 April 2026
Available online 4 May 2026

Keywords:
PSO, LiDAR, SLAM, robot, obstacles

ABSTRACT

This research aims to develop mobile robot-based navigation with PSO. The system will be based on ROS environment to identify the PSO performance in mobile robot navigation. Further analysis on the mobile robot navigation with PSO is discussed in the literature review section of this paper. The aim for this paper is for the mobile robot to solve the path planning problem of mobile. The basic robot's navigation of a robot requires many data or resources to process the output path of the navigation. To achieve this, LiDAR sensor is used to map the environment using SLAM functionality that is available on the mobile robot. A distance sensor will also be used for the mobile robot to avoid obstacles during the navigation process. However, it is found out that errors in terms of the sensor measurement may occur between the computer and real time performance. Thus, PSO is used to overcome this problem. To implement the PSO to the robot, basic research and review need to be done. All the results of this project had been collected in figures in the result section, including the differences before and after PSO implementation to the system. Further analysis and discussion were made on the system's performance.

1. Introduction

Mobile robots have been effectively employed in many fields due to their capacity to do tough jobs in hazardous environments, such as robot rescuing, space exploring, and their multiple prospective uses in our daily lives. Robot route planning is a fundamental topic in mobile robots is a kernel portion of mobile robot technology. Consequently, path planning has been paid great attention in modern. The goal of robot route planning is to create a collision-free path in a particular environment while meeting optimization criteria [1]. The particle swarm optimization approach was used to address this problem. In general, these approaches are divided into two groups depending on environmental aspects: off-line global path planning based on a model of known environments and on-line local path planning based on sensors sensing unknown surroundings.

* *Corresponding author.*
E-mail address: hamzah@ump.edu.my

https://doi.org/10.37934/araset.59.5.206218

A unique technique of local path planning based on PSO will be presented in this study. The optimization goal is defined in this technique based on the locations of the target and the barriers in the environment. Then PSO is applied to address the optimization problem. The sites of the globally best particle in each iteration are picked and obtained by the robot in succession throughout the PSO process. Also, this approach is based on the robot sensor system with limit range of the sensor. Following this, robot path planning processors need to update the information on paths once the robot sensor detects the environment's differences. By using the strategy, the robot can dynamically cope with changes in the unknown environment and achieve some global optimization impact by using goal and obstacle knowledge at the same time.

2. Literature Review

This study is related to the path planning method for mobile robot navigation focusing on ROS, robot navigation and control, the obstacle using LiDAR and path planning based on PSO. These elements will be combined to develop the PSO based mobile robot navigation.

2.1 Robot interface using ROS

Robot operating system (ROS) in an open-source software which one can acquire it freely by the developer without purchasing it. Gergely Magyar et.al in their research clarified that ROS is very compatible with variety of robots compared to RT-Middleware. Besides, it also has simulation environment functionality in contrast with Orocros which do not have that specific ability [2]. Meanwhile, Emmanouil Tsardoulias et al found that ROS uses C++, Phyton or LISP as its programming language and due to that feature, it has the ability to be integrated with another robotic platform [3]. These are commonly used programming language for most of engineering students and thus deemed very suitable for our application. Furthermore, ROS also capable of doing indoor mapping, localization, and navigation for autonomous robot as mentioned by Ruchik Mishra [4]. Entirely, these are the reasons why ROS become the appropriate robotic middleware that will be used in this research. Basically, ROS has two sides. On the operating system side, which provides standard operating system services such as:

i. Hardware abstraction
ii. Low-level device control
iii. Implementation of commonly used functionality
iv. Message-passing between processes
v. Package management

A collection of user-contributed packages that implement typical robot features including Gazebo 11, SLAM, planning, perception, vision, manipulation, and more.

a) ROS Nodes

Single- purposed executable programs. Such an example sensor drivers, actuator drivers, mapper, planner, UI and etc. Individually compiled, executed and managed. Nodes are written using ROS client library such as roscpp C++ library and rospy python client library. Nodes can publish or subscribe to a Topic. Nodes can also provide or use a Service.

b) ROS Master

Provides connection information to nodes so that they can transmit messages to each other. Every node connects to a master at start-up to register details of the message streams they publish and the streams to which that they to subscribe. When a new node appears, the master provides it with the information that it needs to form a direct peer-to-peer connection with other nodes publishing and subscribing to the same message topics.

c) ROS Packages

Software in ROS is organized in packages. A package contains one or more nodes and provides a ROS interface. Most of ROS packages are hosted in GitHub.

d) ROS Environment

ROS relies on the notion of combining spaces using the shell environment. This makes developing against different versions of ROS or against different sets of packages. After install ROS setup. *sh files in '/opt/ros/<distro>/' and also source it $source/opt/ros/indigo/setup.bash

2.2 Navigation and Control System of Mobile Robot Based on ROS

Mobile robot is an autonomous agent that can navigate intelligently using sensor-actuator control techniques. Thus, the control system and navigation are very important aspects that need to be concerned in autonomous mobile robots. Based on a conducted study, the control system based on STM32, and ROS contains hardware circuit designing, control software and upper computer software has been highlighted. According to this study, for lower computer, the velocity and current control of DC motors has been realized with some other tasks such as posture calculation, sensor data transfer is also complete in lower computer [5]-[8]. The upper computer platform is based on ROS (robot operating system), which is an open system for developers of robot [9]. SLAM (simultaneous localization and mapping) and autonomous navigation have been realized on ROS with LiDAR scan [10].

2.3 Path Planning Techniques for Mobile Robots

Apart from operating system and navigation and control system, the path planning techniques has become the essential element for the development of mobile robot in navigation. As been mentioned earlier, path planning is very important to determine a safe plan and collision-free path throughout the system. Generally, mobile robots have become increasingly popular in recent years, offering a wide range of applications in areas such as industry, agriculture, search, and rescue due to extreme of active research and development work on robotic and autonomous technology. However, there are challenges for a robot to navigate efficiently and reliably in an environment without or with less human assistance. The mobile robot should be capable of extracting the necessary information from the environment and taking the necessary action required to plan a feasible path for collision free motion to reach its goal [11]-[15]. The other study has been stated that the use of genetic algorithms is one of the alternative ways to help a controllable mobile robot to find an optimal path between a starting and ending point [16].

2.4 Particle Swarm Optimization (PSO) Algorithms for Mobile Robot System

In the past decade, the Particle Swarm Optimization (PSO) algorithm is a biologically inspired optimization method which has gained significant popularity [17]. Based on that study, researcher has been considered a version of the Particle Swarm Optimization (PSO) algorithm which is appropriate for search tasks of multi-agent systems consisting of small robots with limited sensing capability. The outcome for simulation and implementation shows that the algorithm performs well in a sense that the robots can move towards and aggregate in areas with high gas concentration around the maximum points of the gas concentration profile representing the environment. Besides, there have a study regarding a multi-robot search application involving one or more target. This study was focused on finding optimal parameters for PSO algorithm and the use of 2-level hierarchy which are PSO in inner level and outer level. Specifically, the PSO in inner level solves the finding locations of target problems while PSO in outer level determines the optimal set of parameters for inner level PSO.

2.5 Path Planning for Mobile Robots in Dynamic Environments

There are many challenges for robot navigation in densely populated dynamic environments. By referring to a study, PSO is able to swiftly determine optimal solution for mobile robot path planning problems in dynamic environments. The generated valid paths are depending on PSO algorithm to get the optimal paths as shown in several reports [18]-[20]. The simulation is able to demonstrate the effectiveness and efficiency of the algorithm.

An analysis on PSO on the motion planning and control for a robot in dynamic environment have been demonstrated in various reports as mentioned above. The effectiveness of the proposed controller and collision avoidance strategy is evaluated through a series simulation where the result show that the proposed strategy is efficient in stabilizing the robot in the desired configuration and in avoiding collision with obstacles, even in narrow spaces and with complicated arrangements of obstacles [19].

Besides, there have a survey done on the path planning methods for robot navigation in dense environments [20]. Specifically, the path planning in the navigation framework of mobile robots is composed of global path planning and local path planning, regarding the planning scope and the executability. Notably, the recently developed Velocity Obstacle method and its variants that serve as the local planner are analysed comprehensively.

2.6 Obstacles Avoidance Using LiDAR Sensor

LiDAR is an optical scanning technology that measures the properties of radiated light to find distance and other information from target. Shooting pulses of laser light onto the object's surface is one method for determining the distance of an object from the LiDAR sensor. LiDAR sensor is normally applied to detect and avoid obstacles around it. The LiDAR sensor is mounted on top of an autonomous robot. A Raspberry Pi 3 is also one of the controllers used as a data collector from LiDAR and as a motor controller. Generally, for analysis purposes, the mobile robot is placed in an indoor room with multiple obstacles along its path. Figure 1 below shows the resulted minimum and maximum distance measurements of the LiDAR are 0.12 m and 10.5 m, respectively, with an average error of 0.9%. Figure 1 below shows the robot can successfully navigate and avoid obstacles that lies in its path. However, the robot also has a drawback which is it cannot avoid transparent objects [10].

Distance (m)	Read Distance (m)	Error (%)
0 – 0.11	0	-
0.12	0.121	0.83
0.16	0.161	0.83
0.20	0.201	0.83
0.24	0.241	0.83
0.5	0.505	1
1.5	1.495	0.33
2.5	2.509	0.36
3.5	3.517	0.48
4.5	4.531	0.68
5.5	5.525	0.45
6.5	6.561	0.93
7.5	7.606	1.41
8.5	8.600	1.17
9.5	9.679	1.88
10.5	10.662	1.54
11.5	0	-
12.5	0	-
Average		**0.9**

Fig. 1. The lidar measurements [10]

3. Methodology

3.1 Mobile Robot Navigation

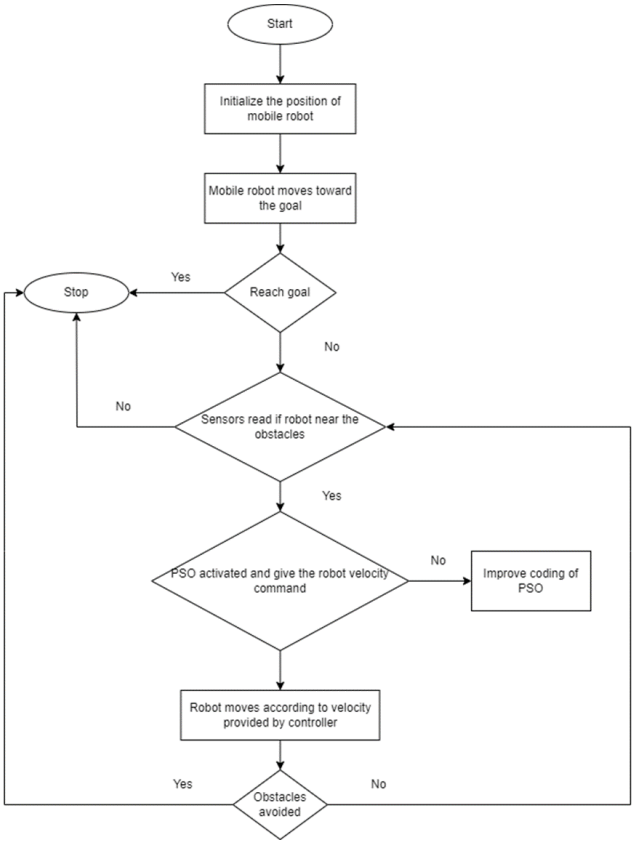

Fig.2. Mobile robot navigation flowchart

The flow chart of mobile robot navigation is shown in Figure 2 above. First, the mobile robot's location is defined in global coordinate system. The mobile robot is then advances toward the specified target as defined by the path planning algorithm. If the mobile robot reaches the target, it will stop; otherwise, the software will check all sensor values attached to the robot. At that time, the obstacle sensor's distance value will be determined accordingly. The system will shut off if there are no sensors reading obtained during observations. To perform the PSO actions, the application obtains excessive information from sensors during its motion. If the PSO operation is not performed, then the PSO should be further tuned to achieve the objective to achieve better path planning performance. The robot will only turn to avoid obstacles based on the results of the PSO performance. When the turning process is finished, the sensor recognizes or checks for the consequent input.

The robot will then move in accordance with the velocities specified by PSO. The velocities is defined to be not too fast and depends on the system settings as well as sensors limitations. The mobile robot keeps the avoidance process continues until there is no longer an obstacle detected when it moves in the environment. If an obstacle appears again, the avoiding procedure continues until there are no obstacles. If no obstacles are detected, then the mobile robot proceeds straight and moves in a specific speed. Following each movement, the sensor is observed once again before proceeding to the next movement phase. The sensor reading demonstrates the PSO controller performance which is linked directly to the system to determine the obstacle distance and continuously presents obstacle difference value.

3.2 Implementation of Particle Swarm Optimization for path planning

i. Algorithm content

A designed route is accomplished by identifying the path points on these circles, connecting the origin and path points, and finally connecting with the target point from the initial location to the target point.

path = $[\theta_s, \rho_s;$ route $(\theta,\rho);\theta_g, \rho_g]$ (1)

where θ_s, ρ_s, θ_g, ρ_g are the beginning and goal polar angles and radiuses, correspondingly, and route (θ,ρ) is the polar angle and radius of the path point on all route point circles.

ii. Establishment of environmental model

The beginning point serves as the origin of a polar coordinate system. Polar radius and polar angle specify the placements of all obstacles.

obsp = $[\theta_1, \rho_1;\theta_2, \rho_2; \ldots;\theta_n, \rho_n]$ (2)

where **obsp** is the position vector of obstacle, θ is the polar angle of obstacle, and ρ is the polar radius of obstacle.

To regard the mobile robot as a mass point, the influence area of the obstacle is expanded into a circle. Its radius is equal to the sum of the radius of the obstacle, the safe distance and the double radius of the mobile robot.

$R_{efr} = R_{obs} + R_{safe} + 2(R_{rob})$ (3)

$\rho^2 + \rho_i^2 + 2\rho\rho_i \cos(\theta-\theta_i) = R_{efr\,i}^{\,2}$ (4)

where R_{obs} denotes the radius of the obstacle, safe R the radius of the safe distance, and R_{rob} the radius of the mobile robot. ρ and θ are the polar radius and polar angle of the obstacle-affected point on the circle.

iii. Construction of fitness function

To assess the route evaluated by PSO, a fitness function must be built to determine if the current path is better than the previous one and whether the current direction is the global optimum path. It will also serve as a criterion for particles to update the local and global extremes. In the mobile robot path planning challenge, mobile robots must avoid obstacles. As a result, the fitness function may be written as:

$$f_{fit} = \frac{f(d)}{f(d')+g(p)} \quad (5)$$

Where f_{fit} represents the fitness value, f(d) represents the Euclidean distance from the starting point to the destination point, f(d') represents the length of the path sought by PSO, and g(p) represents the violation value of the path found by PSO if it allocates or crosses barriers.

iv. Mutation operation

The PSO method is used to find the path in the environmental model.

$$w = wmax - iter.\frac{wmax-wmin}{itermax} \quad (6)$$

Where w max and w min are the maximum and minimum inertia weights, and max iter iter are the current and maximum number of iterations. The difficulty of the optimization issue is low when there are fewer impediments. The PSO algorithm works nicely. The path found after 350 iterations is either the ideal path or extremely close to it. However, when the number of obstacles rises, so does the intricacy of the optimization issue, and the path sought by 350 iterations is quite bad.

Particles become caught in local extremums and find it more difficult to escape. As a result, at a subsequent step of the process, the mutation operation is performed at random on one dimension of the particle. The mutation procedure is as follows.

$$xid = bn + (bm - bn).rand \quad (7)$$

$$if\ rand > pthres \quad (8)$$

Where rand is a random number obeys normal distribution, bm, bn are the upper and lower bounds of the position of particles in each dimension, and *pthres* is the threshold of the probability of mutation.

4. Results and Discussion

This section identifies and analyzes the performance of the mobile robot using PSO in MATLAB environment. The mobile robot will be observed on its behavior during its motion when it moves to the designated point. The research findings and the problems encountered during the research development process are discussed in this section.

The mobile robot simulation is run using MATLAB, ROS, and Gazebo software in sequence. To verify that the robot can fully operate indoors, the functioning of the robot hardware is evaluated in an indoor setting.

The research demonstrates simulation results for the PSO application. Next subsection defines the performance of the proposed technique.

a. MATLAB

Path planning lets a robot find the shortest and most obstacle-free part from a start to reach the goal state. Path planning requires a map of the environment that can be represented in different ways. Thus, MATLAB acts as important tools for mobile robot path planning.

Based on Figure 3, MATLAB depicts the best and shortest path with PSO when the simulation performs later. Next, maximum number of planner iterations for exploring the cost-map, is specified as a positive integer. By increasing this value, it will then increase the number of samples for finding a valid path for the mobile robot. Remark that, if a valid path is not identified, the path planner will exit after exceeding this maximum value as shown in Figure.4.

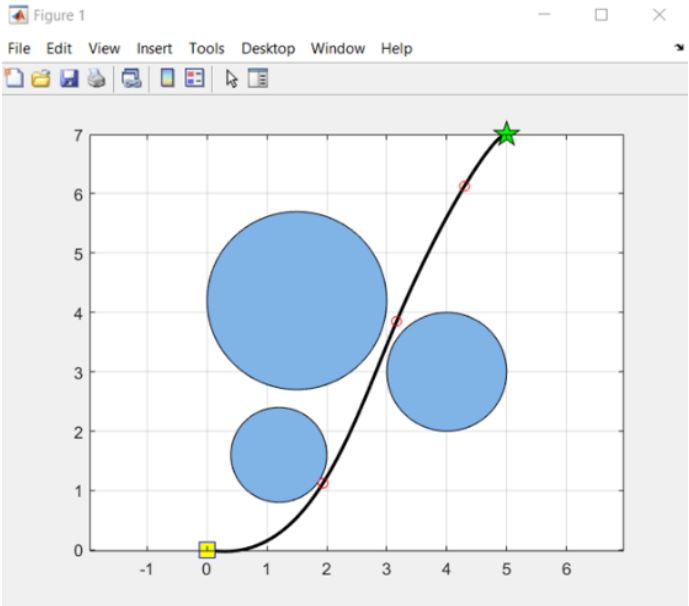

Fig.3. The best and shortest path with PSO

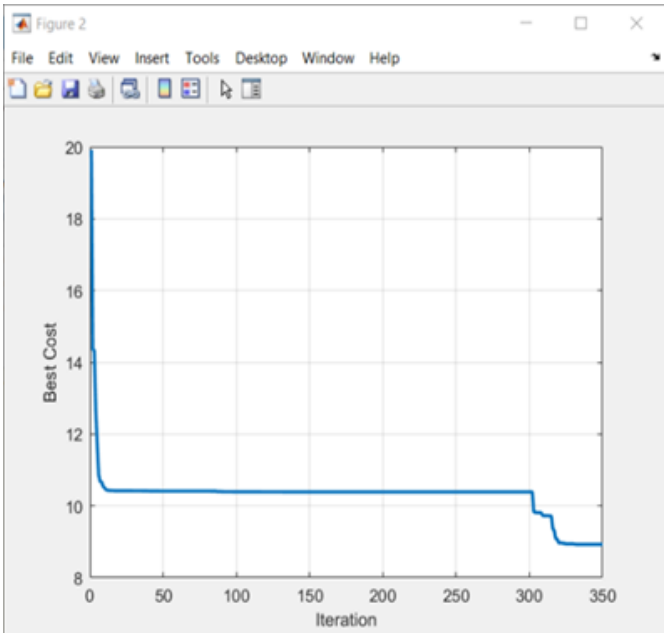

Fig.4. The best and shortest path with PSO

During the design error detection analysis, this algorithm checks the specified minimum and maximum values on intermediate signals throughout the model on the output ports. This process is important to understand the characteristics of the error based on the designed environment.

b. ROS simulation

For the simulation in ROS, it is assumed that a Turtlebot moves inside the environment. Figure 5 depicts the mobile robot attempts to avoid obstacles in a simulation. There were four obstacles which are designed in different shape and size to determine the sensor and PSO efficiency in avoiding the obstacle. The objects are placed randomly in the indoor environment in a planar world. As shown in the figure, the mobile robot detects and has successfully avoided the obstacle 1 at location X1.

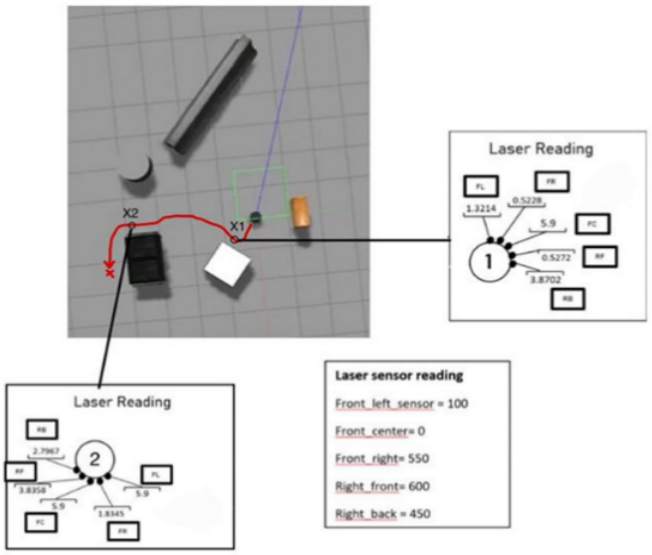

Fig.5. ROS simulation with PSO

Because the value reading sensor of the front left sensor (FL) and front right sensor (FR) is smaller than the sensor of the right front sensor, the Turtlebot will move to the left. As it will drive straight, the front center sensor (FC) reading is 5.9m by default. Because of the greater reading, the right back sensor (RB) will move to the right. The Turtlebot will avoiding barriers 3 at position X2. The Turtlebot then drive straight because the value of the front left sensor (FL) is high, and to the left because the front right sensor reading (FR) is low, to avoid obstacles on the right, and then turn back to the right owing to the front centre sensor reading (FR) (FC).

The distance between the sensor and the barriers is shown by the laser sensor readout illustrated in the figure. Front center is defined as 0 to ensure a Turtlebot to move straight, front left sensor set to 100 as it moderate or near to the obstacles. Front right and front right are set as gaps with 50 values to identify any object at a long distance. Furthermore, the right back with a value of 450 detects obstructions in a modest range of obstacles.

Figure 6 shows when the Turtlebot attempts to avoid three obstacles during its movements. The lower the value of the turtlebot's laser sensor, then the more the turtlebot intends to avoid obstacles. The greater the laser sensor value, the less the turtlebot avoids obstacles. The turtlebot's speed for the T1 is 0.625, while the speed for the turn forward to avoid obstacles is -1.5. The turtlebot's speed for the T2 is 0.625, while the speed for the turn forward to avoid obstacles is 1.5.

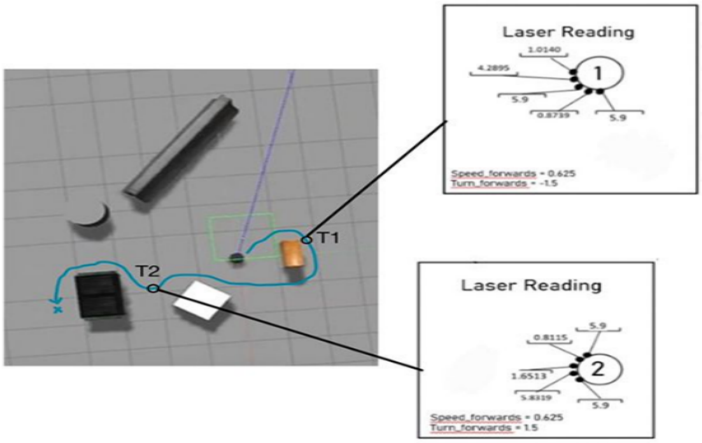

Fig.6. ROS simulation without PSO

Different mobile robot motions are then being investigated to assess the performance of mobile robot as illustrated in Figure 7 above.

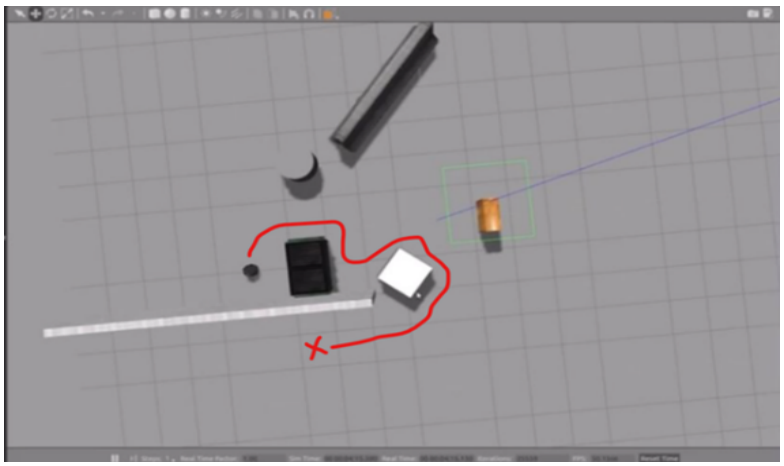

Fig.7. Additional obstacle with PSO

In this situation, the additional barrier has been placed in the environment in order to see either Turtlebot can identify that barrier. Remark that barrier was not presented during the mapping process. After that, the navigation objective has been set to the X mark. The TurtleBot then proceeds to the destination location. It then detects, and then move apart from the new barrier and then avoids two obstacles during its motions. Due to the PSO application, the Turtlebot can take the shortest path to the destination location.

To understand the performance of the Turtlebot when PSO is not applied, similar settings are defined in the same environment. Figure 8 presents the outcomes where the mobile robot is now moves and arrived at the target but with longer distance. This consequently shows that PSO has offer a better result than without the controller.

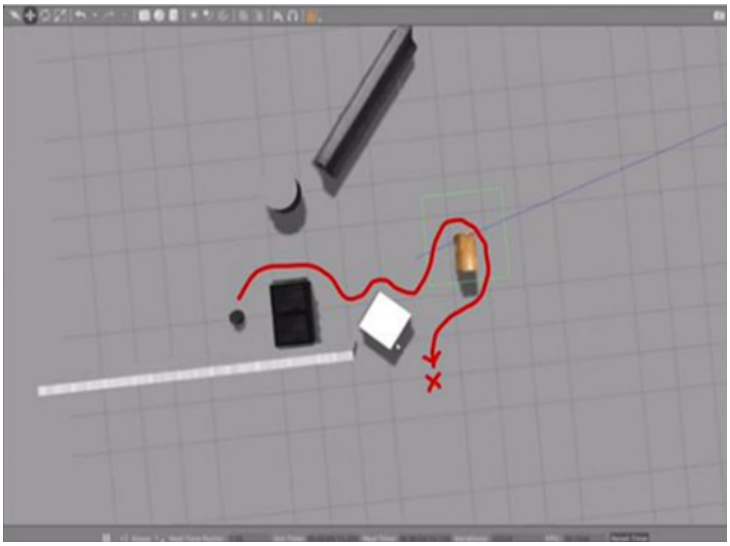

Fig.8. Additional obstacle without PSO

5. Conclusion

As a conclusion, this research highlights the mobile robot navigation to build the environment model and develops the fitness function of PSO. The analysis employs PSO to plan the course of mobile robots. During the investigation, a mutation operation is presented that retains particle search performance while increasing particle variety, allowing particles to emerge from local extremes. To improve the path, any redundant path points from the path found by PSO is deleted. Simulation analyses are used to demonstrate the reliability of the given strategy. Based on the findings, the path chosen by PSO is not necessarily the best option. The PSO will be improved more in the future so that the algorithm can identify a best path with less iterations. In conclusion, the mobile robot with ROS navigation system has been developed with PSO to avoid unknown or unexpected obstacles in an unknown environment. The developed technique is proven to successfully navigate the shortest path from starting point to goal point and avoid the obstacles with particle swarm optimization method.

Acknowledgment

This research is sponsored by MOHE FRGS FRGS/1/2022/TK07/UMP/02/20 and UMP RDU220113 grant.

References

[1] Adzhar, Noraziah, Yuhani Yusof, and Muhammad Azrin Ahmad. 2020. "A Review on Autonomous Mobile Robot Path Planning Algorithms." *Advances in Science, Technology and Engineering Systems Journal* 5 (3): 236–40. https://doi.org/10.25046/aj050330.

[2] SinčákPeter, Pitoyo Hartono, VirčíkováMária, VaščákJán., and JakšaRudolf. 2015. *Emergent Trends in Robotics and Intelligent Systems : Where Is the Role of Intelligent Technologies in the next Generation of Robots?* Cham: Springer International Publishing : Imprint : Springer. https://doi.org/10.1007/978-3-319-10783-7

[3] Tsardoulias, Emmanouil, and Pericles Mitkas. 2017. "Robotic Frameworks, Architectures and Middleware Comparison." *ArXiv (Cornell University)*, November. https://doi.org/10.48550/arxiv.1711.06842.

[4] Mishra, Ruchik, and Arshad Javed. 2018. "ROS Based Service Robot Platform." IEEE Xplore. April 1, 2018. https://doi.org/10.1109/ICCAR.2018.8384644.

[5] Gupta, Meenakshi, Balaji Uggirala, and Laxmidhar Behera. 2008. "Visual Navigation of a Mobile Robot in a Cluttered Environment." *IFAC Proceedings Volumes* 41 (2): 14816–21. https://doi.org/10.3182/20080706-5-kr-1001.02508.

[6] Zhi, Li, and Mei Xuesong. 2018. "Navigation and Control System of Mobile Robot Based on ROS." IEEE Xplore. October 1, 2018. https://doi.org/10.1109/IAEAC.2018.8577901.

[7] Yosif, Zead, Basil Mahmood, and Saad Al-khayyt. 2021. "Assessment and Review of the Reactive Mobile Robot Navigation." *Al-Rafidain Engineering Journal (AREJ)* 26 (2): 340–55. https://doi.org/10.33899/rengj.2021.129484.1082.

[8] Chen, Y, C Li, Y Zhou, F Che, and H Li. 2021. "RESEARCH on SLAM TECHNOLOGY of ROBOT BASED on ROS," January. https://doi.org/10.1049/icp.2021.1318.

[9] Ibragimov, Ilmir Z., and Ilya M. Afanasyev. 2017. "Comparison of ROS-Based Visual SLAM Methods in Homogeneous Indoor Environment." IEEE Xplore. October 1, 2017. https://doi.org/10.1109/WPNC.2017.8250081.

[10] Hutabarat, Dony, Muhammad Rivai, Djoko Purwanto, and Harjuno Hutomo. 2019. "Lidar-Based Obstacle Avoidance for the Autonomous Mobile Robot." IEEE Xplore. July 1, 2019. https://doi.org/10.1109/ICTS.2019.8850952.

[11] Lu, Li, and Dunwei Gong. 2008. "Robot Path Planning in Unknown Environments Using Particle Swarm Optimization." IEEE Xplore. October 1, 2008. https://doi.org/10.1109/ICNC.2008.923.

[12] Nurmaini, Siti, and Bambang Tutuko. 2017. "Intelligent Robotics Navigation System: Problems, Methods, and Algorithm." *International Journal of Electrical and Computer Engineering (IJECE)* 7 (6): 3711. https://doi.org/10.11591/ijece.v7i6.pp3711-3726.

[13] Sánchez-Ibáñez, José Ricardo, Carlos J. Pérez-del-Pulgar, and Alfonso García-Cerezo. 2021. "Path Planning for Autonomous Mobile Robots: A Review." *Sensors* 21 (23): 7898. https://doi.org/10.3390/s21237898.

[14] Azizi, Mahmood Reza, Alireza Rastegarpanah, and Rustam Stolkin. 2021. "Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments." *Robotics* 10 (1): 48. https://doi.org/10.3390/robotics10010048.

[15] Wang, Yang, I.P.W. Sillitoe, and D.J Mulvaney. 2007. "Mobile Robot Path Planning in Dynamic Environments." *Proceedings*, April. https://doi.org/10.1109/robot.2007.363767.

[16] AL-Taharwa. 2008. "A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment." *Journal of Computer Science* 4 (4): 341–44. https://doi.org/10.3844/jcssp.2008.341.344.

[17] Salih Burak Akat, Veysel Gazi, and Lino Marques. 2010. "Asynchronous Particle Swarm Optimization-Based Search with a Multi-Robot System: Simulation and Implementation on a Real Robotic System." *Turkish Journal of Electrical Engineering and Computer Sciences*, January. https://doi.org/10.3906/elk-0906-7.

[18] Fei, Wang, Wang Ziwei, and Lin Meijin. 2021. "Robot Path Planning Based on Improved Particle Swarm Optimization," March. https://doi.org/10.1109/icbaie52039.2021.9390071.

[19] Doctor, S, Ganesh K Venayagamoorthy, and Veera Babu Gudise. 2004. "Optimal PSO for Collective Robotic Search Applications," June. https://doi.org/10.1109/cec.2004.1331059.

[20] Raja, P, and S. Pugazhenthi. 2009. "Path Planning for Mobile Robots in Dynamic Environments Using Particle Swarm Optimization," January. https://doi.org/10.1109/artcom.2009.24.