IOP Conf. Series: Materials Science and Engineering **732** (2020) 012098 doi:10.1088/1757-899X/732/1/012098

Development of path planning of line follower robot with obstacles avoidance based on particle swarm optimization

R Mardiyanto*, A Suhartono and R F Siregar

Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

*ronny@elect-eng.its.ac.id

Abstract. Research on robot path planning based on the PSO (Particle Swarm Optimization) method has been widely developed. PSO is one of optimization algorithm that aims to provide the best solution on the finding of shortest route by utilizing particle population movements. In this paper, we developed a line follower robot with obstacle avoidance based on PSO. The PSO is embedded into Arduino microcontroller to control the navigation as well as avoid the obstacles. The robot is programmed a start point on coordinate (1,1) and a destination point on coordinate (5,5). The robot performances to avoid the obstacles were tested by robot goes to destination point while obstacles were added to the route. The computation time and the shortest route distance of the robot to reach the destination point with and without obstacles have been compared.

1. Introduction

Robot path planning is an essential element in the robotic system. Robot path planning aims to provide the ability and power of the robot to perform motion in accordance with the provisions that have been made. So, the robot can move automatically from the initial point to the goal. Classical robot path planning techniques consist of several problems, i.e. a lot of cost and high computational time. The Heuristic techniques for solving the robot motion planning are attracted the interest of many researchers because simple implementation and fast generation of searches to achieve the desired solution. PSO is a very simple heuristic optimization technique, but it is very powerful and effective in many complex optimization problems. Many advantages of PSO compared to another heuristic technique are high speed of convergence, simplicity, and the parameters are easy to set [1].

In this paper, we propose an approach using the PSO method to plan the shortest route that the robot can pass to reach its destination. PSO method is a robot path planning technique that is good in a dynamic environment to avoid the presence or absence of obstacles. The test aims to compare the computational time of the robot to solve the route with existing and the absence of obstacles and the shortest route that the robot can pass from the starting point to the destination point.

2. Related works

Yoney Kirsal Ever developed path planning for cellular robots using SSO. The SSO algorithm is used to find ideal ways for cellular robots in the workplace with irregular complications. The recommended method is an advanced robot that is successfully managed from the starting point to the end of a complex path that is complete and shortest ways to reach the destination point without hit the obstacles. The error element from SSO has been approved to ensure PSO unites together. This error is used in overcoming the particle route towards a target. The customized rule is applied in SSO to overcome the problem of convenient paths. Conditional statements are used to move from the difficult passed route to another possible route. The SSO Computing can choose the key points to passed from start to destination point without pay attention to useless points. As the result, researcher legitimate SSO as a better than average algorithms to join speed and be efficient in global searches [2].

Zhou and Wang verified the effectiveness of the track planning algorithm, they physically verified the MT-R research intelligent robot platform. The diameter of the MT-R robot is 50 Cm, the maximum speed is 2.5 m / s, and the turning radius is zero. The planning path is converted to the instruction set and set control instructions are sent individually to the robot driver so they can control the robot autonomously. The indoor experimental environment consists of two static barriers, there are starting and target point which symbolised as S(x) and T(x). The optimal algorithm for planning the PSO elastic heuristic pathway was proposed where an A * efficient algorithm was used to collect heuristic information for particle initialization. Using an enhanced PSO elastic algorithm, the initial heuristic path is repeated, and particles assembled along with the optimal path use shrink operations in the elastic strategy. Particle degradation occurs when particles shrink to the threshold of reflection. This phenomenon is overcome by rebound surgery, thus avoiding local extremes between particles. Robot simulations and experiments show that the algorithm here can plan the shortest path efficiently and achieve good performance in solving the two main problems of path planning in real-time and low stability. Path planning can be further optimized for the operation of cellular robots that are safe and smooth, according to the characteristics of their movements [3].

Qiang and Gao carried out complex testing, the PSO-SP algorithm and the PSO algorithm in Literature were tested. The particle coding dimension of the PSO-SP algorithm is specified as 3. The number of interpolation points is set as 50. The scale of the N particle crowd is set as 80, the inertia factor and learning factor are set as: w = [0.9, 0.4], c1 = 1.5, c2 = 1.5. The maximum particle speed is limited to Vmax = 6. Particle coding of the PSO dimension is set as 16, the number of spline curves is identical to the PSO-SP algorithm, but the coding dimension of the PSO-SP algorithm is lower. The 3 best routes produced by PSO-SP and PSO for 30 operations each. The path is acquainted with two very fine algorithms, and this is characteristic of the cubic spline interpolation curve. The PSO-SP found the global path of the optimal path length 141,7652, while the PSO only found 3 local optimal paths with short distances. Table 2 shows the results of the PSO and PSO-SP statistics for 30 operations. The PSO-SP has superior performance in terms of speed of convergence, property convergence and stability [4].

Ayari and Bouamama used two method approaches to complete path planning in their research. The method they use is PSO and D2PSO. Several scenarios have been made according to previous research, they tried to simulated the programs in 200 iterations with 3, 5, 7 and 10 robots and different start points and different destination points. The algorithm's effectiveness had evaluated with the comparation between the path lengths which obtained from the basic distributed PSO and also from D2PSO, and ensuring there is non-collision with the static obstacles which defined in advance. D2PSO used to ensure the robots can move to better and unexplored space without disturbed the speed of convergence and keeping the fundamental of PSO. Researchers get that approaches using PSO and D2PSO can make the robot can be escaping from local optimum with better result and proving that using D2PSO in large number of robots with path planning problem is practical and quite efficient [5].

Tang, Zhu, and Luo used HNTVPSORBSADE the hybrid PSO algorithm, combined of NTVPSO and RBSADE, used to complete the global mobile robot's path planning problems. At the beginning, the rule that defined in NTVPSO is obeyed by all particles to change the speed and position. In the next step, to keep off from the stagnant of the particles, the best position of the particles must be updated by developing the RBSADE. To take balance between NTVPSO's exploitation and exploration capabilities, three kind of control parameters must be updated to better proportion. As a result, NTVPSO push the particle to get the better routes. Improving performance and adjusting RBSADE control parameters easily, operators to perform rank-based mutation, and independent adaptation strategies were developed at RBSADE. Because HNTVPSORBSADE convergence is still needed, the researcher analysing the convergence of HNTVPSO-RBSADE from the theory of dynamic systems. Finally, the principle of parameter selection is used to get the convergence of HNTVPSORBSADE. Four numerical simulations are used to testing the proposed algorithm and the Monte-Carlo experiment is very different with four evolutionary algorithms. From the result of simulation, revealed that the proposed algorithm better then another four algorithms for path optimisation; For future works, researcher can compared between proposed algorithm's calculation time with another algorithm's calculation time [6].

In this paper, we proposed a line tracer robot with dynamic obstacles avoidance. PSO is used to optimize the routes and also avoid the dynamic obstacles. Such robot can be operated as a service robot with ability to carry food as well as drink to customer.

3. Research method

The developed robot uses PSO as the main algorithm to find the best route as well as to avoid the obstacles. The robot route is modelled in the cartesian coordinate system with the coordinate starting point (0, 0) and destination point (5, 5). Robot route planning testing is carried out with obstacles and without obstacles.

3.1. PSO (Particle Swarm Optimization)

PSO is an optimization algorithm using random number or usually called stochastic method. Since it was first discovered, the PSO commonly used extensively to handle the problems about optimization. PSO using stochastic method according to the behaviour of individual movements in flocks, such as flocks of insects, ants, termites, bees or birds. In PSO, the herd have specific size with particle of its starting position with a random multidimensional space. Every particle moves in a certain space and remember the best position that has been found. Every particle gives its value to the others and then adjusts the value of position and also the value of speed based on the best position's information [7].

Although every bird has limitations in terms of intelligence, then as a solution they will follow the rule as follows [7]:

- bird is not too close to another bird.
- The bird will direct its flight towards the overall average of the bird.
- It will position itself with the average position of the other birds by keeping the distance between the birds in the herd not too far away.

There are 3 factors that can reflect the behaviour of bird flocks:

- Cohesion, means fly together in flocks.
- Separation, means each bird don't get too close.
- Alignment, means all birds follow the directions together.

PSO mathematical model can be seen in equation below:

$$Vi^{(t+1)} = \omega.Vi^{t}c1.rand1(.).(Pbesti - Xi^{t}) + c2.rand2(.).(Gbest - Xi^{t}) \quad (1)$$

$$Xi^{(t+1)} = Xi^{t} + Vi^{(t+1)}$$

$$i = 1, 2, 3, \dots Nswarm \quad (2)$$

Description: i: Index of each particle, t: Number of iterations or repetitions, rand1 (.) and rand2 (.): Random value between 0 and 1, Pbesti: Best experience of particle i recorded, Gbest: The best particles among whole population, Nswarm: Number of herds, constants c1 and c2: Weight factor of the term stochastic acceleration that attracts each particle to the position of Pbesti and Gbest, tmax: Maximum number of iterations or repetitions, ω: Weight of inertia, axmax: Maximum inertial weight, ωmin: Minimum inertia weight, K: Number of variables.

IOP Conf. Series: Materials Science and Engineering **732** (2020) 012098 doi:10.1088/1757-899X/732/1/012098

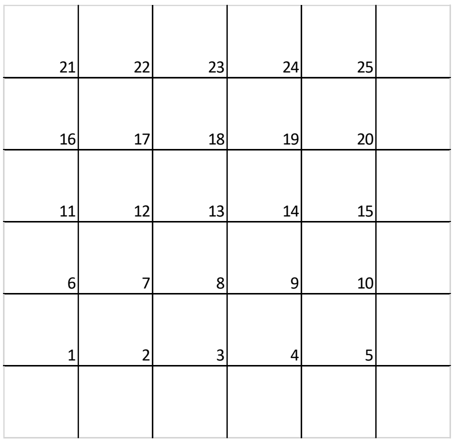

Figure 1. Navigation coordinate.

3.2. Hardware configuration

The hardware configuration consists of a line tracer robot with an Arduino Uno microcontroller with 6 IR sensors (transmitter) and 6 photodiodes (receivers), ultrasonic sensors, and DC motors. The speed of the robot is 17,54 Cm/s in straight route and 330,84 deg/s for rotate.

We give numbers to each coordinate to make it easier to mention coordinates when recording and analyzing. Coordinates (1,1) are given number 1, coordinates (1,2) are given number 2, coordinates (2,1) are given numbers 6, and so on. the distance of point 1 with point 2 is 13 Cm, and the distance between points 1 and 6 is 10 Cm. The navigation coordinate is shown in Figure 1.

4. Results and discussion

In this research, we applied the PSO algorithm to planning the robot path. We carry out two test conditions with and without obstacles. Our robot path is based on cartesian coordinates with a starting point (1, 1) and destination point (5, 5).

On robot testing without obstacles, we conducted 9 experiments with parameters taken, among others: processing time, when the robot took the route from the initial point to the goal, sum of time needed to robot do the computation and the route point that the robot passed. The following table 1 provide the data of testing without obstacles. Result of testing without obstacles is shown in Table 1.

Table 1. Results of testing without obstacles.

Experiments	Processing Time (s)	Time from Starting Point to Destination Point (s)	Total Time (s)	Passed Route Points
1	2,793	7,585	10,378	1,6,11,16,17,18,19,24,25
2	1,084	6,288	7,372	1,6,11,16,21,22,23,24,25
3	0,812	8,455	9,267	1,6,11,12,17,18,23,24,25
4	0,641	6,558	7,199	1,6,11,16,21,22,23,24,25
5	0,965	7,526	8,491	1,6,11,16,17,18,19,24,25
6	0,687	7,483	8,17	1,2,7,12,17,22,23,24,25
7	0,834	7,447	8,281	1,6,11,16,17,18,19,24,25
8	1,263	8,465	9,728	1,6,11,12,17,18,23,24,25
9	2,94	7,482	10,422	1,6,11,16,17,18,19,24,25

The table 1 shows there are 4 variations of 9 trials, we could say that the percentage of the randomize is 44,44%. The fastest processing time is 0,641 and the latest process time is 2,94 with the average time is 1,335. The fastest-moving time is 6,288 and the latest moving time is 8,465 with the average time is 7,477. The fastest total time (sum of processing time and movement time) is 7,199 and the latest total time is 10,422 with the average time is 8,812, there are 4 trials that have total time which is above the average of the total time.

IOP Conf. Series: Materials Science and Engineering **732** (2020) 012098 doi:10.1088/1757-899X/732/1/012098

In testing robots with 1 obstacle on the robot path, we conducted 5 experiments with additional parameters to position the obstacles on the robot route path. The Table 2 provide the data of testing with 1 obstacle.

Table 2. Results of testing with 1 obstacle.

Experiments	Processing Time (s)	Time from Starting Point to Destination Point (s)	Total Time (s)	Passed Route Points	Experiments
1	3,564	8,629	12,193	between 6 and 11	1,6,7,12,13,18,23,24,25
2	2,569	8,858	11,427	between 6 and 11	1,6,7,12,13,14,19,24,25
3	3,975	7,687	11,662	between 6 and 11	1,6,7,8,13,18,23,24,25
4	4,014	7,396	11,41	between 6 and 11	1,6,7,12,17,22,23,24,25
5	4,059	8,635	12,694	between 6 and 11	1,6,7,8,13,18,19,24,25

In testing robots with 2 obstacles on the robot path, we conducted 5 experiments. The table 3 provide the data of testing with 2 obstacles.

Table 3. Results of testing with 2 obstacles.

Experiment	Processing Time (s)	Time from Starting Point to Destination Point (s)	Total Time (s)	Position of Obstacles	Passed Route Points
1	7,982	6,835	14,817	Between 6 and 11, Between 7 and 8	1,6,7,12,13,14,15,20,25
2	2,996	10,307	13,303	Between 6 and 11, Between 7 and 8	1,6,7,12,17,22,23,24,25
3	3,71	7,46	11,17	Between 6 and 11, Between 7 and 8	1,6,7,12,17,22,23,24,25
4	5,605	8,606	14,211	Between 6 and 11, Between 7 and 8	1,6,7,12,13,14,19,24,25
5	5,358	8,048	13,406	Between 6 and 11, Between 7 and 8	1,6,7,12,13,14,15,20,25

In testing robots with 3 obstacles on the robot path, we conducted 5 experiments. The table 4 shows the results of testing with 3 obstacles.

Table 4. Results of testing with 3 obstacles.

Experiment	Processing Time (s)	Time from Starting Point to Destination Point (s)	Total Time (s)	Position of Obstacles	Passed Route Points
1	3,665	8,863	12,528	Between 1 and 6, Between 2 and 3, Between 7 and 12	1,2,7,8,13,18,23,24,25
2	3,238	8,951	12,189	Between 1 and 6, Between 2 and 3, Between 7 and 12	1,2,7,8,13,18,19,20,25
3	2,303	8,794	11,097	Between 1 and 6, Between 2 and 3, Between 7 and 12	1,2,7,8,13,18,23,24,25
4	2,461	8,491	10,952	Between 1 and 6, Between 7 and 12, Between 8 and 9	1,2,7,8,13,18,23,24,25
5	2,124	8,82	10,944	Between 1 and 6, Between 7 and 12, Between 8 and 9	1,2,7,8,13,18,23,24,25

In testing robots with 4 obstacles on the robot path, we conducted 5 experiments. The table 5 provide the data of testing with 4 obstacles.

Table 5. Results of testing with 4 obstacles.

Experiment	Processing Time (s)	Time from Starting Point to Destination Point (s)	Total Time (s)	Position of Obstacles	Passed Route Points
1	3,162	9,397	12,559	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 9 and 14	1,2,7,8,9,10,15,20,25
2	2,946	9,762	12,708	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 9 and 14	1,2,7,8,13,18,19,24,25
3	3,375	9,928	13,303	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 9 and 14	1,2,7,8,13,18,19,24,25
4	3,732	8,224	11,956	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 9 and 14	1,2,7,8,9,10,15,20,25
5	3,207	9,221	12,428	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 9 and 14	1,2,7,8,9,10,15,20,25

In testing robots with 5 obstacles on the robot path, we conducted 5 experiments. The table 6 provide the data of testing with 5 obstacles.

Table 6. Results of testing with 5 obstacles.

Experiment	Processing Time (s)	Time from Starting Point to Destination Point (s)	Total Time (s)	Position of Obstacles	Passed Route Points
1	2,226	8,667	10,893	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 14	1,2,7,8,13,23,24,25
2	7,232	8,513	15,745	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18	1,2,7,8,13,14,19,24,25
3	5,074	9,706	14,78	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18	1,2,7,8,13,14,15,20,25
4	7,553	9,772	17,325	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18	1,2,7,8,13,14,15,20,25
5	4,823	10,608	15,431	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18	1,2,7,8,13,14,19,24,25

In testing robots with 6 obstacles on the robot path, we conducted 5 experiments. The table 7 provide the data of testing with 6 obstacles.

Table 7. Results of testing with 6 obstacles.

Experiment	Processing Time (s)	Time from Starting Point to Destination Point (s)	Total Time (s)	Position of Obstacles	Passed Route Points
1	2,691	9,878	12,569	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18, Between 14 and 15	1,2,7,8,13,14,19,24,25
2	9,703	10,558	20,261	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18, Between 14 and 15	1,2,7,8,13,14,19,20,25
3	9,757	10,284	20,041	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18, Between 14 and 15	1,2,7,8,13,14,19,24,25
4	2,594	9,837	12,431	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18, Between 14 and 15	1,2,7,8,13,14,19,24,25
5	4,597	10,324	14,921	Between 1 and 6, Between 2 and 3, Between 7 and 12, Between 8 and 9, Between 13 and 18, Between 14 and 15	1,2,7,8,13,14,19,24,25

From the trials with some obstacles, we get that the processing time increases because every time the robot finds an obstacle it will be rerouting and it increases the processing time. The movement time increases too because almost every time the robot finds the obstacle, it will turn left or turn right to avoid the obstacle. The processing time is shown in Figure 2.

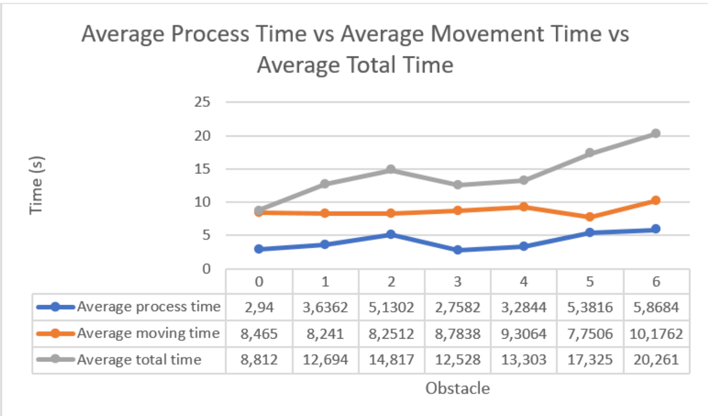

Figure 2. Processing time.

IOP Conf. Series: Materials Science and Engineering **732** (2020) 012098 doi:10.1088/1757-899X/732/1/012098

5. Conclusion

From the experiments, we get that PSO can be used to find the route from the start point to the endpoint. The percentage of randomizing is not too high, it is just 44,44% and decreases every time we add the obstacles. The average processing time, moving time and total time are tending to increase based on the number of obstacles, even though the increment from the moving time is quite small. Average total time with 5 obstacles is double from the average total time without obstacle, this far different value happens because of process time increase quite high.

References

[1] Alam M S, Rafique M U and Khan M U 2015 Mobile robot path planning in static environments using particle swarm optimization *International journal of computer science and electronics engineering (IJCSEE)* **3**(3)

[2] Ever Y K 2017 Using simplified swarm optimization on path planning for intelligent mobile robot *Procedia computer science* **120** 83-90

[3] Wang H and Zhou Z 2019 A Heuristic Elastic Particle Swarm Optimization Algorithm for Robot Path Planning *Information* **10**(3) 99

[4] Qiang N and Gao J 2017 A New Method for Mobile Robot Path Planning based on Particle Swarm Optimization algorithm *2017 2nd International Conference on Advances in Materials, Mechatronics and Civil Engineering (ICAMMCE 2017)* Atlantis Press

[5] Ayari A and Bouamama S 2017 A new multiple robot path planning algorithm: dynamic distributed particle swarm optimization *Robotics and biomimetics* **4**(1) 8

[6] Tang B, Zhu Z and Luo J 2016 Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning *International Journal of Advanced Robotic Systems* **13**(3) 86

[7] Hossain M A and 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