AN IMPROVED PATH PLANNING APPROACH FOR MOBILE ROBOT NAVIGATION BASED ON PARTICLE SWARM OPTIMIZATION

Praveen Kumar Yadav[1], J. C. Mohanta[2], S. R. Mohanty[3]
[1]PG SCHOLAR, [2]ASSISTANT PROFESSOR, [3]ASSISTANT PROFESSOR
[1,2]Department of Mechanical Engineering, MNNIT Allahabad
[3] Department of Electrical Engineering, MNNIT Allahabad
E-mail:praveen.adept.gs@gmail.com[1],jcmohnata@mnnit.ac.in[2],soumaya@mnnit.ac.in[3]

Abstract

This paper describes the motion planning problem of mobile robot based on particle swarm optimisation (PSO). To find a suitable collision free trajectory of a robot is the most important task in path planning problem, which satisfies certain criteria like feasibility, smoothness, safety, minimum path length and so on. Most of the paths developed by the researchers especially for the situation like edge navigation at the turning point, which may be difficult to implement on the real robot in the generated path. Therefore defining a suitable curve to describe a path is become essential for safe navigation. In present work the concept of B-spline curve is used to develop the path of the robot. Various simulations have been carried out and compare the results with the predefined standard benchmark maps problem to prove the effectiveness and efficacy of the developed control scheme.

Index Terms- Path Planning, Particle Swarm Optimization, Bezier Curves, B-spline curve.

I. INTRODUCTION

Mobile robot path planning is one of the most important topic in the field of robotics in last four decades. Path planning is basically the determination of path which will be used by the robot in order to pass over every single point in a given environment with obstacle, the path planning is used to find suitable collision free path, which needs to fulfill certain criteria like minimum path, minimum associated cost, minimum time, feasibility and safety (security), for mobile robot to move from start point to destination point. Researchers are trying to focus on finding the shortest path with minimum time and cost [1-2]. However, the motion of the robot in a sharp edge is still a challenging task for most of the navigation problems.

In order to solve such types of problems, this may be the reason of introducing curve for path planning; also non smooth-motion may have effect on slip [3-6]. In recent year lots of optimization algorithm is introduced which mimics biological behavior of animals and insects found in nature, such as Genetics algorithm, Particle swarm intelligence technique, Ant colony algorithm etc are gradually used in path planning and few of the got better result [7]. Particle swarm intelligence technique is natural heuristic algorithm based on the birds flocking characteristics .In [8] Bezier curve combined with Casteljau algorithm is discussed, particle swarm optimization in [9], genetic algorithm in [10].

Many traditional approaches for mobile robot path planning are artificial potential field method [11], neural network [12], D^* algorithm [13], and so on. With passage of time and development of evolutionary computational algorithm ,many natured inspired optimization algorithm has been proposed to solve path planning problem, Particle swarm optimization (PSO) which was proposed by Eberhart and Kennedy in 1995 [14-15] is based on swarm foraging behaviour. In this paper a novel path planning method based on PSO is proposed, which is used to optimize the necessary parameter of the curve which will finally achieve the shortest feasible path from start point to destination point, the length of the path is used to compare the optimality of different path planning algorithm. The experimental result shows that the proposed PSO algorithm outperform the algorithm taken for the comparison of the result.

II. METHOD

A. Bezier curve.

An engineer known as P.E. Bezier introduced a new curve representation which was based on the control points in 1962 and is named after him the Bezier curve [16].

Given a set of $N+1$ control point $P_0, P_1, \dots\dots, P_N$,
with degree *n* the corresponding Bezier curve C_v can be represented by :

$$P(t) = \sum_{i=0}^{N} P_i\, B_{i,N}(t) \quad t \in [0,1] \tag{1}$$

Where, $B_{i,N}(t)$ is the Berstein polynomial represented by $B_{i,N}(t) = C_N^i t^k (1-t)^{N-i}$

$$i = 0, i \dots . n \tag{2}$$

The parameter equation of every point for three times Bezier curve could be generated by formulas (1) and (2) as follows:

$$p(t) = P_0(1-t)^3 + 3P_1 t(1-t)^2 + 3P_2 t^2(1-t) + P_3 t^3 \tag{3}$$

Where t,is in the range of [0,1], Bezier curve starts at $t = 0$ and ends at $t = 1$.
Properties of Bezier curve [17]can be defined as follows:

- Bezier curve starts at the start point and ends at the end point.
- First derivative of the start and end points are only related to the nearest control points and in the same direction of line of two points.

The calculation formula

$$P'(0) = 3(P_1 - P_0),$$
$$P'(n) = 3(P_n - P_{n-1}) \tag{4}$$

A typical first order continuous Bezier curve can be formed by joining many segments of lower order Bezier curve each segment has four control points. In order to meet the property of first order continuity when using n segments of Bezier curves to describe a path, 2 n points are needed. The path can be generated by using the following Equations as follow:

$$P(t) = \begin{cases} P_0(1-t)^3 + 3P_1^i t(1-t)^2 \\ +3P_2^i t^2(1-t) + P_3^i t^3 \quad i = 1 \\ P_3^{i-1}(1-t)^3 + 3(P_3^{i-1} - P_2^{i-1})t(1-t)^2 \\ +3P_2^i t^2(1-t) + P_3^i t^3 \quad 1 < i < n \\ P_3^{i-1}(1-t) + 3\left(2P_3^{i-1} - P_2^{i-1}\right)t(1-t)^2 \\ +3P_2^i t^2(1-t) + P_1 t^3 \quad 1 = n \end{cases}$$

$$p(t) = [x(t), y(t)]^T \tag{5}$$

where P_0 represents the start point while P_1 stands for the end point. When t changes in the interval (0, 1), we can get a cubic Bezier curve of segment i. These n segments of cubic Bezier curve constitute the entire path of the curve.

B. B-spline curve

As we know that cubic hermit curve is controlled globally i.e. any change in any control point will change the wall curve along its full length, *also it* is not possible to keep the degree of the curve fixed ,while adding additional points any additional points will automatically increase the degree of the curve.

B-Spline curve is the solution of all the problem mentioned above it. B-spline curve is defined by $n+1$ control points P_i is given by

$$P(t) = \sum_{i=0}^{n} P_i\, N_{i,k}(t) \qquad 0 \le t \le t_{max} \tag{6}$$

Where $N_{i,k}(t)$ are the B-spline functions , parameter k controls the degree $(k-1)$ of the resulting B-spline curve and is independent of the number of control points , ,maximum limit of the parameter t is no longer 1 as was arbitrarily chosen. B-spline curve has the following property:
Partition of unity: $\sum_{i=1}^{n} N_{i,k}(u) = 1$
Positivity: $N_{i,k}(u) \ge 0$
Local support: $N_{i,k}(u) = 0 \quad if\ u \notin [u_i, u_{i+k+1}]$
Continuity: $N_{i,k}(u)\ is\ (k-2)\ times\ countinously\ differentiable.$
Some conclusions can be made on the basis of above properties, the first properties ensures that the relationship between the curve and its defining control points is invariant under affine

transformation. The second property guarantees that the curve segment lies completely within the convex hull of P_i . The third property indicates that each segment of B-spline curve is influenced by only k control points or each control point affects only k curve segments.

The B-spline function also has the property of recursion which is given by the equation shown below:

$$N_{i,k}(t) = (t - t_i)\frac{N_{i,k-1}(t)}{t_{i+k-1} - t_i} + (t_{i+k} - t)\frac{N_{i+1,k-1}(t)}{t_{i+k}-t_{i+1}} \quad (7)$$

Where,

$$N_{i,1} = \begin{cases} 1, & t_i \le t \le t_{i+1} \\ 0, & otherwise \end{cases}$$

t_i are called knot values.

C. PSO based path planning

The path planning problem is converted into an optimization problem through using B-spline curve which is used to describe the path, PSO is used to locate the optimal control point in B-spline curve to find the shortest path from start to end point. The basic flow of the algorithm for PSO based path planning is show below:

1) Initialize the particle by randomly assigning each particle an initial velocity and a position within the solution space.
2) Evaluate the desired fitness function which is to be optimized for each particle‘s position.
3) For each particle, update its previously found best position so far, ***Pi,*** if its current position is better than its previous best one.
4) Recognize and update the swarm's overall globally best particle that has the swarm‘s best fitness value, and set/reset its index as ***g*** and its position at ***Pg***.
5) Update the velocities of all the particles.
6) Move each particle to its new position as per the new velocity.
7) Repeat steps 2–6 until convergence or a stopping criterion is met (e.g., the maximum number of allowed iterations is reached; a sufficiently good fitness value is achieved; or the algorithm has not improved its performance for a number of consecutive iterations).

In PSO velocity and position update equation is given below:

$$v_{id(t+1)=}v_{id}(t)c_1R_1[p_{id}(t)-x_{id}(t)]+c_2R_2[p_{gd}(t)-x_{id}(t)] \quad (8)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id(t+1)} \quad (9)$$

Where,

- v_{id} Tells about the velocity.
- x_{id} Denotes the position of i^{th} particle in d^{th} dimension.
- p_{id} Denotes the previous best position of the i^{th} particle in the d^{th} dimension.
- p_{gd} Denotes the position of swarm's global best particle in d^{th} dimension.
- R_1 And R_2 are two n-dimensional vector with numbers uniformly selected in the range of [0.0, 1.0].
- c_1 And c_2 are positive constant weighting parameters.

To solve the path planning problem B-spline curve is used to describe the path, first of all model environment is created with start and end points with number of obstacle in it as per the standard map, randomly three points are taken through which path of the robot has to be pass and proposed PSO algorithm is used to optimize the path, length of the path is calculated by dividing the curve into smaller segment and distance of each segment is calculated , summing up the length of all the segment gives the length of the path formula used is as follow:

$$l_1 = (x_1 - x_0)^2 + (y_1 - y_2)^2$$
$$similarly\ l_2, l_3, \dots, l_{n+1}$$

$$L = l_1 + l_2 + l_3 + \dots + l_{n+1}$$

L gives the length of the path, in order to check the feasibility condition i.e. path is passing or touching the obstacle following equation is used,

$$d = (((xx - x_{obs}(k))^2 + (yy - y_{obs}(k))^2)^{1/2}$$
$$for\ k = 1,2,3, \dots, \quad (10)$$

Where, x_{obs} , y_{obs} are the x & y coordinate of the obstacles, if it touches the obstacle then it will leads to violation criteria which given by

$$v = (1 - \frac{d}{robs(k)}) \quad (11)$$

$$violation = \frac{v_1 + v_2 + \cdots v_n}{n}$$

Cost function is evaluated as follow:

$$cost = L \times (1 + \beta \times violation) \quad (12)$$

Where, β is a constant taken equal to 100, which is used to balance the proportion of path length.

III. SIMULATION AND RESULT

To compare the result of proposed PSO ,simulation result by J. J. Liang , H. Song, B. Y. Qu, and Z. F. Liu[], is taken with similar environment, the benchmark maps [a,b and c]are employed as the environment, the maps and the path planned by PSO are illustrated in figures . The white areas are negotiable and the coloured areas are impassable, the task is planning a shortest collision free path from starting point (5, 5) to the end point (15, 15).

Environment 1

Table 1.0 Parameter specification map a, b & c

parameter	PSO
Maximum Iterations	100
Number of particles	50
Inertia weight, w	1
Damping ratio (w damp)	.98
Global learning coefficient	1.5
Radius of obstacles	.8

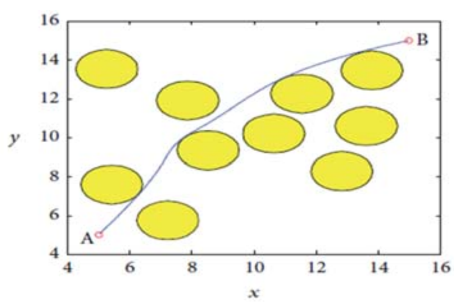


Fig.1.0 Result showing the navigational path of robot by J. J. Liang, et.al.[19]

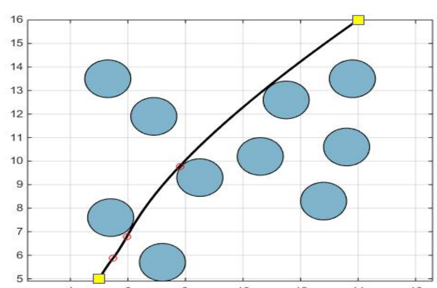


Fig 1.1 Fig. Result showing the navigational path of robot using the proposed algorithm.

Environment 1

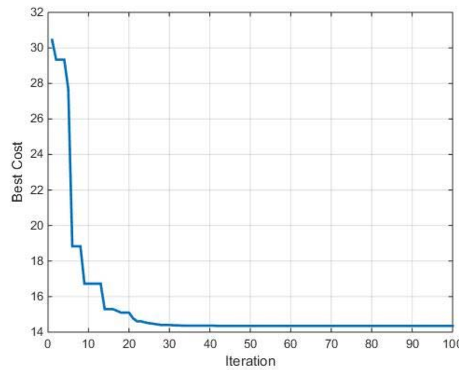


Fig 1.2 cost vs. iteration

The result obtained from the navigational path of robot using the proposed algorithm. is superior and the length obtained is 14.3491 cm, whereas the navigational path length of robot by J. J. Liang, et.al is 18.3684 cm.

Environment 2

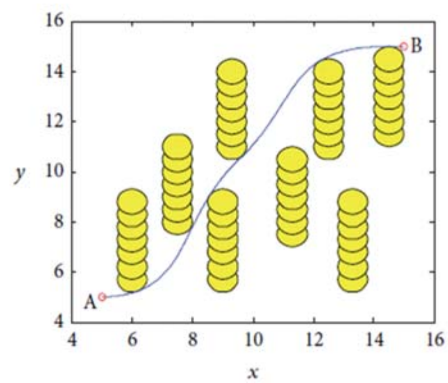


Fig.2.0 Result showing the navigational path of robot by J. J. Liang, et.al.[19]

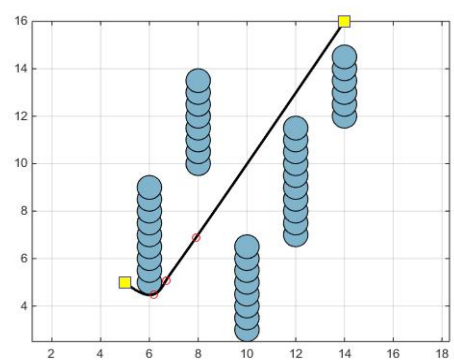
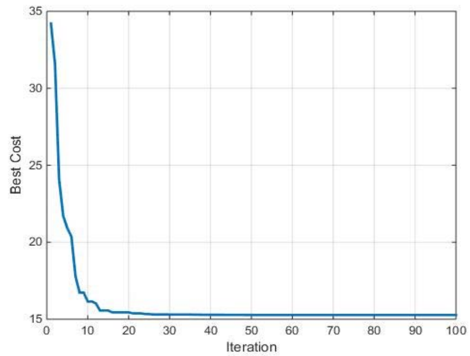


Fig 2.1 Result showing the navigational path of robot using the proposed algorithm

Environment 2



.

Fig 2.2 cost vs. iteration

For this case path length obtained through proposed algorithm is 15.3106 cm, whereas the navigational path length of robot by J. J. Liang, et.al is 16.7417 cm.

Environment 3

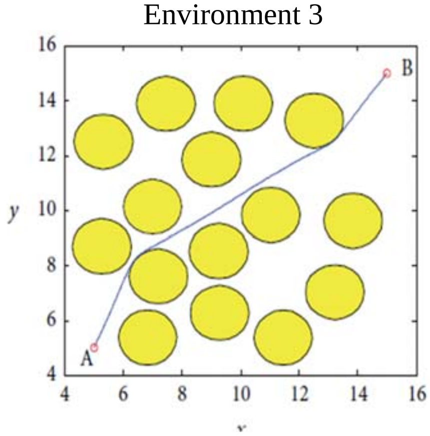


Fig.3.0 Result showing the navigational path of robot by J. J. Liang, et.al.[19]

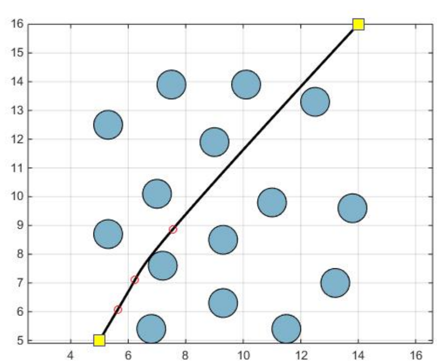


Fig. 3.1Result showing the navigational path of robot using the proposed algorithm. Environment 3.

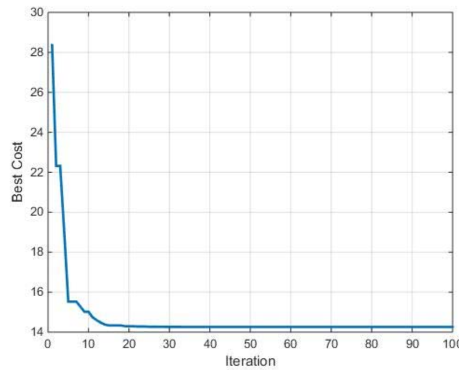


Fig 3.2 cost vs. iteration

For this case total path length is 14.2606 whereas the standard test result is 16.5045 cm.

Table 2.0 Comparison of result

Environment No.	**Results from J. J. Liang, et.al.[19]**	**Proposed method**	
	Distance (cm)	Distance (cm)	Time (sec)
1	18.3684	14.3491	42.368
2	16.7417	15.3106	67.286
3	16.5054	14.2606	44.169

IV. CONCLUSIONS

In this paper, improved PSO is employed to optimize the control points of B-spline curve to solve the path planning problem of mobile robot. Computer Simulations has been performed using MATLAB software for the situation like edge navigation. Results show that during edge navigation a smooth path has been generated by the developed algorithm. In order to validate the proposed control scheme the developed algorithm has been tested in various environments and compared with the results obtained from []. From the comparison it has been found that the proposed algorithm performed better results in terms of path length, smoothness of trajectories, time taken and convergence rate.

REFERENCES

[1] Abiyev, R., Ibrahim, D., & Erin, B. (2010). Navigation of mobile robots in the presence of obstacles. Advances in Engineering Software, 41(10), 1179-1186.

[2] Buniyamin, N., Wan Ngah, W. A. J., Sariff, N., & Mohammad, Z. (2011). A simple local path planning algorithm for autonomous mobile robots. International journal of systems applications, Engineering & development, 5(2), 151-159.

[3]. Zhao, T.: A Transit Path Planning Model Based on The Heterogeneous Road Network. In: International Conference on Geoinformatics, pp. 1–6 (2009).

[4]. Chesi, G.: Designing Image Trajectories in The Presence of Uncertain Data for Robust Visual Serving Path-Planning. In: IEEE International Conference on Robotics and Automation, pp. 1492–1497 (2009)

[5]. Li, G.S.: Study of Technology on Path Planning for Mobile Robots. In: Control and Decision Conference, pp. 3295–3300 (2008)

[6]. Qu, Y.H.: Flight Path Planning of UAV Based on Heuristically Search And Genetic Algorithms. Industrial Electronics Society, 6–10 (2005)

[7] M.J Zhang, Underwater Robot, Beijing: Ocean Press, 2000:Page1-7,123-133.

[9] D. W. Zhu and X. B. Mao, "Path planning algorithm based on improved particle swarm optimization of Bezier curves,"*Application Research of Computers*, vol. 5, pp. 1–6, 2012.

[10] L.Yang, Z. Luo, Z.Tang, and w. Lv, "Path planning algorithm for mobile robot obstacle avoidance adopting Bezier curve based on genetic algorithm," in *Proceedings of the Chinese Control and Decision Conference (CCDC '08)*, pp. 3286–3289, Yantai, China, July 2008.

[11] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[12] N. G. Bourbakis, D. Goldman, R. Fematt, I. Vlachavas, and L. H. Tsoukalas, "Path planning in a 2-D known space using neural networks and skeletonization," in *Proceedings of the IEEE International Conference on Systems,Man, and Cybernetics*, vol. 3, pp. 2001–2005, Orlando, Fla, USA, October 1997.

.[13] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 968–975, May 2002.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, NJ, USA, December 1995.

[15] Z. A. Bashir andM. E. El-Hawary, "Short-term load forecasting using artificial neural network based on particle swarm optimization algorithm," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECD '07)*, pp. 272–275, Vancouver, Canada, April 2007.

[16] F. Yamaguchi, F. Yamaguchi, "Curves and surfaces in computer aided geometric design". Berlin: Springer-Verlag, 1988, pp. 23-24.

[17] Z. A. Bashir andM. E. El-Hawary, "Short-term load forecasting using artificial neural network based on particle swarm optimization algorithm," in *Proceedings of the Canadian Conference.*

[18] J. J. Liang, H. Song, B. Y. Qu, et al., "Path planning based on dynamic multi-swarm particle swarm optimizer with crossover", *Intelligent Computing Theories and Applications*. Heidelberg: Springer Berlin, pp.159-166, 2012. *on Electrical and Computer Engineering*

[19] J. J. Liang,1 H. Song,1 B. Y. Qu,2 and Z. F. Liu2," Comparison of Three Different Curves Used in Path Planning Problems Based on Particle Swarm Optimizer", Hindawi Publishing Corporation. Mathematical Problems in Engineering Volume 2014, Article ID 623156, 15 pages

.