Journal of Physics:
Conference Series

PURPOSE-LED
PUBLISHING™

# Smooth and collision-free path planning for holonomic mobile robot based RRT-Connect

To cite this article: Yaaqob Saad Aljamali and M. Juhairi Aziz Safar 2023 *J. Phys.: Conf. Ser.* **2550** 012032

# Smooth and collision-free path planning for holonomic mobile robot based RRT-Connect

**Yaaqob Saad Aljamali**[1*][0000-0003-0960-8170] **and M. Juhairi Aziz Safar**[1,2]

[1*] Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis, 02600 Pauh Putra, Perlis, Malaysia

[2] Sports Engineering Research Center(SERC)CoE, Universiti Malaysia Perlis, 02600 Pauh Putra, Perlis Malaysia

E-mail: [1] `yaaqobsaad@studentmail.unimap.edu.my`

**Abstract.** Robot navigation is a crucial aspect of any mobile robot system. Many path planners generate a path that is zigzag in nature and contains many sharp or angular turns. Such courses are not adequate for a mobile robot to follow as it has to slow down and then speed up at these sharp turns. Thus, the integration of the smoothing function is a vital aspect to generate smooth and continuous paths. Moreover, a collision check after path smoothing needs to be accounted for a possible collision with obstacles surrounding. Herein, a path pruning algorithm is utilized to reduce the total path waypoints. Then a piecewise cubic B-spline smoothing function is applied to ensure the path is smooth and continuous. In addition, a post-smoothing collision checking and improvement algorithm based on mid-point insertion for the collide segments is proposed. Experiments and comparisons with the geometric RRT-Connect path planner have shown that the inclusion of a path pruning algorithm, smoothing function and post-smoothing collision check generates better results with reduced path length.

**Keywords**: B-Spline; collision detection; mobile robot; point insertion; path pruning; smooth path

## 1. Introduction

Recent trends show that the involvement of humans in industrial activities has decreased following the introduction of autonomous robots and automation technology [1]. The abilities such as solving complex motion planning are demanded to equip autonomous robots with higher performance capabilities replacing the human worker. Motion planning plays an important role in many mobile robot applications ranging from self-driving cars, Unmanned Aerial Vehicles (UAV), military drones, cleaning robots, planetary rovers, and rescue bots [2]. It can be divided into path planning and trajectory planning where path planning is merely geometric planning, whereas the latter assigned time law to the produced path [3]. Moreover, motion planning is divided into global or offline planning and local or online planning [1]. Wherein global planning complete information regarding robots, and the environment is given a prior; mostly this approach is used with a static environment. Whereas, in local planning the robot must be equipped with adequate sensors for planning and reacting to any interference; mostly this approach is used in a dynamic environment.

Herein, the focus will be on path planning wherein only a merely geometric path will be produced. Many algorithms have been presented in literature which can be classified based on their working nature into sampling-based approaches, the combinatorial approach, and biologically inspired approaches [4]. Sampling-based approaches have proven their efficiency in which can return a solution in a short time, and overcome the complexity of path planning for high dimension workspace [2]. Among the most commonly used algorithms are the Probabilistic Roadmap Algorithm (PRM) [5–7] and Rapidly-exploring Random Trees [8]. The focus here will be on RRTs based approach specifically RRT-Connect [9] that utilized Rapidly-exploring Random Trees (RRTs) with a simple greedy heuristic that aggressively tries to connect two trees, one from the initial configuration and the other from the goal.

However, most path planning algorithms including RRT-Connect generate a path consisting of straight lines and sharp turns [10] which is not adequate for the robot to follow. To overcome such a problem path smoothing functions have been proposed such as inscribed circle [11–13], removing redundant waypoints or unnecessary turns [14,15], three-point smoothing method [16]. These classical approaches resulted in curvature discontinuities which can result in overshooting [17], wheel-slippage [18], passenger discomfort [19,20], and mechanical wear [21,22] .
Special curves such as Clothoid have been utilized by several works which can provide continuous curvature planning [23–25] . However, clothoid-based planning uses global waypoints [10]; thus required high order approximations to represent them [26]. Parametric curves such as Bezier and B-spline curves were widely used in computer-aided design [27], which stimulate their use in robotic path smoothing. Cubic Bezier curve was utilized to produce smoothed path for differential drive robot respecting the kinematical constraints [28]. RRT-connect was integrated with goal biasing and sixth order Bezier curve to solve the path planning for non-holonomic mobile robot [29]. However, the order of the Bezier curve is dependent on the number of control points which can lead to high computational costs for high-degree curves. The global waypoints affect the entire curve, and placing new control points might be difficult [10]. B-spline was introduced as a general type of curve than Bezier curves which use several Bezier curves joined end on end. Many works in literature employ B-spline curves either as post-smoothing function [30–32], or integrated into the planning phase such as proposed by [26]. To account for possible post-smoothing collision mid-point insertion was utilized either for every path segment [26, 33] or only collide segments [30, 34].

Inspired by the work [26,30,34] the RRT-connect path planner was integrated with the pruning algorithm Ramer–Douglas–Peucker, then piecewise cubic B-spline algorithm was deployed to smoothen the linear path. Post-smoothing collision detection by discrete samples was performed. In the case of collision detection, a sub-waypoint was added for any collided segment then the smooth path was constructed including the sub-waypoint using cubic B-spline, and this continued until a smoothed collision-free path is produced.

This paper is organized as follows: Section 2 details the methodology followed to conduct the experiment, while Section 3 explains the experimental setup to evaluate the effects of pruning, smoothing, and collision detection algorithms inclusion into RRT-connect concerning total path length and path feasibility. Section 4 presents and discusses the obtained results. The paper is concluded in section 5 with future directions.

## 2. Methodology

Before conducting the experiment assumptions reported in [35] were adopted. The path produced by solely geometric path planners is zigzag in nature, thus not adequate for the robot to maneuver. Herein, the path smoothing algorithm is integrated with path planner RRT-connect, path pruning algorithm, and collision detection technique and refinement for colliding segment so that the final path is feasible as well as collision-free.

Figure 1 explains the different procedures of experimenting, wherein, there are five steps before producing the smoothed collision-free path:
(A) the map is first acquired using any means of mapping tools, then downsized to 500pixels-by-500pixels so that the experimental time is lesser. It is convenient for the map to be represented as logical values "1" and "0", thus the original map is binarized, wherein, the black area represents obstacles ($C_{obs}$) and the white area represents free space ($C_{free}$).

Normally the robot is treated as a mass point in the space configuration and the obstacles are inflated to account for the robot size [36] so that the path solution $P_s$ maintains a safety distance $D_s$ as indicated by Equation 1.

$$d(P_s, C_{obs}) > D_s. \tag{1}$$

Thus; obstacle regions were inflated using morphological erosion operation with a square structuring element whose width equals to robot virtual width. After obstacle enlargement the start and goal configurations are set interactively by the user which will be input to the path planner as $(X_s, Y_s)$ and $(X_g, Y_g)$.

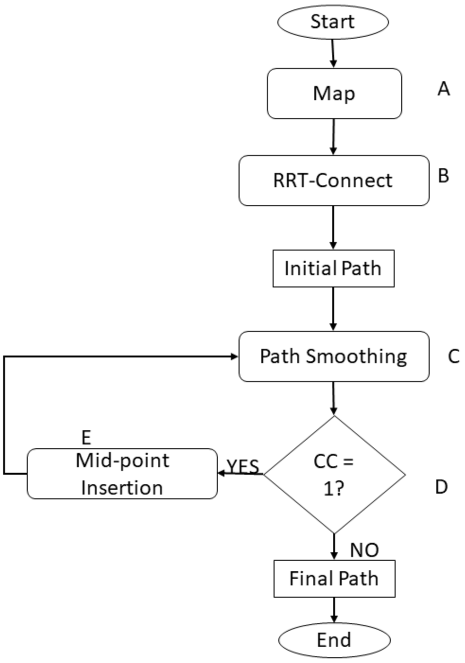


Figure 1: Flow Chart

(B) The utilized path planner is RRT-Connect which was first introduced by [9] wherein two rapidly-exploring random trees (RRTs) rooted at the start and goal configurations incrementally grow toward each other. The RRT-Connect algorithm syntax is shown in Algorithm 1, wherein, it takes as input environment with obstacles inflated, starts and goals configuration, step size $\epsilon$, the maximum number of iterations $k$, and the threshold value. The algorithm will be terminated either a collision-free path from start to goal is found, or the maximum number of iterations is exceeded. The path returned from RRT-Connect contains sharp turns which increases the cost of robot maneuvering, so the post-processing step is an essential requirement for any motion planning algorithm.

**Algorithm 1** The RRT-Connect algorithm

```
    Input: $map_{inflated}$, $q_{init}$, $q_{goal}$, $\epsilon$, threshold, k
    Output: $Path_{init}$, Computation Time, Path Length
$T_a.init(q_{init}); T_b.init(q_{goal})$
while i $\leq$ k do
    $q_{rand} \leftarrow RANDOM\_CONFIG()$;
    if not $(EXTEND(T_a, q_{rand}) = Trapped)$ then
        if $CONNECT(T_a, q_{new}) = Reached$ then
            Return $PATH(T_a, T_b)$;
    SWAP($T_a, T_b$);
Return $Failure$
```

(C) As mentioned previously, the path generated by the RRT-Connect planner is zigzag in nature and has many sharp turns, thus, it is not adequate for the robot to follow. To tackle such a problem many smoothing techniques have been proposed, however, herein, the B-spline algorithm [26, 30] was utilized as the post-smoothing algorithm. Before the post-smoothing the initial path was pruning using Ramer–Douglas–Peucker algorithm [37]. The Ramer-Douglas-Peucker algorithm takes as input the waypoints of the initial path, and the tolerance value which specifies how much a point can deviate from a straight line, and its value is between 1 and 0. The pruned path waypoints - control points - are input to the piecewise cubic B-spline, then the path is sampled uniformly between the specified time interval $[0, 5]$ with an increment of 0.01.

(D) Once the initial smooth path is generated, possible collision is checked with the obstacles surrounding, see Figure 1. If a collision occurs in the new smoothed path, then the next step is to locate the segments with collision and perform mid-point insertion [**?**,30,33]. After inserting the mid-point $P_{mid}$ between $P_{i-1}$ and $P_i$, and between $P_i$ and $P_{i+1}$, previous pruned path waypoints are updated with the new waypoints and input to the smoothing algorithm then checked for possible collision until the whole path becomes collision-free.

## 3. Experiment

All experiments were conducted on a 2.3 GHz Intel Core i3 machine with 6 GB memory running Windows 10 Pro. The algorithm was implemented in MATLAB. An environment of a size of 10m-by-10m was utilized to test the algorithm see Figure 2, wherein, obstacles were shown as black areas while free space as a white plane. The map was discretized so that in each 1 meter there are 50pixels so that the map input to the algorithm will be of size 500pixels-by-500pixels.

The robot that was considered is a non-circular shape with a length of 0.5m and a width of 0.3m. Based on the robot dimensions and the map size, the obstacles inflation amount (15 pixels) was calculated based on Equations 2-3.

$$Pixel/meter = map(colums)/MapRealLength \tag{2}$$

$$RobotVirtualWidth = RobotRealWidth \times PixelPerMeters \tag{3}$$

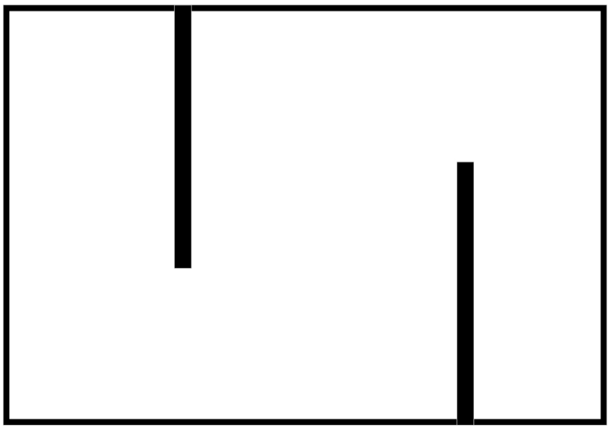


Figure 2: Map environment

As shown in the previous section RRT-Connect algorithm takes as inputs environment with obstacles inflated, start and goal configurations, step size $\epsilon$, the maximum number of iterations $k$, and threshold value which were set as shown in table 1. The path pruning algorithm takes as inputs the path generated and the tolerance value(0.03). The pruned path is input to the cubic B-spline smoothing algorithm. Post-smoothing collision detection is performed, once a collision is detected then the waypoints with a collision are located to the closest free waypoints, and then find the consecutive and a new point is inserted in-between using Equation 4

$$P_{mid} = (\frac{P_{i-1} + P_{i+1}}{2}) \tag{4}$$

Table 1: RRT-Connect Parameters

| Parameters | Value |
|---|---|
| Map | 500-by-500 |
| $q_{initial}$,$q_{goal}$ | $(X_s, Y_s)$ and $(X_g, Y_g)$ |
| Step Size ($\epsilon$) | 20 pixels |
| Threshold Value | 20 pixels |
| Maximum Iterations ($k$) | 10000 |

## 4. Result and Discussion

This section presents the effect of path smoothing, post-smoothing collision detection inclusion in the RRT-Connect algorithm concerning path length, and path feasibility. The total path length metric is a vital factor for algorithm efficiency since it affects the total travel time as well as the consumed energy. Path feasibility will be measured by its collision status, even though, the safety margin is not maintained after the smoothing which will be the subject of future work. The algorithm has been tested for 10 trails with random start and goal configurations before and after path optimization using path smoothing and post-smoothing collision check, see table 2.

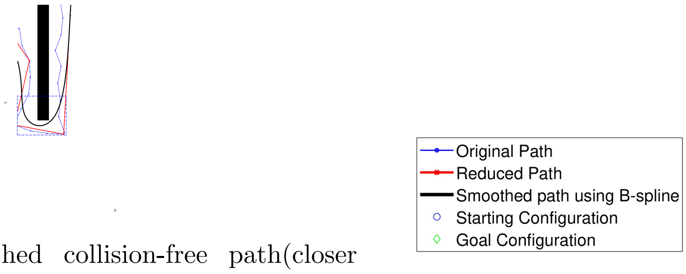
Table 2 shows the results obtained from 10 tails, wherein, the decrease in the path length is observed clearly. The values in bold experienced post-smoothing collision check since the collision was detected after the first smoothing has taken place, see Figure 3 as an example. As shown in the process block diagram Figure 1 once the collision has been detected the collided segments are located then new control points are formed by inserting mid-points for each collide segment. The total path length is calculated using the formula for distance $d$ between two consecutive points $(x_i, y_i)$ and $(x_{i+1}, y_{i+1})$ see Equation 5. Another factor that plays a major role in optimizing the path length is the path pruning algorithm utilized, wherein, the total waypoints are decreased by removing the redundant points based on the $\epsilon$ value set, see table 3. Needless to say, the location of the start and goal configurations plays a role in the total path length as well as the curvature of the smoothed path.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5}$$

Table 2: Path length before and after optimization

| **Trails** | **Path Length (Pixels)** | | **Collision Status** |
|---|---|---|---|
| | **Before Optimization** | **After Optimization** | |
| **1** | **1226.73** | **0.18** | **0** |
| 2 | 1326.50 | 1120.60 | 0 |
| 3 | 1001.70 | 845.89 | 0 |
| **4** | **1010.05** | **0.20** | **0** |
| 5 | 1164.20 | 969.54 | 0 |
| 6 | 536.42 | 481.69 | 0 |
| 7 | 577.67 | 493.05 | 0 |
| 8 | 580.72 | 510.68 | 0 |
| **9** | **844.47** | **0.02** | **0** |
| 10 | 1244.63 | 1043.98 | 0 |

(a) Initial smoothed path with collision (b) Smoothed collision-free path



(c) Smoothed collision-free path(closer look) (d) Legend

Figure 3: Effect of post-smoothing collision check on path feasibility and total path length

Table 3: Effects of path pruning algorithm

| **Trails** | **Total Waypoints** | | |
|---|---|---|---|
| | **Initial Path** | **Pruned Path** | **Post-Collision Pruned path** |
| **1** | **62** | **12** | **14** |
| 2 | 67 | 15 | 15 |
| 3 | 51 | 12 | 12 |
| **4** | **52** | **10** | **11** |
| 5 | 59 | 19 | 19 |
| 6 | 28 | 8 | 8 |
| 7 | 30 | 10 | 10 |
| 8 | 30 | 6 | 6 |
| **9** | **43** | **7** | **11** |
| 10 | 63 | 17 | 17 |

## 5. Conclusion

This paper discussed the effect of path smoothing, path pruning, and post-smoothing collision check when integrated with geometric path planners such as RRT-Connect concerning performance metrics (1. Total Path Length, and 2. Collision Status). The effects were observed by performing path planning for ten different start and goal configurations in a pre-defined environment with and without path optimization. Results showcased the effect of utilizing the path pruning algorithm before path smoothing in reducing the total waypoints hence the decrease in total path length. While the smoothing algorithm utilized -cubic B-spline- result in a continuous and feasible path in most cases while other collide with obstacles. Therefore, post-collision checking and collide-segment reformation were integrated to result in a smooth collision-free path. A limitation that was found is the feasibility of the smoothed path concerning the minimum safety distance between the smoothed path and the obstacles surrounding should be formally addressed in future work.

## Acknowledgments

This work was financed by the Ministry of Higher Education,Malaysia, through the Fundamental Research Scheme Grant (Ref: FRGS/1/2019/TK03/UNIMAP/02/6)

## References

[1] Gasparetto A, Boscariol P, Lanzutti A and Vidoni R 2015 *Motion and Operation Planning of Robotic Systems* 3–27
[2] Elbanhawi M and Simic M 2014 *IEEE Access* **2** 56–77
[3] Kunchev V, Jain L, Ivancevic V and Finn A *Int. Conf. on Knowledge-Based and Intell. Info. and Eng. Syst.*
[4] Debnath S K, Omar R, Bagchi S, Sabudin E N, Shee Kandar M H A, Foysol K and Chakraborty T K *Pro. of the 11th National Tech. Seminar on Unmanned Syst. Technol. 2019*
[5] Kavraki L and Latombe J C *Pro. of IEEE/RSJ Int. Conf. on Intell. Robot. and Syst. (IROS'94)*
[6] Amato N M and Wu Y *Pro. of IEEE Int. Conf. on Robot. and Autom.*
[7] Kavraki L E, Svestka P, Latombe J C and Overmars M H *IEEE Trans. Robot.*
[8] LaValle S M *et al.* 1998 Rapidly-exploring random trees: a new tool for path planning Tech. rep. Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA
[9] Kuffner J J and LaValle S M *Proc. 2000 ICRA.IEEE Int. Conf. Robot. Autom.Symp. Proc.(Cat. No. 00CH37065)*
[10] Ravankar A, Ravankar A A, Kobayashi Y, Hoshino Y and Peng C C *Sens.*
[11] Liu J, Li Z, He L P and Shi W
[12] Khaksar W, Uddin M Z and Torresen J *Int. Conf. on Robot. Intell. Technol. and Applic.*
[13] Lei G and Zheng Y *Int. J. Circuits Syst. Signal Process.*
[14] Liu X, Yao L and Dong H 2021 *2021 International Conference on Robotics and Control Engineering* pp 13–18
[15] Rajput U and Kumari M 2017 *Int. J. Bio Inspired Comput.* **9** 106–13
[16] Mandava R K, Bondada S and Vundavilli P R 2019 *Soft Computing for Problem Solving* (Springer) pp 139–50
[17] Lau B, Sprunk C and Burgard W 2009 *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE) pp 2427–33
[18] Magid E, Keren D, Rivlin E and Yavneh I 2006 *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE) pp 2296–301
[19] Morales Y, Kallakuri N, Shinozawa K, Miyashita T and Hagita N *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*
[20] Gulati S and Kuipers B *2008 IEEE Int. Conf. on Robotics and Automation*
[21] Berglund T, Brodnik A, Jonsson H, Staffanson M and Soderkvist I *IEEE Trans. Autom. Sci. Eng.*
[22] Maekawa T, Noda T, Tamura S, Ozaki T and Machida K i *Comput. Aided Des.*
[23] Heinemann T, Riedel O and Lechler A *Procedia Manuf.*
[24] Brezak M and Petrović I *IEEE Trans. Robot.*
[25] Fraichard T and Scheuer A *IEEE Trans. Robot.*
[26] Elbanhawi M, Simic M and Jazar R *IEEE Trans. Intel.l Transp. Syst.*
[27] De Boor C and De Boor C 1978 *A Practical Guide to Splines* vol **27** (springer-verlag New York)
[28] Luan P G and Thinh N T *Int. J. Mech. Eng. Robot. Res.*
[29] Liu H, Zhang X, Wen J, Wang R and Chen X *IFAC-Pap.*
[30] Noreen I *Sym.*
[31] Lei T, Luo C, Ball J E and Rahimi S 2020 *2020 IEEE Congress on Evolutionary Computation (CEC)* (IEEE) pp 1–6
[32] Zeng D, Yu Z, Xiong L, Zhao J, Zhang P and Fu Z *SAE Tech. Pap.*
[33] Elbanhawi M, Simic M and Jazar R N *J. Intell. Robot. Syst.*
[34] Chang S R and Huh U Y *Int. J. Adv. Robot. Syst.*
[35] Aljamali Y, Safar M, Basaruddin K, Yazid H, Basha S, Sa'ad F and Hassan M *Lect. Notes Electr. Eng.*
[36] Guruji A K, Agarwal H and Parsediya D 2016 *Procedia Technol.* **23** 144–49 ISSN 2212-0173 3rd International Conference on Innovations in Automation and Mechatronics Engineering 2016, ICIAME 2016 05-06 February, 2016

[37] Douglas D H and Peucker T K 1973 *Cartographica: the international journal for geographic information and geovisualization* **10** 112–22