MÜHENDİSLİK ve DOĞA BİLİMLERİ FAKÜLTESİ

Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi
International Journal of Engineering Research and Development

UMAGD, (2024) 16(1), 1-19.

10.29137/umagd.1278980

Cilt/Volume:16 Sayı/Issue:01 Ocak/January 2024

Araştırma Makalesi / Research Article

PRM Path Smoothening by Circular Arc Fillet Method for Mobile Robot Navigation

Mobil Robot Navigasyonu için Dairesel Yay Yumuşatma Yöntemi ile Yol Düzeltme

Meral Kılıçarslan Ouach[1*], Tolga Eren[1], Evrencan Özcan[1]

[1]*Kırıkkale University, Faculty of Engineering and Natural Science, 71451, Yahşihan, Kırıkkale, Türkiye*

Başvuru/Received: *08/04/2023* ***Kabul / Accepted:*** *25/10/2023* ***Çevrimiçi Basım / Published Online:*** *31/01/2024*
Son Versiyon/Final Version: *31/01/2024*

Abstract

Motion planning and navigation, especially for mobile robots operating in complex navigational environments, has been a central problem since robotics started. A heuristic way to address it is the construction of a graph-based representation (a path) capturing the connectivity of the configuration space. Probabilistic Roadmap is a commonly used method by the robotics community to build a path for navigational mobile robot path planning. In this study, path smoothening by arc fillets is proposed for mobile robot path planning after obtaining the path from PRM in the presence of the obstacle. The proposed method runs in two steps; the first one is generating the shortest path between the initial state to one of the goal states in the obstacle presence environment, wherein the PRM is used to construct a straight-lined path by connecting the intermediate nodes. The second step is smoothening every corner caused by node presence. Smoothening the corners with arc fillets ensures smooth turns for the mobile robots. The suggested method has been simulated and tested with different PRM features. Experiment results show that the constructed path is not just providing smooth turning; it is also shorter and quicker to finish for a robot while avoiding obstacles.

Key Words

"Probabilistic Roadmap Method (PRM) planner, Path smoothening, Arc fillet method"

Öz

Hareket planlama ve navigasyon özellikle karmaşık navigasyon ortamlarında çalışan mobil robotlar için robot teknolojisinin başlangıcından bu yana merkezi bir sorun olmuştur. Bunu ele almanın sezgisel bir yolu konfigürasyon alanının bağlantısını yakalayan grafik tabanlı bir gösterimin (bir yol) oluşturulmasıdır. Olasılıksal Yol Haritası robotik topluluğu tarafından navigasyonel mobil robot yol planlamasına yönelik bir yol oluşturmak için yaygın olarak kullanılan bir yöntemdir. Bu çalışmada Olasılıksal Yol Haritasından yol alındıktan sonra engelin varlığı durumundaki mobil robot yol planlaması için yay şeritleri ile yol yumuşatma önerilmiştir. Önerilen yöntem iki adımda çalışır; birincisi başlangıç durumu ile engel varlığı ortamındaki durumlardan biri arasındaki en kısa yolu oluşturmaktır; burada Olasılıksal Yol Haritası ara düğümleri bağlayarak düz çizgili bir yol oluşturmak için kullanılan hedeftir. İkinci adım düğüm varlığından kaynaklanan her köşeyi düzeltmektir. Köşelerin yay dolgularıyla yumuşatılması mobil robotlar için düzgün dönüşler sağlar. Önerilen yöntem farklı Olasılıksal Yol Haritasın özellikleriyle simüle edilmiş ve test edilmiştir. Deney sonuçları ile inşa edilen yolun sadece düzgün dönüş sağlamadığı aynı zamanda bu yol engellerden kaçınarak bir robotun işini bitirmenin daha kısa ve daha hızlı olduğunu göstermiştir.

Anahtar Kelimeler

"Olasılıksal yol haritası yöntemi planlayıcı, Yol yumuşatma, Yay doldurma yöntemi"

***Responsible Author:** m.kilicarslan@kku.edu.tr*

Nomenclature

Abbreviation,	Definition	VD	Voronoi Diagram
$\mathcal{G}$	Graph	RRT	Rapidly-exploring Random Trees
$\mathcal{V}$	Vertex	APF	Artificial Potential Field
$\mathcal{E}$	Edge	GA	Genetic Algorithm
PRM	Probabilistic Roadmap	ABC	Artificial Bee Colony
CS	Configuration Space	EPA	Evolutionary Programming Algorithm
dist	Distance	QPI	Quadratic Polynomial Interpolation
PP	Path Planning	SVM	Support Vector Machine
PSO	Particle Swarm Optimization	MD	Morphological Dilation

1. Introduction

Setting up a collision-free path and motion planning for autonomous vehicles to move from an initial state to a final destination, especially for obstacle hindered environments, has been challenging work in robotics over the last decade (Choset, 2005; Latombe , 1991; LaValle , ProQuest (Firm), 2006). Even though the latest robots may possess substantial differences in sense, size, actuation, workspace, application, etc., the problem of navigating through a complex environment is compound and crucial in almost all robotics applications. This problem also appears relevant to other domains such as autonomous exploration, computational biology, agriculture, search and rescue, etc. Considering the affecting factors on path planning (PP) like different kinds of robots and navigational environments, multiple robots, and dynamic or static obstacles, finding the shortest path with the highest degree of smoothness prevents collision with obstacles and other robots is still a challenging problem.

The goal of PP for mobile robots is to find a collision-free path from the starting point to the target point and optimize it based on specific methods (Sugihara & Smith, 1997) and (Raja & Pugazhenthi, 2012). Thus, determining an optimal path is difficult within environments containing many obstacles. As a result, depending on the complexity of the environment, this problem is categorized as NP-hard or NP-complete (Canny, 1988) and (Nasrollahy & Javadi, 2009). Therefore, PP studies generally specify the assumed environment type, known/unknown environments based on a pre-identified map. Nevertheless, the environment can be dynamic or static depending on whether the obstacle position changes when the mobile robot moves; however, a reliable map is essential for navigation without which robots cannot achieve their goals.

Another critical feature of robot PP is path smoothening. Although usually, the simplified PP solutions compute piecewise linear paths that neglect robot restrictions, it can sometimes be possible that the planned path may not be practical for a particular robot because the robot kinematic or other restrictions are not taking account of or the path may include sharp corners in which the robot cannot perform successfully. Thus, smoothening the path is very much important in these conditions. The path smoothening supports synchronization of robot acceleration and velocity with robot kinematics in practical applications such as: in case of multiple robot navigation, not only the navigation path but also the firm control over the robot acceleration and velocity are important points, or for the robot manipulator's path planning, the smoothening is necessary for effectively controlling the acceleration and velocity at the path corners so that the resulting accelerations or velocities can be maximized within their limitations, etc. Therefore, there are many reasons for path smoothening.

However, studies in the literature mentioned in Section 3 show that considerably less effort was made to smooth the path or have time complexity For example, it is mentioned in (Gang & Wang, 2016) that a circular arc is usually configured to replace the joints of the path segments so that a smooth path can be performed. On the other hand, Mohanta and Keshari in (Mohanta & Keshari, 2019) state that drawing a circular arc can be problematic if path joints are very close to the obstacles, and determining the appropriate radius of the circular arcs is itself difficult. However, utilizing a modified geometric approach proposed in this paper, determining a suitable circular arc radius becomes easy to compute and perform. Moreover, path safety is ensured by inflating the obstacles before applying PRM.

In this paper, new modified circular arcs are introduced to replace path segment joints so that optimal smooth path can be performed and smooth turning of the robot movement. This method uses the advantage of Probabilistic Roadmap (PRM), one of the most popular applications for computing navigational paths. The presented arc fillet method is a powerful study, especially for smoothening the robot navigational path, which can be counted as the novelty of this study. The success of the proposed approach is demonstrated through many simulations, and its superiorities are discussed thoroughly.

The primary elements of the overall framework of the proposed method consist of workspace preparation, path calculation, and path smoothing. The proposed framework is represented in Figure 1, and the general algorithm for the PRM Arc Fillet method is shown in a flowchart in Figure 14.

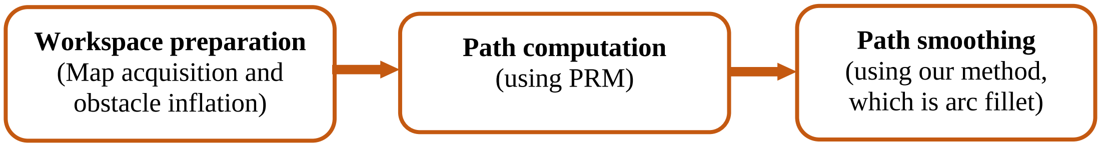

Workspace preparation
(Map acquisition and obstacle inflation)
Path computation
(using PRM)
Path smoothing
(using our method, which is arc fillet)

Figure. 1. Diagram of the workflow of the proposed method

In Section 2 of the study, a comprehensive definition of PRM has been given. Section 3 is extended by ensuring a broad overview of the work in the literature due to the relevance of the problem to PP methods. Section 4 shows the calculations of the optimal radius of circular arcs and optimal intersection points of arcs with the path edges. Path smoothing simulations at different PRM features are shown, and comparisons are made in Section 5. Finally, the obtained results and discussions are compiled in Section 6 and Section 7 concludes the outcomes, interests, and future scope of the research.

2. Methodological Foundations

2.1. Probabilistic Roadmap (PRM)

Classic PRM planner is a motion planning algorithm that makes sample points randomly defined as location nodes at the robot's configuration space and connects these points to construct the roadmap graph that captures connectivity of the collision-free subset of the configuration space (CS) (Ladd & Kavraki, 2004). It is highly appreciated to determine a pre-planned mobile robot path that solves the problem of calculating a path between the starting point of a robot to the goal point(s) while avoiding collisions with obstacles. PRM is combined by graph search algorithm (breadth-first search) to make graphs at CS during the construction phase and Dijkstra algorithm to obtain the shortest path between starting and goal points during the query phase.

Dijkstra Algorithm: Dijkstra's algorithm finds the shortest path between the start point and the endpoint among the location nodes in a workspace (Hamdy A. Taha, 1997). It is based on Euclidian distance as a measurement when the distance between nodes needs to be calculated. Dijkstra's algorithm, to start with, marks the distance from the starting point to every other node. Initially, the starting point, which is the source, is labeled as zero distance, while every other node is marked as infinity (see Algorithm 1). By exploring every node, their infinite value is changed to a constant number. Dijkstra's algorithm repeats a two-step process; the first step is updated estimates, and the second is the next nodes chosen to be explored. From the starting point, the shortest distance is selected among the nodes connected to the starting point, and then this point is labeled as the second point. This step is repeated until reaching the endpoint to find the shortest path.

Algorithm 1: Operation of Dijkstra

```
Input: $\mathcal{N}$ random samples from $CS$
Output: A trajectory $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in $CS$
$\mathcal{V} \leftarrow \{SampleFree_i\}_{i=1,\ldots\ldots,\mathcal{N}}$
$\mathcal{E} \leftarrow \emptyset$
dist[source]$\leftarrow$0
for each $v \in \mathcal{G}$ do
    dist[$v$] $\leftarrow$ infinity
    prev[$v$] $\leftarrow$ null
    add $v$ to $\mathcal{V}$
    while $\mathcal{G}$ is not empty, then
        $u \leftarrow v$ with min dist[$u$]
        remove $u$ from $\mathcal{V}$
    end while
end for
return $\mathcal{G}$
```

The goal of PRM is to calculate an optimal collision-free trajectory. The Standard algorithm for construction PRM is reviewed in Algorithm 2. Some formal definitions used in the description of the algorithm are as follows; let CS be the configuration space where $d \in \mathbb{N}$ is its dimension. The PRM establishes a roadmap stated as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose vertices are selections from CS and the edges are collision-free routes between vertices, as shown in Figure 2. Then the PRM launches the vertex set with $\mathcal{N}$ random samples from CS and tries to connect the nearest nodes. If no path is found, then nodes at CS must be increased at the construction phase.

Algorithm 2: Operation of PRM

```
Input: $\mathcal{N}$ random samples from CS
Output: A trajectory $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in CS
$\mathcal{V} \leftarrow \{\text{SampleFree}_i\}_{i=1,\ldots,\mathcal{N}}$
$\mathcal{E} \leftarrow \emptyset$
for each $v \in \mathcal{V}$ do
    $\mathcal{U} \leftarrow$ Apply_Dijkstra($\mathcal{G}$, $v$)
    for each $u \in \mathcal{U}$ do
        if CollisionCheck($v$, $u$) then
            $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v, u), (u, v)\}$
        end if
    end for
end for
if $\mathcal{G}$ is not computed, then
    return no path is found
else
    return $\mathcal{G}$
end if
```

Mobile robot path planning by PRM is required in these steps, i.e., import or create map depiction of the obstacle present environment, plan an obstacle-free path from a start and goal points at CS, finally follow the planned path from starting point to endpoint. In this study, maps are imported from the MATLAB library as simple and complex maps. The PRM complex map was inflated as obstacles to have a safe environment for the robot during the navigation. Map inflation extends each occupied position to a specified amount in the input by radius selected in meters as obstacles. Radius is rounded up a similar neighboring cell based on the resolution of the map. This map inflation increases the size of the occupied positions in the map. After path construction finishes, the inflated map returns to its original. This method is useful when the path is reshaped after PRM output.

Mobile robot path planning and executing the robot movements in an obstacle-free path using PRM involves various steps. The distinct representation of the mobile robot surroundings is the first important step. An area map obtained from images or manually can also be created using range sensor data from the robot and an occupancy grid that obstacle presence locations. In the second step, the PRM path planner calculates the connectivity between different map areas, extracting the obstacle-free path using the various CS nodes. If the robot movement project can work successfully in the simulation environment, they can implement it in real-life experiments.

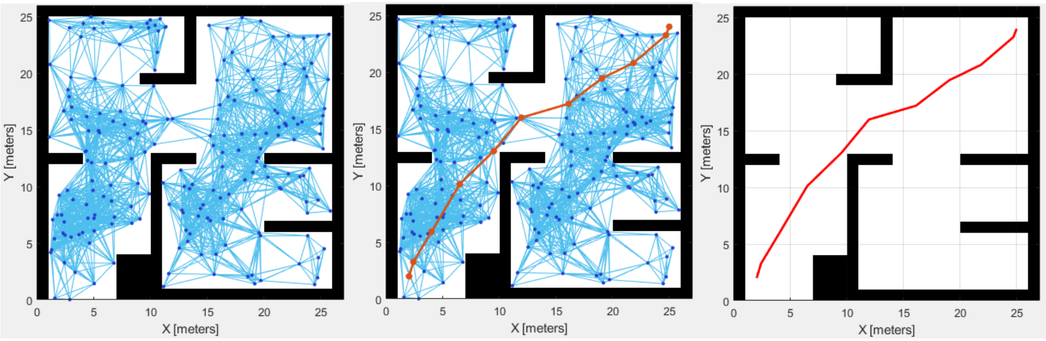


Figure 2. A Simple construction of PRM (a) Network formation (b) Path formation (c) Final path

As required, the probabilistic roadmap method is related; it is formed by two phases, i.e., (1) construction phase and (ii) query phase. In the construction phase, a trajectory (graph) is built by corresponding to the motions made in CS. Firstly, random nodes are generated in CS. Then they are attached to some neighbors, typically neighbors less than some predetermined distance (see Figure 3) or the k-nearest neighbors (see Figure 2). Finally, connections and configurations are added to the graph until the roadmap is compact. The query phase operates; the start and goal positions are connected by a graph derived by Dijkstra's shortest path algorithm.

A separate case study of PRM path formation that contains phases is demonstrated in Figure 2. In network formation, nodes are joined using the breadth-first search graph traversal technique after a specified number of nodes are randomly created in the given map, as shown in Figure 2a for the construction phase. Then Dijkstra's algorithm runs to determine of selection the shortest path as shown in Figure 2b for query phase, and the final path can be seen better in Figure 2c.

Node numbers can determine a straighter and shorter path at CS and a connection distance for the nodes at the path. Increasing node numbers brings computational complexity; hence the construction time of the path would be higher. At the same node numbers, different connection distance affects the shape of the path. When connection distance increases, the connection between nodes also increases during node formation; this gives more choices amongst nodes to form a path during path formation; hence the path would be straighter and shorter, but duration also increases. For example, at the PRM in Figure 3, the connection distance is less than Figure 2, but the path is straighter and shorter (Both figures have the same number of nodes, and these nodes are positioned at the same locations).

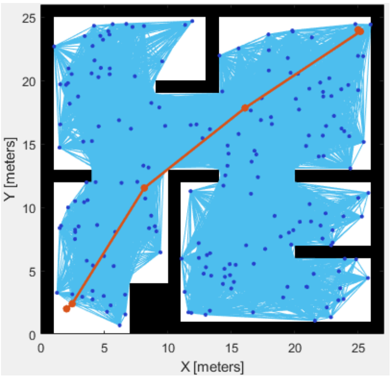


Figure 3. Effects of connection distance on PRM

A common version of the PRM is the lazy-collision PRM (Bohlin & Kavraki, 2000; Sánchez & Latombe, 2002) or its asymptotically optimal variant lazy-collision PRM*. In those versions of PRM, the ColisionCheck action in line 6 of Algorithm 1 is cut during the construction stage. Every time a path between two vertices is searched and detected in the graph, it is confirmed if at least one of its edges collides with an obstacle during the query stage. If one of the edges in the path is in a collision with an obstacle, that edge is eliminated from the graph, and a new path is constructed.

PRM benefits have been appreciated in various mobile robot path planning applications in 2D and 3D spaces, i.e., multiple robot path planning, pick and place robot end-effector path planning, etc. PRM is useful to predetermine obstacle-free the shortest path; this length can also be shortened by increasing the number of randomly specified nodes. If the number of nodes increases during the construction phase, then the length of path divisions reduces, in return, an increase in the number of intermediate nodes. As a result, an increase in the number of sharp edges is found at the path. The pre-planned route is nothing but the link of many straight lines connecting the start and goal positions. Straight-line connections cause sharp edges throughout the entire path. Whether the number of nodes is enhanced or not, PRM calculated path cannot prevent sharp edges.

2.2. Circular Arc Fillet

To be self-content, we briefly review an arc fillet definition. Arc fillet has good geometric properties and has been widely used in computer graphics and cad drawings applications in engineering and architectural projects. This algorithm detects the beginning angle and the angle subtended by the arc and the arc's direction. The lines' new beginning and ending points will be computed to join the arc smoothly (Miller, 1992). One of the significant advantages of the arc fillet is that it can be applied to any line that intersects at any angle. Therefore, the arc fillet is a good tool for curve fitting, and it smooths the line segments by turning them into arcs.

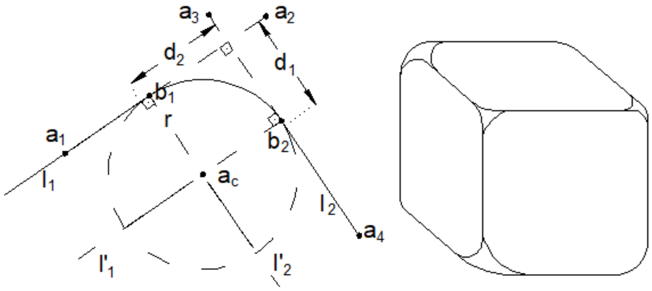


Figure 4. Joining two lines **(a),** and three lines **(b)** with a circular arc fillet

Constructing a circular arc fillet that joins two lines, l_1 ($\overline{a_1 a_2}$) and l_2 ($\overline{a_3 a_4}$) (see Figure 4a) can be performed step-by-step:

1) The line equations in the form $ax + by + c = 0$ need to be found for the lines that intersect. The center of the fixed arc a_c must lie at a distance r of both lines.
Define the distance d_1 from l_1 to the midpoint of l_2 and d_2 from l_2 to the midpoint of l_1. The midpoints are used because in practice one point may be mutual to both lines. The signs of d_1 and d_2 determine on which sides of the particular lines the arc center a_c resides (See Figure 4a).
2) Find $l_1' \parallel l_1$ at d1 and $l_2' \parallel l_2$ at d_2. The center of the expected arc ac lies at the intersection of l_1' and l_2'.
3) Calculate the beginning and ending points, b_1 and b_2 on the arc.
4) Find the starting angle, s (concerning the x-axis). Angle *s = tan-1(* $b_1 a_c$ *).* The two-argument arctangent is used to uniquely determine α in the range $0 \leq s < 2\pi$. Utilize the vector dot product to the directed line segments b_1a_c and b_2a_c to find the angle α subtended by the arc from a_c.
5) Use the vector cross product sign to define the direction to draw the arc b_1b_2.
6) The line may extend or be clipped at points b_1 and b_2 so the endpoints of the nearest line point of intersection of l_1 and l_2 will coincide with the endpoints of the arc. The fillet will result from drawing a line from a_1 to b_1 the arc from b_1 to b_2 and the line from b_2 to a_4.

The arc fillets are well connected to form a smooth path planning for mobile robots in the path planning problem.

3. Literature Review

The comprehensive research components of this study consist of mobile robot path planning using PRM and improving the navigational path by smoothening the corners. In the study, the terms "PRM path planning", "probabilistic roadmap", "mobile robot path planning" and "path smoothening" were searched using ScienceDirect, IEEE Xplore, SCOPUS, Web of Science, and Google Scholar to specify references that could influence the conceptional framework meticulously. To maintain the comparability of these findings, only studies that have utilized mobile robot path plannings in the last ten years with smoothening techniques have been considered.

Table 1. Literature summary on path smoothing methods

Ref.	Publication Type*	Path deriving method**	Path smoothing method**
(Song et al., 2021)	JA	Improved PSO	High degree Bézier Curve
(Ayawli et al., 2021)	JA	MD, VD and A*	Cubic spline interpolation
(Aria, 2020)	JA	PRM and RRT	Reed sheep planner
(Sun et al., 2020)	JA	VD and RRT-Connect	Cubic spline
(Mohanta & Keshari, 2019)	JA	PRM	Fuzzy control system
(Song et al., 2019)	JA	Modified PSO	η3-splines
(Tharwat et al., 2019)	JA	Chaotic PSO	Bézier curve
(Nazarahari et al., 2019)	JA	APF	Enhanced GA
(Cheok et al., 2019)	CP	-	Lyapunov stability
(Elhoseny et al., 2018)	JA	Modified GA	Bézier curve
(Wang et al., 2017)	CP	A partial and global A*	A partial and global A*
(Sudhakara et al., 2017)	CP	PRM	Spline method
(Janjoš et al., 2017)	CP	RRT	Quartic spline
(Han & Seo, 2017)	JA	Surrounding point set	Former and latter points
(Gang & Wang, 2016)	JA	PRM	Circular arc
(Song et al., 2016)	JA	GA	Bézier curve
(Ravankar et al., 2016)	JA	PRM	Hypocycloid curve
(Simba et al., 2015)	JA	A*	Bézier curve
(Davoodi et al., 2015)	JA	Multi objective GA	Multi objective GA
(Contreras-Cruz et al., 2015)	JA	ABC	EPA
(Su & Phan, 2014)	CP	-	Fuzzy inference system
(Huh & Chang, 2014)	CP	PRM	Modified QPI
(K. Yang et al., 2013)	JA	-	Cubic-Bézier spiral
(Kapitanyuk & Chepinsky, 2013)	JA	-	Piecewise method
(C.-Y. Yang et al., 2012)	JA	Voronoi tessellation	SVM
(Ju & Cheng, 2011)	CP	GA	GA

* Conference Paper (CP) and Journal Article (JA).
** Full names of methods are presented in Nomenclature.

This research has resulted in the following assessment layout:

- Objective and problem definition: It determines the motivation for mobile robot path planning to be considered and included in the scope of application.
- Path smoothening method: It includes path-creating methods.
- Assessment criteria: Includes which methods and techniques were examined for the application goal.
- Assessment of results: Includes the level of shortening the path, path smoothing degree, and calculation times.

The path smoothing process in the literature mentioned in Table 1 has some disadvantages such as too much computation time or only working when the path has a few nodes or not considering path deriving method to be optimal, and many of paths would be smoother and shorter even after smoothing process is performed

This paper proposes optimal smooth path planning, which takes the path from PRM in a complex environment to contribute to research in the field. It shows that some adjustments at PRM affect the performance of smoothening process on time complexity and for the shorter distance. Two steps were studied in this paper:

1. The first step is determining feasible connection distance between nodes at pure paths obtained from PRM.
2. The second step is smoothening every node at the path derived from PRM using our method, which is the arc fillet method.

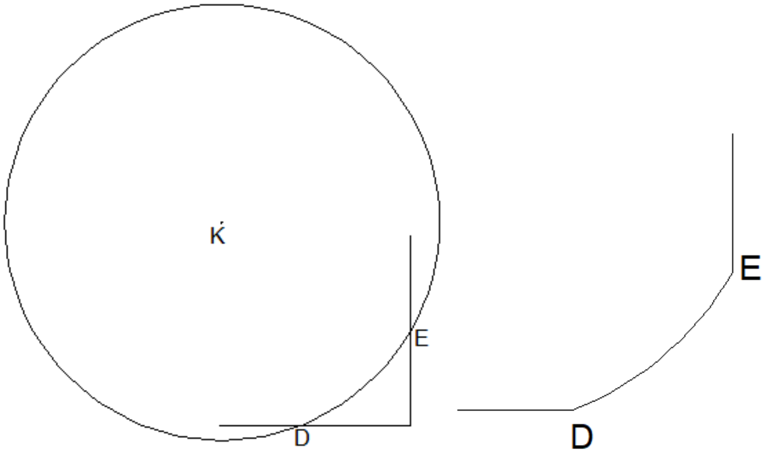
The proposed path smoothing method is easy to perform, has minimum calculation time, and can be successfully applied to many applications like robot manipulator movement planning, mobile robot path planning, multiple robot path planning, intelligent unmanned aerial vehicle navigation, etc. As a mobile robot path construction method, PRM was chosen in this study for its advantages as a tool to derive a path.

4. The Proposed Method

Smoothening process of the edges of a path consists of two steps explained below. The smoothening process is to be performed by helping a circular arc that belongs to a circle. The present research aims to find the right circle and position it in the right place. The purpose of smoothening here is not just for the sharp edges; it can be applied to all sorts of edges with angles varying from till 180o.

4.1. Optimal Cutting Points

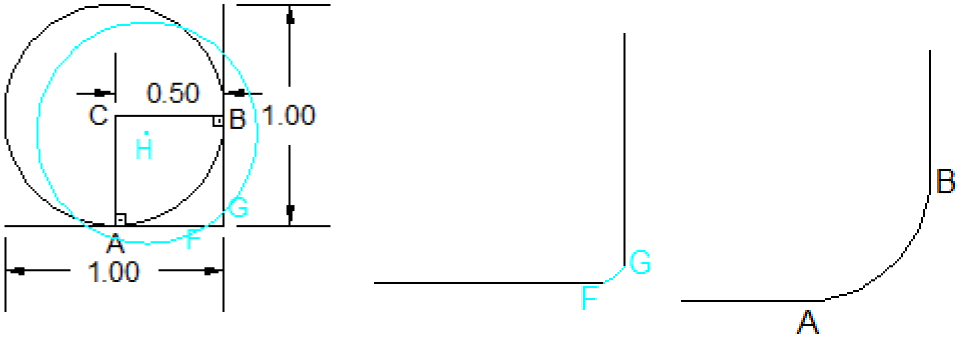
There mustn't be any remaining sharp edges after smoothening the corners at a given section of the path; this depends not only on the radius of the circle but also on the cutting points.



(a) (b)

Figure 5. A random circle cutting the edges of a path

In Figure 5a, the K centered circle cuts the edges of a path from D and E points. After trimming the remaining part from the node side, the modified path can be seen in Figure 5b. Unfortunately, the modified path doesn't seem to be smooth enough and still needs to be smoothened.



(a) (b) (c)

Figure 6. Two same circles cutting the edges of a path from different points

In Figure 6 there are two circles placed on a given edge of a path. Although they have the same diameter, cutting points make a difference in the smoothness of the new path. After trimming the remaining part from the corner side, two small corners appear at the F and G points in Figure 6b. Only in Figure 6c, arc AB connects the edges without any interruptions; in other words, no different small corners appear after trimming.

The optimal circle whose arc is compatible with the edges for smoothness must tangentially touch the edges. The circle mustn't cut the edges to avoid having extra mini corners. For a path that has many nodes, the formulation of the distance formula between two nodes would be:

$$disth_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (1)$$

At Equation (1) the distance between two nodes is calculated from node number *i* to *m*, where m denotes the number of nodes at the path.

To find the points that smoothening arcs start and end with, first, we need to define which edge is shorter. Let's suppose these points are a_i and b_i. Then let the ratio of distances is *t*, measured from a_i and b_i to the corner divide to the whole edge.

If $dist_i < dist_{i+1}$ half of the shorter edge (*h*) is $dist_i/2$. Then the ratio of distance;

$$t = h/2 \quad (2)$$

The initial coordinates for the arc smoothening at the shorter edge are as follows:

$$a_{xi} = (x_i + x_{i+1})/2 \quad (3)$$
$$a_{yi} = (y_i + y_{i+1})/2 \quad (4)$$

The final coordinates of the smoothing arc on the longer edge are as follows:

$$b_{xi} = (1-t)x_{i+1} + tx_{i+2} \quad (5)$$
$$b_{yi} = (1-t)y_{i+1} + ty_{i+2} \quad (6)$$

If $dist_i >= dist_{i+1}$ half of the shorter edge (*h*) is $dist_{i+1}/2$. The ratio of distance is still the same equation in Equation (2). The initial coordinates of the smoothing arc at the shorter edge would be:

$$a_{xi} = (1-t)x_{i+1} + tx_i \quad (7)$$
$$a_{yi} = (1-t)y_{i+1} + ty_i \quad (8)$$

The final coordinates of the smoothing arc at the longer edge would be:

$$b_{xi} = (x_{i+1} + x_{i+2}) / 2 \quad (9)$$

$$b_{yi} = (y_{i+1} + y_{i+2}) / 2 \quad (10)$$

4.2. Optimal Diameter

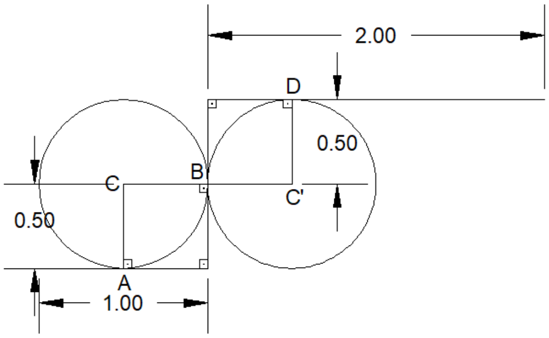


Figure 7. Defining the circle diameter

In Figure 7, the half of the edge with point B creates identical circles of C and C'. The common point of these circles is point B. When the edges have different lengths (the edge that has point B and the edge that has point D), the circle that smoothens the corner depends on the shorter edge. For maximum smoothness, the path continues by drawing an arc from A to B, and another arc starts at B point without stopping. For optimum smoothness, the circle's diameter equals to half the length of the shorter edge; this is also the same case for any edges whose angle varies.

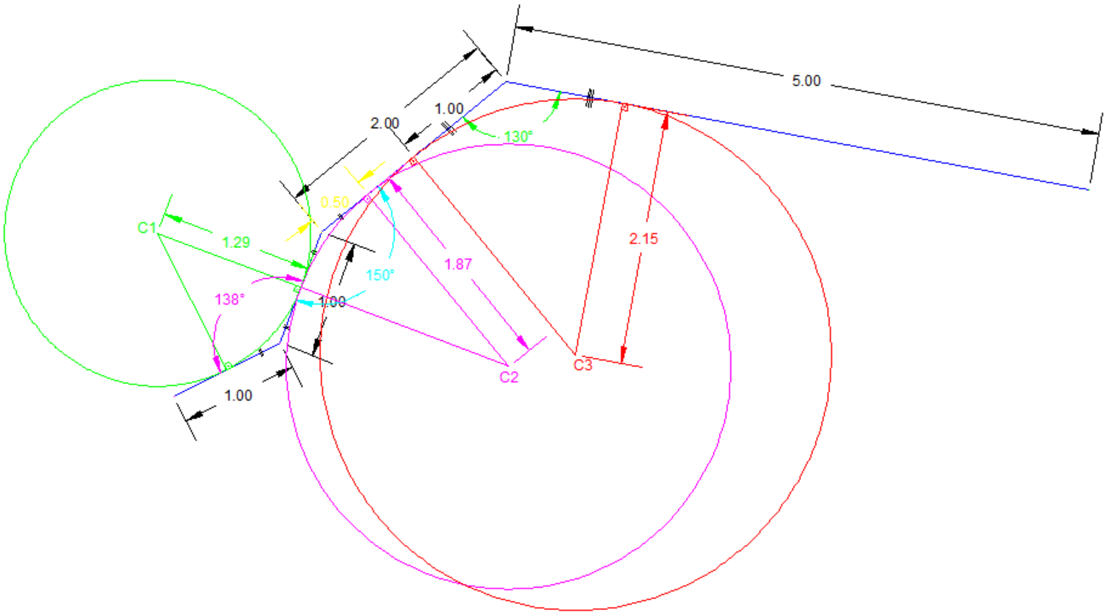


Figure 8. A path has different angles between edges

In Figure 8, the edges of the path have three different lengths, and the angles between them are also different; this leads to different smoothening circles having different sizes. The edges that C1 centered circle tangent to, have the same length. One of the edges of these is also tangent to C2 centered circle. The size of the circle depends on the angle between the edges and the tangent points at the edges.

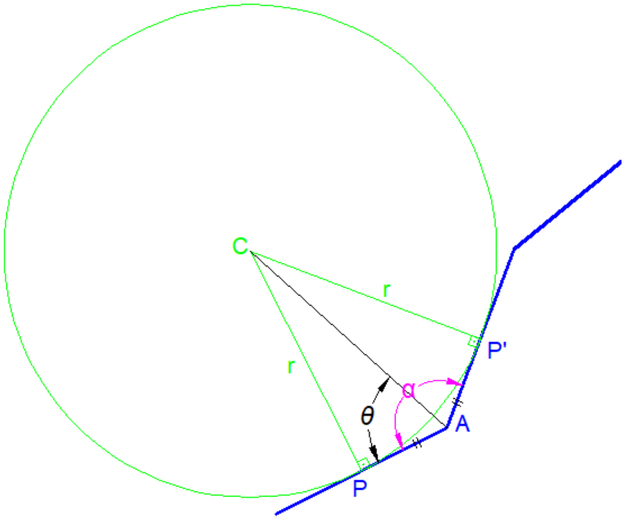


Figure 9. Finding the circle diameter

In Figure 9, the triangle formed by *CAP* and *CAP*' is identical because circle *C* is tangent to the edges of the path. The measurement of *CP* and *CP'* serve as the diameter of this circle. The angles of *CAP* and *CAP'* are equivalent due to the similar nature of the triangles. Therefore, these angles (θ) equal half the angle between edges (α). The length of *AP* and *AP*' is equal to half the shorter edge. In order to determine the radius of the smoothing circle, it is necessary to calculate the angle between edges through the utilization of the following formula:

$$\alpha_i = \mathrm{Arctan}(\frac{y_i - y_{i+1}}{x_i - x_{i+1}}) - \mathrm{Arctan}(\frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}}) \qquad (11)$$

At Equation (11) α_i is the angle between i_{th} and $(i+1)_{\mathrm{th}}$ edges where coordinates at i_{th} nodes which are (x_i,y_i) shaped. If the angle is negative, then;

$$\alpha_{inew} = 2\Pi\alpha_i \qquad (12)$$

In order to determine the angle between the edges and the vector from the center of the circle to the node, it is necessary to calculate the radius first. Specifically, at the i_{th} node, the angle can be represented as:

$$\theta_i = \frac{\alpha_i}{2} \qquad (13)$$

Through the utilization of basic trigonometric principles, the radius of the smoothing circle situated between edges can be determined.

$$r_i = \tan(\alpha_i).|AP| \qquad (14)$$

As depicted in Figure 9, half of the shorter edge can be observed in Equation (14). Additionally, by utilizing Equation (15) the center coordinates of the circle's center can be determined. In this equation, the variable C represents the circle's center, while P denotes the point at which the circle tangentially touches one of the edges.

$$(P_x - C_x)^2 + (P_y - C_y)^2 = r^2 \qquad (15)$$

This circle also tangentially touches the next edge. Therefore, using the same equation for the next point reduces the number of circles that touch these two points.

$$(P'_x - C_x)^2 + (P'_y - C_y)^2 = r^2 \qquad (16)$$

From Figure 9, the center coordinates of the center can be found by Equation (16). Here, the C point is the circle's center, and P is the tangential touch to the one edge.

Through this method, we have successfully narrowed down the number of circles that touch the defined points on the edges to two. These two circles are identical in terms of radius and connect the edges to the same two tangential points. However, as depicted in in Figure 10 these circles have distinct orientations.

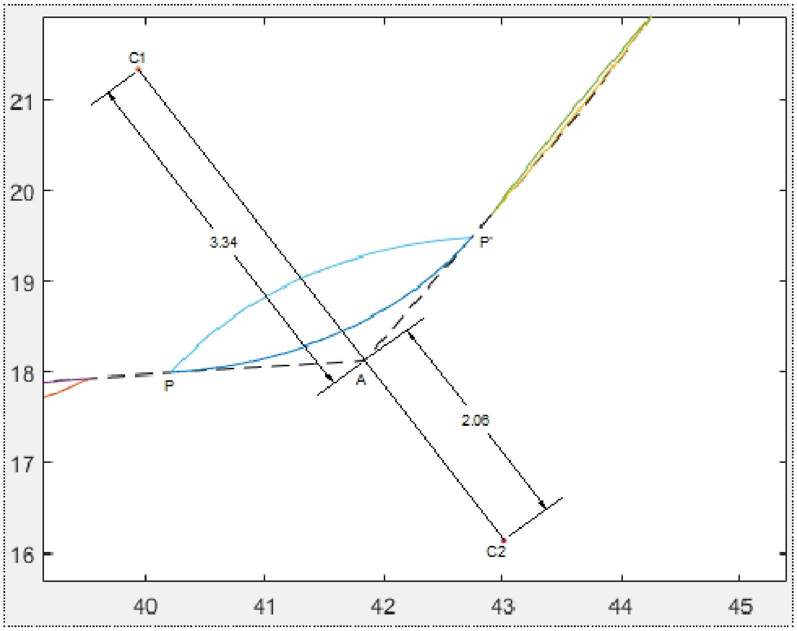


Figure 10. Finding the correct arc

In Figure 10, C1 and C2 centered arcs touch the same path edges to the same points (Point *P* and *P'*). The correct arc that smoothens the path has the center that has more distance from corner *A* than the incorrect arc. Equation (16) and (16) give these double center coordinates of these arcs. Let's suppose these coordinates are $C1_x$ and $C1_y$ for the *C1* centered arc and $C2_x$ and $C2_y$ for the *C2* centered arc. And we named the correct arc's center as *C*. Then:

$$C_i = \begin{cases} C2_i & \text{if} \quad \sqrt{(x_i - C1_{ix})^2 + (y_i - C1_{iy})^2} < \sqrt{(x_i - C2_{ix})^2 + (y_i - C2_{iy})^2} \\ C1_i & \text{if} \quad \sqrt{(x_i - C1_{ix})^2 + (y_i - C1_{iy})^2} > \sqrt{(x_i - C2_{ix})^2 + (y_i - C2_{iy})^2} \end{cases} \tag{17}$$

At Equation (17), in a path, i_{th} numbered node's coordinates were named as x_i and y_i. The correct arc that smoothens the corners was assigned as C_i at node number *i*.

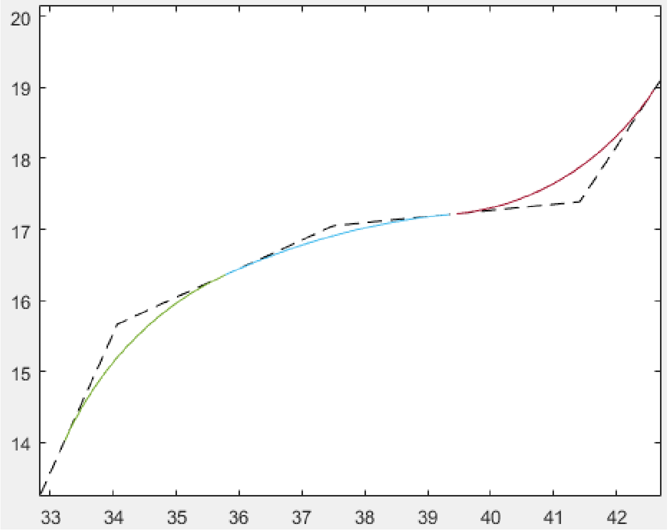


Figure 11 Final smoothening arcs of a path

After processing corners for smoothening in a path, results can be seen successfully in Figure 11. The process from circles to arc and the final path can be seen in Figure 12 and Figure 13 respectively.

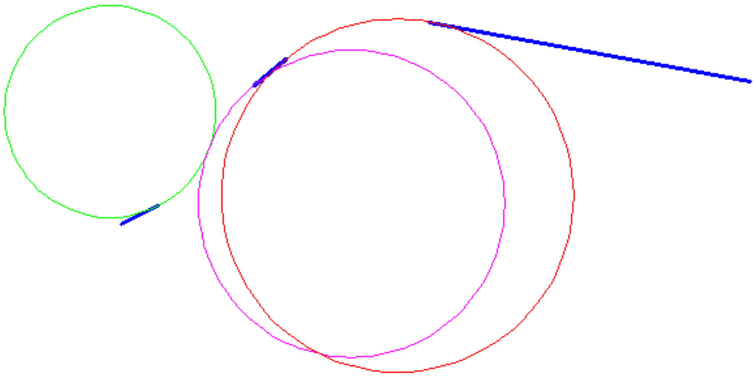


Figure 12. Trimming the path with the optimized circles

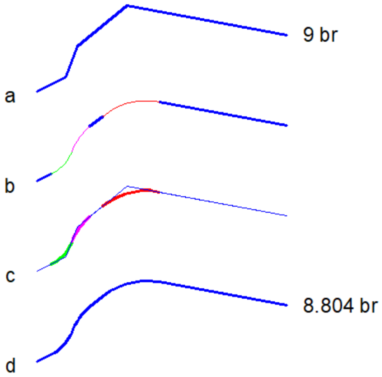


Figure 13. Remainder after smoothening the edges

The suitable circles in Figure 12 trimmed the corners at the path in Figure 8. After lifting out the circles from the corners, the smoothened path can be seen in Figure 13d. By smoothening the path, the path length is also diminished. The total length of the whole path can be found by:

$$dist = \sum_{i=1}^{m} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{18}$$

At Equation (18), the distance of the path is calculated by the *dist* equation from node number *i* to *m*, where *m* denotes the number of nodes at the path.

The path planning scheme is explained with the help of the flowchart shown in Figure 14.

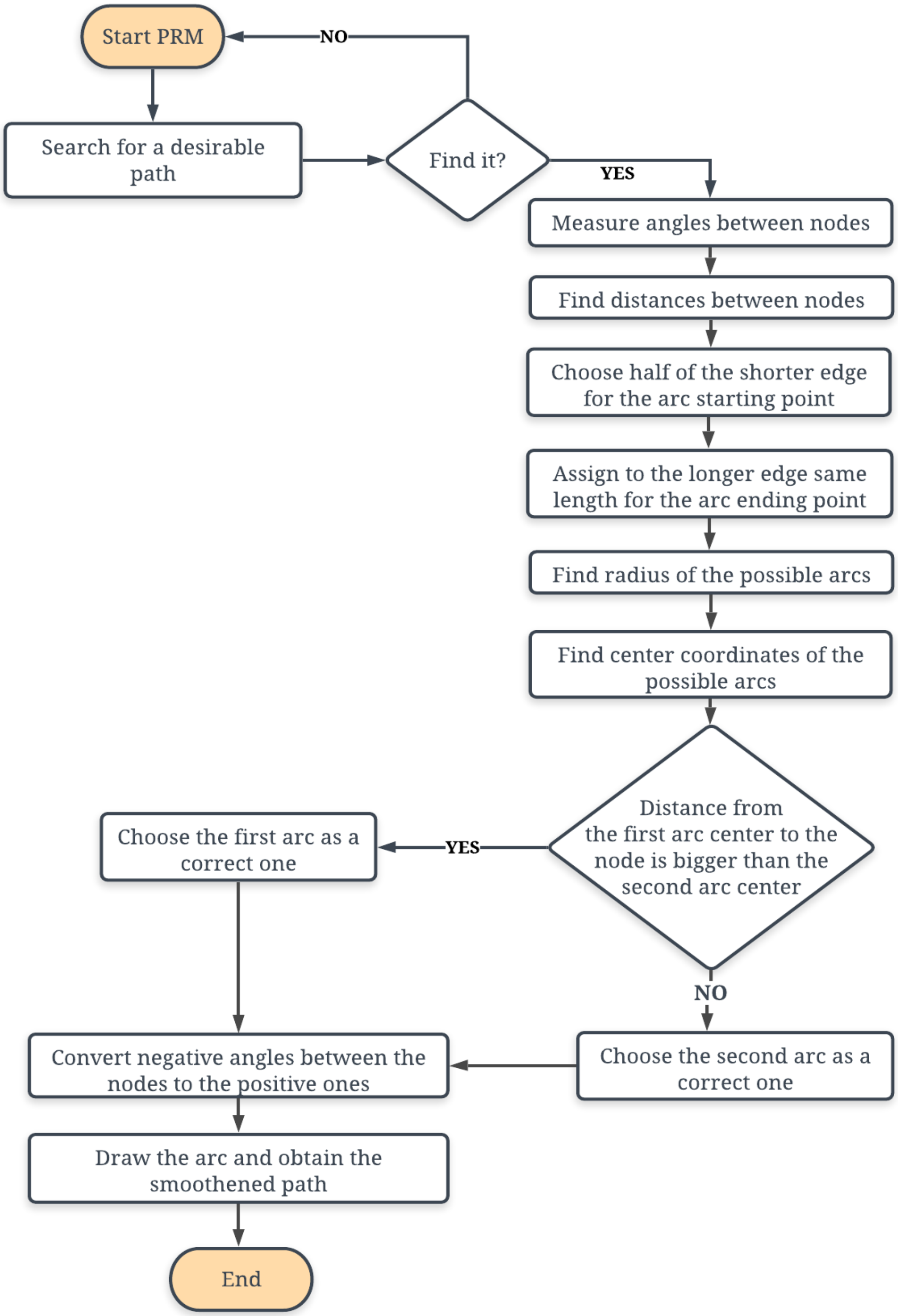

Start PRM
NO
Search for a desirable path
Find it?
YES
Measure angles between nodes
Find distances between nodes
Choose half of the shorter edge for the arc starting point
Assign to the longer edge same length for the arc ending point
Find radius of the possible arcs
Find center coordinates of the possible arcs
Distance from the first arc center to the node is bigger than the second arc center
YES
Choose the first arc as a correct one
NO
Choose the second arc as a correct one
Convert negative angles between the nodes to the positive ones
Draw the arc and obtain the smoothened path
End

Figure 14 Flow diagram defining the PRM smoothening path

5. Case Study: Mobile Robot Navigation

Workspace preparation: The robot's safety may not be fully guaranteed if its dimensions are not considered while building the path. The path may be built closer to obstacles or could also be built in narrow passages more diminutive than the robot's size, resulting in a collision. Besides the robot may not be able to fully utilize the created path because of noise and other unknowns caused by hardware restrictions and other defects both systematic and non-systematic. There may be variations resulting in crashes. Using PRM to generate a path does not take into account the robot's size, which may cause crashes and the robot getting stuck in narrow passages. While addressing this, and taking care of uncertainties, e.g. obstacles were inflated based on safety space requirements as mentioned in Section 2.

Many simulations have been tested on a standard complex map at MATLAB (binary occupancy map) for the same starting and endpoints. At a different number of PRM nodes and different connection distances, both pure PRM path and smoothened path behaviors have been observed. For upper and lower limits of connection distance, map and obstacle measurement limits have also been considered.

After PRM shaped the path, we smoothened the path and we made a simulation robot for walking the smoothened path from beginning to end. We observed some outcomes in the coming sections. Map, simulation robot and controller properties at MATLAB are given in in Table 1, Table 2, Table 3 and Table 4.

Table 1. Map

Type	Binary occupancy
Data Type	ogical
X limits (meters)	[0 52]
Y limits (meters)	[0 41]
Resolution	1

Table 2. Controller

Type	Pure pursuit
Max. Angular Velocity	2
Look Ahead Distance	0.3
Desired Linear Velocity	0.6

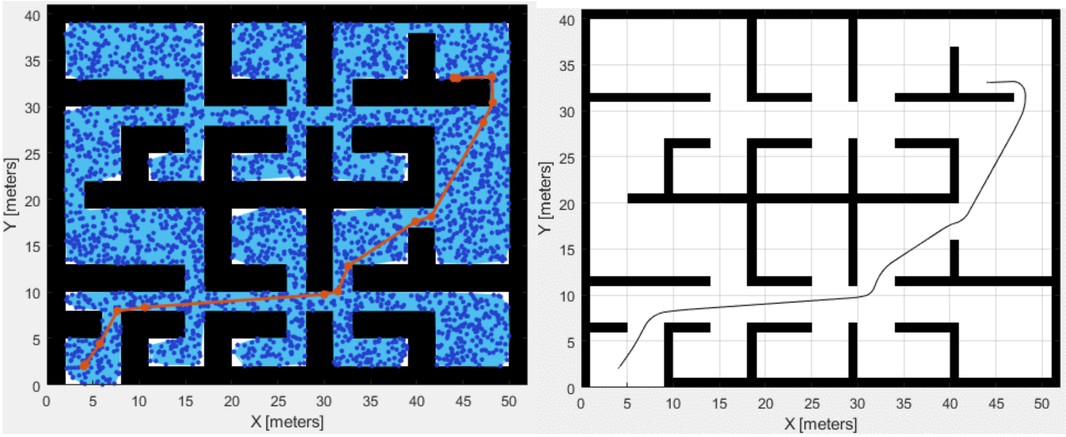
Table 3. Robot

Type	Differential drive
Track Width	1
Sample Time	0.1
Frame Size	1.25

Table 4. Viz rate

Type	Rate control
Desired rate	10
Desired period	0.1

Simulation environment:



(a) **(b)**

Figure 15. PRM by inflated map, number of nodes:3000, connection distance:100

In Figure 15a, PRM was utilized on an inflated map. Initially 3000 nodes were utilized to map the obstacle-free area. The chosen connection distance was set at 100 meters, resulting in a path formation time of 140.36 seconds. The length of the generated path was 140.36 meters comprising 15 nodes. In Figure 15b additional smoothing was applied to the path, with a duration of 1.03 seconds. The length of the smoothened path was 64.14 meters and the simulation robot traversed this path from the start to the end point within 107.38 seconds. The properties of these figures can be found in Table 5.

Table 5. Comparison of paths

		1st Exmp.	**2nd Exmp.**	**Av.of 20 Samp.**	**Av.of 20 Samp.**	**3rd Exmp.**	**4th Exmp.**	**Av.of 20 Samp.**	**Av.of 20 Samp.**
PRM	PRM Nodes	3000	3000	3000	3000	1000	1000	1000	1000
	Conn. Dist.	5	100	5	100	5	100	5	100
	Time	7.15	140.36	38.12	171.35	9.18	14.55	2.36	14.86
	Path Length	65.97	65.33	66.32	66.13	67.46	68.01	68.60	67.65
	Path Nodes	25.00	15.00	23.30	14.35	23.00	14.00	23.00	13.35
Smooth Path	Path Length	64.95	64.14	65.11	64.95	65.74	65.89	66.69	65.71
	Time	0.54	1.03	1.82	1.06	6.69	0.93	1.77	1.00
	Robot Walking Time	112.15	107.38	109.70	108.66	110.74	109.81	111.92	111.12
Total Time		119.85	248.77	149.64	281.07	126.60	125.29	116.05	126.98

Table 5 presents the results of an investigation into the effects of varying PRM node numbers and connection distances on the formation of paths using both pure path by PRM and smoothened versions. The study utilized PRM node numbers of 3000 and 1000 as well as connection distances of 5 meters and 100 meters. The properties of the paths are depicted in figures within the Simulation Environments section with 20 samples of each category being simulated to establish the average behavior of the variables under examination.

6. Results and Discussion

This section has made comparisons among the different PRM nodes at CS and connection distances between nodes while shaping the path. Path smoothing performances have been observed during four case studies in simulation experiments. When connecting the nodes to shape the path, the nearest possible nodes are connected, or the furthest possible nodes are connected. Even these connection distances, while shaping the path in the first place, also affects the behaviors of both the pure path and the smoothened one. The PP performance was mainly evaluated on path lengths, the number of path nodes or connection distance between nodes, and time is taken to cover the trajectories.

6.1. PRM Node Count

While shaping the path by PRM, the number of nodes at the movement area affects the path shape and time complexity.

For the pure path shaped by PRM:

- At the same connection distance, while PRM nodes are increasing, the time for shaping the path out of them also increases.
- If there are more nodes at CS, then the path has more segments.
- Pure path length shaped by PRM is shorter at more nodes at the CS.

For the smoothened path:

- When more PRM nodes shape the path, then the smoothened path is shorter.
- Time taking smoothening the path is more when more PRM nodes shape the path.
- When a simulation robot walks at the smoothened path, it takes less time to finish it when more PRM nodes shape the path.
- Total time of shaping PRM, smoothening the path, and the robot walking on it is more when more PRM nodes shape the path.

6.2. PRM Node Connectivity

While shaping the path by PRM, the connection distance of the nearest node at the path affects the path shape and time complexity.

- For the pure path shaped by PRM:
- When the connection distance between the nodes is more while shaping the path, the time needed for shaping the path is also more.
- The path length is more when the connection distance is less.
- Nodes at the path are more when the connection distance is less.

For the smoothened path:

- The path length is more when the connection distance is less.
- Time taking smoothening the path is more when the connection distance is less.
- When the simulation robot walks at the smoothened path, it takes less time to finish it when the connection distance is more.
- Total time of shaping PRM, smoothening the path, and the robot walking on it is more when the connection distance is greater.

7. Conclusion and Future Work

This research proposes a new easy and practical circular arc fillet methodology that smooths and the navigation along the PRM path, makes some adjustments to derive optimal path from PRM. The proposed methodology defeats many drawbacks of PRM based navigational mobile robot trajectory, i.e., PRM obtained mobile robot paths are segmented in linear lines, and connecting of two linear lines generates the corner at the PRM path nodes; causes sharp turning risks, and limits hassle-free controlled velocity navigation for robots; provides an optimal shortest path between the start point and the goal point. After the path is derived from PRM, optimal circles are found to smoothen the corners. Circular arc parameters such as radius and circle intersections with the edges need to be calculated for optimal circles. Before constructing the path, obstacles are inflated to have a safe environment after smoothing the path. The smoothing methodology is performed on four different PRM features. The modified circular arc method outperforms to handle any problematic sharp turnings in obstacle present environment.

It is mentioned in (Mohanta & Keshari, 2019), that drawing a circular arc can be difficult if path joints are very close to the obstacles, and determining a suitable radius of the circular arcs is itself a difficult job. However, this study demonstrates that determining the appropriate radius of the circular arcs becomes easy to compute and perform with the new geometric approach proposed in this paper. Furthermore, smoothened path safety is ensured by inflating obstacles before applying PRM.

Smoothening the path is affected by some factors, even different PRM variables initially while shaping the pure path. Results show that when the path has fewer nodes, in other words, when the connection distance between the nodes is more, the path is straighter; as a result, the path is shorter. When the path has more nodes and less connection distance, it requires additional work to smoothen extra nodes. However, shaping the original pure path by PRM at the beginning with more connection distance requires more time.

For future work, before smoothing the path, PRM path construction time with more connection distance can be reduced by processing the path with some techniques. After improving the pure path, the smoothing process can be applied to a straighter path. This study has not been tested in a dynamic environment, and maybe it would be considered in future works.

At this work, PRM algorithm was used to obtain a mobile robot navigational path. Circular arc fillets eliminated path corners. Finally, the path became ready to have smooth and shorter navigation. This smoothening process can be applied to any PP algorithm that gives path coordinates as an outcome and can be successfully performed in real life mobile robot PP applications such as autonomous robot navigation PP, multiple robot PP, robot manipulator movement planning, intelligent unmanned aerial vehicle navigation, etc.

References

Aria, M. (2020). New sampling based planning algorithm for local path planning for autonomous vehicles. *Journal of Engineering Science and Technology*, *15*, 66–76.

Ayawli, B. B. K., Appiah, A. Y., Nti, I. K., Kyeremeh, F., & Ayawli, E. I. (2021). Path planning for mobile robots using Morphological Dilation Voronoi Diagram Roadmap algorithm. *Scientific African*, *12*, e00745.https://doi.org/10.1016/j.sciaf.2021.e00745

Bohlin, R., & Kavraki, L. E. (2000). Path planning using Lazy PRM. *Proceedings-IEEE International Conference on Robotics and Automation*, *1*, 521–528. https://doi.org/10.1109/ROBOT.2000.844107

Canny, J. (1988). *The complexity of robot motion planning*. MIT Press.

Cheok, K. C., Iyengar, K., & Oweis, S. (2019). Smooth trajectory planning for autonomous leader-follower robots. *Proceedings of 34th International Conference on Computers and Their Applications, CATA 2019*, *58*, 301–309. https://doi.org/10.29007/n6kt

Choset, H. M. (2005). *Principles of robot motion : theory, algorithms, and implementation*. MIT Press.

Contreras-Cruz, M. A., Ayala-Ramirez, V., & Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing*, *30*, 319–328. https://doi.org/https://doi.org/10.1016/j.asoc.2015.01.067

Davoodi, M., Panahi, F., Mohades, A., & Hashemi, S. N. (2015). Clear and smooth path planning. *Applied Soft Computing*, *32*, 568–579. https://doi.org/https://doi.org/10.1016/j.asoc.2015.04.017

Elhoseny, M., Tharwat, A., & Hassanien, A. E. (2018). Bezier Curve Based Path Planning in a Dynamic Field using Modified Genetic Algorithm. *Journal of Computational Science*, *25*, 339–350. https://doi.org/https://doi.org/10.1016/j.jocs.2017.08.004
Gang, L., & Wang, J. (2016). PRM Path planning optimization algorithm research. *WSEAS Trans. Syst. Control*, *11*, 81–85.

Hamdy A. Taha. (1997). Operations research: An introduction. *Computers & Mathematics with Applications*, *33*(5), 132. https://doi.org/https://doi.org/10.1016/S0898-1221(97)82948-3

Han, J., & Seo, Y. (2017). Mobile robot path planning with surrounding point set and path improvement. *Applied Soft Computing Journal*, *57*, 35–47. https://doi.org/10.1016/j.asoc.2017.03.035

Huh, U.-Y., & Chang, S.-R. (2014). A G(2) Continuous Path-smoothing Algorithm Using Modified Quadratic Polynomial Interpolation. *International Journal Of Advanced Robotic Systems*, *11*. https://doi.org/10.5772/57340

Janjoš, F., Reichart, R., & Niermeyer, P. (2017). Smooth Path-Generation Around Obstacles Using Quartic Splines and RRTs. *IFAC-PapersOnLine*, *50*(1), 9108–9113. https://doi.org/10.1016/j.ifacol.2017.08.1708

Ju, M.-Y., & Cheng, C.-W. (2011). Smooth Path Planning Using Genetic Algorithms. *2011 9th World Congress On Intelligent Control And Automation (Wcica 2011)*, 1103–1107.

Kapitanyuk, Y. A., & Chepinsky, S. A. (2013). Control of mobile robot following a piecewise-smooth path. *Gyroscopy and Navigation*, *4*(4), 198–203. https://doi.org/10.1134/S207510871304007X

Ladd, A. M., & Kavraki, L. E. (2004). Measure theoretic analysis of probabilistic path planning. *IEEE Transactions on Robotics and Automation*, *20*(2), 229–242. https://doi.org/10.1109/TRA.2004.824649

Latombe , J.-C. (1991). *Robot Motion Planning*. https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3080882

LaValle , ProQuest (Firm), S. M. (2006). *Planning algorithms*. Cambridge University Press.

Miller, R. D. (1992). IV.5 - Joining Two Lines With A Circular Arc Fillet. In D. Kirk (Ed.), *Graphics Gems III (IBM Version)* (pp. 193–198). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-08-050755-2.50044-0

Mohanta, J. C., & Keshari, A. (2019). A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation. *Applied Soft Computing*, *79*, 391–409. https://doi.org/https://doi.org/10.1016/j.asoc.2019.03.055

Nasrollahy, A. Z., & Javadi, H. H. S. (2009). Using Particle Swarm Optimization for Robot Path Planning in Dynamic Environments with Moving Obstacles and Target. *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, 60–65. https://doi.org/10.1109/EMS.2009.67

Nazarahari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications*, *115*, 106–120. https://doi.org/https://doi.org/10.1016/j.eswa.2018.08.008

Ouach, M. K., Eren, T., & Özcan, E. (2022). PRM path smoothening by circular arc fillet method for mobile robot navigation. *Arxiv.Org*.

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

Ravankar, A., Ravankar, A. A., Kobayashi, Y., & Emaru, T. (2016). SHP: Smooth Hypocycloidal Paths with Collision-free and Decoupled Multi-robot Path Planning. *Internatıonal Journal Of Advanced Robotıc Systems*, *13*. https://doi.org/10.5772/63458

Sánchez, G., & Latombe, J.-C. (2002). On delaying collision checking in PRM planning: Application to multi-robot coordination. *International Journal of Robotics Research*, *21*(1), 5–26. https://doi.org/10.1177/027836402320556458

Simba, K. R., Uchiyama, N., Aldibaja, M., & Sano, S. (2015). Vision-based smooth obstacle avoidance motion trajectory generation for autonomous mobile robots using Bézier curves. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *231*(3), 541–554. https://doi.org/10.1177/0954406215616986

Song, B., Wang, Z., & Sheng, L. (2016). A new genetic algorithm approach to smooth path planning for mobile robots. *Assembly Automation*, *36*(2), 138–145. https://doi.org/10.1108/AA-11-2015-094

Song, B., Wang, Z., & Zou, L. (2021). An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Applied Soft Computing*, *100*, 106960. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106960

Song, B., Wang, Z., Zou, L., Xu, L., & Alsaadi, F. E. (2019). A new approach to smooth global path planning of mobile robots with kinematic constraints. *International Journal of Machine Learning and Cybernetics*, *10*(1), 107–119. https://doi.org/10.1007/s13042-017-0703-7

Su, K.-H., & Phan, T.-P. (2014). Robot Path Planning and Smoothing Based on Fuzzy Inference. *2014 IEEE International Conference On System Science And Engineering (ICSSE)*, 64–68.

Sudhakara, P., Ganapathy, V., & Sundaran, K. (2017). Probabilistic roadmaps-spline based trajectory planning for wheeled mobile robot. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 3579–3583. https://doi.org/10.1109/ICECDS.2017.8390129

Sugihara, K., & Smith, J. (1997). Genetic algorithms for adaptive motion planning of an autonomous mobile robot. *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. "Towards New Computational Principles for Robotics and Automation,"* 138–143. https://doi.org/10.1109/CIRA.1997.613850

Sun, Y., Zhang, C., Sun, P., & Liu, C. (2020). Safe and Smooth Motion Planning for Mecanum-Wheeled Robot Using Improved RRT and Cubic Spline. *Arabian Journal for Science and Engineering*, *45*(4), 3075–3090. https://doi.org/10.1007/s13369-019-04283-x

Tharwat, A., Elhoseny, M., Hassanien, A. E., Gabel, T., & Kumar, A. (2019). Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm. *Cluster Computing*, *22*(s2), 4745–4766. https://doi.org/10.1007/s10586-018-2360-3

Wang, K., Dang, S., He, F., & Cheng, P. (2017). *A Path Planning Method for Indoor Robots Based on Partial a Global A-Star Algorithm*. *130*(Fmsmt), 395–398. https://doi.org/10.2991/fmsmt-17.2017.83

Yang, C.-Y., Yang, J.-S., & Lian, F.-L. (2012). Safe And Smooth: Mobile Agent Trajectory Smoothing By Svm. *International Journal Of Innovative Computing Information And Control*, *8*(7B), 4959–4978.

Yang, K., Jung, D., & Sukkarieh, S. (2013). Continuous curvature path-smoothing algorithm using cubic Bezier spiral curves for non-holonomic robots. *Advanced Robotics*, *27*(4), 247–258. https://doi.org/10.1080/01691864.2013.755246