2016 16th International Conference on Control, Automation and Systems (ICCAS 2016)
Oct. 16–19, 2016 in HICO, Gyeongju, Korea

Path Smoothing Extension for Various Robot Path Planners

Abhijeet Ravankar[1], Ankit A. Ravankar[2], Yukinori Kobayashi[2], Takanori Emaru[2]

[1,2]Division of Human Mechanical Systems and Design
[1]Graduate School of Engineering, Hokkaido University, Sapporo, Japan
[2]Faculty of Engineering, Hokkaido University, Sapporo, Japan
(abhijeet@eis.hokudai.ac.jp) * Corresponding author

Abstract: Many path planning algorithms have previously been proposed for mobile robots to navigate from a start to a goal location in a given map. These planners generate a path which keeps a safe distance from the obstacles in the map. However, most of the global path planners generate a path with sharp and angular turns which is not desired for robot motion as robots must stop at these turns. A smooth path is desired for robot motion which allows the robot to move at nearly constant velocity. We present a novel path smoothing extension which uses the geometry of hypocycloids to smooth out the sharp and angular turns of the robot's path and generates a smooth path for the robot to traverse. The proposed technique works as an 'extension' and can be used in conjunction with any of the previously proposed global path planners like D*, A*, Dijkstra, or PRM (Probabilistic Roadmap) planners. The proposed extension also generates 'nodes' on the robot's path which can be used as points of retreat for the robot to avoid collision with other robots. Unlike other path smoothing algorithms which generates a wavy path for the robot and brings them close to the walls, the proposed path smoothing extension keeps straight paths of the robot straight, and smooths only the turns. We discuss the results in both simulated and real environments about the smooth paths generated by the proposed extension with different global path planners along with multirobot collision avoidance.

Keywords: Smooth robot paths, smoothing extension, multi-robot collision avoidance, robot navigation.

1. INTRODUCTION

Mobile service robots often need to move from one location to another location to perform tasks like moving an object from one place to another in a warehouse, clean the floor of the room, patrol the area, deliver food and other necessary items to the patients in a hospital, and many others. These robots move autonomously and perceive the external world through the sensors attached to them. These robots often construct a map of the environment and simultaneously localize themselves in it [1]. Given a map, many path planning algorithms have been proposed for mobile robots to navigate to the desired location. Most of these algorithms generate a path for the robot taking the dynamic obstacles of the environment into consideration, and prevents the robot from hitting the obstacles. Path planning is generally carried out in two steps of: (1) global planning, in which, an overall path from start to the goal location is calculated, and (2) local planning, in which, the angular turns are smoothed out. Many algorithms for global robot path planning have been proposed and a complete summary of the various path planning algorithms can be found in [2].

Classifying broadly, most of the global path planners have two potential limitations:

a) Sharp and Angular Turns: Most of the previously proposed algorithms such as $A*$ algorithm [3], $D*$ algorithm [4], shortest path algorithm, probabilistic roadmap planner (PRM) [5], rapidly exploring random tree (RRT) [6], thinning [7], or Voronoi diagrams etc. generate a path which is angular and not desired for robot motion. This is because, a robot cannot take sharp turns abruptly at high velocities, and needs to stop or slow down considerably in order to make these turns.

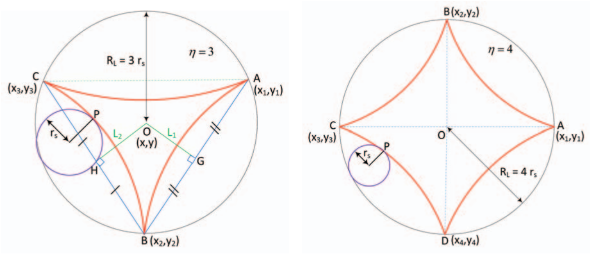
b) Ignore Multi-robot Scenarios: Path planning problem is further complicated in case of multiple robots working in the environment [8]. Multiple robots introduce a probability of colliding with each other, particularly at blind turns. Such situations must also be resolved apart from smoothing the paths of the robots. Most of the path planning techniques either ignore this problem, or treat it as a separate problem. An integrated approach is not found in the literature.

Local planning deals with smoothing out the angular turns of the overall global path. Some of the previously proposed algorithms make use of interpolation [9], B-spline [10], quintic polynomials [11], bezier curves and clothoids [12] extensively. A good summary of various path smoothing techniques can be found in [13].

Some of the major limitation of the previously proposed path smoothing techniques are:

a) Ignore Topological Admissibilities: Most of the previously proposed path smoothers ignore topological admissibilities. In other words, they mainly consider robot as a point entity, and generate a wavy path by applying interpolation or spline smoothing across selected points in the passages. Sometimes, this wavy path comes close to the walls of the straight corridors, which is not desired.

b) Difficult to Integrate with Other Global Planners: Most of these path smoothing techniques have their own ways of generating key-points across which smooth-



(a)

(b)

Fig. 1: Geometry of hypocycloid curves shown in red. (a) Deltoid curve: $R_L = 3 \times r_s$. (b) Asteroid curve: $R_L = 4 \times r_s$.

ing is carried out and have limitations when used in conjunction with other global path planners.

This paper presents a novel path smoothing extension which uses hypocycloid curves [14] to smooth out the sharp and angular turns of the robot's path generating a smooth path for the robot to traverse. This is an extension of our previous work [15] which worked mainly with skeleton paths. The present work is extended, so that, not only skeleton paths but paths generated with any of the global path planners such as $A*$ algorithm [3], $D*$ algorithm [4], shortest path algorithm, probabilistic roadmap planner (PRM) [5], rapidly exploring random tree (RRT) [6], thinning [7], or Voronoi diagrams etc. can also be smoothed. The proposed technique works as an 'extension' and can be used in conjunction with any of the previously proposed path planners. The proposed extension also generates 'nodes' on the robot's path which can be used as points of retreat for the robot to avoid collision with other robots. Unlike other path smoothers which generates a wavy path for the robot bringing them close to walls, the proposed path smoothing extension keeps straight paths of the robot straight, and smooths only the turns.

The new contributions of the paper are summarized below:

1) Works with various global planners: The proposed extension works with any of the previously proposed global path generators. This makes it a good choice for various mobile robot applications.
2) Smooths only the sharp turns: Paths which are straight are not un-necessarily smoothed. This helps to keep the robot to move straight in narrow passages without coming close to any of the walls.
3) Computationally fast to generate: The proposed extension can generate smoothed paths fast and in real-time for most of the mobile robot navigation tasks.
4) Generates nodes for collision avoidance: Apart from smoothing, 'nodes' or locations in map are generated which can be used by the robots as points of retreat to avoid collision.

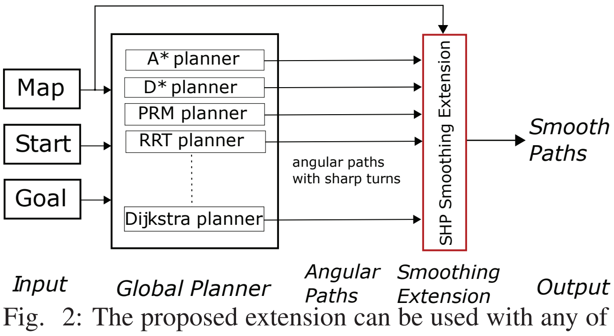


Fig. 2: The proposed extension can be used with any of the global path generators to produce smooth paths

2. A BRIEF INTRODUCTION OF HYPOCYCLOIDS

A hypocycloid is a geometrical curve which is produced by a fixed point P which lies on the circumference of a small circle of radius r_s rolling inside a larger circle of radius $R_L > r_s$ [14]. As shown in Fig.1, the rolling circle has a radius of r_s and the large circle has a radius of R_L, and the curve is defined in general by,

$$\begin{aligned} x(\theta) &= (R_L - r_s)\cos(\theta) + r_s \cdot \cos\left(\frac{R_L - r_s}{r_s}\theta\right) \\ y(\theta) &= (R_L - r_s)\sin(\theta) + r_s \cdot \sin\left(\frac{R_L - r_s}{r_s}\theta\right). \end{aligned} \tag{1}$$

We define η as the number of cusps which are the total number of sharp corners where the curve is not differentiable. Fig.1(a) shows a 3-cusped hypocycloid called as a deltoid or tricupsoid. Fig.1(b) shows a 4-cusped hypocycloid called as an asteroid. An n cusps hyocycloid can be generated by setting the radius of smaller circle is set to $r_s = \frac{R_L}{n}$. This is because n rotations of the smaller circle brings it back to the original position, generating n cusps while traversal [16]. For a deltoid, $R_L = 3 \times r_s$ and for an asteroid, $R_L = 4 \times r_s$.

3. PATH SMOOTHING EXTENSION

As shown in Fig.2, the input to the proposed extension are the map with the start and goal locations. The map can be represented in any form, but should indicate the obstacles and free areas in some form or the other. The start and goal locations are provided as spatial coordinates. This information is given to any of the global planners which generates a path which has angular turns. This path and the map is given as an input to the proposed extension and it smooths out the angular portions of the path generating a smooth path for the robot. The working of the proposed extension is explained below.

3.1 Detecting Sharp Turns in Global Path

Figure 3 shows a typical global path generated by PRM, or D* path planners. These algorithms generate a sequence of n points $P = \{P_1, P_2, \cdots, P_n\}$, where $P_i =$

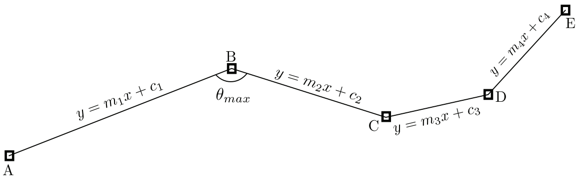


Fig. 3: A global path with sharp turns at B, C, and D.

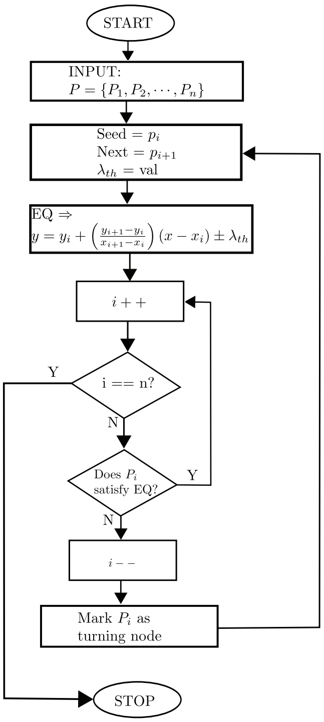

START
INPUT:
$P = \{P_1, P_2, \cdots, P_n\}$
Seed = p_i
Next = p_{i+1}
λ_{th} = val
EQ ⇒
$y = y_i + \left(\frac{y_{i+1}-y_i}{x_{i+1}-x_i}\right)(x - x_i) \pm \lambda_{th}$
$i++$
i == n?
Y
N
Does P_i satisfy EQ?
Y
N
$i--$
Mark P_i as turning node
STOP

Fig. 4: Flowchart of turn-point detection from angular paths.

(x_i, y_i) represents the spatial coordinate in the two dimensional XY-plane. It can be seen that a robot traversing this path from A to E encounters sharp turns at points B, C, and D. Since the proposed extension smooths out the turns, it is necessary to detect these turns beforehand. The entire path from A to E consists of several line segments $\overline{AB}, \overline{BC}, \overline{CD}$, and $\overline{DE}$. Each of the four line segment is represented by,

$$y = m_i \cdot x + c_i,\ i \in \{1, \cdots, 4\}, \tag{2}$$

where m_i and c_i are the slopes and the intercepts of the four lines. Depending upon the accuracy of the global planner employed, if we consider the section of m points P_i between the points A and B ($m < n$), they may or may not accurately lie on the same line $\overline{AB}$. This may arise due to rounding-off errors, or the precision scale (for ex. single precision or double precision floating point) employed. However, considering a correction range $\pm\lambda_{th}$, the m points will lie on the straight line $\overline{AB}$ within this correction range. The equation of line $\overline{AB}$ can be obtained by considering a seed point (starting point) A ($P_i = (x_i, y_i)$), and the next point ($P_2 = (x_{i+1}, y_{i+1})$).

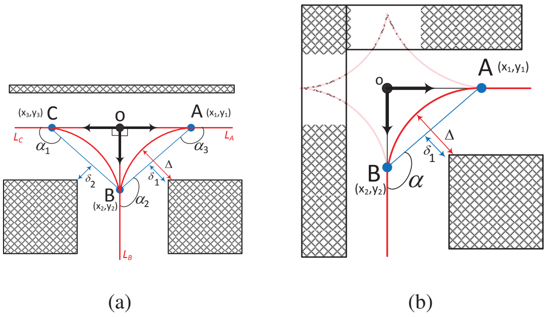


Fig. 5: Smooth path generation. (a) With deltoid, or lower half of asteroid, (b) A portion of hypocycloid to generate curve at corner.

Equation of $\overline{AB}$ is,

$$y = y_i + \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)(x - x_i) \pm \lambda_{th}. \tag{3}$$

Next, we check if the subsequent point P_{i+2} also lies on the same line given by Eq.(3). If P_{i+2} lies on the same line, i is incremented and the next point (P_{i+3}) is checked if it lies on the same line. If a point P_q ($q < n$) does not lie on the same line given by Eq.(3), it indicates the start of a new line segment. It also means that the last point on the previous line segment is P_{q-1} ($q < n$). The seed point is set to P_q, and the next point is set to the point P_{q+1}. This process is continued for all n points, and the turning points are determined. The points of turns are called as 'nodes' in this work. This process is described in the form of a flowchart in Fig.4.

3.2 Smoothing the Angular Paths

Once the nodes are calculated, the process of smoothing the angular paths generated by the global path planners is carried as shown in the flowchart of Fig.6. Each step of the process is explained below:

PROC 1: First, parameters of the smoothing extension are set. This includes obstacle clearance threshold distance (δ), a node diffusion factor (d_f), and counter for total points P_i.

PROC 2,3: The angular path with sharp turns is read and nodes are found as described in Section 3.1. If many nodes are found in vicinity, they are clustered into a single node.

PROC 4: We define diffusion as moving a point along the diverging paths. Each node is incrementally 'diffused' by a factor d_f. This diffusion is controlled by the parameter δ. In Fig.5(a) node 'o' (detected as a turning-node) has been diffused into points A (x_1, y_1), B (x_2, y_2) and C (x_3, y_3). If d represents the distance between the left obstacle and line segment $\overline{BC}$, and right obstacle and

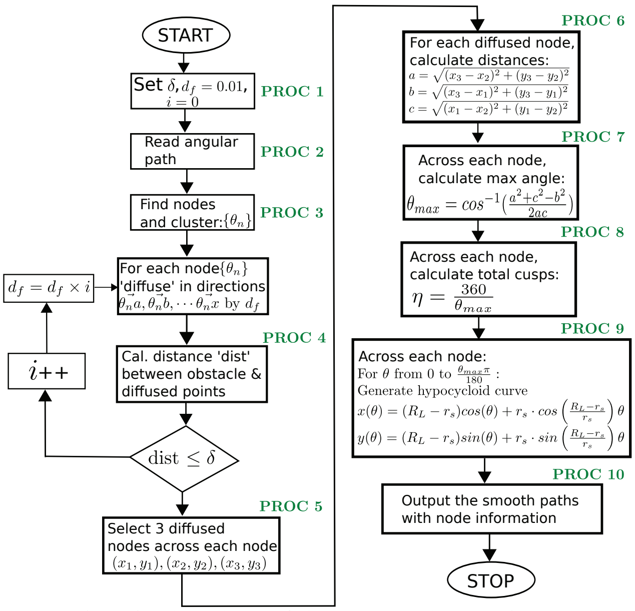

START
Set δ, $d_f = 0.01$, $i = 0$
PROC 1
Read angular path
PROC 2
Find nodes and cluster:$\{\theta_n\}$
PROC 3
For each node$\{\theta_n\}$ 'diffuse' in directions $\vec{\theta_n a}, \vec{\theta_n b}, \cdots \vec{\theta_n x}$ by d_f
$d_f = d_f \times i$
i++
PROC 4
Cal. distance 'dist' between obstacle & diffused points
dist $\leq \delta$
PROC 5
Select 3 diffused nodes across each node $(x_1, y_1), (x_2, y_2), (x_3, y_3)$
PROC 6
For each diffused node, calculate distances:
$a = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$
$b = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}$
$c = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
PROC 7
Across each node, calculate max angle:
$\theta_{max} = cos^{-1}(\frac{a^2+c^2-b^2}{2ac})$
PROC 8
Across each node, calculate total cusps:
$\eta = \frac{360}{\theta_{max}}$
PROC 9
Across each node:
For θ from 0 to $\frac{\theta_{max}\pi}{180}$:
Generate hypocycloid curve
$x(\theta) = (R_L - r_s)cos(\theta) + r_s \cdot cos\left(\frac{R_L - r_s}{r_s}\right)\theta$
$y(\theta) = (R_L - r_s)sin(\theta) + r_s \cdot sin\left(\frac{R_L - r_s}{r_s}\right)\theta$
PROC 10
Output the smooth paths with node information
STOP

Fig. 6: Flowchart of the proposed extension

$\overline{AB}$, diffusion is performed incrementally in very small steps of d_f until $d = \delta$ is satisfied.

PROC 5,6,7: From the diffused points A (x_1, y_1), B (x_2, y_2) and C (x_3, y_3), we find the angle between the line segments θ_{max} using cosine rule as,

$$\theta_{max} = \cos^{-1}\left(\frac{a^2 + c^2 - b^2}{2ac}\right),$$

$$\begin{aligned} &\text{where,} \\ a &= \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}, \\ b &= \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}, \\ c &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \end{aligned} \tag{4}$$

PROC 8,9: The total cusps (η) of hypocycloid are calculated by $\frac{360}{\theta_{max}}$. For each node, the turns are smoothed by generated corresponding hypocycloids by varying θ from 0 to $\frac{\theta_{max}\pi}{180}$ using the equations of hypocycloid (Eq.(1)). For corner sections, as shown in Fig.5(b), there are only two points, and the third (virtual) point is assumed to exist at equidistant location and is used to generate the hypocycloid. In Fig.5, the smooth paths are shown in red color.

PROC 10: The smooth paths and the diffused node location are stored as output of the smoothing extension.

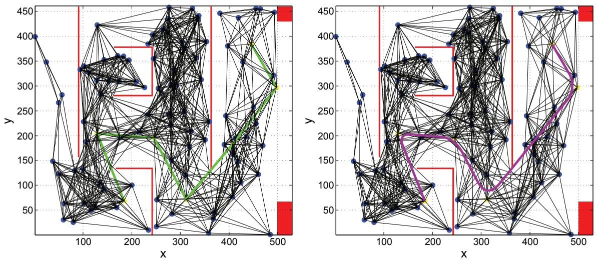
3.3 Using Diffused Nodes to Avoid Collision

The diffused nodes (ex. points A, B, and C in Fig.5(a)) can be used as points of retreat for the robot to avoid collision with other robots moving on the same path. For multi-robot scenarios, once the global path and the smoothed paths have been calculated, the robots start their motion. Mobile robots are generally equipped with laser range sensors, or vision sensors to perceive the environment. In case of intersecting paths, the robots stop at a safe distance from each other, and exchange information comprising of: (a) goal location, and (b) task-priority. A robot with low task priority moves to a non-intersecting node to give way to the other robot. In case, there are no priorities assigned, or if the priorities are equal, retreating robot can randomly be selected or other schemes (ex. prioritizing the robot which has traveled the maximum distance, or has less battery power [8]) can be employed. In case there is no node to retreat, the situation is called as a deadlock. To prevent deadlock, the robot simply retreats to an unoccupied free space in map until the way is cleared. Centers of deltoid or hypocycloids can also be used as retreat points for the robots provided that their area (A_{hypo}[†]) is safely smaller than the area of the robot (A_{rob}).

4. RESULTS

This section presents the results of smoothing using the proposed extension, and using the generated nodes for collision avoidance of multiple robots.

4.1 Smoothing PRM, D*, and Skeleton Angular Paths



(a)

(b)

Fig. 7: PRM path smoothing. (a) Angular path generated by PRM shown in green. (b) Smoothed PRM path shown in purple color.

Fig.7 shows the result of smoothing the PRM generated path. An angular PRM path is shown in green in Fig.7(a) from start location $S_{\{x,y\}} = (180, 70)$ to goal location $G_{\{x,y\}} = (450, 380)$. It is clear that the robot has to take abrupt turns at many points during entire path. Using the

[†] Area of the hypocycloid is given by,

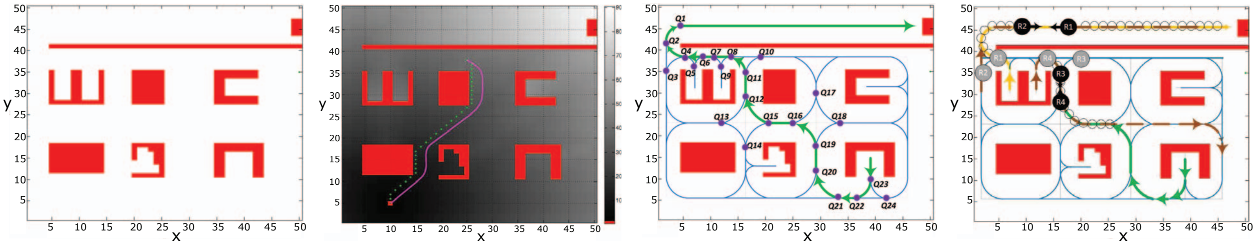
$$A_{hypo} = \int_0^{2\pi} Y(\theta) \left(\frac{\partial}{\partial\theta} X(\theta)\right) d\theta.$$

$\therefore$ Area of deltoid, $A_{del} =$

$$= \int_0^{2\pi} R_L \frac{2\sin(\theta) - \sin(2\theta)}{3} \left(\frac{\partial}{\partial\theta} \frac{R_L(2\cos(\theta) + \cos(2\theta))}{3}\right) d\theta$$

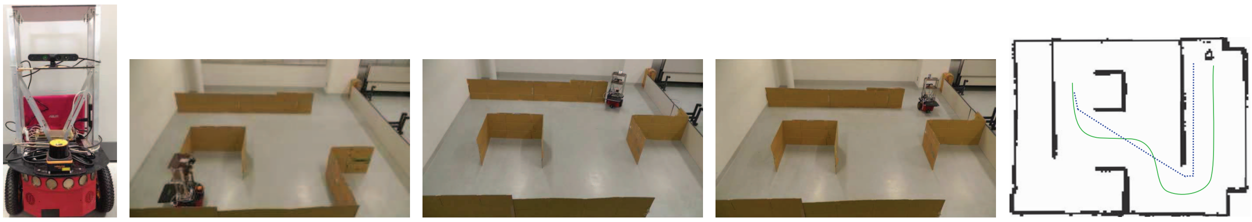
$$= \frac{2}{9}\pi R_L^2.$$

Similarly, area of astroid, $A_{ast} = \frac{3}{8}\pi R_L^2$



(a) (b) (c) (d)

Fig. 8: Results in simulated environment. (a) Environment layout with obstacles in red and free space in white. (b) $D*$ path (in green dots) from start location $S_{\{x,y\}} = (25, 37)$ to goal location $G_{\{x,y\}} = (10, 5)$. Shifted smoothed path in purple color. (c) Smoothed skeleton paths with nodes traversed by robot from from $S_{\{x,y\}} = (38, 15)$ to goal location $G_{\{x,y\}} = (47, 46)$. (d) Multi-robot collision avoidance. Robots in gray show retreat to non-overlapping nodes and communication. Circles indicate overlapping paths.



(a) (b) (c) (d) (e)

Fig. 9: Results in real environment. (a) Pioneer P3DX robot used in experiments. (b) Real environment for tests. (c) D* path brings the robot too close to obstacles and has sharp turn. (c) Shifted and smoothed path removes the sharp turns. (d) Comparison of D* vs smoothed paths. Dotted blue shows D*, whereas solid green shows the smooth path traversed.

proposed extension, the turn points in Fig.7(a) are calculated and they are smoothed by generating hypocycloid curves, as shown in Fig.7(b).

The layout of the environment to test D* path smoothing is shown in Fig.8(a). It shows a complex environment where obstacles are shown in red, and white space represents free space. Fig.8(b) shows the $D*$ path (in green dots) from start location $S_{\{x,y\}} = (25, 37)$ to goal location $G_{\{x,y\}} = (10, 5)$. This path is too close to the obstacles and dangerous for the robot to traverse. It is therefore shifted and then smoothed using the proposed extension. The smoothed path is shown in purple color in Fig.8(b).

Figure 9 shows the results of path smoothing with real robots in real environment using the proposed extension. Figure 9(a) shows the P3DX robot used in the experiment and Fig.9(b) shows the environment. In Fig.9(c), robot was made to traverse an angular D* path. It can be seen in Fig.9(c) that the robot comes close to obstacles and has to take sharp turns. Figure 9(d) shows the paths smoothed with proposed extension and keeps the robot safe from the obstacles. A comparison of D* and smoothed path is shown in Fig.9(e).

4.2 Collision Avoidance on Smoothed Paths

Fig.8(d) shows situations where two robots have common intersecting paths and possibility of collision. It is assumed that the passages are narrow and there is not enough free space for the robots to cross-over each other. Robot $R1$ is on a return course from location $S_{\{x,y\}} = (45, 45)$ to goal location $G_{\{x,y\}} = (7, 30)$ indicated by a yellow path in Fig.8(d). Path of robot $R2$ is indicated in brown from location $S_{\{x,y\}} = (3, 30)$ to goal location $G_{\{x,y\}} = (45, 45)$. $R1$ and $R2$ exchange a tuple which consists of their path information and priorities. The common path is found out by both the robots and is indicated by circles on the path in Fig.8(d). Priority of $R1$ was set to be higher than $R2$ in the simulation. Hence, $R2$ retreats to a safe node Q_3. The nodes are shown in Fig.8(c). $R2$ on its path from start location $S_{\{x,y\}} = (3, 30)$ had cached nodes $Q_1 \cdots Q_6$ (refer to Fig.8(c)). However, since nodes Q_1, Q_2, Q_4 and Q_6 lies on the path of $R1$, $R2$ retreats to the nearest non-overlapping safe node Q_3. $R1$ upon reaching node Q_4, notifies its status to $R2$ and $R2$ continues to its goal. In this situation, if the priorities of $R1$ and $R2$ are the same, $R2$ still retreats because $R1$ would have cached safe nodes Q_3, Q_2 and Q_1, and $R2$ is nearer to both of them with cross-over not possible.

Similarly, $R3$ is configured with a start location of $S_{\{x,y\}} = (13, 30)$ to goal location $G_{\{x,y\}} = (48, 15)$, and $R4$ is configured with a start location of $S_{\{x,y\}} = (40, 15)$ to goal location $G_{\{x,y\}} = (13, 30)$. When $R3$ and $R4$ approach each other, they exchange a tuple with each others configuration from which the common intersecting path is calculated. As show in Fig.8(d) $R3$ has *cached* safe nodes $Q_{9,7,8,10}$, whereas $R4$ has cached many safe nodes (Fig.8(c)). In case of same task priority (or no priority assigned), the ratio of the length of path already traversed to the length of path left to traverse is more for $R4$ than $R3$, hence, $R4$ gets priority. $R3$ retreats to one of the safe nodes Q_{10} which does not lie in the path of $R4$. $R3$ resumes motion once notified by $R4$. In case $R3$ was assigned a higher priority, $R4$ would retreat to the nearest safe node Q_{13}.

A major advantage of the proposed extension is that the diffused nodes can be used by the robots to retreat, and multi-robot collision avoidance is feasible through simple communication. Thus, it eliminates the need for a centralized controller for multiple robots. Since the decision to retreat and detour is taken in real time by local communication, it eliminates the need to recalculate the path of all the robots in case of dynamically changing the goal location (or tasks) of multiple robots. All the paths in the aforementioned exampled could be smoothed using the proposed extension in less than 100 ms, which is good enough for most real-time robot navigation.

5. CONCLUSION

This paper presented a novel extension for smoothing the angular paths of the robots. The proposed extension can be used with any of the previously proposed and widely used global planners like PRM, or A* algorithms. The extension employs the geometry of hypocycloids to smooth the angular turns. Compared to other path smoothing techniques, the proposed extension smooths only the sharp turns, and leaves the straight paths intact, which is advantageous in narrow passages the environment. The proposed extension is computationally efficient and can generate smooth paths in real-time. Apart from generating smooth paths, the proposed extension also generates nodes for the robots to avoid collision with other robots in a multi-robot navigation scenario. Experimental results show that the proposed extension can smooth the paths for the robots. We mathematically discussed the generation of smooth paths and nodes. However, an important characteristic of the proposed extension is that, the users of the extension can think of it as a 'black box' without worrying about the internal workings of the extension, and can still use the extension to smooth the angular paths.

REFERENCES

[1] A. Ravankar, A. A. Ravankar, Y. Kobayashi, and T. Emaru, "On a hopping-points svd and hough transform-based line detection algorithm for robot localization and mapping," *International Journal of Advanced Robotic Systems*, vol. 13:98, 2016.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006. Available at http://planning.cs.uiuc.edu/.

[3] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, pp. 100–107, July 1968.

[4] A. Stentz and I. C. Mellon, "Optimal and efficient path planning for unknown and dynamic environments," *International Journal of Robotics and Automation*, vol. 10, pp. 89–100, 1993.

[5] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, pp. 566–580, Aug 1996.

[6] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," tech. rep., 1998.

[7] C. Arcelli and G. S. Di Baja, "A width-independent fast thinning algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-7, pp. 463–474, July 1985.

[8] A. Ravankar, A. A. Ravankar, Y. Kobayashi, L. Jixin, T. Emaru, and Y. Hoshino, "An intelligent docking station manager for multiple mobile service robots," in *Control, Automation and Systems (ICCAS), 2015 15th International Conference on*, pp. 72–78, Oct 2015.

[9] M. Interpolation, "Wikipedia," 2015. Available at https://en.wikipedia.org/wiki/Interpolation.

[10] K. Komoriya and K. Tanie, "Trajectory design and control of a wheel-type mobile robot using b-spline curve," in *Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings., IEEE/RSJ International Workshop on*, pp. 398–405, Sep 1989.

[11] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto, "Local path planning and motion control for agv in positioning," in *Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings., IEEE/RSJ International Workshop on*, pp. 392–397, Sep 1989.

[12] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *Robotics, IEEE Transactions on*, vol. 20, pp. 1025–1035, Dec 2004.

[13] G. Farin, *Curves and Surfaces for CAGD: A Practical Guide*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 5th ed., 2002.

[14] E. W. Weisstein, "Hypocycloid from mathworld– a wolfram web resource.," 2016. Available at http://mathworld.wolfram.com/Hypocycloid.html.

[15] A. Ravankar, A. A. Ravankar, Y. Kobayashi, and T. Emaru, "Shp: Smooth hypocycloidal paths and decoupled multi-robot path planning with collision avoidance," *International Journal of Advanced Robotic Systems*, vol. 13:133, 2016, DOI:10.5772/63458.

[16] E. H. Lockwood, *Book of Curves*. Cambridge University Press, 1961. Cambridge Books Online.