3D Path Planning: Pruning with Constraint Satisfaction (7A)

Kushagra Khare IMT2015022
Rachit Jain IMT2015034

November 13, 2018

1 RRT* Dubins

We can combined Dubins with RRT* to get the desired results,i.e, a path satisfying non-holonomic constraints.
So took into account these Dubins Path equations into RRT* :-
x = vcos(θ)
$y = v * \sin(\theta)$
$\theta = u$

Here
(x,y) is the position of the car, θ is the direction and u $\in [-\tan\phi, \tan\phi]$

2 RRT* Reeds-Shepp

So now we can combine Reeds-Shepp with RRT* to get the desired results,i.e, a path satisfying all non-holonomic constraints. As discussed earlier the equations for a car for non-holonomic constraints were:-
dx = v*cos(ϕ) $* \cos(\theta)$
$dy = v * \cos(\phi) * \sin(\theta)$
$d\theta = (v/L) * \sin(\phi)$
$distance(p,q) = \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2 + A * min[(p.\theta - q.\theta)^2), (p.\theta - q.\theta + 2\pi)^2, (p.\theta - q.\theta - 2\pi)^2]}$.

Here
L is the length of the car.
A = L*L.

After finding the nearest node (vn) of the randomized configuration (q), we have to find the optimal input(v, ϕ) to get the new node (qn) closest to q. This is a optimization problem, defined as following:-

distance(qn,q) $= \sqrt{(qn.x - q.x)^2 + (qn.y - q.y)^2 + A * min^2[(qn.\theta - q.\theta), (qn.\theta - q.\theta + 2\pi), (qn.\theta - q.\theta - 2\pi)}$
$Where,$
$qn.x = vn.x + v * dt * \cos(\phi) * \sin(vn.\theta)$
$qn.y = vn.y + v * dt * \cos(\phi) * \sin(vn.\theta)$

$qn.\theta = vn.\theta + v * dt * \sin(\phi)$

The above distance metric used is that of Reeds-Shepp algorithm. The combination of RRT* and Reeds-Shepp gives us a smooth, constraint and optimal path in real-time. Here now every time we select a random point we will also select a θ(direction) which will also decide the path to be taken.
But while implementation we faced some issues:-

- Due to very big resolutions screen the algorithm takes a lot of time to converge to a solution and also sometimes not converges to optimal path.
- Also there are a lot of paths to enumerate from to select an optimal path, which adds to the compplexity.

References

[1] Jane Li. Non-holonomic Planning
http://users.wpi.edu/ zli11/teaching/rbe550$_2$017/$slides$/9 − Non − $holonomic$%20$Planning.pdf$

[2] Leszek Podsedkowski (1998) Path Planner for nonholonomic mobile robot with fast replanning procedure
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=681024

[3] Jun Qu. Nonholonomic Mobile Robot Motion Planning
http://msl.cs.uiuc.edu/ lavalle/cs576$_1$999/$projects$/$junqu$/

[4] Bill Triggs. Motion Planning for Nonholonomic Vehicles: An Introduction
https://hal.inria.fr/inria-00548415/document

[5] J. A. Reeds AND L. A. Shepp. Optimal paths for a car that goes both forwards and backwards.
https://pdfs.semanticscholar.org/932e/c495b1d0018fd59dee12a0bf74434fac7af4.pdf

[6] Dubins Path.
https://gieseanw.wordpress.com/2012/10/21/a-comprehensive-step-by-step-tutorial-to-comp