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

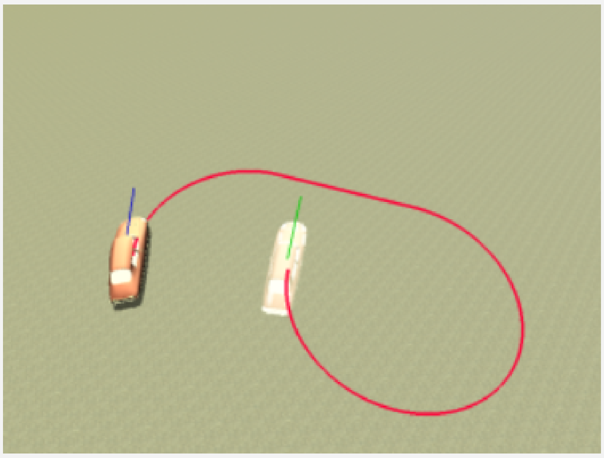
Kushagra Khare
IMT2015022

Rachit Jain
IMT2015034

October 23, 2018

1 Dubins Path

Dubins path is refered to as the shortest curve between two points in the two-dimensional Euclidean plane (i.e. x-y plane) with a constraint on the curvature of the path and with prescribed initial and terminal tangents to the path. Also there is an assumption that the vehicle traveling the path can only travel forward.

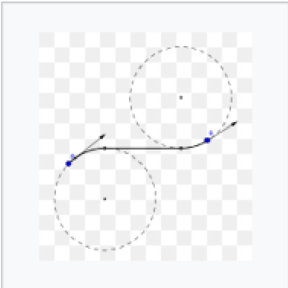


The Dubin's Car only moves forward, never backwards, even if reversal leads to a shorter path.

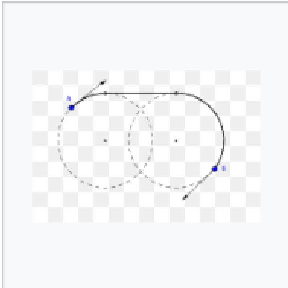
Dubins car:- It is a car that can only move forward, and it always moves at a unit velocity.
So now a Dubins car has only 3 controls,i.e, **max right**, **max left** and **straight**.
So the equations for a car are:-
x = vcos(θ)
$y = v * \sin(\theta)$
$\theta = u$

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

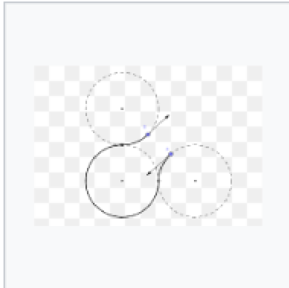
Lester Dubins proved in his paper that there are only 6 combinations of these controls that describe ALL the shortest paths, and they are: RSR, LSL, RSL, LSR, RLR, and LRL.



An RSL Dubins path



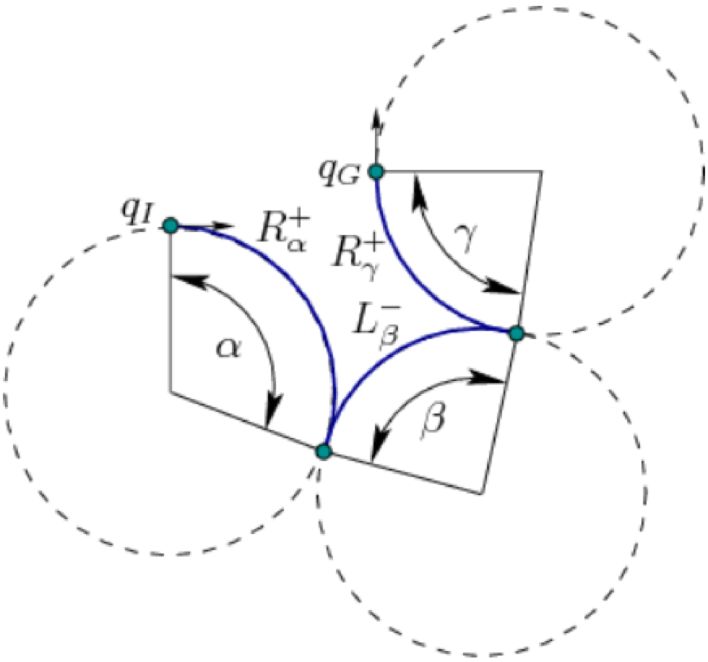
An RSR Dubins path



An LRL Dubins path

2 Reeds-Shepp curves

This is similar to Dubins path with an additional movement control of **moving back**. The shortest path is equivalent to the path that takes minimum time, as for the Dubins car(with reverse movement).



So the equations for a car now are:-
x = u1cos(θ)

$y = u1\sin(\theta)$
$\theta = u1 * u2$

Here:-
u2 $\in [-\tan\phi, \tan\phi]$
$u1 \in \{-1, 1\}$ .u1 is the gear selection ,i.e, u1 = 1 means forward gear and u1 = -1 means reverse gear. There are 48 different combinations of these controls that describe ALL the shortest paths.

Symbol	Gear: u_1	Steering: u_2
S^+	1	0
S^-	-1	0
L^+	1	1
L^-	-1	1
R^+	1	-1
R^-	-1	-1

Figure 15.8: The six motion primitives from which all optimal curves for the Reeds-Shepp car can be constructed.

3 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(θ)
$dy = v * \cos(\phi) * \sin(\theta)$
$d\theta = (v/L) * \sin(\phi)$
$distance(p, q) = \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2 + A * min[(p.\theta - q.\theta)^2), (p.\theta - q.\theta + 2\pi)^2, (p.\theta - q.\theta - 2\pi)^2]}.$

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.

References

[1] Jane Li. Non-holonomic Planning
`http://users.wpi.edu/ zli11/teaching/rbe550`$_2017/slides/9 - Non - holonomic\%20Planning.pdf$

[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`$_1999/projects/junqu/$

[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`