Motion planning:

1) Just looking for shortest paths doesn't work! Math is nasty.

- Distinguish between
  - purely kinematic effects.
    (i.e. make a sequence of configurations that gets from start to end w/o collision)
  - dynamic effects
    (i.e. limits on velocity, angular velocity, control inputs, etc.)

- Dynamics makes planning a lot harder.
  - eg. a car going at 30 ms^{-1} towards a wall 2m away has had it
  - the future matters

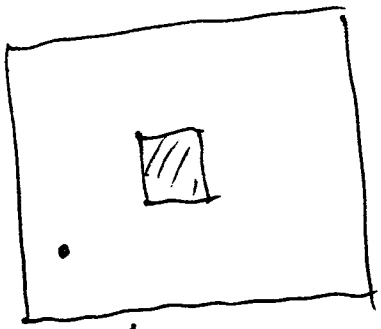
Configuration space:

- a parameter space encoding legal configs of robot (- car!)
- in 2D: $(x, y, \theta) \in SE(2)$
- others may be needed.

~~Obstacle~~

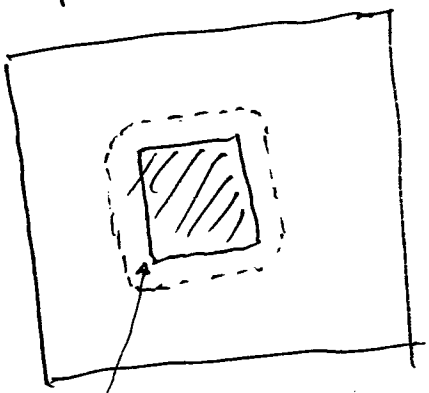
Configuration space obstacles:

- any point in C.S. st. robot intersects some rigid object
- NOT just a plane map.



geometry

" CS for point robot



a spherical robot can't go here, either.

We will ... config space and (6) detect obstacles.

- Simple sampling based planner
  - draw samples in C.S.
    - keep only those that are outside C.S. obstacles.
  - join pairs of samples w/ interpolates
    - keep only those that do not intersect C.S. obstacles

- You now have a graph - Road map

- Planning.
  - join start to some nearest samples (check for intersections)
  - 2 to goal
  - walk graph for good path

several approach

- You could put a lot of work into making a good graph if you're dealing with the space a lot.
- You could precompute + store controls on edges
- rich collection of variants

Issues:

- How to draw samples.
- What to join, to what.
- how to interpolate
- getting to goal.

- pretty much anything might work.
- low discrepancy seqs worth a try.
- require samples to have density property (i.e as $N \to \infty$, every point has a sample increasingly close)
- Nasty spaces can cause problems

Connecting samples:

framework:

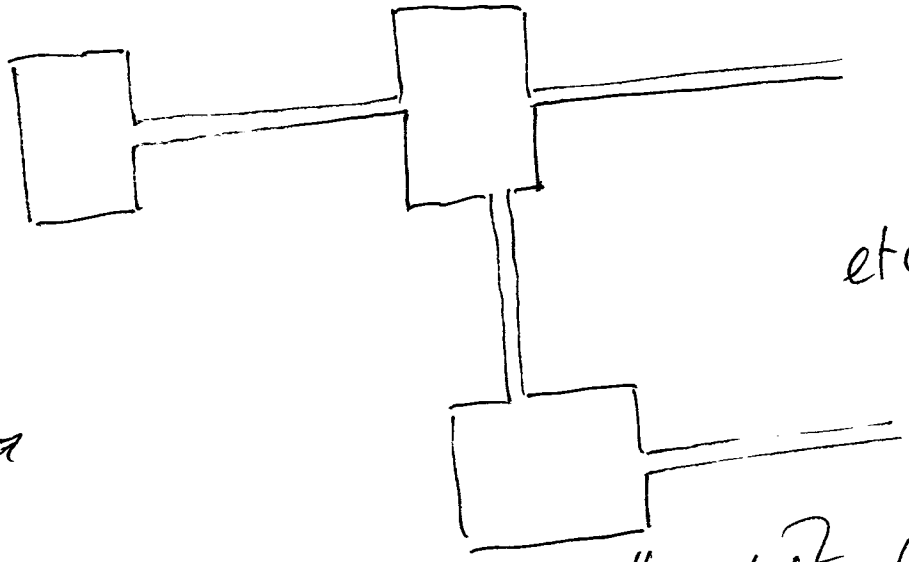
start $G_0 = 0$, S = samples

- pick $\& \alpha \in S$, UAR.
- join α to G_n to get G_{n+1}

Rules for joining α to G

- connect to K closest $v \in G$, (K typically 15; do this first)
  - may have some long edges
- find K nearest pts in each connected component of G (usually $K=1$)
  - notice the 2D intuition that the # of cc's is small is dubious
  - may help with narrow gaps
- radius: take all verts of G in a ball of radius r centered on α.
- visibility (details below)

why connected components



- in this K closest will not get corridors; CC's might.
- recall: a geometry with few narrow obstacles could have many when you take robot geometry into account!

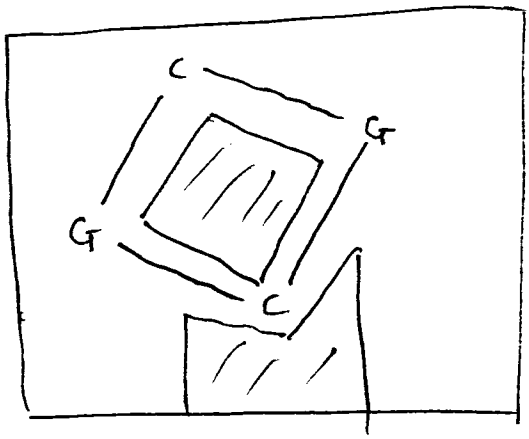
Visibility

- for 2D geometries, visibility between points is easily computed quickly

Vertices are either

guards. - a guard can see no other guard

Connectors - a connector can see at least 2 guards



eg:

Alg.

Draw α

. 3 cases.

- α can see no guards.
  then α is a guard.
  try to attach to G
- α can see two or more guards; try to join as connector
- otherwise, ditch.

• Adv:
  much fewer vertices

• Disadv:
  poor initial choice of guards could lead to poor roadmap

(make many and choose?)

- rapidly expanding Random tree (Lavalle sizes).

What if there are dynamics involved?

we focus on two cases:

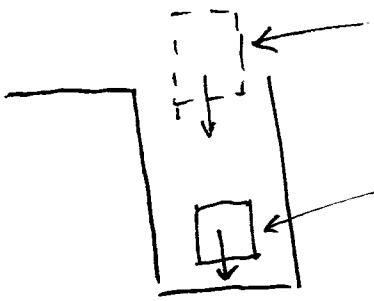
Dubins car

- 2D config.
- $\dot{x} = V\cos\theta$
  $\dot{y} = V\sin\theta$
  $\dot{\theta} = u$ ← $|u|$ is bounded, $V \geq 0$ (can't never)

It can be shown that shortest paths between states are circular arcs joined to line segments.

- Dubins, but can reverse
- shortest paths are still unions of arcs and straight lines
- collection of paths is much richer

(R.S. this is much harder for D.



But it's also stuck here!

D is stuck here

generally a path must be

$\underline{x}(t)$ where

$$\dot{x} = f(x, u)$$

↑ control.

if x contains velocity, etc.

$\Pi_{CS}(x(t))$ = path in C-S.

BUT there are constraints on u,

Sometimes on x

- eg turning angle
- max speed.

S.P. x_s

EP x_e

I seek $x(t)$, $u(t)$ st

$\dot{x} = f(x, u)$

$x(0) = x_s$

$x(t_e) = x_e$

$H(x, u) \leq 0$

and no collision

This is a nasty Boundary Value Problem

- use few/adapted samples, solve BVP
- construct motion primitives, search combos of primitives

- In either case, can build a roadmap
- Goal ~~has~~ won't be on map.
  — BVP.