Lecture 17: "Complete" Planning

AI / graph - search community

Def. complete if guaranteed to find a solution
if one exists

"global optimality" if it finds the optimal plan

1) Decompose non-convexity in problem, then
search all decompositions

2) Randomized motion planning

q_{goal}

Polygons are
P. $\{x \mid Ax \leq b\}$

$x[n+1] = Ax[n] + Bu[n]$

$$\min_u \sum (x[n] - x_{goal})^T Q (x[n] - x_{goal}) + \dots$$

$$\text{s.t. } x[n+1] = \overset{Ax[n]}{\cancel{AE}} + Bu[n]$$

Exterior:

$a_1 x \geq b_1$ OR $a_2 x \geq b_2$ OR $a_3 x \geq b_3$

Disjunctive constraints
w/ integer variables (binary)

$c[n] \in \{0,1\}$

$\sum c_i[n] \overset{\leq m-1}{\cancel{\geq 1}}$

at least one
constraint on

$a_1 x \geq b - c_1[n] M$

$\uparrow$ large number

$c_1[n] = 0$ constraint on

$c_1[n] = 1$ constraint off

nonlinear dynamics

$$x[n+1] = f(x[n], u[n])$$

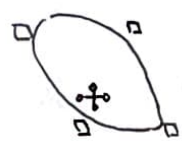
use linear dynamics to bound nonlinear constraints

$$x[n+1] \in \text{convex region around } f(x[n], u[n])$$

e.g., linear approximation bound

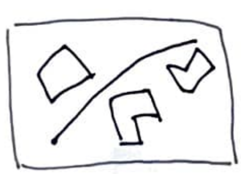
IRIS (Fast approximate convex segmentation)

Find convex region object is inside



ellipsoid approximation

Randomized motion planning

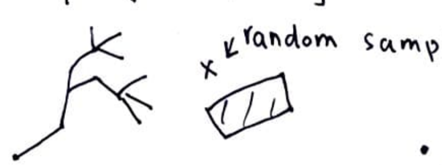


kinematics

collision checker

"sample-based planner"

RRT (Rapidly-exploring Random Tree)

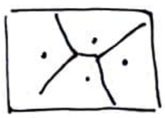


1) Random sample

2) find closest point on current tree

3) grow it towards sample point

RRT has a Voronoi bias

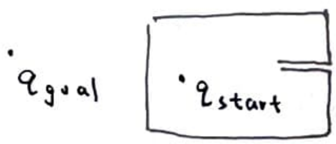


boundary has equal distances to two points

prob $\propto$ |region of voronoi|

↑ to select a point

compared to ~~na~~ grid search



narrow passage makes grid search not complete

reject collision config

→ uniform distribution in free space

→ $t \to \infty$ uniform distribution over free space

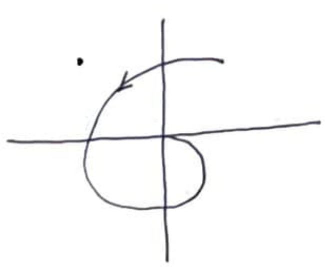
→ probabilistic complete

on humannoids James Knuffer sleak

reject not quasi-static configurations

pendulumn dynamics example

sample ~~number~~ number very high



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

Euclidean distance does not work

Kino dynamics Planners

(Control- based planners

OMPL

1) Better sampling distributions

2) Better distance metric

3) Better extend (e.g., trajectory optimization ?)

Approximate
"Distance" metrics

Min time to go (boundary value problem)

don't satisfies distance metric

cost to go

LQR RRT (quick cost calculation)

unconstrainted dynamics → Value-iteration finds optimal policy

use cost to go to guide RRT search

Reachability - Guided RRT

Sample configs likely to reachable

Build two trees

For every node in main tree

grow natural candidates

↳ reachability tree

closer to reachability tree ✓

closer to original tree X

RG - RRT

sample main RRT tree

extend each node to several reachable nodes

new sample

closer to reachable nodes - tree ✓

closer to main tree ✗ reject

biased samples towards reachable regions

RG - RRT

changed Voronoi bias

only from Voronoi regions for which differential constraints

permit the expansion of the node towards sample

after the ~~closed~~ closest point is identified X^r_{new} ← reachable

it is added to main tree + ~~reachalle~~ reachable points