LQR-Trees [Tedrake, 2010]

George Kontoudis, Shriya Shah

ME5984 Motion Planning Analysis
Spring 2017

Mechanical Engineering Department, Virginia Tech

April 16, 2017

Outline

Motivation

- Design robust algorithms for non-linear feedback motion planning
- Non-linear underactuted systems such as robot manipulator or bipedal walking
- Computation of planning regions of attraction (funnels) for non-linear underactuated dynamical systems
- Applicable to real robots

Definition of Lyapunov Functions

For a given dynamical system

$$\dot{x} = f(x), f(0) = 0$$

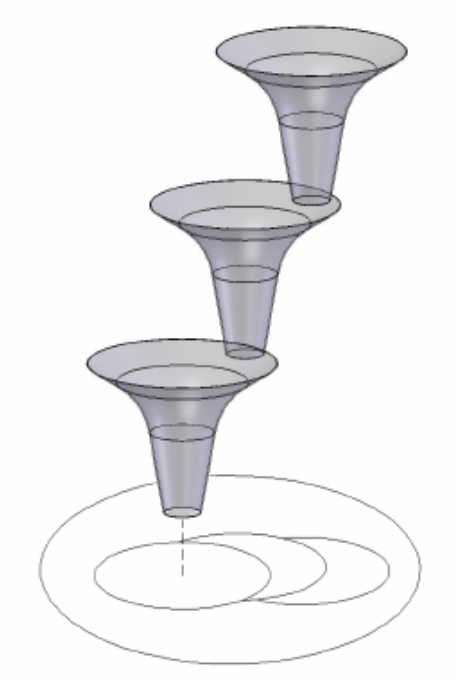
a Lyapunov function is $V(x)$, $V \in C$ where

- $V(x) > 0$, positive definite
- $\dot{V}(x) = \frac{dV}{dx}\frac{dx}{dt} < 0$, negative definite

If conditions met in some state space ball B_r, then origin is a.s.

Sequential Composition of Lyapunov Functions

- Each funnel acts like a valid Lyapunov function
- A.s. of each Lyapunov falls in the region of attraction of the next lower level
- The lowest function stabilizes in the goal point



Sequential composition of funnels [Burridge, 1999]

Sums of Squares

We want to check inequalities and validate Lyapunov functions using sums-of-squares (SoS) method [Parrilo, 2000]

- $x^4 + 2x^3 + 3x^2 - 2x + 2 \geq 0, \quad \forall x \in \mathbb{R}$, by employing SoS

$$x^4+2x^3+3x^2-2x+2 = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^T \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} = X^T A X$$

- Eigenvalues of A are $\lambda_1 = 3.88$, $\lambda_2 = 1.65$, $\lambda_1 = 0.47$, so the inequality stands $\forall x \in \mathbb{R}$

Sums of Squares Properties

General structure of (SoS) for a 4-*th* order polynomial is

$$fx^4 + 2ex^3 + (d + 2c)x^2 + 2bx + a = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$$

- Extend to multivariable polynomials
- Check non-negativity by searching positive semidefinite matrix

Feedback Synthesis by SoS Optimization

Given a system $\dot{x} = f(x) + g(x)u$ we want to generate

- Feedback control law $u = \pi(x)$
- Lyapunov fcn $V(x)$, s.t. $V(x) > 0$, $\dot{V}(x) = \frac{\partial V}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial V}{\partial x}\dot{x} < 0$

BUT, this is a difficult problem as the set of $V(x)$, $\pi(x)$ may not be convex sets

- Rely on LQR synthesis
- Design a series of locally-valid controllers
- Compose these controllers utilizing feedback motion planning

The Minimum Principle

The first order or necessary condition for optimality is called *Maximum (Minimum) Principle*

- Given a function we want to minimize $f(x, y, z)$ on a level surface (constraint) $g(x, y, z)$ we get

$$\nabla f = \lambda \nabla g$$

- To convert a constrained problem to an unconstrained we construct the Hamiltonian function

$$H(x, u, t, \lambda) = L(x, u, t) + \lambda^{\mathsf{T}} f(x, u, t)$$

Goal Stabilization

- For a given non-linear dynamical system

$$\dot{x} = f(x, u, t), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m$$

- Set goal state x_G, where $f(x_G, u_G) = 0$, and $\bar{x} = x - x_G$, $\bar{u} = u - u_G$
- Linearize around (x_G, u_G), $\dot{\bar{x}} \approx A\bar{x}(t) + B\bar{u}(t)$
- Infinite horizon LQR minimum energy cost-to-go fcn (performance index)

$$J_\infty = \frac{1}{2}\int_0^\infty [\bar{x}^\mathsf{T}(t)Q\bar{x}(t) + \bar{u}^\mathsf{T}(t)R\bar{u}(t)]dt,$$

$$Q = Q^\mathsf{T} \geq 0, R = R^\mathsf{T} > 0$$

Hamiltonian System

Set the Hamiltonian

$$H(x, u, t) = L(x, u, t) + \lambda^\intercal f(x, u, t)$$

- State equation

$$\dot{x} = \frac{\partial H}{\partial \lambda}$$

- Costate equation

$$-\dot{\lambda} = \frac{\partial H}{\partial x} = \frac{\partial f^\intercal}{\partial x}\lambda + \frac{\partial L}{\partial x}$$

- Stationarity condition

$$0 = \frac{\partial H}{\partial u} = \frac{\partial f^\intercal}{\partial u}\lambda + \frac{\partial L}{\partial u}$$

Riccati equation and Optimal Control Law

- Infinite horizon LQR problem results the optimal cost

$$J^*(\bar{x}) = \frac{1}{2}\bar{x}^\intercal S \bar{x}$$

- $S > 0$, w/ the solution given from ARE

$$0 = Q - SBR^{-1}B^\intercal S + SA + A^\intercal S$$

- Optimal feedback closed loop control policy

$$\bar{u}^* = -R^{-1}B^\intercal S\bar{x} = -Kx$$

Goal State Convergence

The domain of attraction of the LQR over some sub-level set

$$B_G(\rho) = \{x | 0 \leq V(x) \leq \rho\}$$

To guarantee a.s. we require $V(x)$ to be a valid Lyapunov function

- $V(x) > 0, \quad x \in B_G(\rho)$
- $\dot{V}(x) < 0, \quad x \in B_G(\rho)$

Assign $V(x) = J^*(\bar{x}) = \frac{1}{2}\bar{x}^\intercal S \bar{x}$

- By definition positive definite
- $\dot{V}(x) = \dot{J}(\bar{x}) = \frac{dV}{dx}\frac{dx}{dt} = 2\bar{x}^\intercal S \dot{x} = 2\bar{x}^\intercal S f(x_G + \bar{x}, u_G - K\bar{x})$

Lyapunov Verification Using SoS

We require

$$\dot{J}^*(\bar{x}) < 0, \quad \forall \bar{x} \neq 0 \in B_G(\rho), \quad \dot{J}^*(0) = 0$$

- First, modify the inequality from negative to non-positive

$$\dot{J}^*(\bar{x}) \leq -\epsilon ||\bar{x}||_2^2, \quad \forall \bar{x} \in B_G(\rho), \quad \epsilon \in \mathbb{R}^+$$

- Second, include the constraint with Lagrange multiplier $h(\cdot)$

$$\dot{J}^*(\bar{x}) + h(\bar{x})(\rho - J^*(\bar{x})) \leq -\epsilon ||\bar{x}||_2^2$$

Lagrange Mulitplier Searching

- If $f^{(cl)}(x,u) = f(x, u_G - K(x - x_g))$, search for $h(\cdot)$ polynomial with sufficient order for $\dot{J}^*(\bar{x})$, using SoS

$$\begin{aligned} \text{find} \quad & h(\bar{x}) \\ \text{subject to} \quad & \dot{J}^*(\bar{x}) + h(\bar{x})(\rho - J^*(\bar{x})) \leq -\epsilon ||\bar{x}||_2^2 \\ & h(\bar{x}) \geq 0 \end{aligned}$$

- If $f^{(cl)}(x) \approx \hat{f}^{(cl)}(\bar{x})$, where $\hat{f}^{(cl)}(\bar{x})$ is the Taylor expansion (algebraic approximation) and $\hat{\dot{J}}(\bar{x}) = 2\bar{x}^\intercal S \hat{f}^{(cl)}(\bar{x})$

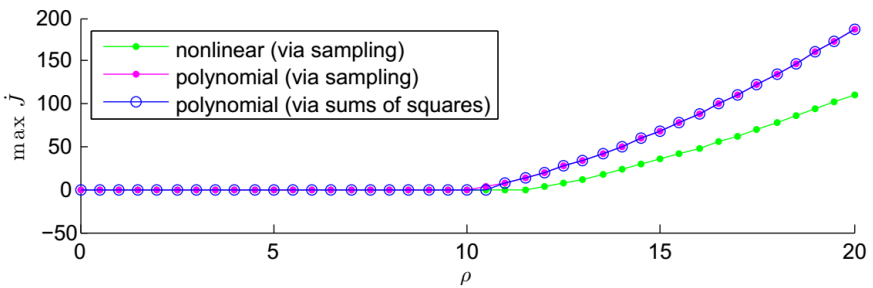
$$\begin{aligned} \text{find} \quad & h(\bar{x}) \\ \text{subject to} \quad & \hat{\dot{J}}(\bar{x}) + h(\bar{x})(\rho - \hat{J}^*(\bar{x})) \leq -\epsilon ||\bar{x}||_2^2 \\ & h(\bar{x}) \geq 0 \end{aligned}$$

Optimization for ρ

Set a convex optimization problem for the region of attraction

$$\begin{aligned} \max \quad & \rho \\ \text{subject to} \quad & \dot{\hat{J}}^*(\bar{x}) + h(\bar{x})(\rho - \hat{J}^*(\bar{x})) \leq -\epsilon||\bar{x}||_2^2 \\ & h(\bar{x}) \geq 0 \\ & \rho > 0 \end{aligned}$$

- At each step the Lagrange multiplier searching is performed
- If the program is feasible ρ increased



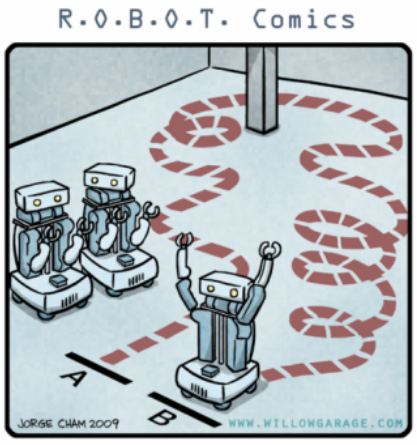
Polynomial verification of damped single pendulum [Tedrake, 2010]

Trajectory Optimization

Time-Invariant LQR Verification

What is Motion Planning?

- Can a robot go from point A to point B?
- How should a robot go from point A to point B?
- What is the optimal path?



Motion Planning explained!
[Burridge, 1999]

Problem Formulation

The Motion planning problem can be stated as follows. Given:

- A start pose X_S of the Robot.
- A desired goal pose X_G of the Robot.
- The geometric description of the Robot.
- The geometric description of the World.

The motion path planning probelm is to find a path that moves the robot from the start position to the goal position gradually while avoiding the obstacles.

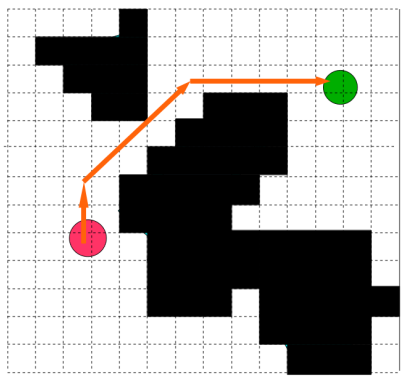
Configuration space

- The Motion planning is performed in the configuration space of the robot.
- Configuration is a complete specification of every point of the robot.
- Configuration space is the space of all possible configurations.

A very simple example

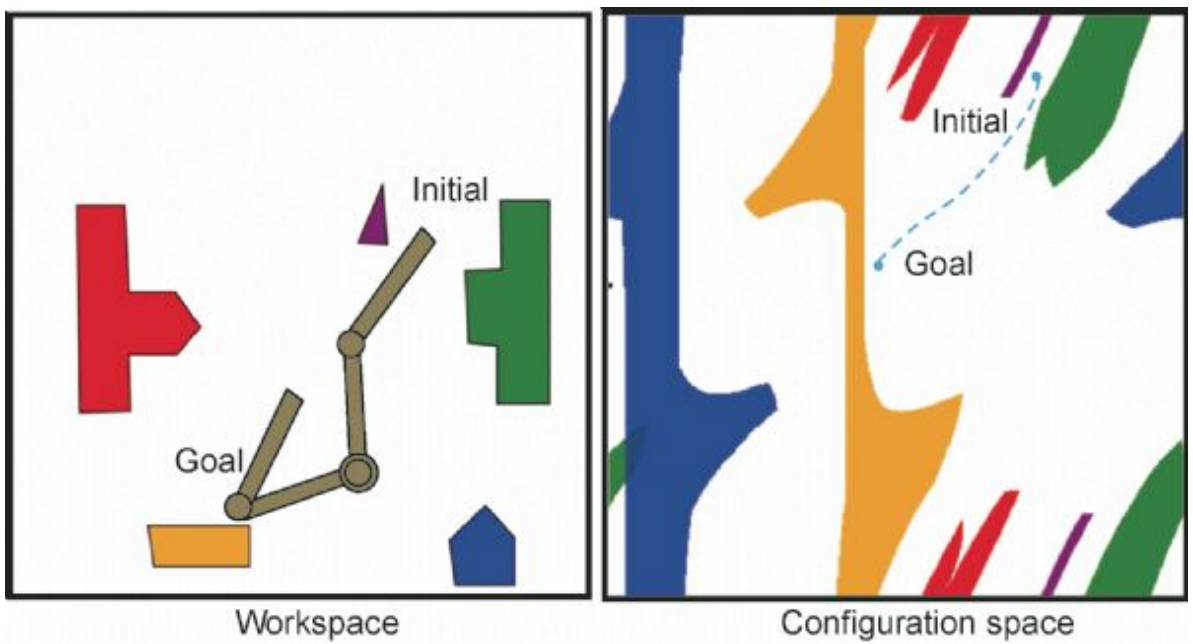
A point Robot that can translate in 2D. Such a Robot can live in a grid world (C-Space) as shown below.

- The robot can move to adjacent grids.
- There is perfect sensing and localization.



Motion Planning on grid World

A manipulator

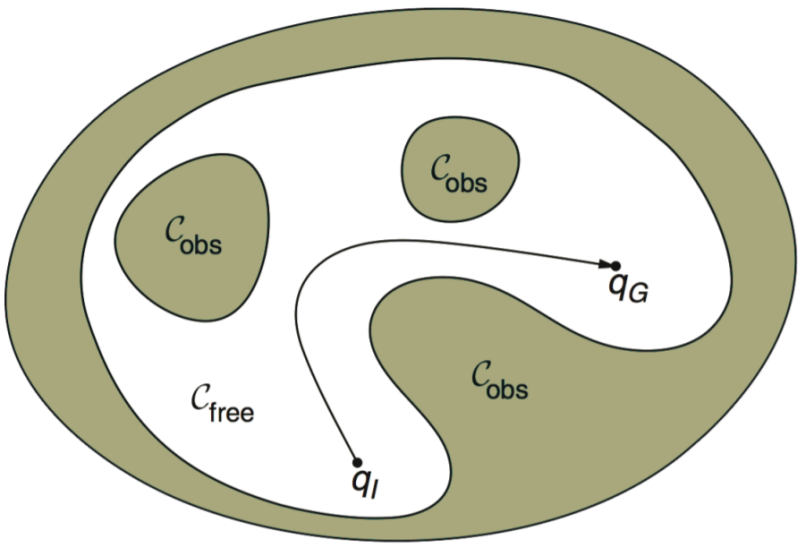


The workspace and configuration space of a 2 link manipulator

Planning Algorithms

Once we have a c-Space, the motion planning problem now is to find a contionous path

$\tau : [0, 1]$ in C_{free}
With, $\tau(0) = q_I$, $\tau(1) = g_G$



Planning in C-Space

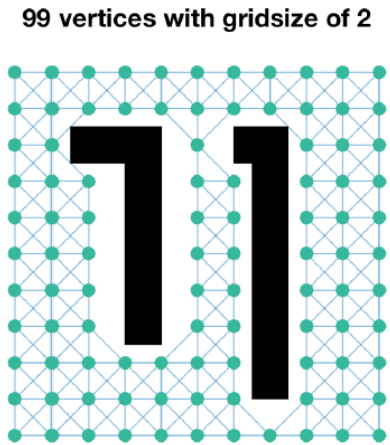
C-Space Discretizations

- In order to implement path planning algoritms, the continous space needs to be discretized.
- There are 2 approaches to discretizing c-space:
  - Naive discretization
  - Combinatorial planning
  - Sampling based planning

Naive Discretizations

Discretize the grid in *nxn* regions for a 2D grid.

- Grows exponentially therefore does not scale well.
- Resolution is user dependent.



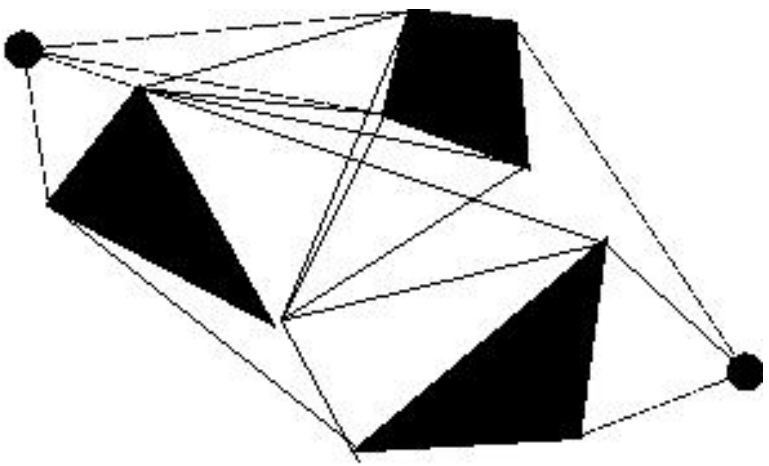
Planning in C-Space

Outline

- These techniques capture the connectivity of the C_{free} into a graph and finds the solution using search.
- It produces a graph where the vertex is a configuration in C_{free} and each edge is a collision free path through the free space.
- There are 4 major techniques
  - Visibility graph
  - Voronoi diagram
  - Exact cell decomposition
  - Approximate cell decomposition

Visibility Graph

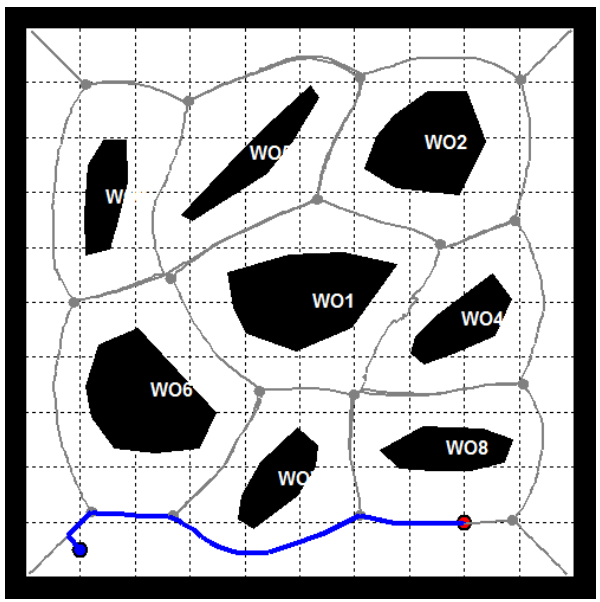
- The path is made of lines connecting the start and goal pose though the vertices of the obstacles in the C-Space.
- Run Time: $O(n^2 \log n)$



Visibility Graph

Voronoi Diagram

- Find the place with the same maximal clearance from the nearest obstacles.
- Run Time: $O(n \log n)$



Conclusion

- These techniqies are complete i.e. they will find a path if there exists one and report failure otherwise.
- Becomes intractable as the C-Space dimensionality increases.

Outline

- The algorithm assumes no prior knowledge of the free region in the C-space.
- Use a collision detection algorithm that probes the space to see whether a given configuration is achievable or not.
- There are 2 major techniques
  - Probabilistic road maps (PRM)
  - Rapidly exploring random trees (RRT)

Rapidly exploring random trees

- The algorithm assumes no prior knowledge of the free region in the C-space.
- Use a collision detection algorithm that probes the space to see whether a given configuration is achievable or not.

Algorithm

- Design a function that produces a sequence of discrete samples x_i in the C-space.
- Check if x_i is in the free region of the configuration space.
- If x_i is in the free space then add it to the graph and connect it to the existing vertices in some way.
- If x_i is not in the free space then discard it
- Keep repeating till we find a path to the goal.

Algorithm

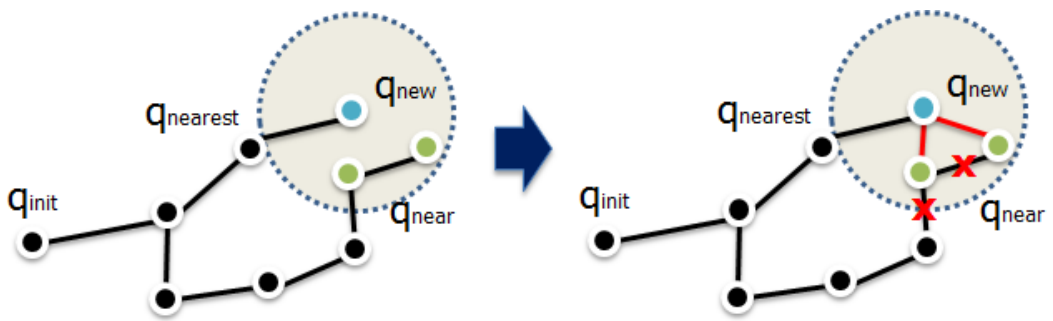
Starting at the start state, randomly sample a point in the C-space within a radius r_0

Algorithm

- Find the closest vertex in the graph G using a distance function

$$\rho = CxC \text{ in } [0, \text{inf})$$

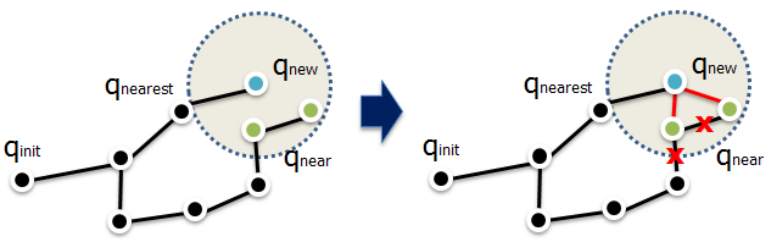
- There are several strategies to find the nearest neighbor like taking the closest vertex or check intermediate points and split edge at nearest vertex.

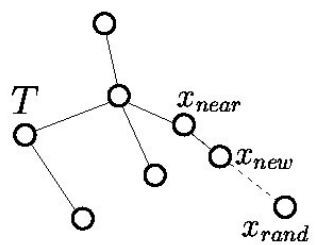


Algorithm

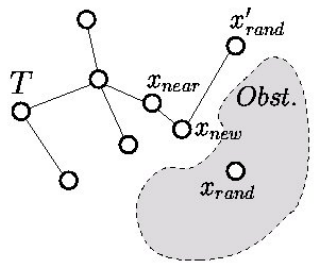
Connect the nearest point with the random point using a local planner that connects q_{new} with q_{init} while making sure that:

- There is no collision : Add edge
- Collision: Find new vertex q_i as close as possbile to the obstacle.





(Basic RRT)

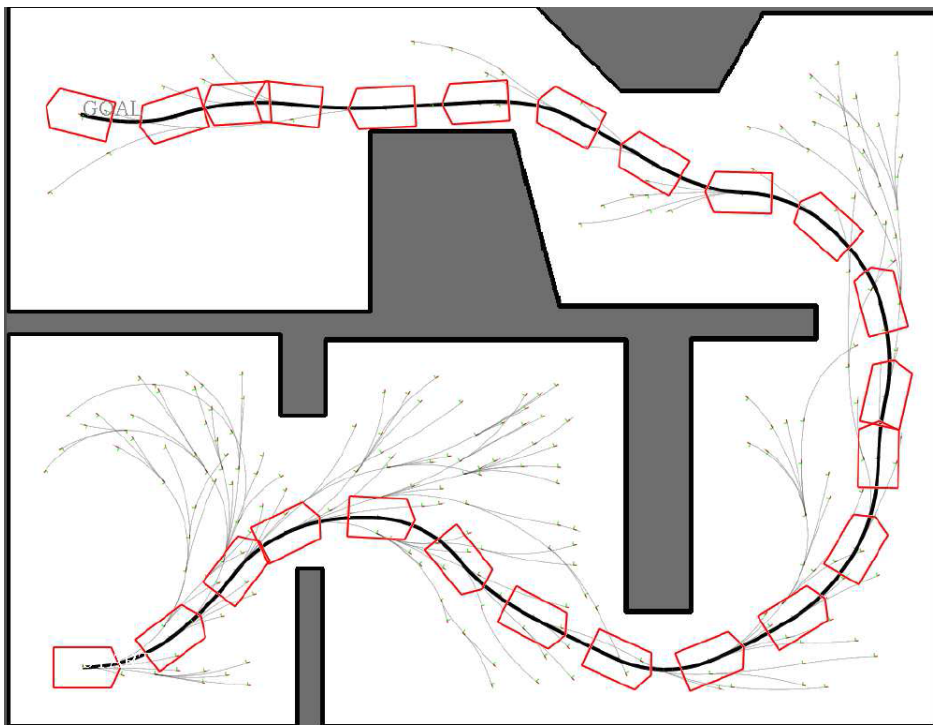


(OBRRT)

Practical considerations

- At every N iterations, pick the next random sample to be the goal and see it the goal is reachable.
- Grow two RRTs, one from q_s and one from q_G.
- PROS:
  - Blalance between greedy search and exploration
  - easy to implement
- CONS:
  - Sensitive to distance metric
  - Only probabilistically complete
  - Not optimal

RRT



https://www.youtube.com/watch?v=E-IUAL-D9SY

Conclusions

References

R. Tedrake, I. Manchester, M. Tobenkin, and J. Roberts

LQR-trees: Feedback motion planning via sums-of-squares verification

International Journal of Robotics Researchs, 1038–1052, SAGE Publications, 2010.

R. Burridge, A. Rizzi, and D. Koditschek

Sequential composition of dynamically dexterous robot behaviors

International Journal of Robotics Researchs, 534–555, SAGE Publications, 1999.

P.Parrilo

Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization

Ph.D. Theis, MIT, 2000.

Thank You!