Sampling-based Motion planning algorithm to handle a narrow passage problem

Dissertation Defense

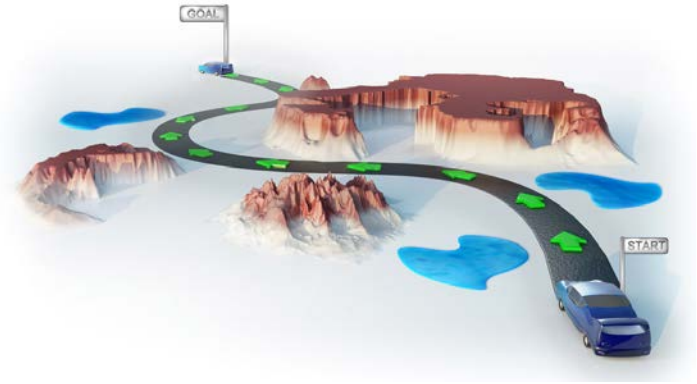
이 정 환 (Junghwan Lee)
2014-05-12

Contents

- Introduction
- Goal
- Motion planning algorithm for handling narrow passages
  - Narrow passage detection
  - Sampling-bias for kinematic constraint
  - Database-based extension for kinodynamic constraint
- Summary

Robot motion planning

- Given obstacles, a robot, and its motion capabilities, compute robot motions satisfying constraints from the start and goal
- Applications:
  - Autonomous vehicle
  - Industrial assembly problem
  - Protein folding



Images from Kavraki Lab, Rice Univ., and CSAIL, MIT

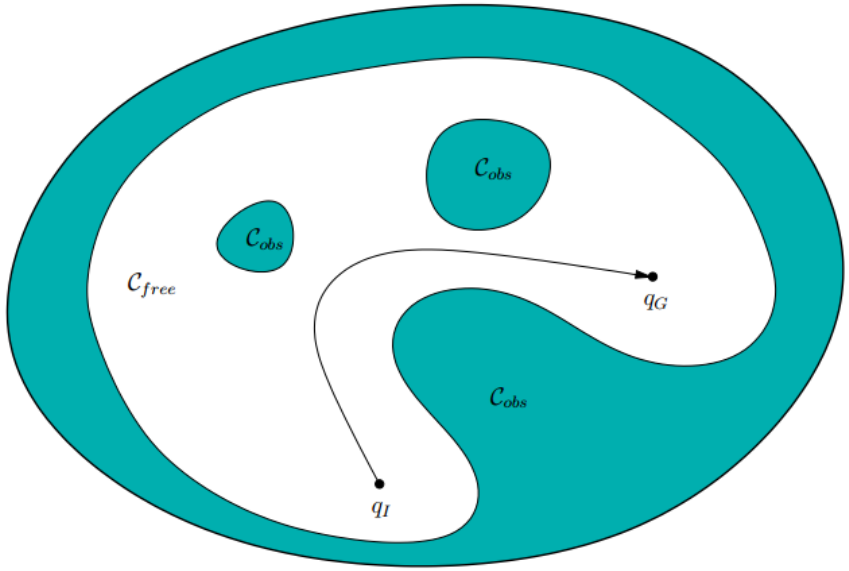
Configuration space

- The ***configuration space*** is the set of all allowable robot ***transformations***

- a Free-flying robot in 2D space
  - x, y: positions of the center of robot
  - θ: orientation of a robot
- a Robot Manipulator with revolute joints
  - x, y, z: positions of the base
  - $\theta_1, \ldots, \theta_i$: orientations of joints

Robot motion planning

- Given robot and obstacle models, C-space C, and $q_I, q_G \in C_{free}$
- Compute a path from q_I to q_G in C_{free} satisfying given constraints



Types of Constraints

- Geometric constraint:
  - Not colliding with obstacles
- Kinematic constraint:
  - When a robot is not rigid
  - e.g. robot manipulator consisting of multiple bodies with joints
- Kinodynamic constraint:
  - Non-holonomic and differential constraints
  - Underactuated robot model
    - Movement is constrained by a control
  - Interacting with dynamic systems
  - e.g. Most of robots in real world

Sampling-based algorithms

- PRM [Kavraki et al. 96], RRT [LaValle et al. 98]

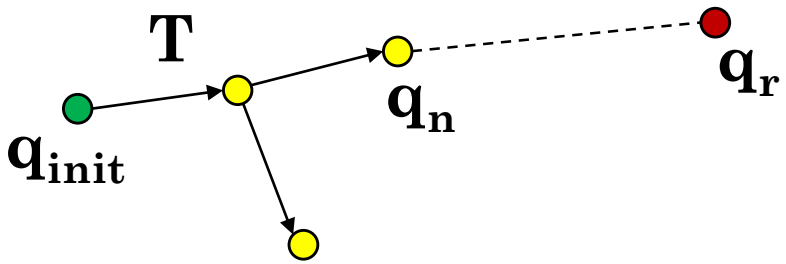
- Incrementally sample and search the free space
- Randomized planning
- Probabilistically complete:
  - If a solution exists, the ***probability*** that it ***finds a solution*** tends to ***one*** as ***the number of iterations*** tends to ***infinity***
- Practically solve high dimensional problem (around 10 dofs)

RRT [LaValle et al. 98]

Given initial, goal and tree T in C-space

1. Randomly sample a point
2. Select the nearest neighbor node from the tree
3. Construct a path between two points
4. Collision check for a path in geometric space
   - If free, add edge and vertex
5. Repeat until solution is found

Local planning



RRT [LaValle et al. 98]

- Example

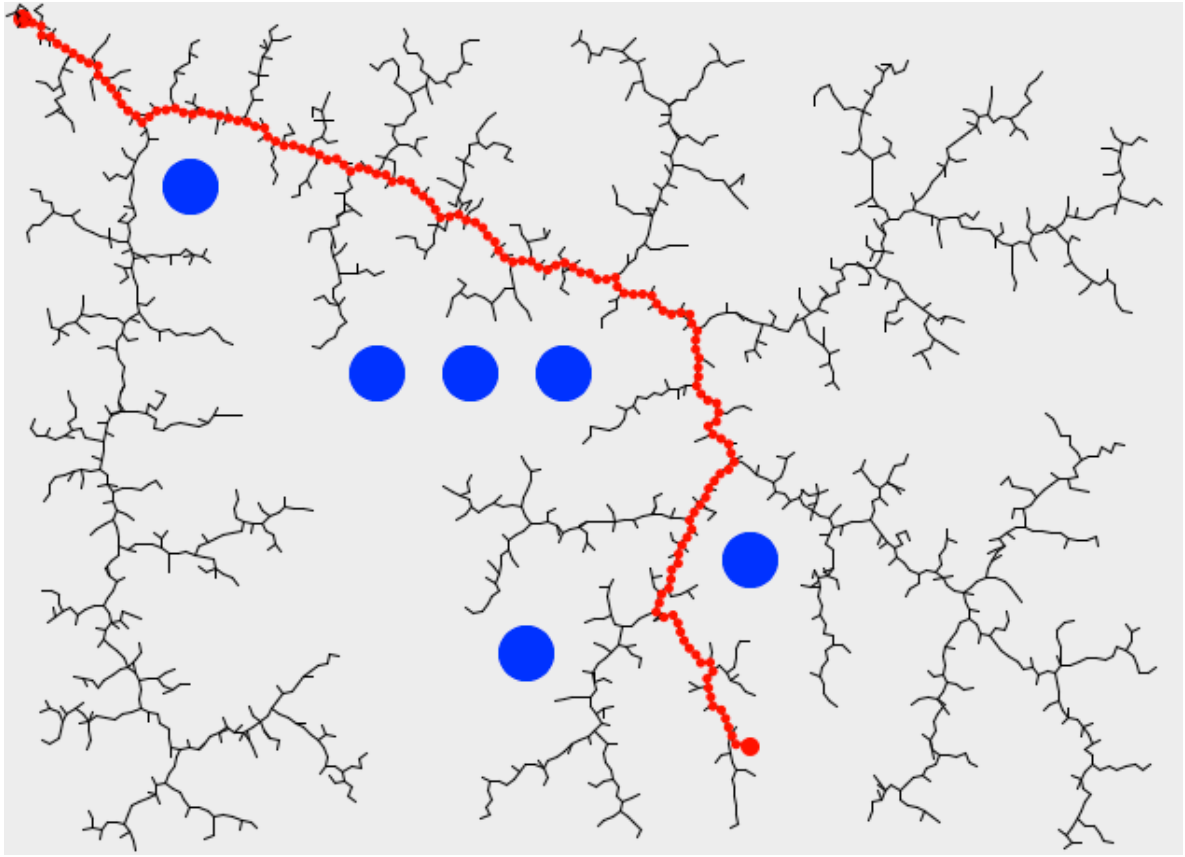
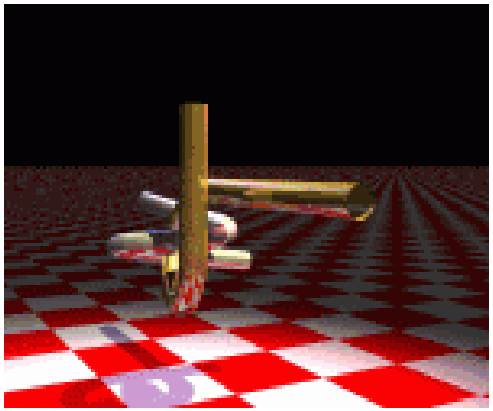
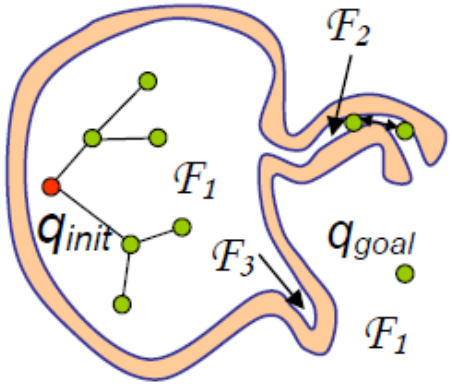


Image from nakkaya.com

Narrow passage problem

- Typical challenges of sampling-based algorithm





- When a robot should go through narrow passages to reach a goal state
- Narrowness is defined in the configuration space

Narrow passage problem

- Becomes more challenging as more constraints are related
  - *Geometric constraint*
  - *Kinematic constraint*
    - Increase the dimension of the problem
  - *Kinodynamic constraint*
    - Dimension increase almost two-fold
      - Velocity components are added
    - Local planning becomes complex and time-consuming

Sampling-based motion planning algorithms to handle a narrow passage problem

Propose sampling-based planning algorithms to efficiently solve the narrow passage problem for robots with various types of constraints.

Sampling-based motion planning algorithms to handle a narrow passage problem

- *Narrow passage detection algorithm*
  - Perform selective retractions
- *Productive region-oriented biased sampling algorithm*
  - Biased sampling for a robot manipulator
- *Motion database-based extension algorithm*
  - Efficient local planning under kinodynamic constraints

I. Narrow passage detection algorithm

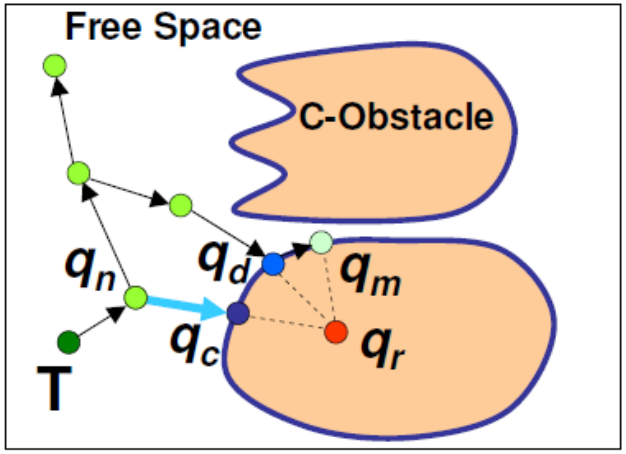
Selective retraction based planning

Related work

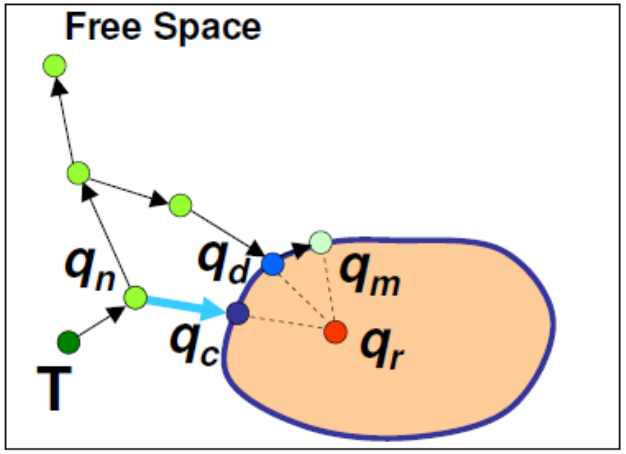
- Filtering technique
  - Adaptive sampling by filtering out some samples
  - Gaussian PRM [Boor et al. 99], Visibility PRM [Simeon et at. 00], Ball Tree [Shkolnik & Tedrake 11], etc.

- Retraction-based technique
  - Retracts in-collision samples towards more useful regions
  - OBRRT [Rodriguez et al. 06], RRRT [Zhang & Manocha 08], etc.

RRRT: pros. and cons.

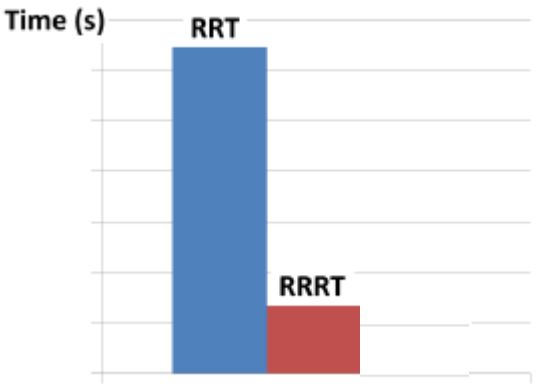


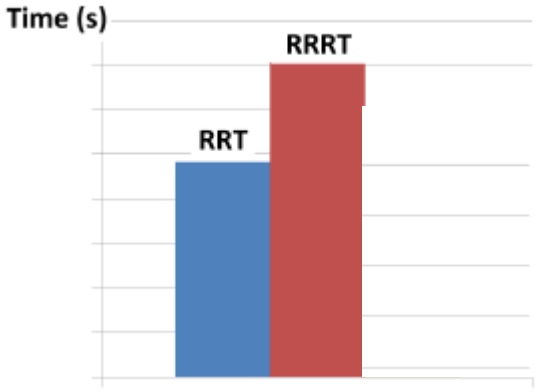
with narrow passages



without narrow passages

images from [Zhang & Manocha 08]





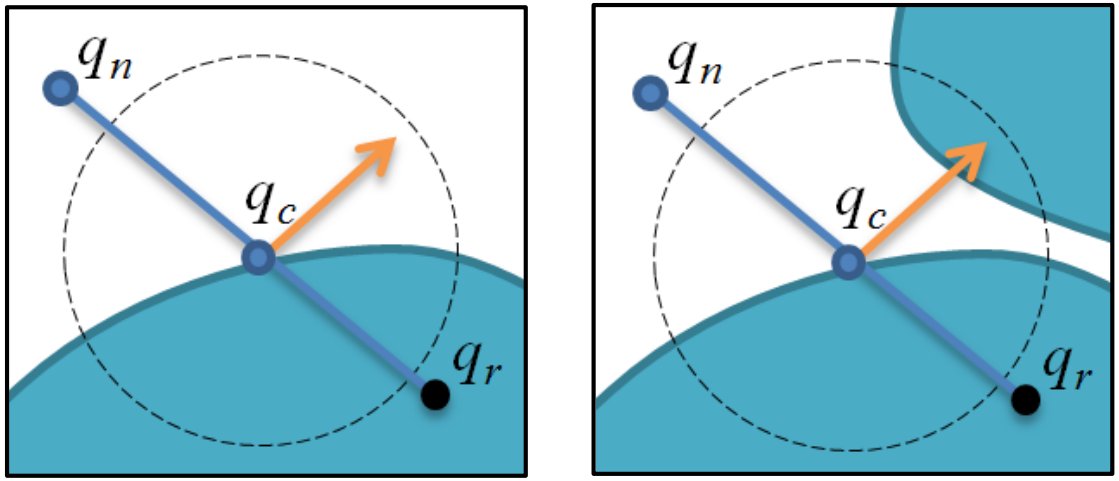
Motivation

- Cons of RRRT:
  Excessive sampling on the boundary of obstacles
  - Computational overhead
  - Not helpful for the problem without narrow passages

- If a planner can ***identify narrow passages***,
  it can ***selectively perform*** retraction operations
  only on narrow passages

Bridge line-test

- To identify narrow passage regions and determine whether retraction operations will be performed
- Bridge line-test
  1. Generate a random line
  2. Check whether the line meets any obstacle



Bridge line-test: completeness

- It can fail with some probability
  to identify a narrow passage region (false negative)

- ***Re-test*** Bridge line-test at a node
  where the tree expansion is stuck during iteration
  - in-contact node selected as nearest neighbor of sample

- The accuracy of Bridge line-test become
  probabilistically ensured after multiple re-testing

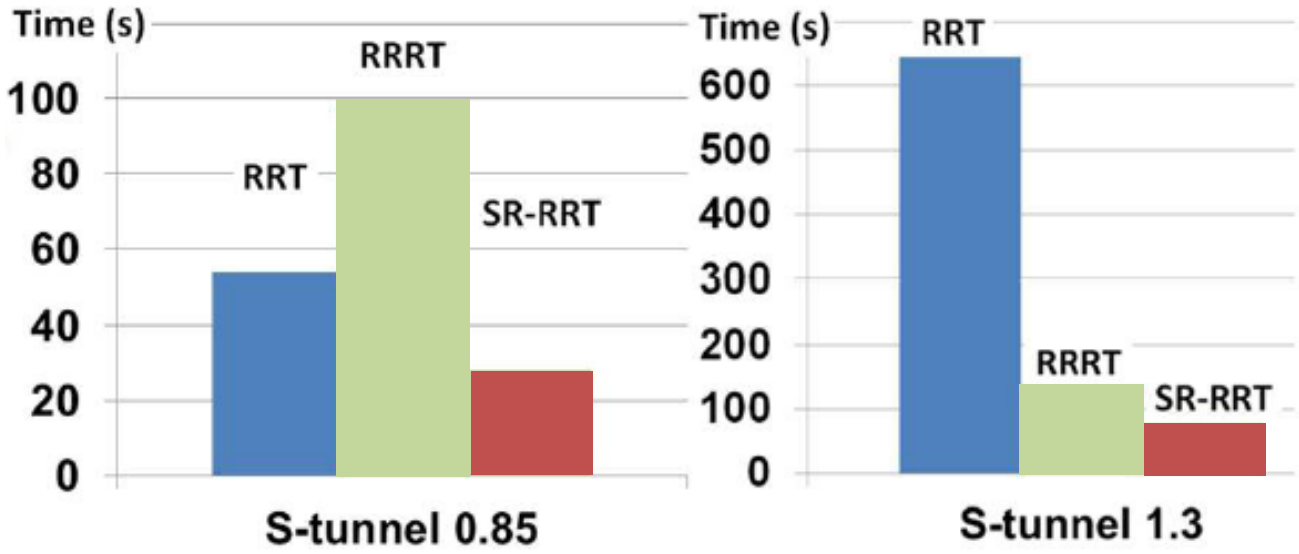
SR-RRT

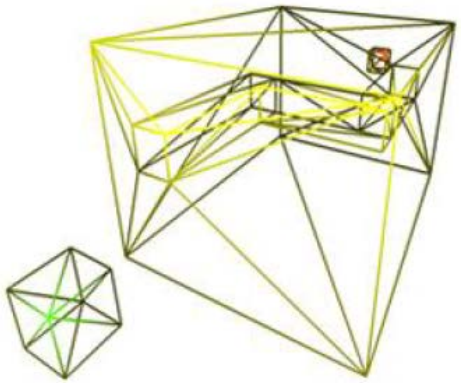
- **S**elective **R**etraction-based **RRT** Planner
- For a free-flying rigid robot
  in 3D workspace including narrow passages



- Presented in IEEE International Conference on Robotics and Automation (ICRA) 2012 &
  IEEE Transactions on Robotics 2014

Results


Time (s)
100
80
60
40
20
0
RRT
RRRT
SR-RRT
S-tunnel 0.85
Time (s)
600
500
400
300
200
100
0
RRT
RRRT
SR-RRT
S-tunnel 1.3

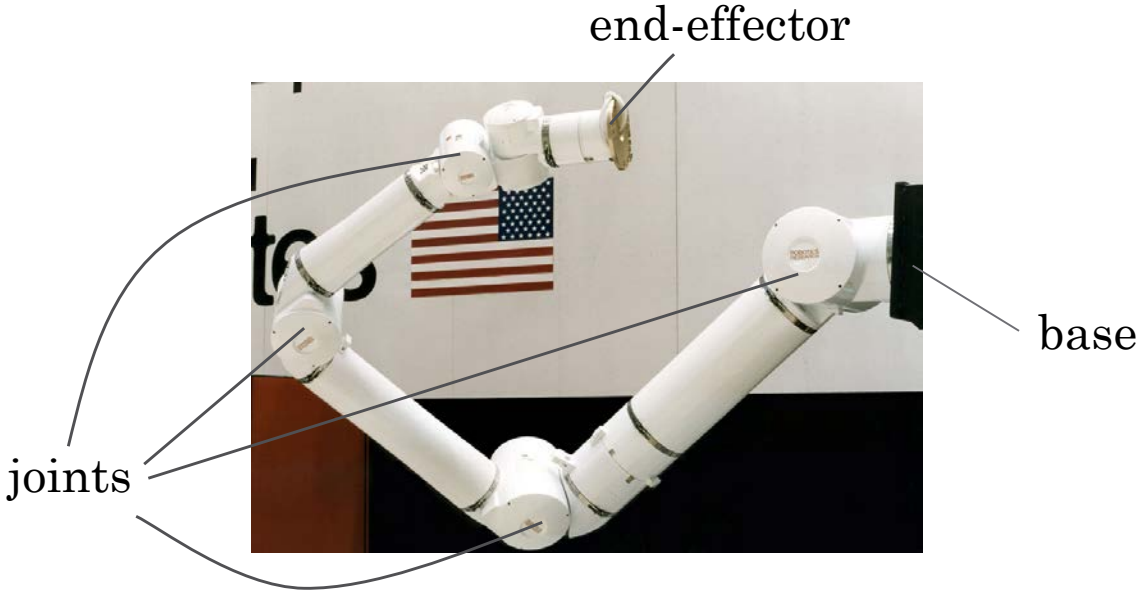


II. Biased sampling for a robot manipulator

Productive region oriented method

Robot manipulator

- Consists of multiple bodies by a kinematic chain

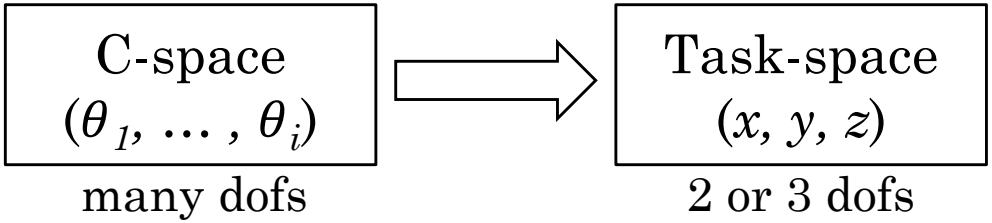


- C-space
  - x, y, z: positions of the base
  - θ_1, … , θ_i: orientations of joints

Task-space Planning

[Shkolnik and Tedrake 09], [Behnisch et al. 10]

- To free from "the curse of dimensionality"
- Dimension reduction technique



- Perform RRT in task-space

  Given initial, goal and tree T
  1. Random sampling
  2. Select the nearest neighbor
  3. Local planning ***in C-space***
  4. Repeat

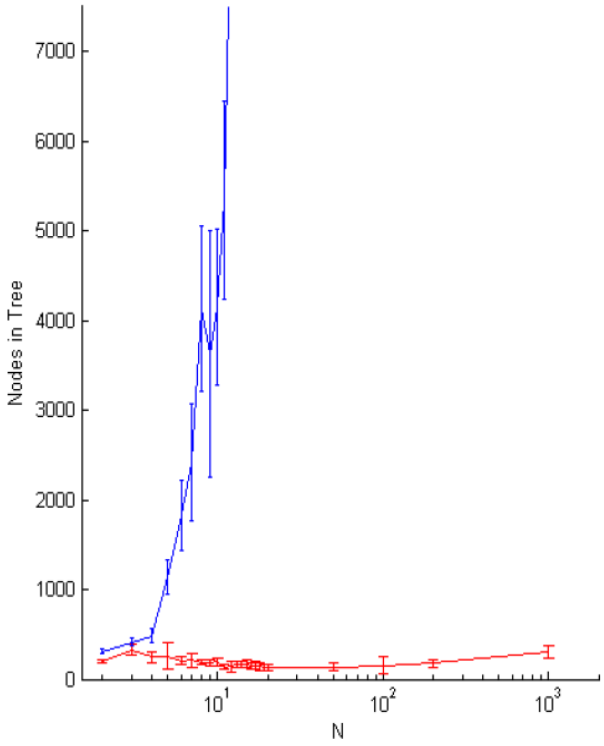
Task-space Planning

[Shkolnik and Tedrake 09], [Behnisch et al. 10]

- Use ***inverse kinematics solution*** to recover an original space from reduced space
  - e.g. *Jacobian pseudoinverse* method

- Effectively solve high dimensional problems

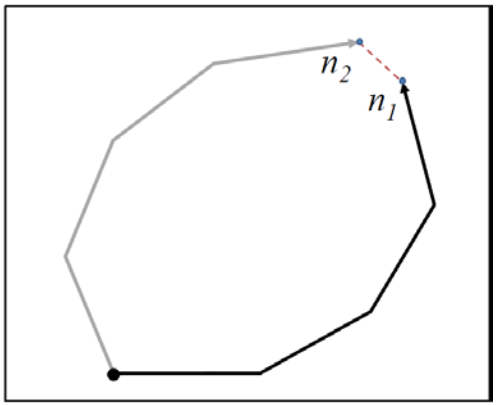
- RRT variations can be used in T-space

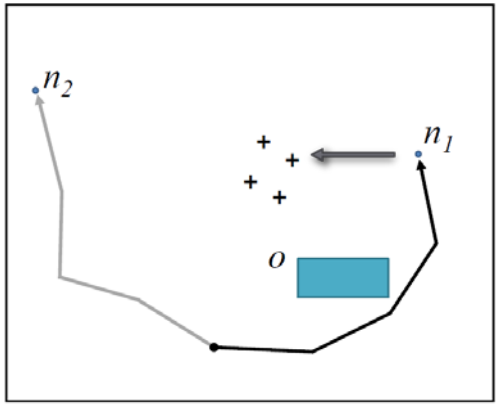


Motivation

Given initial, goal and tree T
1. Random sampling
2. Select the nearest neighbor
3. Local planning
4. Repeat

- Nearness in T-space doesn't reflect nearness in C-space

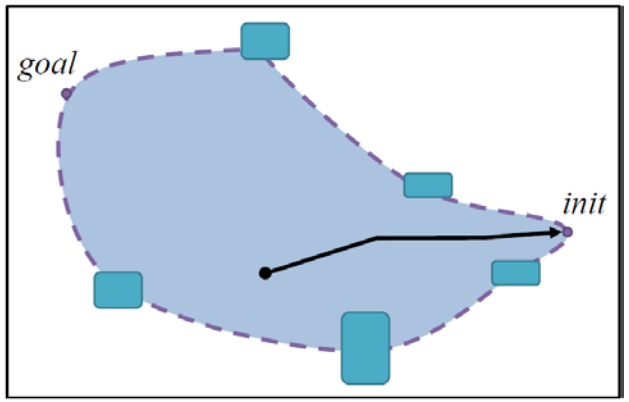




- Affects convergence rate of planner
  when an obstacle restricts the movement of a robot
- For T-space planning, convergence rate is very important
  - Local planner for T-space takes much more time
  - Because of extra computation: Jacobian computation

Productive region-oriented task space planning

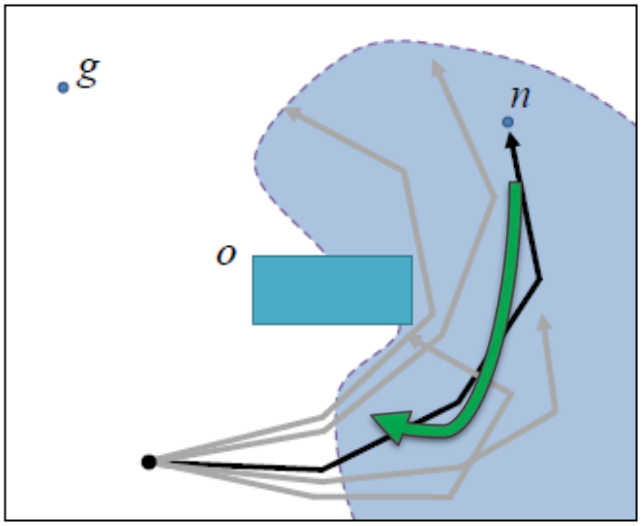
- Productive region
  - A set of T-space states which have a high probability of leading a robot to a goal



- ***Bias sampling into productive region*** will improve the overall performance of the planner
- Exact computation is as hard as motion planning problem

Maximum Reachable Area (MRA)

- Defined for each task space node in the tree
- A set of states where a robot can reach by using an employed local planner



PROT

- **P**roductive **R**egions **O**riented **T**ask-space path planner
- For a robot manipulator with kinematic constraints

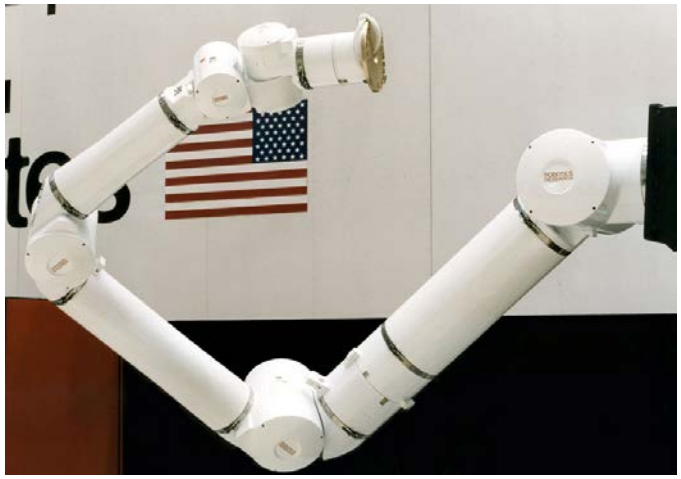
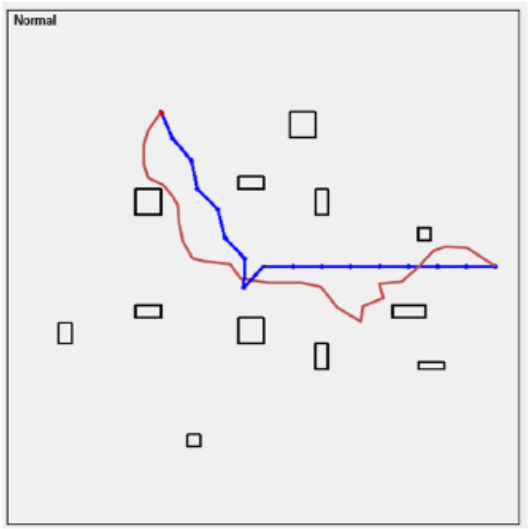


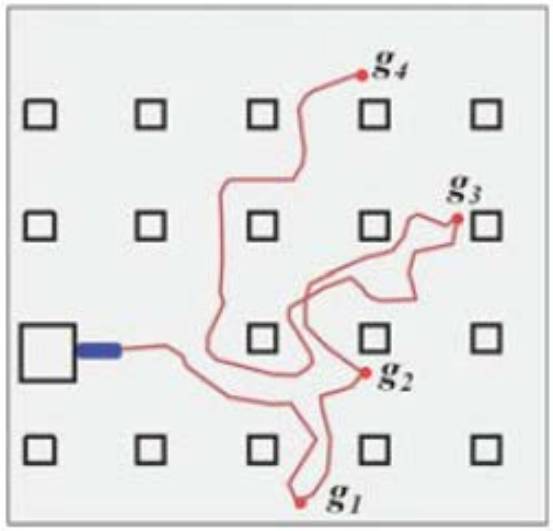
Image from roboticsresearch.com

- To appear in IEEE International Conference on Robotics and Automation (ICRA) 2014

Experimental results



8-dofs	RRT	TS-RRT	PROT
# of iterations	-	850.8	358.1
# of nodes	>448270	340.3	128.9
time (s)	>300	10.91	5.29
20-dofs	RRT	TS-RRT	PROT
# of iterations	-	34227.3	10746.3
# of nodes	-	14240.9	3534.1
time (s)	-	66.22	23.02



	TS-RRT	PROT
# of iterations	6563.9	2752.8
# of nodes	1300.5	527.3
time (s)	250.46	69.24
Success ratio	35%	80%

Sampling-based motion planning algorithms to handle a narrow passage problem

- Thesis proposal (Dec. 2013)
  - *Narrow passage detection algorithm*
    - Perform selective retractions
  - *Productive region-oriented biased sampling algorithm*
    - Biased sampling for a robot manipulator

- ***Motion database-based extension algorithm***
  - Efficient local planning under kinodynamic constraints

III. Motion database-based extension algorithm

For kinodynamic planning problem
with complex dynamics

Motion planning with kinodynamic constraints

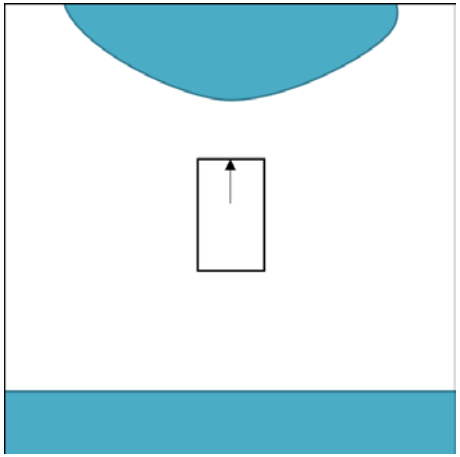
- Planning for real robotic systems requires not only *geometric constraints* but also *kinodynamic constraints*

- Kinodynamic constraint:
  - Non-holonomic & differential constraints
  - Underactuated robot model
    - Movement is constrained by a control
  - Interacting with dynamic systems

Motion planning with kinodynamic constraints

- Much more challenging in narrow passage problem
  - Dimension increase almost two-fold (C-space → state space)
    - Velocity components are added
  - Local planning becomes complex and time-consuming

- A path becomes more narrow in C-space, because the movement is constrained by restricted controls



Motion planning with kinodynamic constraints

- Theoretically, sampling-based method works well, if ***proper motion equations are given***

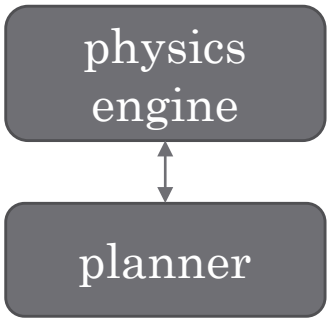
- With modified local planner:
  - Finding a control which extends the robot toward a sampled state by integration of equations

Given initial, goal and tree T
1. Random sampling
2. Select the nearest neighbor
3. ***Local planning***
4. Repeat

- Major weakness:
  - ***Lack of generality and accuracy***
  - Designing accurate equations is restrictive

Integrated with physics-based simulator

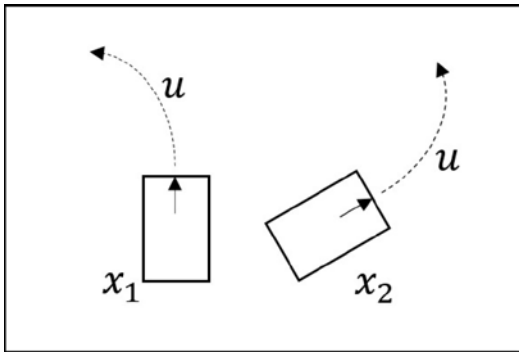
- Using ***physics-based simulation engine*** as a black-box local planner
  - Instead of motion equation specification
- Pros.
  - Exact derivation is not required
- Cons.
  - More computationally expensive than integration of equations



- Benefits outweigh the overheads with several heuristics:
  - [Kostas et al. 08], [Plaku et al. 10], [Sucan & Kavraki 12]

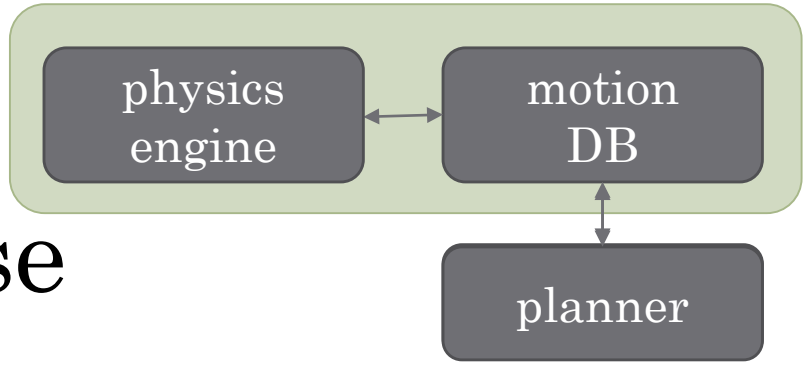
Motivation

- Executing simulations takes most of running time
  - Worse as the complexity of the environments
- Simulation queries which are ***distinct*** in state space could be ***identical*** in robot's local coordinate space.



- Lots of duplicated simulations → degrade performance

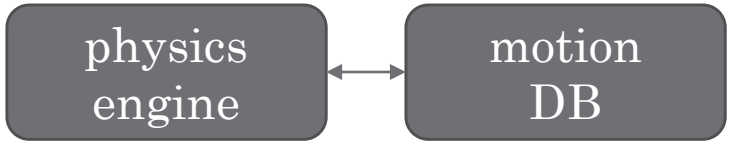
Planning with Motion database



- Pre-compute several motions and reuse in run-time

- Building motion database in simulation state space
  - Projected space into robot's local coordinate space
  - Lower dimensional space than the state space

- Expected benefits:
  - Eliminate duplications of simulation
  - Replace expensive simulation queries with data retrieval from the database
  - Reusable for different setting

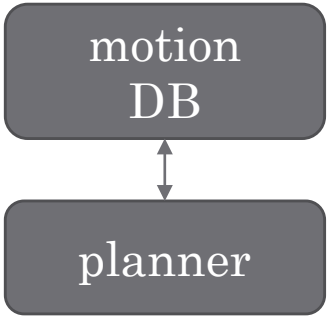


Building motion database

- Simulation state space
  - Lower dimensional space than state space
  - Consisting of components independent to geometric configuration
  - e.g. a rigid mobile robot
  - linear/angular velocities in local coordinate
- Motion generation
  - By executing simulator from given start state
  - Stores swept volume in geometric space and end state
- Conversion between two space is done by geometric transformations

Planning with the database

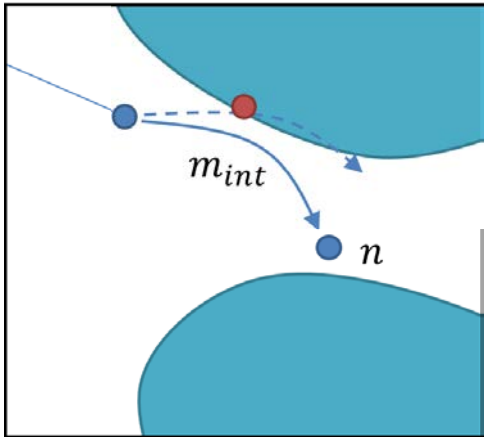
- Local planning module is replaced with:
  - Retrieve pre-computed motions
    - Converts given query to simulation state space
    - Find a motion by epsilon-NN search
    - Choose a motion by motion interpolation
  - Check collisions using swept volume of motion



- Much more efficient than executing simulations

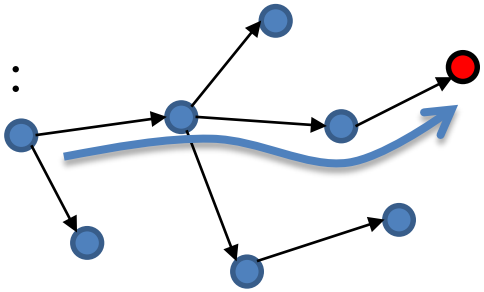
Interpolation errors

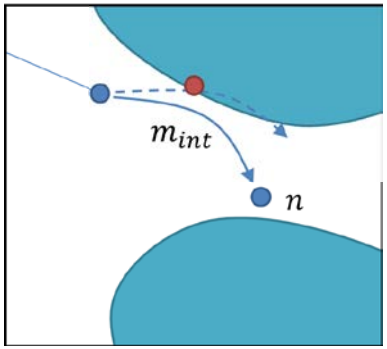
- Motion interpolation occurs because:
  - Database unlikely has an exactly matched motion with a query, in continuous domain
- Generally, doesn't affect the tree exploration if an error is small enough
- However, problem can occur near obstacle regions



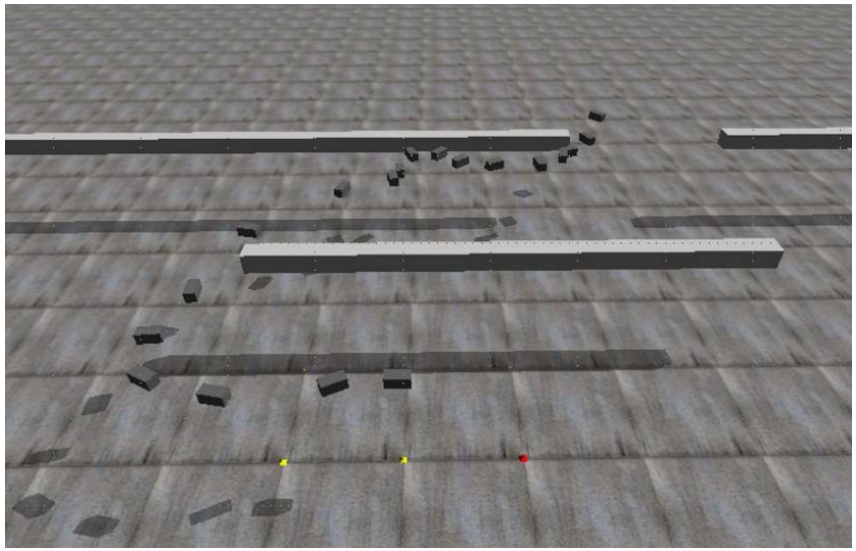
Updating motions

- Motion update is done by real simulation
- For overall efficiency, update motions if :
  - Accumulated errors are above $threshold_a$,
    - Sum of errors from the root node
  - Distance to obstacle is below $threshold_o$,
    - For handling narrow passage
  - Distance to goal state is below $threshold_g$





Experiment results



	RRT	ours 1	ours 2
Total time (s)	125.11	140.96	88.22
Simulation time (s)	94.61	50.53	25.51
% of simulation	75.62	35.85	28.92
Retrieval time (s)	0	10.92	8.99
% of retrieval	0.00	7.74	10.19

ours1: with 35,000 motions (10s)
ours2: with 100,000 motions (30s)

Summary

- Propose efficient sampling-based motion planning algorithms for handling narrow passage problem
  - *Narrow passage detection algorithm*
    - Perform selective retractions
  - *Productive region-oriented biased sampling algorithm*
    - Biased sampling for a robot manipulator
  - *Motion database-based extension algorithm*
    - Efficient local planning under kinodynamic constraints

Publications

- First-authored papers:
  - **Junghwan Lee**, OSung Kwon, Liangjun Zhang, and Sung-eui Yoon ,
    Selective retraction-based RRT planner for various environments,
    IEEE Transaction on Robotics, 2014
  - **Junghwan Lee** and Sung-Eui Yoon,
    PROT: Productive Regions-Oriented Task space path planning for hyper-redundant manipulators,
    IEEE International Conference on Robotics and Automations (ICRA), 2014
  - **Junghwan Lee**, OSung Kwon, Liangjun Zhang, and Sung-eui Yoon
    SR-RRT: Selective Retraction-based RRT Planner,
    IEEE International Conference on Robotics and Automations (ICRA), 2012
- Co-authored papers:
  - Donghyuk Kim, **Junghwan Lee** and Sung-Eui Yoon,
    Cloud RRT*: Sampling Cloud based RRT*,
    IEEE International Conference on Robotics and Automations (ICRA), 2014
  - Tieu Lin Loi, Jae-Pil Heo, **Junghwan Lee** and Sung-Eui Yoon,
    VLSH: Voronoi-based Locality Sensitive Hashing
    IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013
  - Duksu Kim, Jinkyu Lee, **Junghwan Lee**, InSik Shin, John Kim, Sung-eui Yoon
    Scheduling in Heterogeneous Computing Environments for Proximity Queries
    IEEE Transactions on Visualization and Computer Graphics (TVCG), 2013
- Future plan: submit a paper
  - Motion database-based selection algorithm

Acknowledgements

- Advisor, Sung-eui Yoon
- Other committee members
- SGLab members