Nonholomic Motion Planning, and Rapidly-Exploring Random Trees (RRTs)

Robot Motion Planning

ITCS 6152/8152: Spring 2016

Srinivas Akella

Department of Computer Science

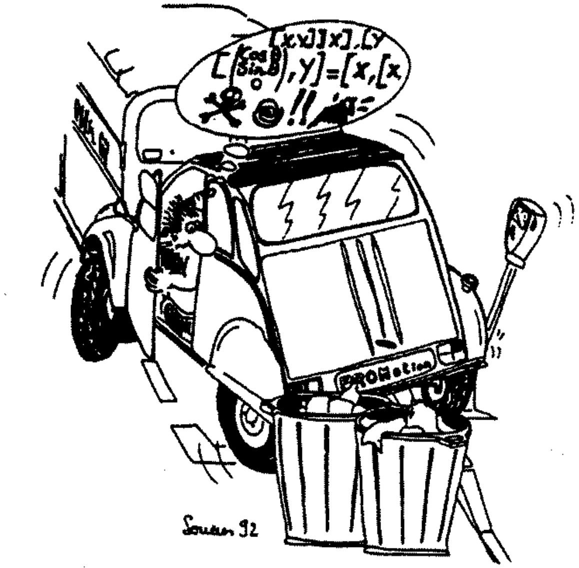
University of North Carolina, Charlotte

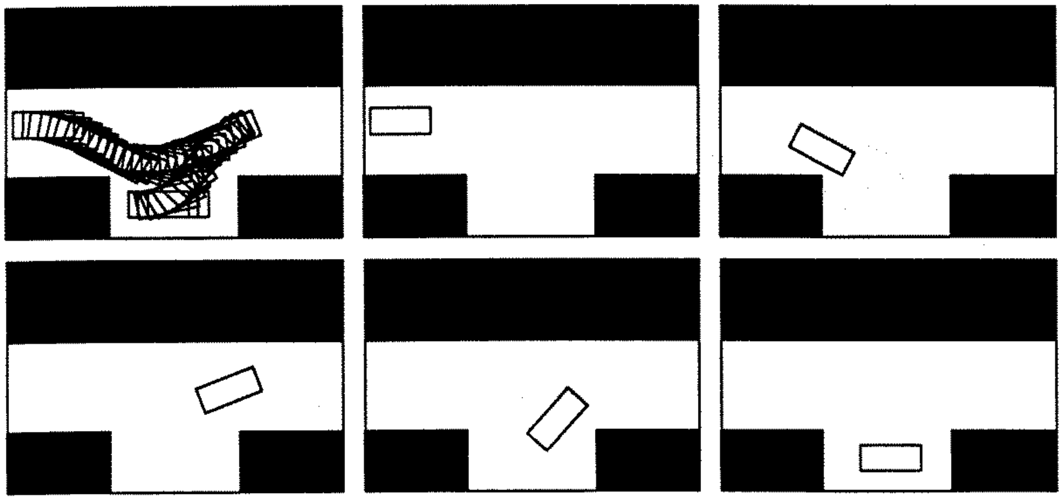
- Stanley (2005)

Navlab (1986)

Nonholonomic Motion Planning

Motion planning for car-like robots





Example task: Parallel parking

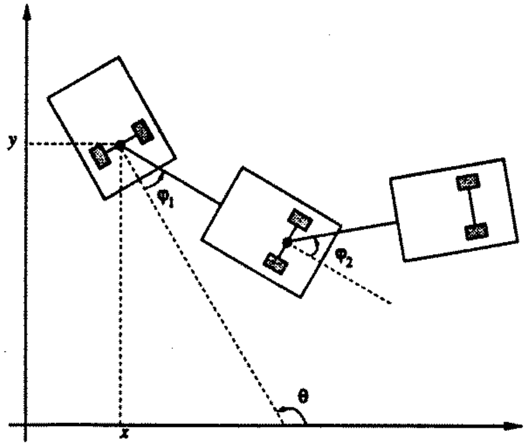
Also motion planning for spacecraft, underwater robots, aircraft

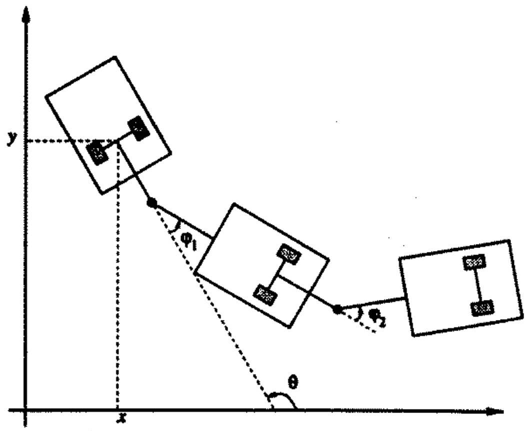
Double Trailer

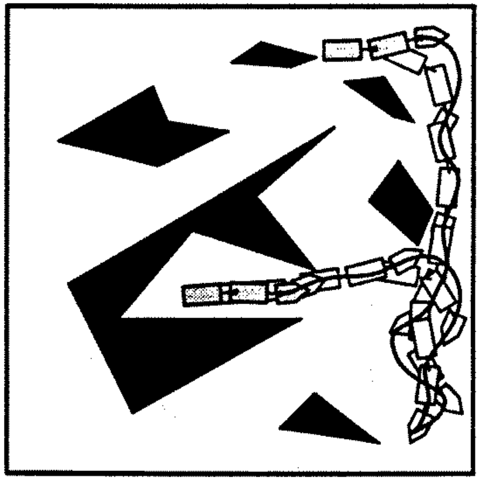
Wikipedia

Nonholonomic Robots

Tractor trailer systems

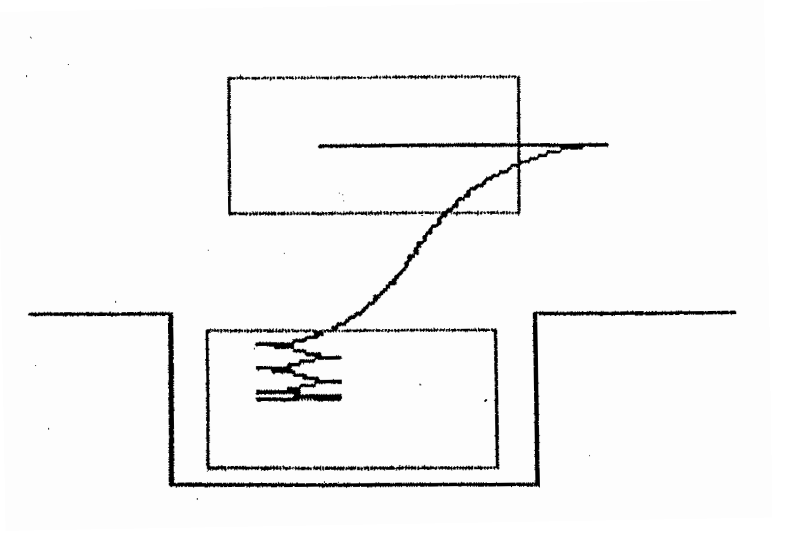






Example: Baggage trucks at airports

Nonholonomic Motion Planning



HOLONOMIC ROBOTS

A HOLONOMIC ROBOT IS A ROBOT THAT CAN MOVE IN ANY DIRECTION IN ITS CONFIGURATION SPACE

ASSUME A HOLONOMIC ROBOT $\mathcal{A}$ IS AT A CONFIGURATION q AND THAT ITS C-SPACE $\mathcal{C}$ IS OF DIMENSION n

THERE IS AN n-DIMENSIONAL SPACE OF VELOCITY VECTORS AT q ; ALL VELOCITY DIRECTIONS ARE FEASIBLE

EXAMPLES:

- FREE FLYING BODY IN $\mathbb{R}^2$ OR $\mathbb{R}^3$
- ARTICULATED ROBOT WITH PRISMATIC OR REVOLUTE JOINTS

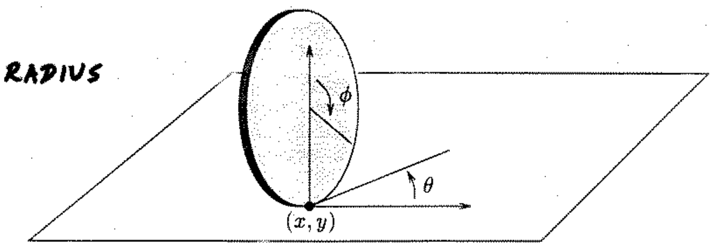
A kinematic constraint is a holonomic constraint if it can be expressed in the form f (q; t) = 0

NONHOLONOMIC CONSTRAINTS

CONSIDER A DISK ROLLING (WITHOUT SLIPPING) ON A PLANE

CONFIGURATION OF DISK $q = (x, y, \theta, \phi)$

DISK RADIUS IS ρ



SINCE THE DISK ROLLS WITHOUT SLIPPING, CONSTRAINTS ON MOTION OF DISC ARE

$$\dot{x} - \rho \cos\theta \, \dot{\phi} = 0$$

$$\dot{y} - \rho \sin\theta \, \dot{\phi} = 0$$

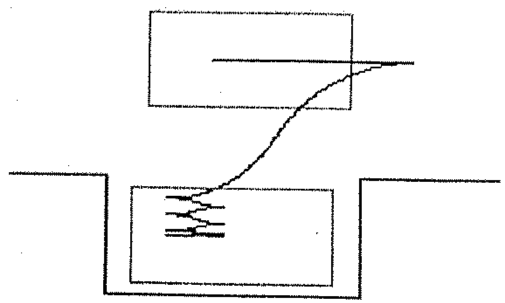
THESE NON-INTEGRABLE CONSTRAINTS ARE CALLED NONHOLONOMIC CONSTRAINTS

Examples: Unicycle, pizza cutter

NONHOLONOMIC SYSTEMS

THE CONSTRAINTS ON A NONHOLONOMIC SYSTEM ARE CHARACTERIZED BY A SET OF NON-INTEGRABLE EQUATIONS INVOLVING SYSTEM STATE AND TIME DERIVATIVES OF STATE

THE MOTION FREEDOMS OF THE ROBOT ARE NOT INDEPENDENT



A FEASIBLE PATH FOR PARALLEL PARKING A CAR-LIKE ROBOT

FOR A NONHOLONOMIC SYSTEM, ANY PATH IN C_{free} DOES NOT NECESSARILY CORRESPOND TO A FEASIBLE PATH SATISFYING THE NONHOLONOMIC CONSTRAINTS

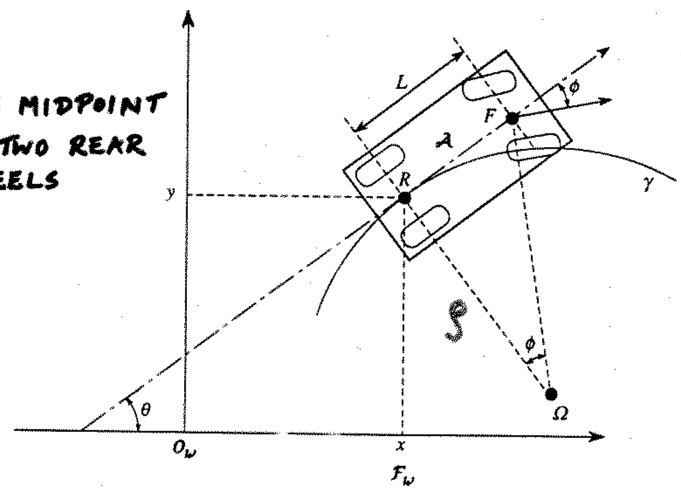
Nonholonomic constraints are of the form f(q, qdot, t) = 0

CAR-LIKE ROBOTS

CONSIDER A CAR-LIKE ROBOT IN THE PLANE.
CONFIGURATION $q = (x, y, \theta)$

THE WHEELS OF THE ROBOT ROLL WITHOUT SLIPPING ON THE PLANE

R IS MIDPOINT OF TWO REAR WHEELS



SINCE THE SIDEWAYS VELOCITY OF THE REAR WHEELS IS ZERO,

$$\dot{x} \sin\theta - \dot{y} \cos\theta = 0$$

NON-INTEGRABLE NONHOLONOMIC CONSTRAINT

STATE TRANSITION EQUATION

IT IS CONVENIENT TO REWRITE THE CONSTRAINTS IN A FORM THAT GIVES A DIRECT EXPRESSION FOR THE SET OF ALLOWED VELOCITIES

WE WISH TO WRITE A STATE TRANSITION EQUATION

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

WHERE x REPRESENTS THE STATE AND u REPRESENTS THE INPUT, AND $\dot{x}$ IS THE DERIVATIVE OF x WITH RESPECT TO TIME

FOR THE CAR-LIKE ROBOT, THE NONHOLONOMIC CONSTRAINT $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$ IS SATISFIED BY

$$\dot{x} = s\cos\theta$$
$$\dot{y} = s\sin\theta$$

s = SPEED OF ROBOT

SINCE $s = \rho\dot{\theta}$ AND $\rho = \frac{L}{\tan\phi}$,

$$\dot{\theta} = \frac{s}{L}\tan\phi$$

STATE TRANSITION EQUATION FOR CAR-LIKE ROBOT IS

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} s\cos\theta \\ s\sin\theta \\ \frac{s}{L}\tan\phi \end{pmatrix}$$

Unicycle

- Unicycle

Source: Walmart

Source: www.bikeforest.com

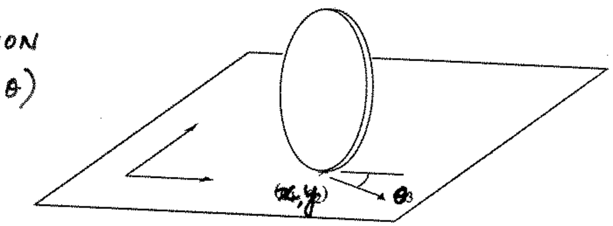
UNICYCLE

A UNICYCLE IS A NONHOLONOMIC SYSTEM WITH MOTION CONSTRAINT

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0$$

CONFIGURATION

$q = (x, y, \theta)$



THE UNICYCLE HAS TWO CONTROLS:

- ROLLING FORWARD OR BACKWARD WITHOUT SLIPPING
- TURNING IN PLACE

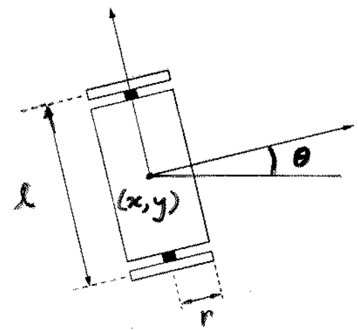
STATE TRANSITION EQUATION CAN BE WRITTEN AS

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} \underset{\substack{\uparrow \\ \text{ROLLING} \\ \text{VELOCITY}}}{u_1} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \underset{\substack{\uparrow \\ \text{ROTATIONAL (TURNING)} \\ \text{VELOCITY}}}{u_2}$$

CONSIDER THE TWO VECTOR FIELDS $\bar{X} = (\cos\theta \quad \sin\theta \quad 0)^T$ AND $\bar{Y} = (0 \quad 0 \quad 1)^T$, WHICH SATISFY THE MOTION CONSTRAINT EQUATION, AS BASIS VECTOR FIELDS

DIFFERENTIAL DRIVE ROBOT

A DIFFERENTIAL DRIVE ROBOT HAS A SINGLE AXLE CONNECTING TWO INDEPENDENTLY DRIVEN WHEELS



LET u_R BE THE ANGULAR VELOCITY OF THE RIGHT WHEEL AND u_L BE THAT OF THE LEFT WHEEL

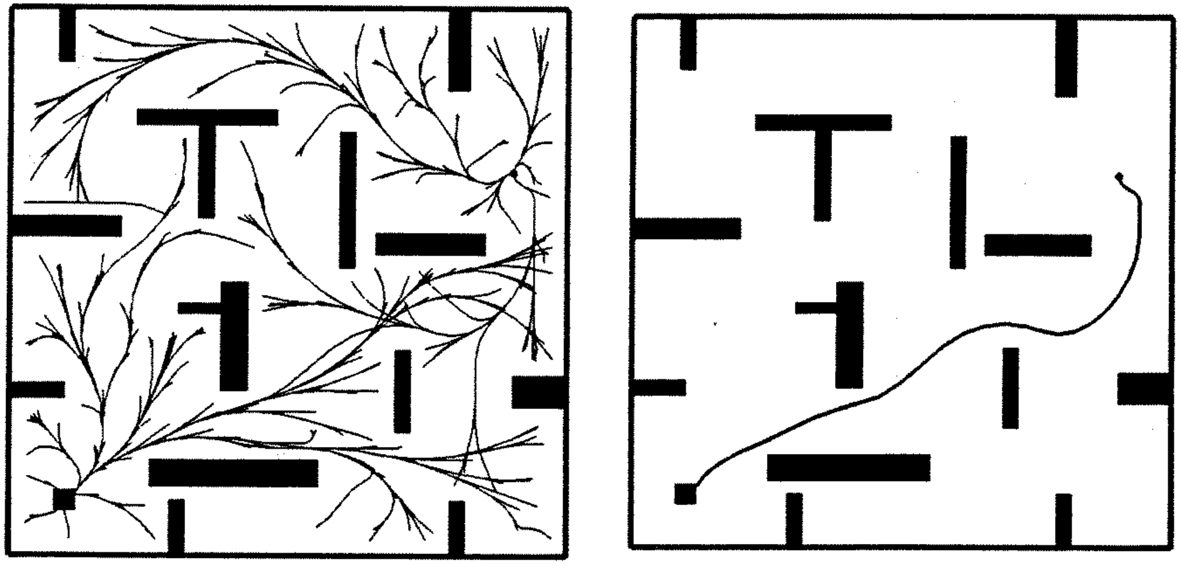
THE STATE TRANSITION EQUATION IS

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{r}{2}(u_R + u_L)\cos\theta \\ \frac{r}{2}(u_R + u_L)\sin\theta \\ \frac{r}{l}(u_R - u_L) \end{pmatrix}$$

Nonholonomic Path Planning

- Two-phase planning:
  - Compute a collision-free path ignoring nonholonomic constraints
  - Transform path into feasible nonholonomic path
- Direct planning:
  - Build a tree of configurations until one is close enough to the goal

Rapidly Exploring Random Trees



Classes of Motion Planning Methods

- Roadmap methods: Represent connectivity of freespace by network of 1D curves
- Cell decomposition methods: Decompose free space into cells, and represent connectivity of free space by adjacency graph of cells
- Potential field methods: Define potential function over c-space so it has minimum at goal, and follow steepest descent to goal
- Sampling based methods: Sample configurations in free space, and connect them in roadmap graph

Motivation

- Nonholonomic constraints
- Dynamics

- Handling differential constraints

Kinodynamic Planning

KINODYNAMIC PLANNING: MOTION PLANNING THAT TAKES INTO ACCOUNT DYNAMIC CONSTRAINTS AS WELL AS KINEMATIC CONSTRAINTS

REPLACE CONFIGURATION SPACE BY STATE SPACE (OR PHASE SPACE)

A POINT IN STATE SPACE INCLUDES BOTH CONFIGURATION PARAMETERS AND VELOCITY PARAMETERS

KINODYNAMIC PLANNERS MUST DETERMINE CONTROL INPUTS TO DRIVE A ROBOT FROM AN INITIAL CONFIGURATION AND VELOCITY TO A GOAL CONFIGURATION AND VELOCITY, WHILE OBEYING PHYSICALLY BASED DYNAMICAL MODELS AND AVOIDING OBSTACLES IN THE ROBOT'S ENVIRONMENT

Rapidly Exploring Random Trees

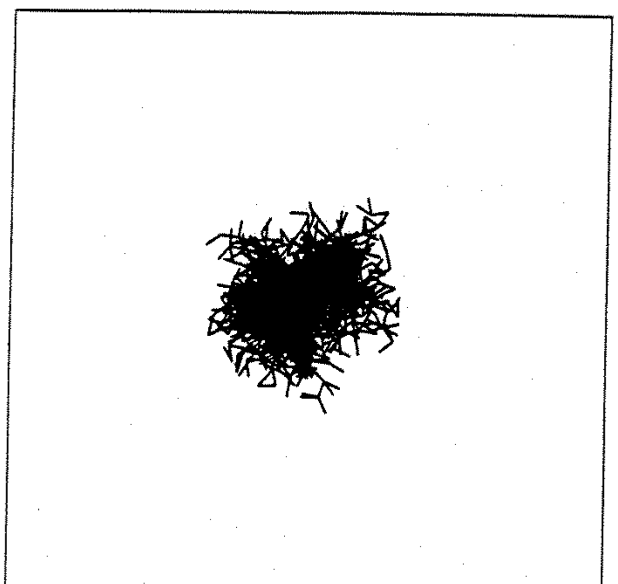
RAPIDLY EXPLORING RANDOM TREES (RRTS) WERE INTRODUCED AS A SIMPLE SAMPLING SCHEME AND DATA STRUCTURE TO QUICKLY EXPLORE HIGH-DIMENSIONAL SEARCH SPACES WITH BOTH ALGEBRAIC CONSTRAINTS (DUE TO OBSTACLES) AND DIFFERENTIAL CONSTRAINTS (DUE TO NONHOLONOMY AND DYNAMICS)

RRT CHARACTERISTICS:

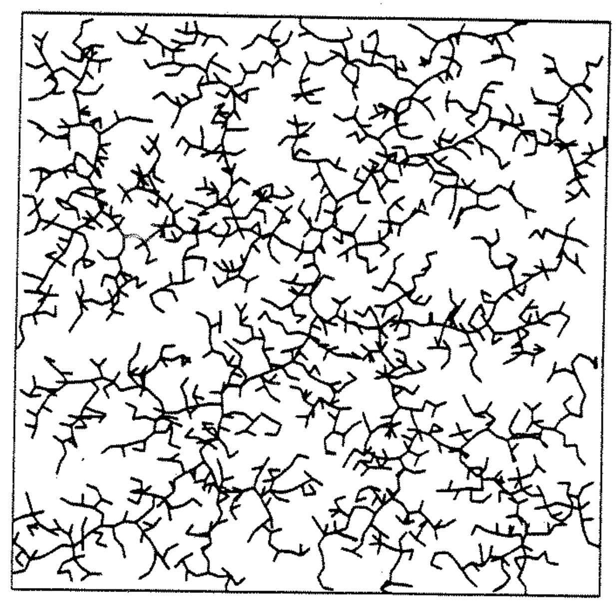
- DESIGNED FOR SINGLE QUERY PATH PLANNING
- RAPIDLY EXPLORE AND ARRIVE AT A UNIFORM COVERAGE OF SPACE
- SIMILAR TO PRMS, BUT CAN HANDLE CHALLENGING NONHOLONOMIC PLANNING AND KINODYNAMIC PLANNING PROBLEMS

RRTs

KEY IDEA: BIAS EXPLORATION TOWARD UNEXPLORED REGIONS OF THE SPACE BY SAMPLING POINTS IN THE SPACE AND INCREMENTALLY "PULLING" THE SEARCH TREE TOWARD THEM



NAIVE RANDOM TREE



RAPIDLY EXPLORING RANDOM TREE

RRT Construction

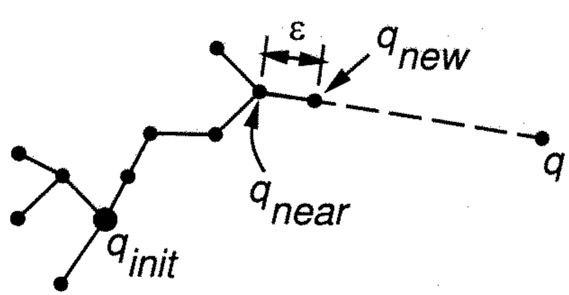
AN RRT IS A TREE ROOTED AT q_{INIT}, AND HAS K VERTICES

BUILD_RRT(q_{init})

$\mathcal{T}$.init(q_{init});
for $k = 1$ **to** K **do**
  $q_{rand} \leftarrow$ RANDOM_CONFIG();
  EXTEND($\mathcal{T}, q_{rand}$);
Return $\mathcal{T}$

EXTEND($\mathcal{T}, q$)

$q_{near} \leftarrow$ NEAREST_NEIGHBOR($q, \mathcal{T}$);
if NEW_CONFIG($q, q_{near}, q_{new}, u_{new}$) **then**
  $\mathcal{T}$.add_vertex(q_{new});
  $\mathcal{T}$.add_edge($q_{near}, q_{new}, u_{new}$);
  if $q_{new} = q$ **then**
    Return *Reached*;
  else
    Return *Advanced*;
Return *Trapped*;

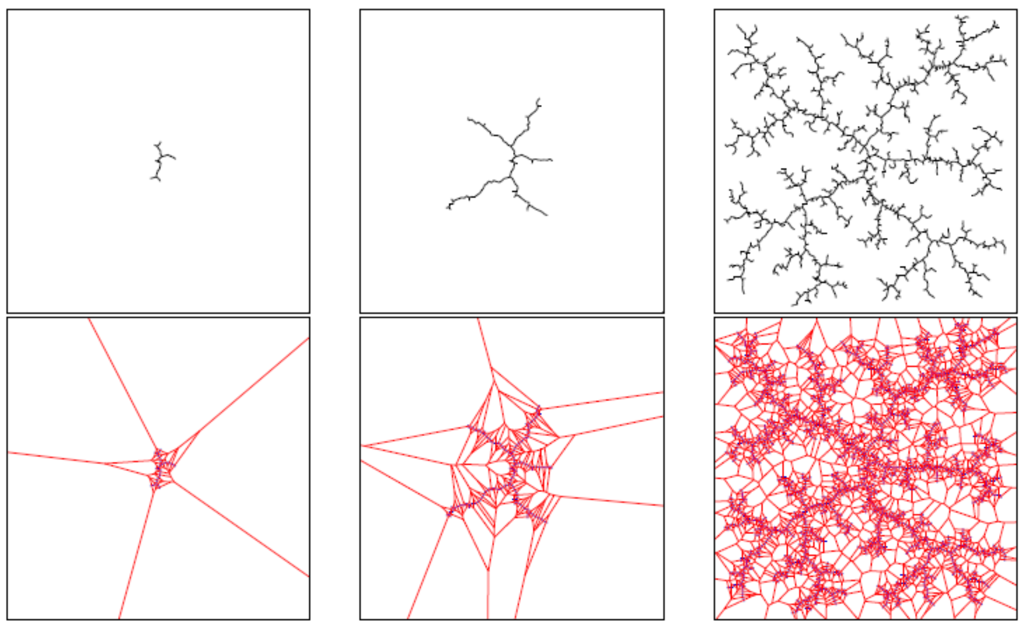


Step size: ε

RRT Exploration

RRTS BIAS SEARCH TOWARD UNEXPLORED REGIONS OF SPACE

THREE STAGES DURING CONSTRUCTION OF AN RRT FOR A HOLONOMIC PLANNING PROBLEM

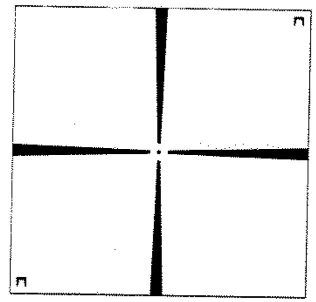


VORONOI REGIONS OF RRT VERTICES

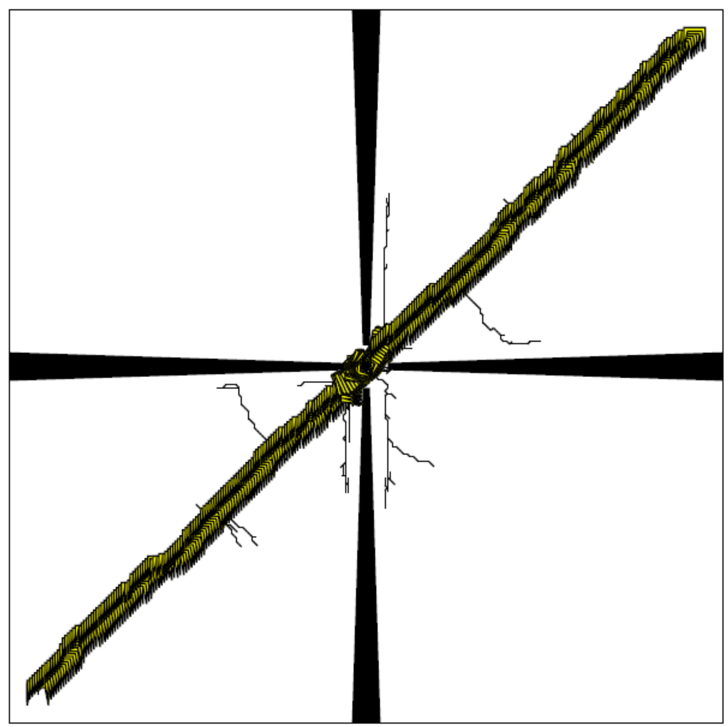
PROBABILITY THAT A VERTEX IS SELECTED FOR EXPANSION IS PROPORTIONAL TO THE AREA OF ITS VORONOI REGION

Holonomic Planning Example

RRTs can be used for challenging holonomic planning problems



A narrow passage example



RRT Variations

INSTEAD OF ATTEMPTING TO EXTEND AN RRT BY AN INCREMENTAL STEP ϵ, THE CONNECT FUNCTION ITERATES THE EXTEND STEP UNTIL q OR AN OBSTACLE IS REACHED

CONNECT CAN REPLACE EXTEND IN BUILD_RRT, YIELDING AN RRT THAT GROWS VERY QUICKLY

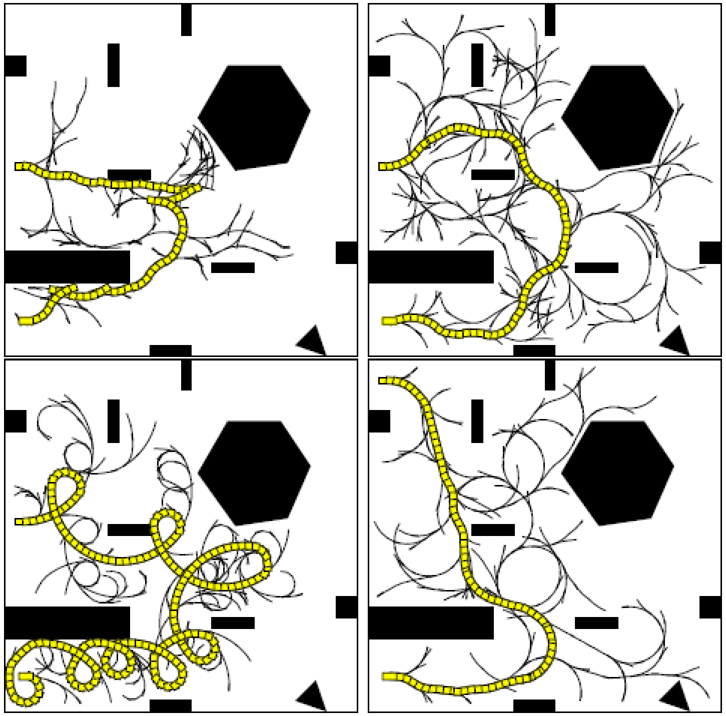
```
CONNECT(T, q)
  repeat
      S ← EXTEND(T, q);
  until not (S = Advanced)
  Return S;
```

THE CONNECT FUNCTION IS A GREEDY HEURISTIC THAT ATTEMPTS TO MOVE OVER A LONGER DISTANCE

CONNECT APPEARS TO BE BEST USED FOR HOLONOMIC PLANNING PROBLEMS THAT INVOLVE NO DIFFERENTIAL CONSTRAINTS

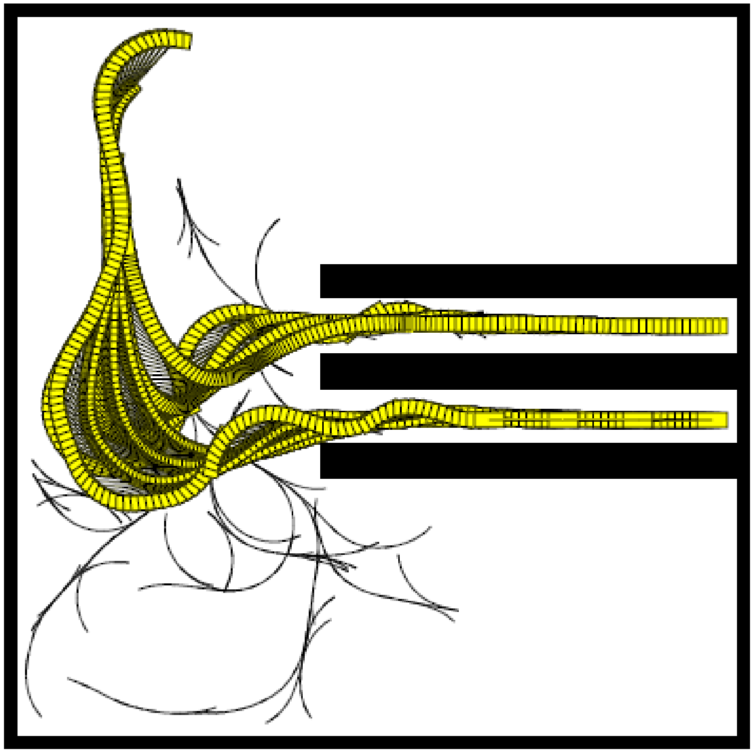
Nonholonomic Planning Example

PLANAR CAR-LIKE ROBOT (3 DEGREES OF FREEDOM)
MOVING AT CONSTANT SPEED



2D PROJECTIONS OF RRTs, ALONG WITH COMPUTED PATH

7-DOF NONHOLONOMIC PLANNING EXAMPLE



4-DOF CAR PULLING THREE TRAILERS

Bidirectional Planning

IMPROVED PERFORMANCE CAN BE OBTAINED BY GROWING TWO RRTs, ONE FROM q_{INIT} AND THE OTHER FROM q_{GOAL}

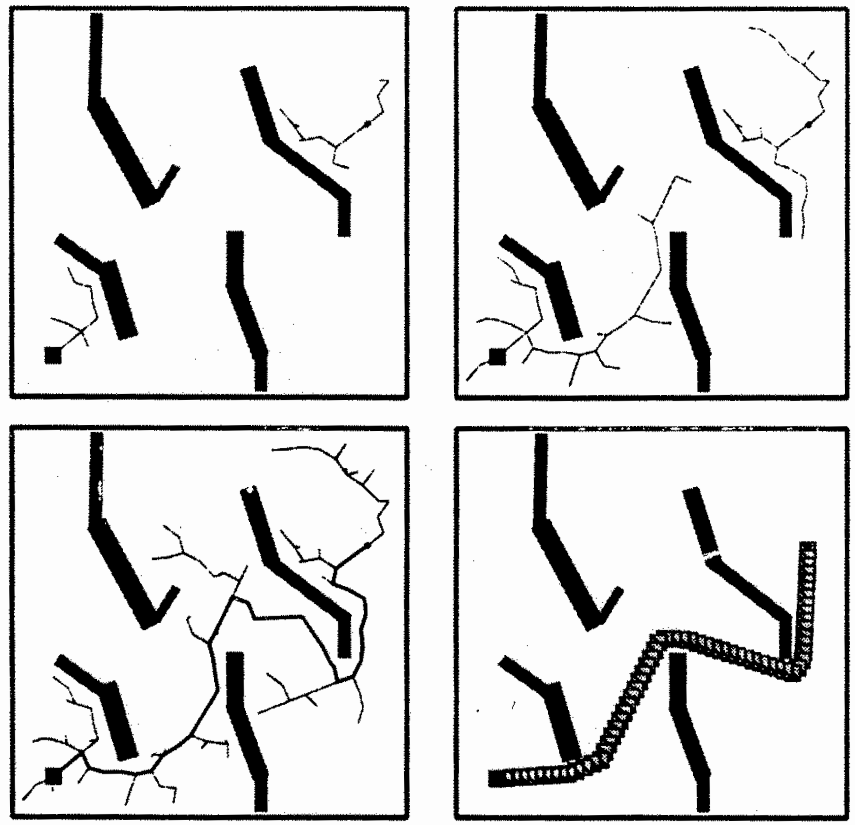
A SOLUTION IS FOUND IF THE TWO RRTs MEET

RRT_BIDIRECTIONAL(q_{init}, q_{goal})
$\mathcal{T}_a$.init(q_{init}); $\mathcal{T}_b$.init(q_{goal});
for $k = 1$ **to** K **do**
  $q_{rand} \leftarrow$ RANDOM_CONFIG();
  if not (EXTEND($\mathcal{T}_a, q_{rand}$) = *Trapped*) **then**
    if (EXTEND($\mathcal{T}_b, q_{new}$) = *Reached*) **then**
      Return PATH($\mathcal{T}_a, \mathcal{T}_b$);
  SWAP($\mathcal{T}_a, \mathcal{T}_b$);
Return *Failure*

RRT CONSTRUCTION IS BIASED TO ENSURE THE TREES MEET WELL BEFORE COVERING THE ENTIRE SPACE

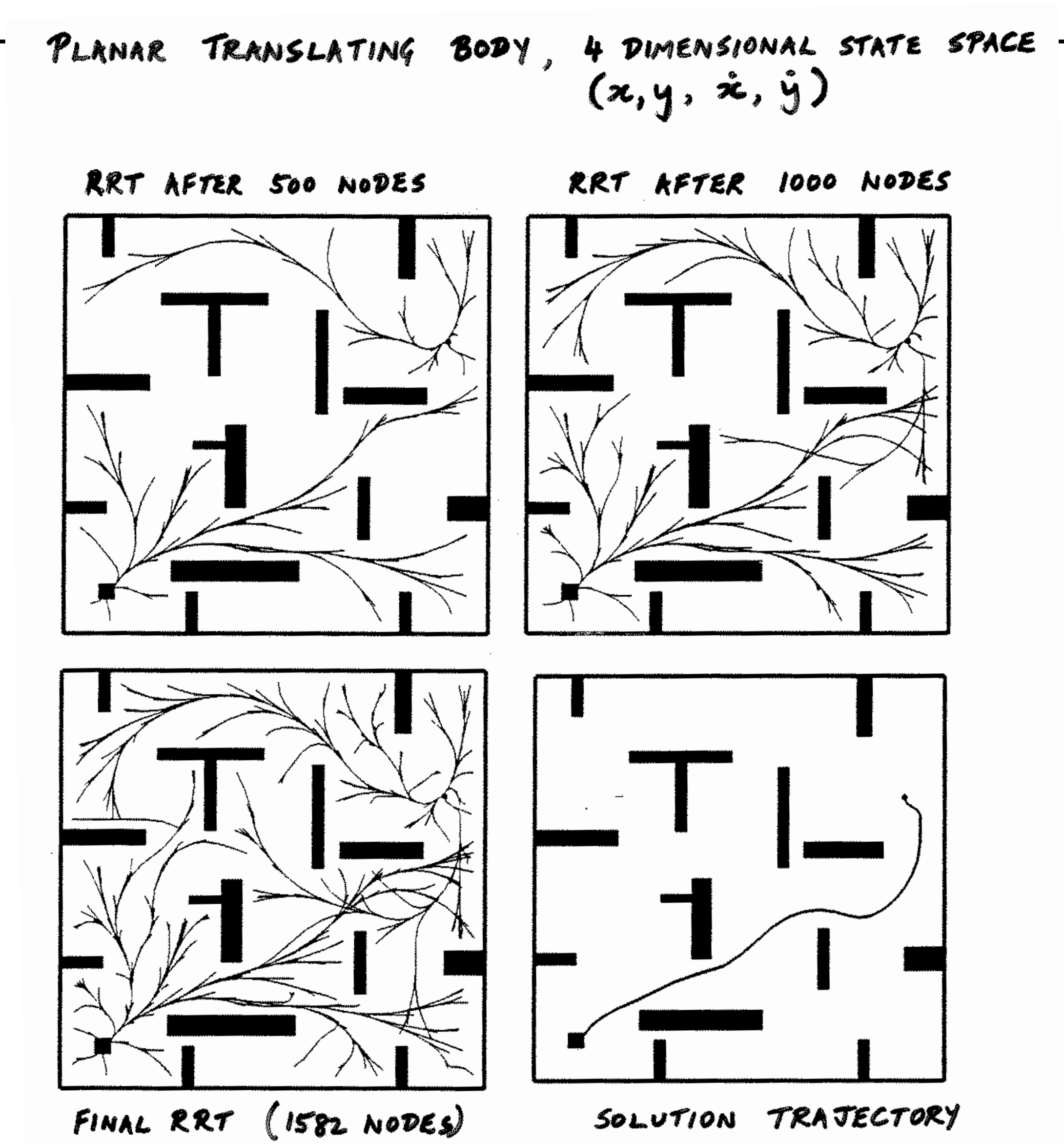
Bidirectional RRTs

BIDIRECTIONAL RRT EXAMPLE

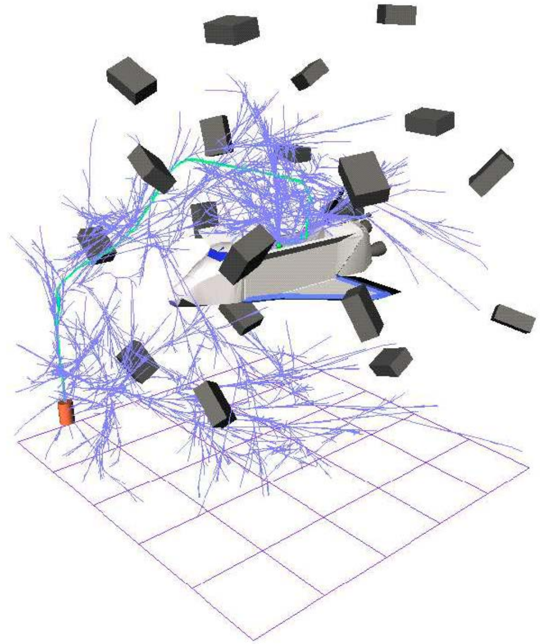


NOTE: PATHS GENERATED USING RRTs CAN BE SMOOTHED FOR ROBOT EXECUTION

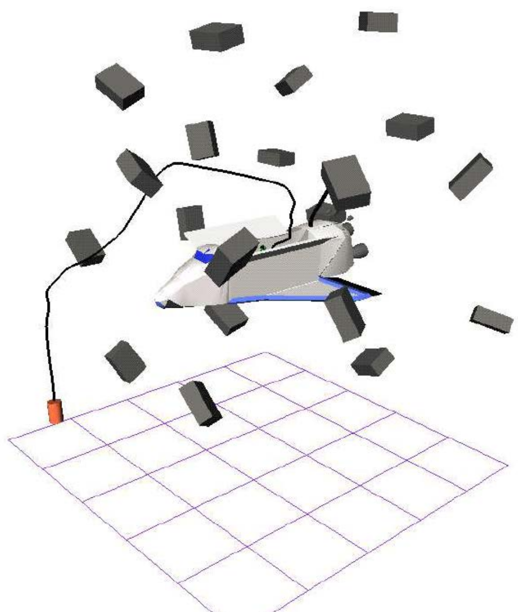
Kinodynamic Planning Example



Kinodynamic Planning Example



Satellite docking example, 12-dimensional state space



Reading

RRT:

- Chapter 5.5 and 14.4.3, LaValle
- Chapter 7.2.2, Choset et al.

Nonholonomic motion planning:

- Chapter 13-13.1 and 15.3, LaValle
- Chapter 12 (esp. 12.5), Choset et al.