Robot Motion Planning

Jan Faigl

Department of Computer Science and Engineering
Faculty of Electrical Engineering, Czech Technical University in Prague

1^{st} lecture

A4M36PAH - Planning and Games

Part I – Organization

General Info

Lectures

Course Evaluation

Assignments

Part II – Robot Motio Planning

Part I

Course Organization

General information

A4M36PAH - Plánování a hry

AE4M36PAH - Planning and games

- Course web pages `https://cw.felk.cvut.cz/doku.php/courses/a4m36pah/start`
- Lectures: in a block during 4th semestr's week

  `http://cw.felk.cvut.cz/doku.php/courses/a4m36pah/lectures`
- Lecturers:

  Carmel Domshlak – Automatic Planning expert from Technion

  Michal Pěchouček

Lectures Plan – Spring 2013

Lecture	Date	Time	Room	Lecturer	Topic
#1	11. 2.	16:15-17:45	T2:C3-132	Jan Faigl	Motion Planning
#2, #3	4. 3.	16:15-17:45 18:00-19:30	T2:C3-132 Dejvice	Carmel Domshlak	Planning as general problem solving
#4, #5	5. 3.	16:15-17:45 18:00-19:30	T2:C3-132 Dejvice	Carmel Domshlak	Domain-independent heuristics
#6, #7	6. 3.	16:15-17:45 18:00-19:30	KN:E-301 Karlak	Carmel Domshlak	Planning as SAT
#8, #9	7. 3.	16:15-17:45 18:00-19:30	KN:E-301 Karlak	Carmel Domshlak	Partial order planning (part I)
#10, #11	8. 3.	12:45-14:15 14:30-16:00	KN:E-107 Karlak	Carmel Domshlak	Partial order planning (part II)

http://cw.felk.cvut.cz/doku.php/courses/a4m36pah/lectures

Tutorial Lecturers

Michal Čáp

Adam Horký

Jan Hrnčíř

Jan Tožička

Course Evaluation

- **Assignments**: Two assignments, each for 15 points
- **Credit** (Zápočet): Both assignments have to be successfully submitted
- **Exam**: Up to 70 points
- **Final mark**: Final mark $= \sum assignments + exam$

Maximum points 30 + 70

Assignments

- **Assignment 1**
    - Handed out 18. 2. 2013
    - Deadline 3. 3. 2013

15 points

- **Assignment 2**
    - Handed out 8. 4. 2013
    - Deadline 21. 4. 2013

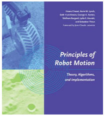
15 points

Part II

Today’s Lecture – Motion Planning

Literature

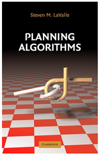
- Robot Motion Planning, *Jean-Claude Latombe,* Kluwer Academic Publishers, Boston, MA, 1991.



Principles of Robot Motion

Theory, Algorithms, and Implementation

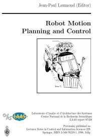
- Principles of Robot Motion: Theory, Algorithms, and Implementations, *H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun,* MIT Press, Boston, 2005.
  `http://www.cs.cmu.edu/~biorobotics/book`



Steven M. LaValle

PLANNING ALGORITHMS

- Planning Algorithms, *Steven M. LaValle,* Cambridge University Press, May 29, 2006.
  `http://planning.cs.uiuc.edu`



Jean-Paul Laumond (Editor)

Robot Motion Planning and Control

- Robot Motion Planning and Control, *Jean-Paul Laumond,* Lectures Notes in Control and Information Sciences, 2009.
  `http://homepages.laas.fr/jpl/book.html`

Robot Motion Planning

Motivational problem:

- How to transform high-level task specification (provided by humans) into a low-level description suitable for controlling the actuators?

To develop algorithms for such a transformation.

The motion planning algorithms provide transformations how to move a robot (object) considering all operational constraints.

It encompasses several disciples, e.g., mathematics, robotics, computer science, control theory, artificial intelligence, computational geometry, etc.

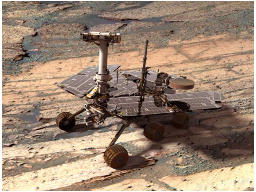
Robot Motion Planning

Motivational problem:

- How to transform high-level task specification (provided by humans) into a low-level description suitable for controlling the actuators?

To develop algorithms for such a transformation.

The motion planning algorithms provide transformations how to move a robot (object) considering all operational constraints.

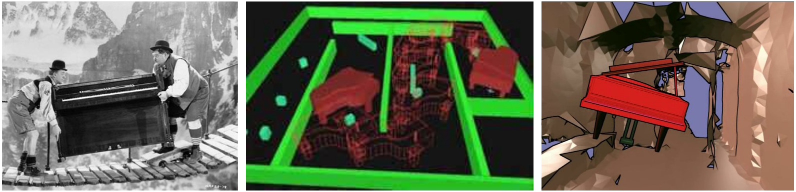


It encompasses several disciples, e.g., mathematics, robotics, computer science, control theory, artificial intelligence, computational geometry, etc.

Piano Mover's Problem

A classical motion planning problem

Having a CAD model of the piano, model of the environment, the problem is how to move the piano from one place to another without hitting anything.



Basic motion planning algorithms are focused primarily on rotations and translations.

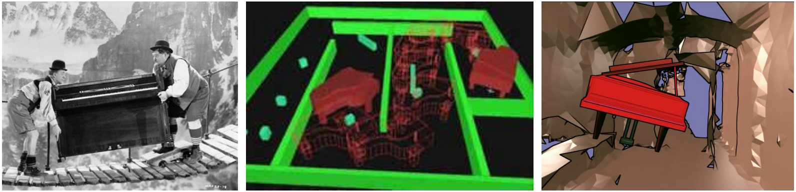
- We need notion of model representations and formal definition of the problem.
- Moreover, we also need a context about the problem and realistic assumptions.

The plans have to be admissible and feasible.

Piano Mover's Problem

A classical motion planning problem

Having a CAD model of the piano, model of the environment, the problem is how to move the piano from one place to another without hitting anything.

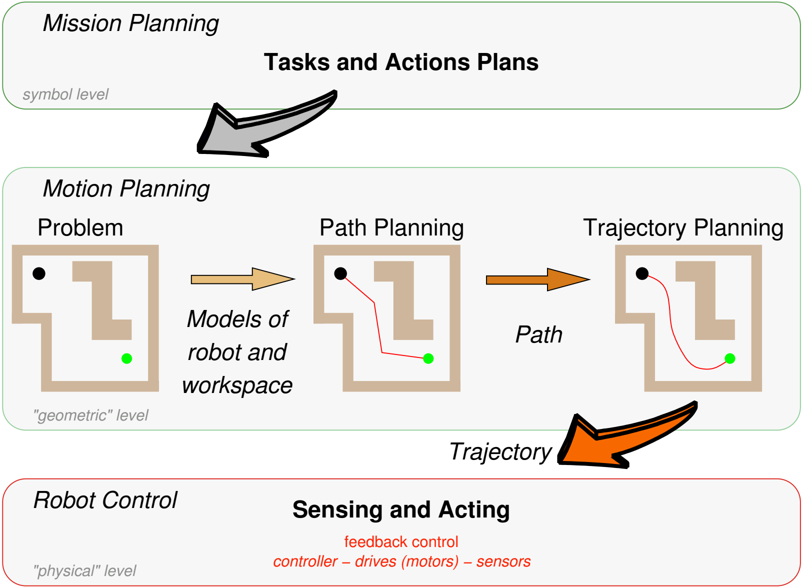


Basic motion planning algorithms are focused primarily on rotations and translations.

- We need notion of model representations and formal definition of the problem.
- Moreover, we also need a context about the problem and realistic assumptions.

The plans have to be admissible and feasible.

Robotic Planning Context



Mission Planning

Tasks and Actions Plans

symbol level

Motion Planning

Problem

Models of robot and workspace

Path Planning

Path

Trajectory Planning

"geometric" level

Trajectory

Robot Control

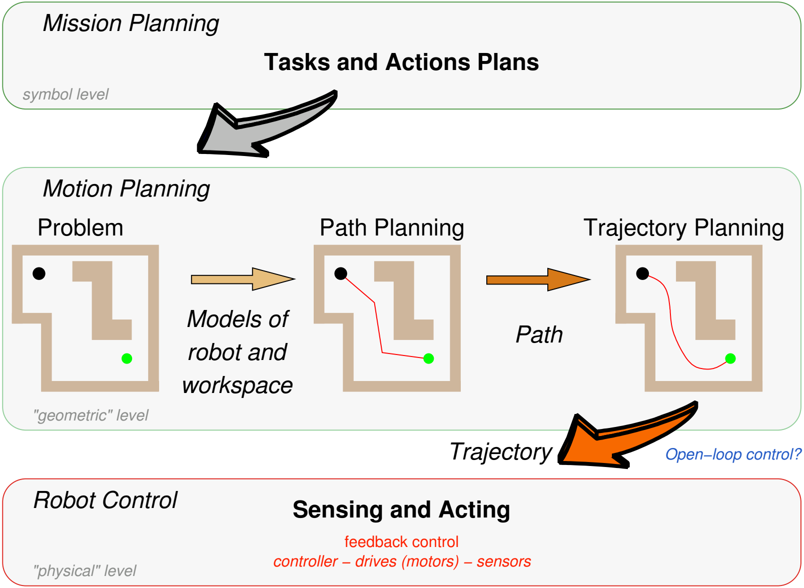
Sensing and Acting

feedback control

controller – drives (motors) – sensors

"physical" level

Robotic Planning Context



Mission Planning

Tasks and Actions Plans

symbol level

Motion Planning

Problem

Path Planning

Trajectory Planning

Models of robot and workspace

Path

"geometric" level

Trajectory

Open–loop control?

Robot Control

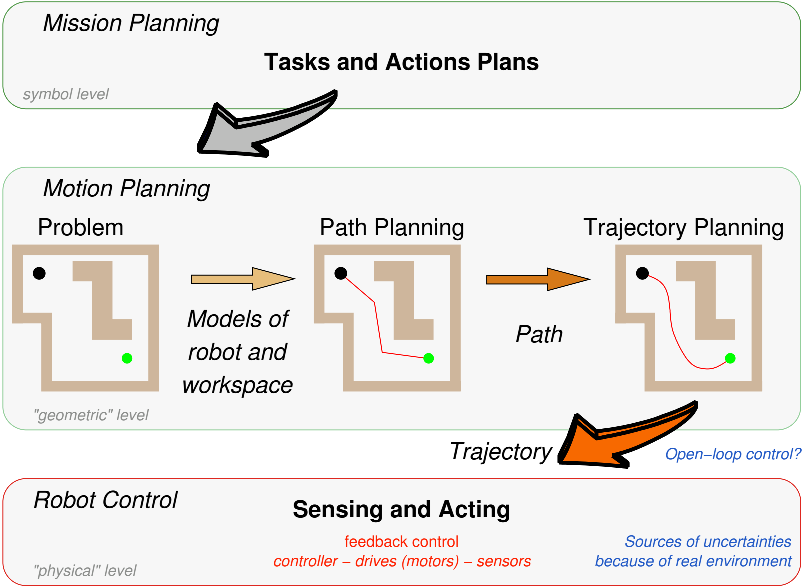
Sensing and Acting

feedback control

controller – drives (motors) – sensors

"physical" level

Robotic Planning Context



Mission Planning

Tasks and Actions Plans

symbol level

Motion Planning

Problem

Models of robot and workspace

Path Planning

Path

Trajectory Planning

"geometric" level

Trajectory

Open–loop control?

Robot Control

Sensing and Acting

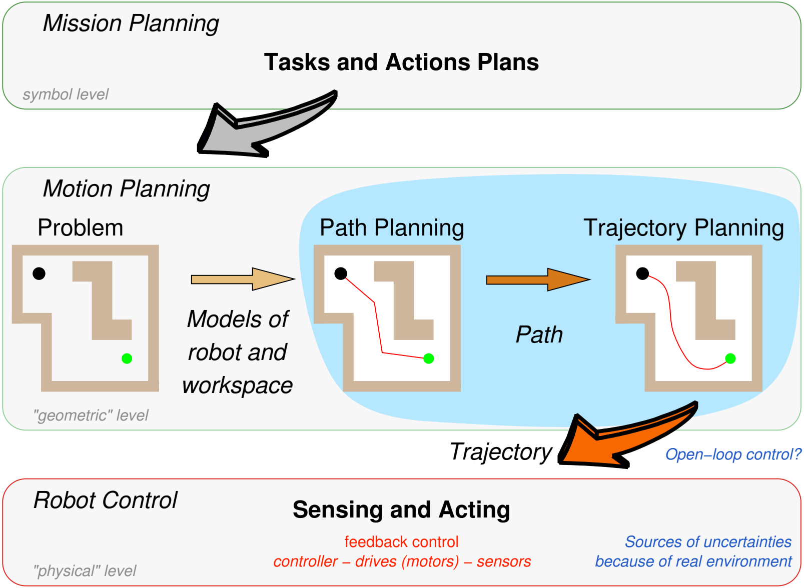
feedback control

controller – drives (motors) – sensors

Sources of uncertainties because of real environment

"physical" level

Robotic Planning Context



Mission Planning

Tasks and Actions Plans

symbol level

Motion Planning

Problem

Models of robot and workspace

Path Planning

Path

Trajectory Planning

"geometric" level

Trajectory

Open–loop control?

Robot Control

Sensing and Acting

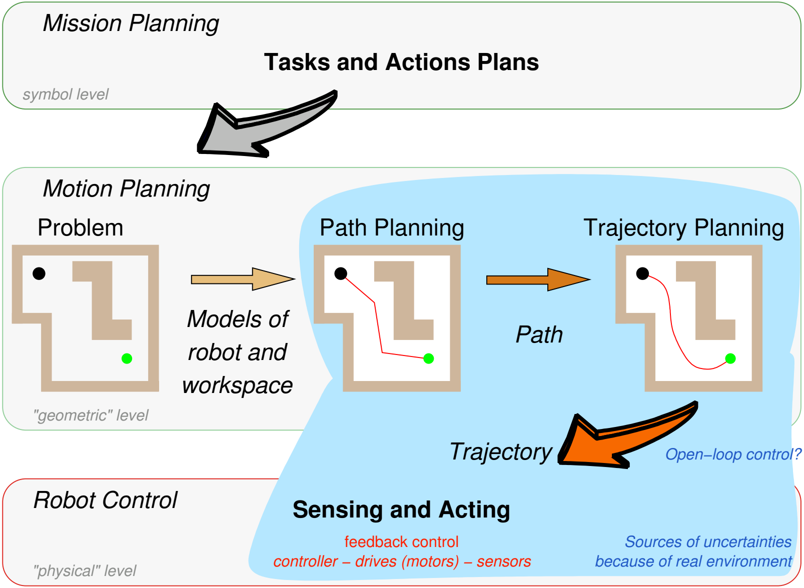
feedback control

controller – drives (motors) – sensors

Sources of uncertainties because of real environment

"physical" level

Robotic Planning Context



Mission Planning

Tasks and Actions Plans

symbol level

Motion Planning

Problem

Models of robot and workspace

Path Planning

Path

Trajectory Planning

Trajectory

Open–loop control?

"geometric" level

Robot Control

Sensing and Acting

feedback control
controller – drives (motors) – sensors

Sources of uncertainties because of real environment

"physical" level

Real Mobile Robots

In a real deployment, the problem is a more complex.

- The world is changing
- Robots update the knowledge about the environment

localization, mapping and navigation

- New decisions have to made
- A feedback from the environment

Motion planning is a part of the mission replanning loop.

Josef Štrunc, Bachelor thesis, CTU, 2009.

An example of robotic mission:

Multi-robot exploration of unknown environment

How to deal with real-world complexity?

Relaxing constraints and considering realistic assumptions.

Real Mobile Robots

In a real deployment, the problem is a more complex.

- The world is changing
- Robots update the knowledge about the environment

  localization, mapping and navigation
- New decisions have to made
- A feedback from the environment

  Motion planning is a part of the mission replanning loop.

Josef Štrunc, Bachelor thesis, CTU, 2009.

An example of robotic mission:

Multi-robot exploration of unknown environment

How to deal with real-world complexity?

Relaxing constraints and considering realistic assumptions.

Real Mobile Robots

In a real deployment, the problem is a more complex.

- The world is changing
- Robots update the knowledge about the environment

localization, mapping and navigation

- New decisions have to made
- A feedback from the environment

Motion planning is a part of the mission replanning loop.

Josef Štrunc, Bachelor thesis, CTU, 2009.

An example of robotic mission:

Multi-robot exploration of unknown environment

How to deal with real-world complexity?

Relaxing constraints and considering realistic assumptions.

Notation

- $\mathcal{W}$ – **World model** describes the robot workspace and its boundary determines the obstacles $\mathcal{O}_i$.

  2D world, $\mathcal{W} = \mathbb{R}^2$
- A **Robot** is defined by its geometry, parameters (kinematics) and it is controllable by the motion plan.
- $\mathcal{C}$ – **Configuration space** ($\mathcal{C}$-space)

  A concept to describe possible configurations of the robot. The robot's configuration completely specify the robot location in $\mathcal{W}$ including specification of all degrees of freedom.

  E.g., a robot with rigid body in a plane $\mathcal{C} = \{x, y, \varphi\} = \mathbb{R}^2 \times S^1$.
  - Let $\mathcal{A}$ be a subset of $\mathcal{W}$ occupied by the robot, $\mathcal{A} = \mathcal{A}(q)$.
  - A subset of $\mathcal{C}$ occupied by obstacles is

    $$\mathcal{C}_{obs} = \{q \in \mathcal{C} : \mathcal{A}(q) \cap \mathcal{O}_i, \forall i\}$$
  - Collision-free configurations are

    $$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}.$$

Notation

- $\mathcal{W}$ – **World model** describes the robot workspace and its boundary determines the obstacles $\mathcal{O}_i$.

  2D world, $\mathcal{W} = \mathbb{R}^2$

- A **Robot** is defined by its geometry, parameters (kinematics) and it is controllable by the motion plan.
- $\mathcal{C}$ – **Configuration space** ($\mathcal{C}$-space)

  A concept to describe possible configurations of the robot. The robot's configuration completely specify the robot location in $\mathcal{W}$ including specification of all degrees of freedom.

  E.g., a robot with rigid body in a plane $\mathcal{C} = \{x, y, \varphi\} = \mathbb{R}^2 \times S^1$.

  - Let $\mathcal{A}$ be a subset of $\mathcal{W}$ occupied by the robot, $\mathcal{A} = \mathcal{A}(q)$.
  - A subset of $\mathcal{C}$ occupied by obstacles is

    $$\mathcal{C}_{obs} = \{q \in \mathcal{C} : \mathcal{A}(q) \cap \mathcal{O}_i, \forall i\}$$

  - Collision-free configurations are

    $$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}.$$

Notation

- $\mathcal{W}$ – **World model** describes the robot workspace and its boundary determines the obstacles $\mathcal{O}_i$.

  2D world, $\mathcal{W} = \mathbb{R}^2$

- A **Robot** is defined by its geometry, parameters (kinematics) and it is controllable by the motion plan.

- $\mathcal{C}$ – **Configuration space** ($\mathcal{C}$-space)

  A concept to describe possible configurations of the robot. The robot's configuration completely specify the robot location in $\mathcal{W}$ including specification of all degrees of freedom.

  E.g., a robot with rigid body in a plane $\mathcal{C} = \{x, y, \varphi\} = \mathbb{R}^2 \times S^1$.

  - Let $\mathcal{A}$ be a subset of $\mathcal{W}$ occupied by the robot, $\mathcal{A} = \mathcal{A}(q)$.
  - A subset of $\mathcal{C}$ occupied by obstacles is

$$\mathcal{C}_{obs} = \{q \in \mathcal{C} : \mathcal{A}(q) \cap \mathcal{O}_i, \forall i\}$$

  - Collision-free configurations are

$$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}.$$

Notation

- $\mathcal{W}$ – **World model** describes the robot workspace and its boundary determines the obstacles $\mathcal{O}_i$.

  2D world, $\mathcal{W} = \mathbb{R}^2$
- A **Robot** is defined by its geometry, parameters (kinematics) and it is controllable by the motion plan.
- $\mathcal{C}$ – **Configuration space** ($\mathcal{C}$-space)

  A concept to describe possible configurations of the robot. The robot's configuration completely specify the robot location in $\mathcal{W}$ including specification of all degrees of freedom.

  E.g., a robot with rigid body in a plane $\mathcal{C} = \{x, y, \varphi\} = \mathbb{R}^2 \times S^1$.
  - Let $\mathcal{A}$ be a subset of $\mathcal{W}$ occupied by the robot, $\mathcal{A} = \mathcal{A}(q)$.
  - A subset of $\mathcal{C}$ occupied by obstacles is

    $$\mathcal{C}_{obs} = \{q \in \mathcal{C} : \mathcal{A}(q) \cap \mathcal{O}_i, \forall i\}$$
  - Collision-free configurations are

    $$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}.$$

Notation

- $\mathcal{W}$ – **World model** describes the robot workspace and its boundary determines the obstacles $\mathcal{O}_i$.

  2D world, $\mathcal{W} = \mathbb{R}^2$

- A **Robot** is defined by its geometry, parameters (kinematics) and it is controllable by the motion plan.
- $\mathcal{C}$ – **Configuration space** ($\mathcal{C}$-space)

  A concept to describe possible configurations of the robot. The robot's configuration completely specify the robot location in $\mathcal{W}$ including specification of all degrees of freedom.

  E.g., a robot with rigid body in a plane $\mathcal{C} = \{x, y, \varphi\} = \mathbb{R}^2 \times S^1$.

  - Let $\mathcal{A}$ be a subset of $\mathcal{W}$ occupied by the robot, $\mathcal{A} = \mathcal{A}(q)$.
  - A subset of $\mathcal{C}$ occupied by obstacles is

  $$\mathcal{C}_{obs} = \{q \in \mathcal{C} : \mathcal{A}(q) \cap \mathcal{O}_i, \forall i\}$$

  - Collision-free configurations are

  $$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}.$$

Path / Motion Planning Problem

- **Path** is a continuous mapping in $\mathcal{C}$-space such that

$$\pi : [0, 1] \to \mathcal{C}_{free}, \text{ with } \pi(0) = q_0, \text{ and } \pi(1) = q_f,$$

Only geometric considerations

- **Trajectory** is a path with explicate parametrization of time, e.g., accompanied by a description of the motion laws ($\gamma : [0, 1] \to \mathcal{U}$, where $\mathcal{U}$ is robot's action space).

It includes dynamics.

$$[T_0, T_f] \ni t \rightsquigarrow \tau \in [0, 1] : q(t) = \pi(\tau) \in \mathcal{C}_{free}$$

The planning problem is determination of the function $\pi(\cdot)$.

Additional requirements can be given:

- Smoothness of the path
- Kinodynamic constraints

E.g., considering friction forces

- Optimality criterion

shortest vs fastest (length vs curvature)

Path / Motion Planning Problem

- **Path** is a continuous mapping in $\mathcal{C}$-space such that

$$\pi : [0,1] \to \mathcal{C}_{free}, \text{ with } \pi(0) = q_0, \text{ and } \pi(1) = q_f,$$

Only geometric considerations

- **Trajectory** is a path with explicate parametrization of time, e.g., accompanied by a description of the motion laws ($\gamma : [0,1] \to \mathcal{U}$, where $\mathcal{U}$ is robot's action space).

It includes dynamics.

$$[T_0, T_f] \ni t \rightsquigarrow \tau \in [0,1] : q(t) = \pi(\tau) \in \mathcal{C}_{free}$$

The planning problem is determination of the function $\pi(\cdot)$.

Additional requirements can be given:

- Smoothness of the path
- Kinodynamic constraints

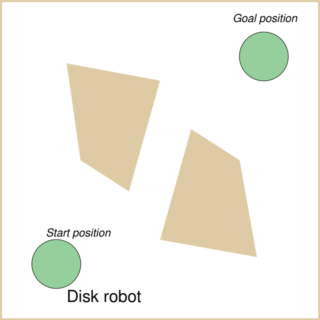
E.g., considering friction forces

- Optimality criterion

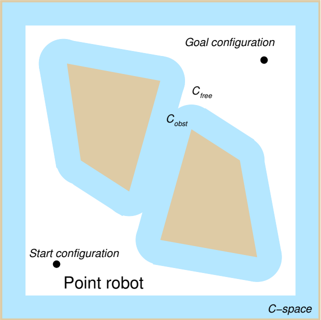
shortest vs fastest (length vs curvature)

Planning in $\mathcal{C}$-space

Robot motion planning robot for a disk robot with a radius ρ.



Motion planning problem in geometrical representation of $\mathcal{W}$

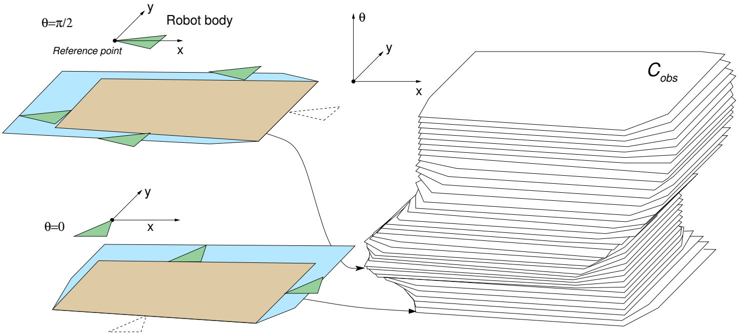


Motion planning problem in $\mathcal{C}$-space representation

$\mathcal{C}$-space has been obtained by enlarging obstacles by the disk $\mathcal{A}$ with the radius ρ.

By applying Minkowski sum: $\mathcal{O} \oplus \mathcal{A} = \{x + y \mid x \in \mathcal{O}, y \in \mathcal{A}\}$.

Example of $\mathcal{C}_{obs}$ for a Robot with Rotation



A simple 2D obstacle → has a complicated $\mathcal{C}_{obs}$

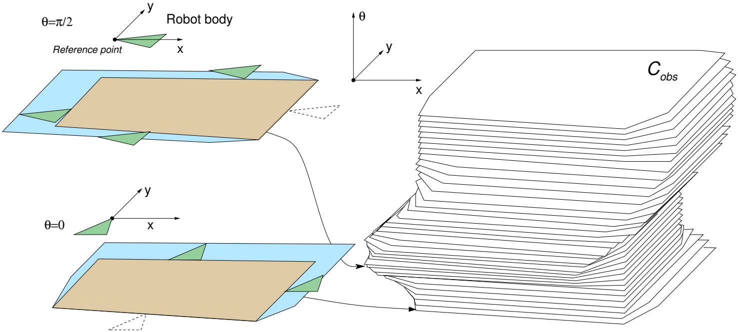
- Deterministic algorithms exist

Requires exponential time in $\mathcal{C}$ dimension,

J. Canny, PAMI, 8(2):200–209, 1986

- Explicit representation of $\mathcal{C}_{free}$ is impractical to compute.

Example of $\mathcal{C}_{obs}$ for a Robot with Rotation



A simple 2D obstacle → has a complicated $\mathcal{C}_{obs}$

- Deterministic algorithms exist

Requires exponential time in $\mathcal{C}$ dimension,

J. Canny, PAMI, 8(2):200–209, 1986

- Explicit representation of $\mathcal{C}_{free}$ is impractical to compute.

Representation of $\mathcal{C}$-space

How to deal with continuous representation of $\mathcal{C}$-space?

Continuous Representation of $\mathcal{C}$-space

↓

Discretization

processing critical geometric events, (random) sampling

roadmaps, cell decomposition, potential field

↓

Graph Search Techniques

BFS, Gradient Search, A*

Representation of $\mathcal{C}$-space

How to deal with continuous representation of $\mathcal{C}$-space?

Continuous Representation of $\mathcal{C}$-space

↓

Discretization

processing critical geometric events, (random) sampling

roadmaps, cell decomposition, potential field

↓

Graph Search Techniques

BFS, Gradient Search, A^*

Planning Methods Overview

(selected approaches)

- Roadmap based methods

Create a connectivity graph of the free space.

    - Visibility graph
    - Cell decomposition
    - Voronoi diagram

- Potential field methods

Classic path planning algorithms

Randomized path/motion planning approaches

- Probabilistic roadmaps (PRM)
- Expansive-Spaces Tree (EST)
- Rapidly-Exploring Random Tree (RRT)

Allow to consider kinodynamic constraints.

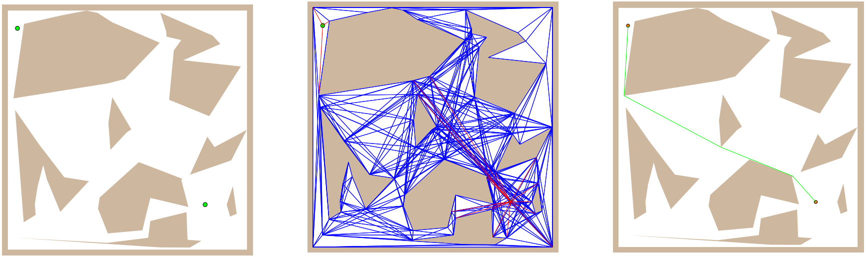
- Optimal sampling based Planner - RRT*

S. Karaman and E. Frazzoli, IJJR, 30(7):846-894, 2011

Visibility Graph

1. Compute visibility graph
2. Find the shortest path

E.g., by Dijkstra's algorithm



Problem

Visibility graph

Found shortest path

Constructions of the visibility graph:

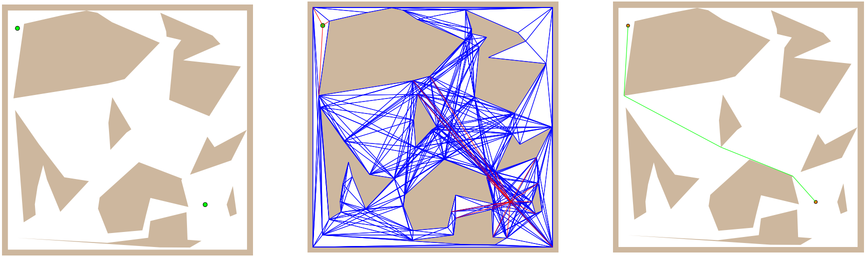
- Naïve – all segments between *n* vertices of the map $O(n^3)$
- Using rotation trees for a set of segments – $O(n^2)$

M. H. Overmars and E. Welzl, 1988

Visibility Graph

1. Compute visibility graph
2. Find the shortest path

E.g., by Dijkstra's algorithm



Problem Visibility graph Found shortest path

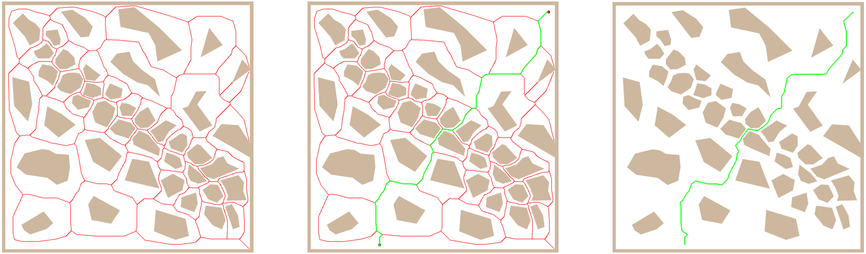
Constructions of the visibility graph:

- Naïve – all segments between n vertices of the map $O(n^3)$
- Using rotation trees for a set of segments – $O(n^2)$

M. H. Overmars and E. Welzl, 1988

Voronoi Diagram

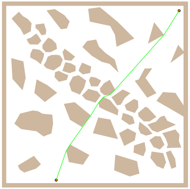
1. Roadmap is Voronoi diagram that maximizes clearance from the obstacles
2. Start and goal positions are connected to the graph
3. Path is found using a graph search algorithm



Voronoi diagram Path in graph Found path

Visibility Graph vs Voronoi Diagram

Visibility graph

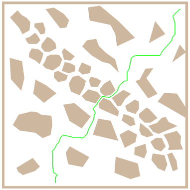


- Shortest path, but it is close to obstacles. We have to consider safety of the path.

An error in plan execution can lead to a collision.

- Complicated in higher dimensions

Voronoi diagram



- It maximize clearance, which can provide conservative paths
- Small changes in obstacles can lead to large changes in the diagram
- Complicated in higher dimensions

A combination is called Visibility-Voronoi – R. Wein, J. P. van den Berg, D. Halperin, 2004

For higher dimensions we need other roadmaps.

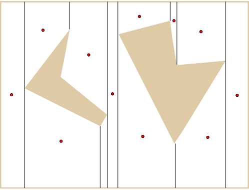
Cell Decomposition

1. Decompose free space into parts.

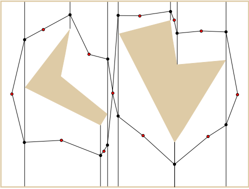
   Any two points in a convex region can be directly connected by a segment.

2. Create an adjacency graph representing the connectivity of the free space.
3. Find a path in the graph.

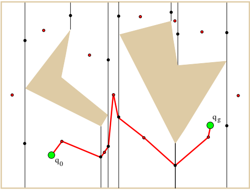
Trapezoidal decomposition



Centroids represent cells



Connect adjacency cells



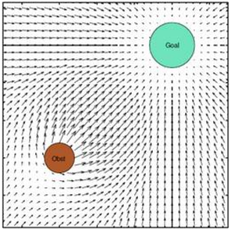
Find path in the adjacency graph

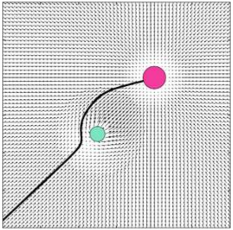
Other decomposition (e.g., triangulation) are possible.

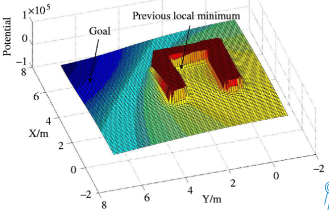
Artificial Potential Field Method

- The idea is to create a function f that will provide a direction towards the goal for any configuration of the robot.
- Such a function is called navigation function and $-\nabla f(q)$ points to the goal.
- Create a potential field that will attract robot towards the goal q_f while obstacles will generate repulsive potential repelling the robot away from the obstacles.

The navigation function is a sum of potentials.







Such a potential function can have several local minima.

Avoiding Local Minima in Artificial Potential Field

- Consider harmonic functions that have only one extremum

$$\nabla^2 f(q) = 0$$

- Finite element method

Dirichlet and Neumann boundary conditions

J. Mačák, Master thesis, CTU, 2009

Probabilistic Roadmaps

A discrete representation of the continuous $\mathcal{C}$-space generated by randomly sampled configurations in $\mathcal{C}_{free}$ that are connected into a graph.

- **Nodes** of the graph represent admissible configuration of the robot.
- **Edges** represent a feasible path (trajectory) between the particular configurations.

Probabilistic complete algorithms: with increasing number of samples an admissible solution would be found (if exists)

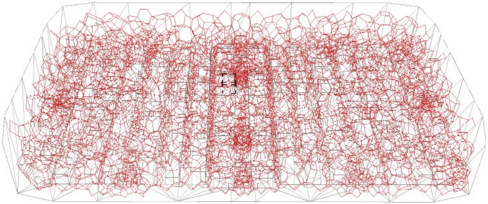
Having the graph, the final path (trajectory) is found by a graph search technique.

Probabilistic Roadmaps

A discrete representation of the continuous $\mathcal{C}$-space generated by randomly sampled configurations in $\mathcal{C}_{free}$ that are connected into a graph.

- **Nodes** of the graph represent admissible configuration of the robot.
- **Edges** represent a feasible path (trajectory) between the particular configurations.

Probabilistic complete algorithms: with increasing number of samples an admissible solution would be found (if exists)



Having the graph, the final path (trajectory) is found by a graph search technique.

Probabilistic Roadmap Strategies

Multi-Query

- Generate a single roadmap that is then used for planning queries several times.
- An representative technique is PRM

Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces
Lydia E. Kavraki and Petr Svestka and Jean-Claude Latombe and Mark H. Overmars,
IEEE Transactions on Robotics and Automation, 12(4):566–580, 1996.

Single-Query

- For each planning problem constructs a new roadmap to characterize the subspace of $\mathcal{C}$-space that is relevant to the problem.
- One of such techniques is Rapidly-Exploring Random Tree (RRT)

Steven M. LaValle, 1998

Multi-Query Strategy

1. Learning phase
    1.1 Find random samples of $\mathcal{C}$-space (consider only the free configurations)
    1.2 Connect the random configurations $q \in \mathcal{C}_{free}$ using a local planner
        - A connection (edge) between two nodes must represent an admissible path between the configurations
        - Consider only nodes within ϵ>0 distance from the node

          For small ϵ a straight line segment can be considered.
        - Collision detection can be performed for several configurations between the configurations being connected.

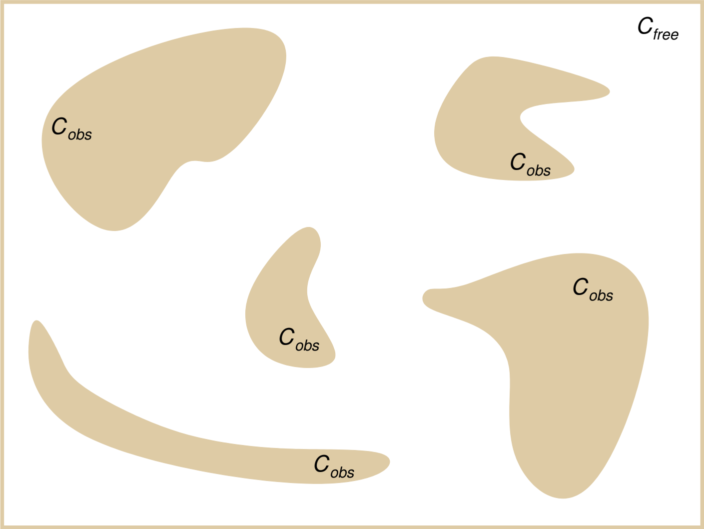
          E.g., sampling of the connecting straight line segment.
2. Query phase
    2.1 Connect start and goal configurations with the PRM

    E.g., using a local planner

    2.2 Use the graph search to find the path

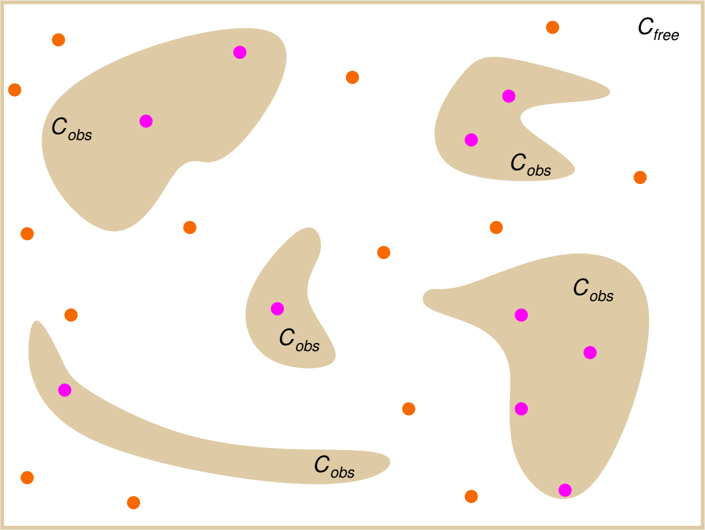
PRM Construction

Given problem domain



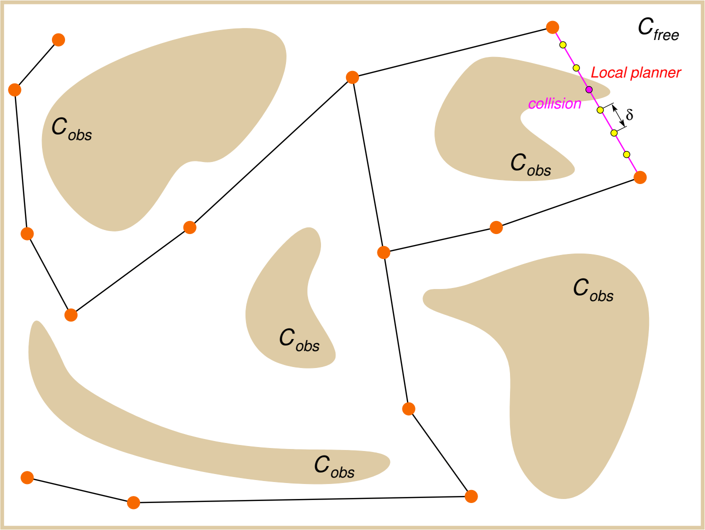
PRM Construction

Random configuration



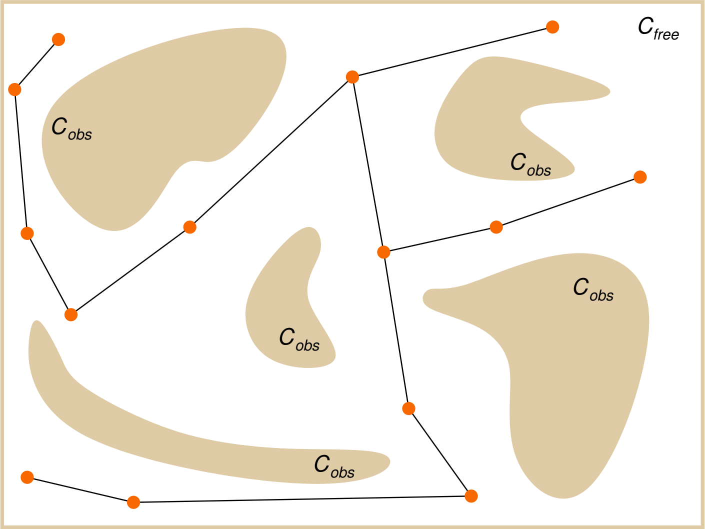
PRM Construction

Connecting random samples



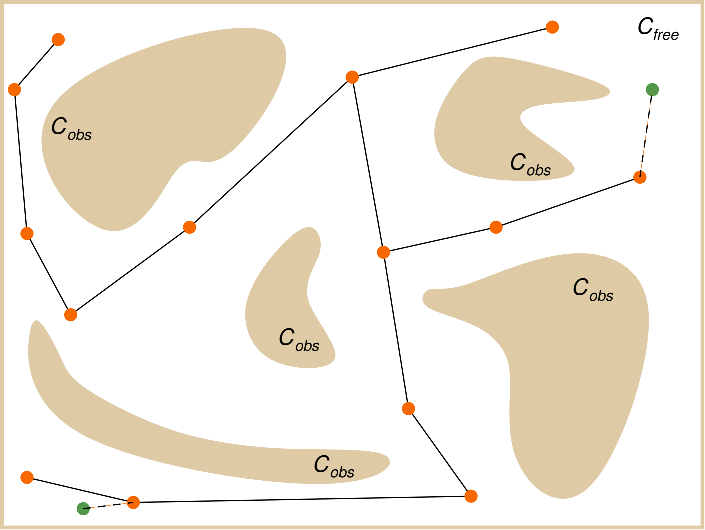
PRM Construction

Connected roadmap



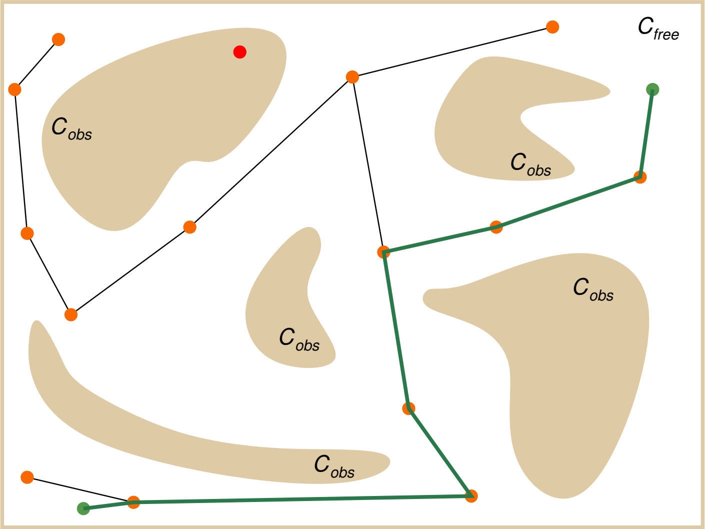
PRM Construction

Query configurations



PRM Construction

Final found path



Notes about Random Sampling

- A solution can be found using only a few samples.

Do you know the Oraculum? (from Alice in Wonderland)

- Sampling strategies are important
  - Near obstacles
  - Narrow passages
  - Grid-based
  - Uniform sampling must be carefully considered.

James J. Kuffner, Effective Sampling and Distance Metrics for 3D Rigid Body Path Planning, ICRA, 2004.

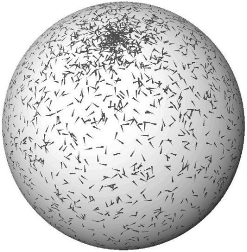
Notes about Random Sampling

- A solution can be found using only a few samples.

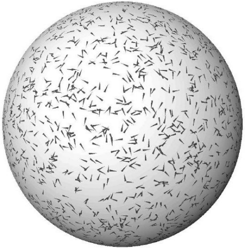
Do you know the Oraculum? (from Alice in Wonderland)

- Sampling strategies are important
  - Near obstacles
  - Narrow passages
  - Grid-based
  - Uniform sampling must be carefully considered.

James J. Kuffner, Effective Sampling and Distance Metrics for 3D Rigid Body Path Planning, ICRA, 2004.



Naïve sampling



Uniform sampling of SO(3) using Euler angles

Rapidly Exploring Random Tree (RRT)

It incrementally builds a graph (tree) towards the goal area.

It does not guarantee precise path to the goal configuration.

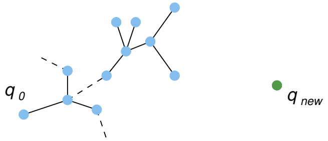
1. Start with the initial configuration q_0, which is a root of the constructed graph
2. Generate a new random configuration q_{new} in $\mathcal{C}_{free}$
3. Find the closest node q_{near} to q_{new} in the tree

   E.g., using KD-tree implementation like ANN or FLANN libraries
4. Extend q_{near} towards q_{new}

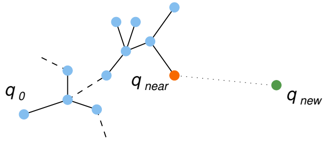
   Extend tree by a small step, but often a direct control $u \in \mathcal{U}$ that will move robot the position closest to q_{new} is selected (applied for δt).
5. Go to Step 2, until the tree is within a sufficient distance from the goal configuration

Or terminates after dedicated running time.

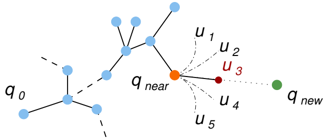
RRT Construction



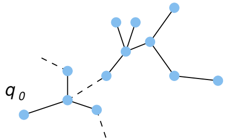
#1 new random configuration



#2 the closest node



#3 possible actions from q_{near}



#4 extended tree

Properties of RRT Algorithms

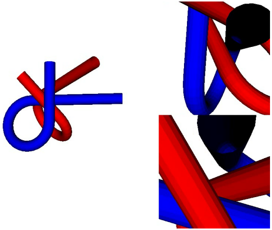
- Rapidly explores the space

  q_{new} will more likely be generated in large not yet covered parts.
- Allows considering kinodynamic/dynamic constraints (during the expansion).
- Can provide trajectory or a sequence of direct control commands for robot controllers.
- A collision detection test is usually used as a "black-box".

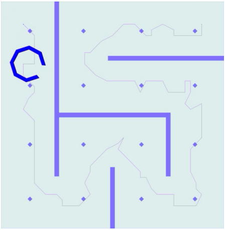
  E.g., RAPID, Bullet libraries.
- Similarly to PRM, RRT algorithms have poor performance in narrow passage problems.
- RRT algorithms provides feasible paths.

  It can be relatively far from optimal solution, e.g., according to the length of the path.
- Many variants of RRT have been proposed.

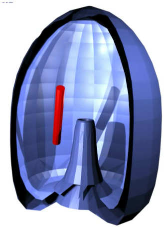
RRT – Examples 1/2



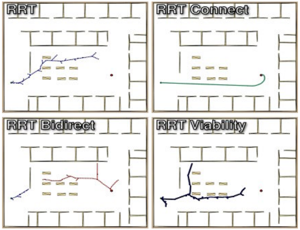
Alpha puzzle benchmark



Apply rotations to reach the goal



Bugtrap benchmark

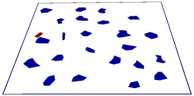


Variants of RRT algorithms

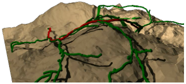
Courtesy of V. Vonásek and P. Vaněk

RRT – Examples 2/2

- Planning for a car-like robot

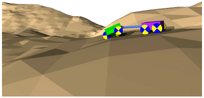


- Planning on a 3D surface



- Planning with dynamics

(friction forces)



Courtesy of V. Vonásek and P. Vaněk

Car-Like Robot

- Configuration

$$\vec{\boldsymbol{x}} = \begin{pmatrix} x \\ y \\ \phi \end{pmatrix}$$

position and orientation

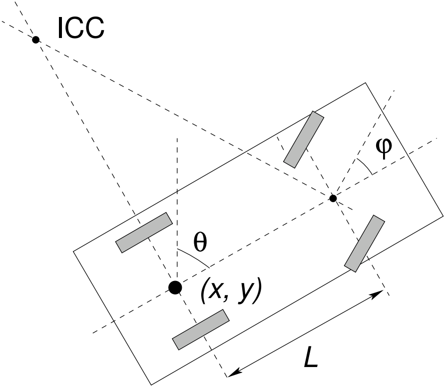
- Controls

$$\vec{\boldsymbol{u}} = \begin{pmatrix} v \\ \varphi \end{pmatrix}$$

forward velocity, steering angle

- System equation

$$\begin{aligned} \dot{x} &= v\cos\phi \\ \dot{y} &= v\sin\phi \\ \dot{\varphi} &= \frac{v}{L}\tan\varphi \end{aligned}$$



Kinematic constraints $\dim(\vec{\boldsymbol{u}}) < \dim(\vec{\boldsymbol{x}})$

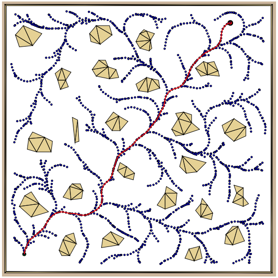
Differential constraints on possible $\dot{q}$*:*

$$\dot{x}\sin(\phi) - \dot{y}\cos(\phi) = 0$$

Control-Based Sampling

- Select a configuration q from the tree T of the current configurations
- Pick a control input $\vec{\boldsymbol{u}} = (v, \varphi)$ and integrate system (motion) equation over a short period

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \varphi \end{pmatrix} = \int_{t}^{t+\Delta t} \begin{pmatrix} v \cos \phi \\ v \sin \phi \\ \frac{v}{L} \tan \varphi \end{pmatrix} dt$$



- If the motion is collision-free, add the endpoint to the tree

E.g., considering k configurations for $k\delta t = dt$.

Maneuvers Based Planning

- Dubins car
  - Constant velocity: $v(t) = v$
  - Steering angle: $\varphi \in [-\Phi, \Phi]$
- There are **6** types of trajectories connecting any configuration in $\mathbb{R}^2 \times \mathbb{S}^1$ (without obstacles)

Dubins curves

$$\{LRL, RLR, LSL, LSR, RSL, RSR\},$$

where L – left turn, R – right turn, S – straight ahead.

L. E. Dubins, 1957

- Reeds-Shepp car – reverse direction allowed (46 types of trajectories)

J. A. Reeds, L. A. Shepp, 1990

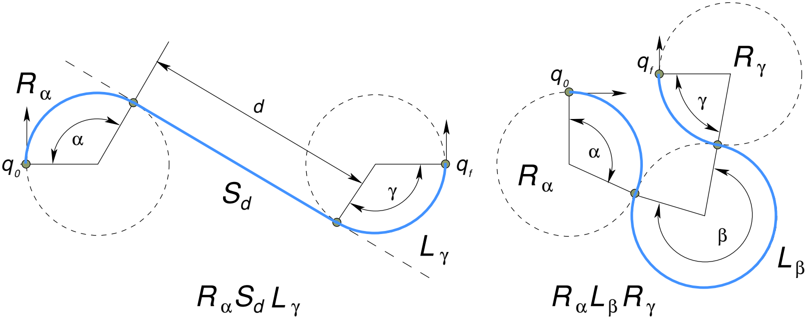
Parametrization of Dubins Curves

- Parametrization of each trajectory phase:

$$\{L_\alpha R_\beta L_\gamma, R_\alpha L_\beta R_\gamma, L_\alpha S_d L_\gamma, L_\alpha S_d R_\gamma, R_\alpha S_d L_\gamma, R_\alpha S_d R_\gamma\}$$

for $\alpha \in [0, 2\pi)$, $\beta \in (\pi, 2\pi)$, $d \geq 0$

Notice the prescribed orientation at q_0 and q_f.



Testing all 6 types of curves, we can define a metric for the RRT.

Efficient Sampling–Based Motion Planning

- PRM and RRT are theoretically probabilistic complete
- They provide a feasible solution without quality guarantee

  Despite that, they are successfully used in many practical applications
- In 2011, a study of the asymptotic behaviour has been published

  It shows, that in some case, they converges to a noon-optimal value with a probability 1.
- Based on the study, new algorithms have been proposed: RRG and optimal RRT (RRT^{star})

Sampling-based algorithms for optimal motion planning
Sertac Karaman, Emilio Frazzoli
International Journal of Robotic Research, 30(7):846–894, 2011.

`http://sertac.scripts.mit.edu/rrtstar`

Efficient Sampling–Based Motion Planning

- PRM and RRT are theoretically probabilistic complete
- They provide a feasible solution without quality guarantee

  Despite that, they are successfully used in many practical applications
- In 2011, a study of the asymptotic behaviour has been published

  It shows, that in some case, they converges to a noon-optimal value with a probability 1.
- Based on the study, new algorithms have been proposed: RRG and optimal RRT (RRT^{star})

Sampling-based algorithms for optimal motion planning
Sertac Karaman, Emilio Frazzoli
International Journal of Robotic Research, 30(7):846–894, 2011.

`http://sertac.scripts.mit.edu/rrtstar`

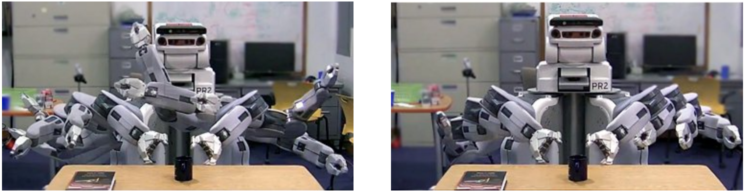
Efficient Sampling–Based Motion Planning

- PRM and RRT are theoretically probabilistic complete
- They provide a feasible solution without quality guarantee

  Despite that, they are successfully used in many practical applications
- In 2011, a study of the asymptotic behaviour has been published

  It shows, that in some case, they converges to a noon-optimal value with a probability 1.
- Based on the study, new algorithms have been proposed: RRG and optimal RRT (RRT^{star})

Sampling-based algorithms for optimal motion planning
Sertac Karaman, Emilio Frazzoli
International Journal of Robotic Research, 30(7):846–894, 2011.



`http://sertac.scripts.mit.edu/rrtstar`

Inspection/Surveillance Planning

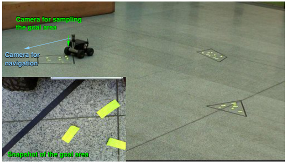
Mission: *Visit a set of goal areas as frequently as possible.*

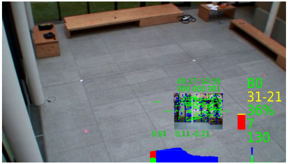
Not only shortest path, but also precise navigation to the goal areas.

- Planning multi-goal path considering model of navigation
- The problem is to find not only a short path but also a path that will improve localization precision

  How precisely the goal is visited

- Mathematical model of localization uncertainty evaluation
- The model is integrated into path planning algorithm





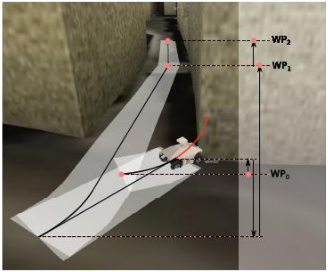
Courtesy of T. Krajník

Adaptive Path Planner

Context: *Cooperative surveillance mission*

- Overall roadmap for a high-level path planning
- A low-level motion planning using maneuvers (Dubins curves)
- Replanning if a possible collision is detected

AgentScout project



E.g., new road block is detected

Summary

- Introduction to motion planning
- Overview of sampling-based planning methods
  - Basic roadmap methods
    - Visibility graph
    - Voronoi diagram
    - Cell decomposition
  - Artificial potential field method
  - Probabilistic roadmap methods (PRM, RRT, RRT*)
- Maneuvers – Dubin’s curves

Are you interested in motion planning or in robotics?

Looking for motivated students for bachelor, master or doctoral theses. Contact me via faiglj@fel.cvut.cz