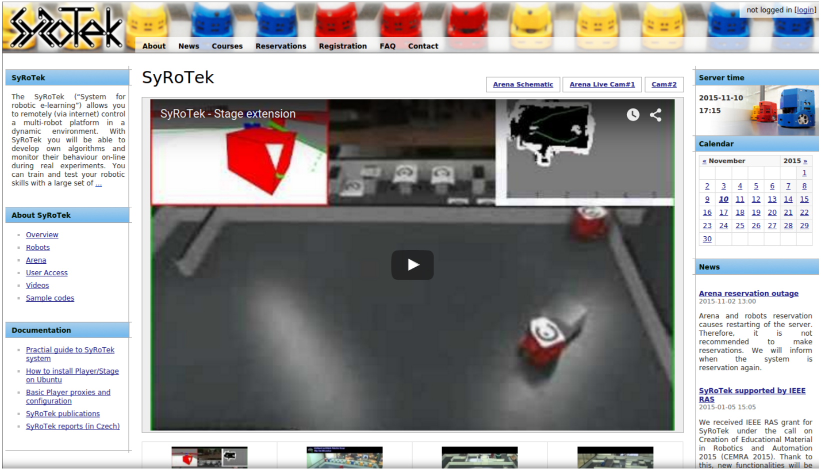
Planning in Robotics - Part I

Miroslav Kulich

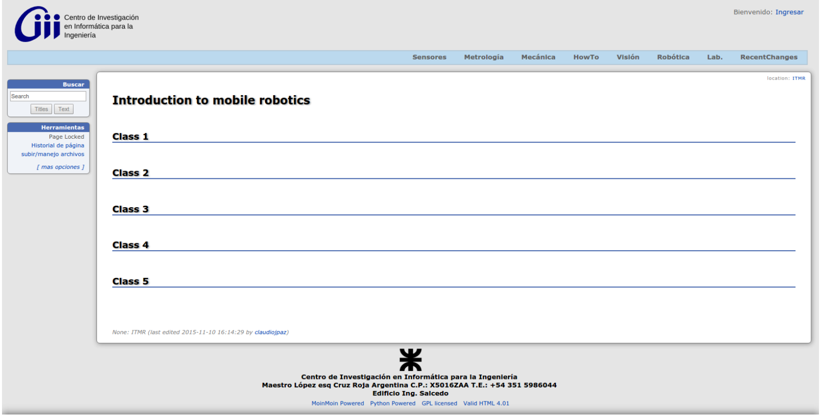
Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
Intelligent and Mobile Robotics Group

http://imr.ciirc.cvut.cz/people/Mirek

http://syrotek.felk.cvut.cz



http://ciii.frc.utn.edu.ar/ITMR



Lecture outline

- Few terms and definitions
- Configuration space
- Roadmaps
  - Visibility graph
  - Cell decomposition
  - Voronoi diagrams
- Potential field
- Probabilistic methods
  - Probabilistic roadmaps
  - Rapidly Exploring Random Trees
  - Local planning/obstacle avoidance

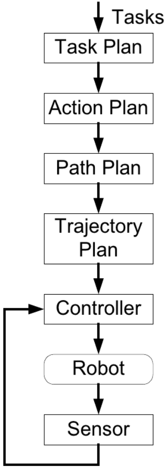
Terminology: path vs. trajectory

- Often confused and used as synonyms informally.
- Path: ordered locus of points in the space (either joint or operational) which the manipulator should follow. Path is a pure geometric description of motion.
- Trajectory: a path on which timing law is specified, e.g., velocities and accelerations in its each point.

hedgehog puzzle mars

Robot Motion Planning

- Path planning
  - Geometric path.
  - Issues: obstacle avoidance, shortest path.
- Trajectory generating
  - "Interpolate" or "approximate" the desired path by a class of polynomial functions and
  - Generate a sequence of time-based "control set points" for the control of manipulator from the initial configuration to its destination.



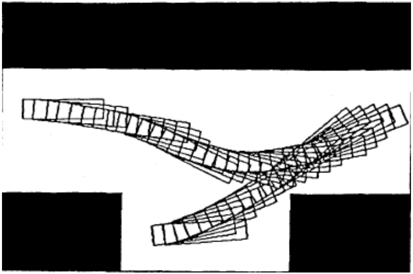
Holonomicity in robotics

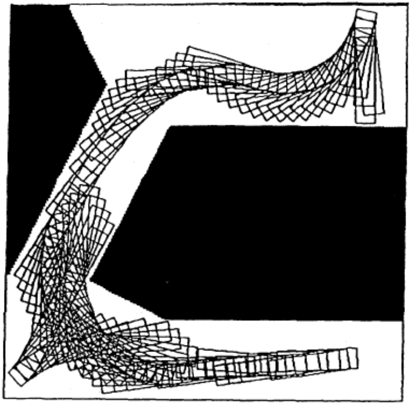
- Holonomicity refers to the relationship between the controllable and total degrees of freedom of a given robot (or part thereof).
- Holonomic: if the controllable degrees of freedom is equal to the total degrees of freedom.
- Non-holonomic: if the controllable degrees of freedom are less than the total degrees of freedom.
- Redundant robot: if it has more controllable degrees of freedom than degrees of freedom in its task space.

Example: A car = non-holonomic

- Three degrees of freedom: its position in two axes, and its orientation relative to a fixed heading.
- Only two controllable degrees of freedom: acceleration/braking and the angle of the steering wheel.
- A car's heading (the direction in which it is traveling) must remain aligned with the orientation of the car, or 180° from it if the car is in reverse. It has no other allowable direction, assuming there is no skidding or sliding. Thus, not every path in phase space is achievable.

Path for car-like robot





A human arm is holonomic

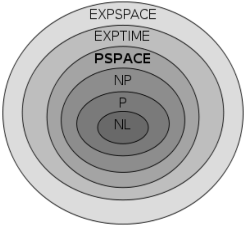
- A human arm is a holonomic.
- It is a redundant system because it has 7 degrees of freedom (3 in the shoulder - rotations about each axis, 2 in the elbow - bending and rotation about the lower arm axis, and 2 in the wrist, bending up and down (i.e. pitch), and left and right (i.e. yaw)).
- There are only 6 physical degrees of freedom in the task of placing the hand (x, y, z, roll, pitch and yaw), while fixing the seven degrees of freedom fixes the hand.

Basic problem

- Problem statement
  - Compute a collision-free path for a rigid or articulated moving object among static obstacles.
- Input
  - Geometry of a moving object (a robot, a digital actor, or a molecule) and obstacles.
  - How does the robot move?
  - Kinematics of the robot (degrees of freedom).
  - Initial and goal robot configurations (positions & orientations).
- Output
  - Continuous sequence of collision-free robot configurations connecting the initial and goal configurations.

Hardness results

- Several variants of the path planning problem have been proven to be PSPACE-hard.
- A complete algorithm may take exponential time. (A complete algorithm finds a path if one exists and reports no path exists otherwise).



Completeness in motion planning

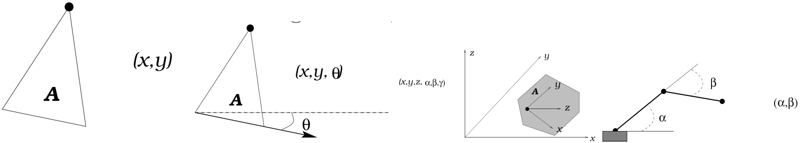
- Exact
  - Usually computationally expensive.
  - May determine bounds of a problem's complexity.
- Heuristic
  - Aimed at generating a solution in a short time.
  - May fail to find solution or find poor one at difficult problems.
  - Important in engineering applications.
- Resolution complete (discretization).
- Probabilistically complete (probabilistic completeness $\rightsquigarrow 1$).

Scope of motion planning algorithms

- Global
  - Take into account all environment information.
  - Plan a motion from start to goal configuration.
- Local
  - Avoid obstacles in the vicinity of the robot.
  - Use information about nearby obstacles only.
  - Used when start and goal are close together.
  - Used as component in global planner, or
  - Used as safety feature to avoid unexpected obstacles not present in environment model, but sensed during motion.

Configuration space

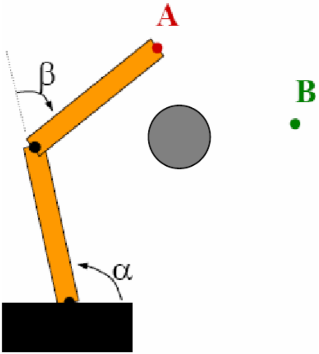
- A key concept for motion planning is a configuration: a complete specification of the position of every point in the system
- A simple example: a robot that translates but does not rotate in the plane: what is a sufficient representation of its configuration?

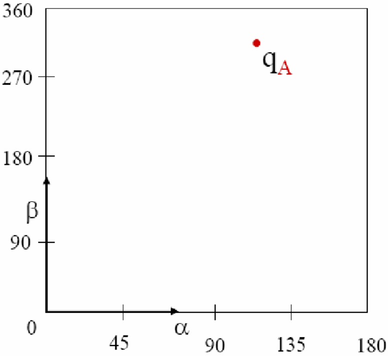


Notation

- The space of all possible configurations is the configuration space or C-space.
- The dimension of C-space $=$ the number of parameters representing a configuration (degree of freedom).
- Workspace is either the ambient space, or the set of reachable points by an end-effector $\mathcal{W}$ (Euclidean 2D, 3D).
- Robot $\mathcal{A}$: compact subset of R^n.
- Region $\mathcal{W}$ taken by a robot in a configuration q: $R(q)$.
- Examples:
  - Rotating bar fixed at a point
  - A rotating bar that translates along the rotation axis
  - What is its C-space?
  - What is its workspace?

Configuration space - manipulator





- Suppose and obstacle in the robot workspace.
- Where can we put q_B?

Obstacles in C-space

- Let q denote a point in a configuration space Q.
- The path planning problem is to find a mapping $c : [0, 1] \to Q$ so that no configuration along the path intersects an obstacle.
- Obstacle in a workspace: $\mathcal{O}$
- A configuration space obstacle Q_O is the set of configurations q at which the robot $R(q)$ intersects $\mathcal{O}_i$:

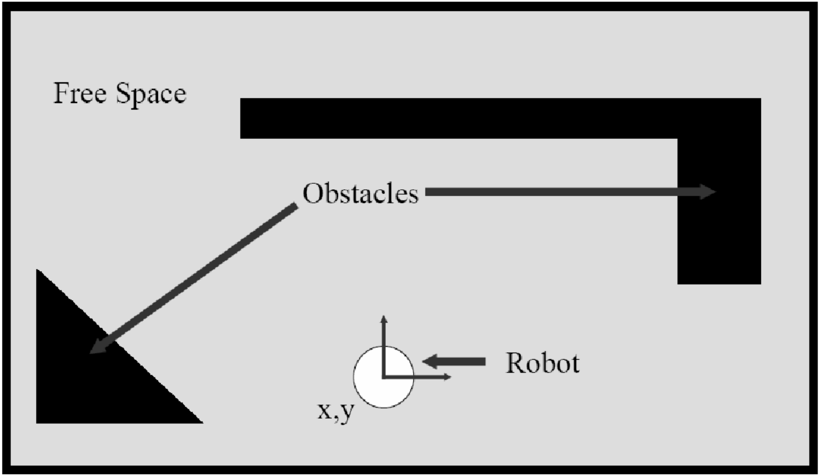
$$Q_{\mathcal{O}_i} = \{q \in Q | R(q) \cap \mathcal{O}_i \neq \emptyset\}$$

- The free configuration space (free space) Q_{free} is

$$Q_{free} = Q - (Q_{O_i})$$

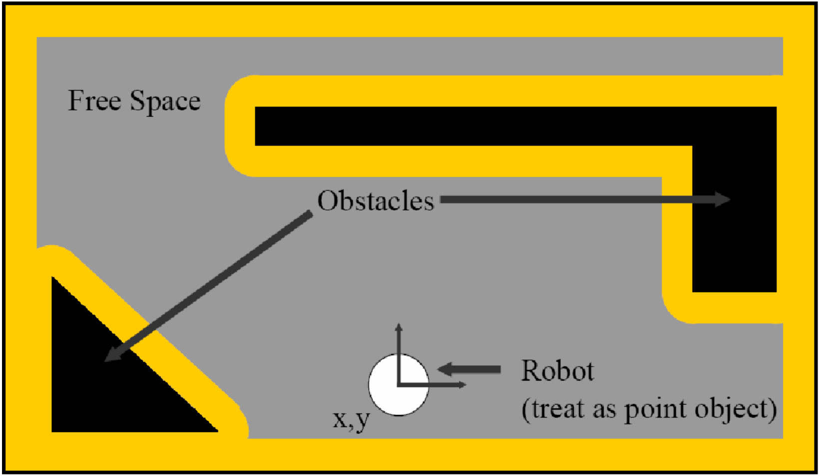
- The free space is generally an open set.
- A free path is a mapping $c : [0, 1] \to Q_{free}$
- A semifree path is a mapping $c : [0, 1] \to cl(Q_{free})$ (cl stands for closure)

Example - a circular robot

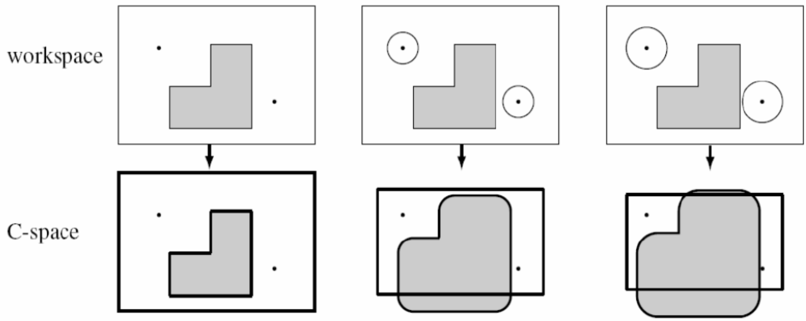


Example - configuration space

(Accommodation of robot size)



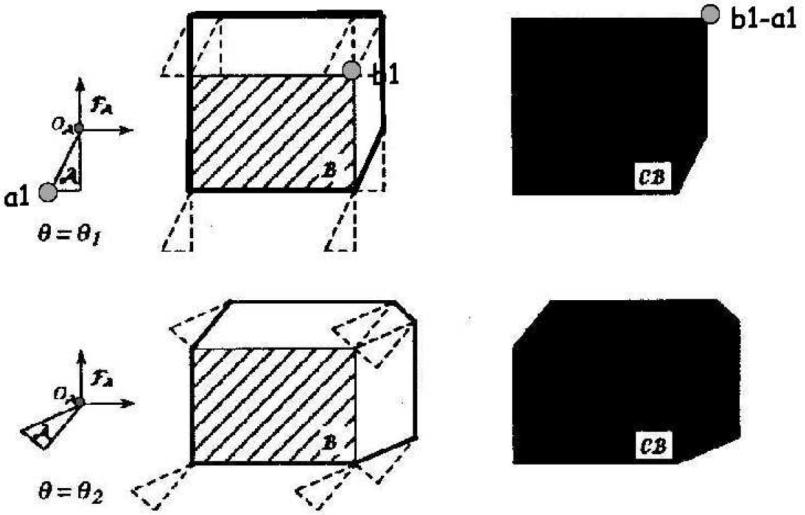
Trace the boundary of the workspace



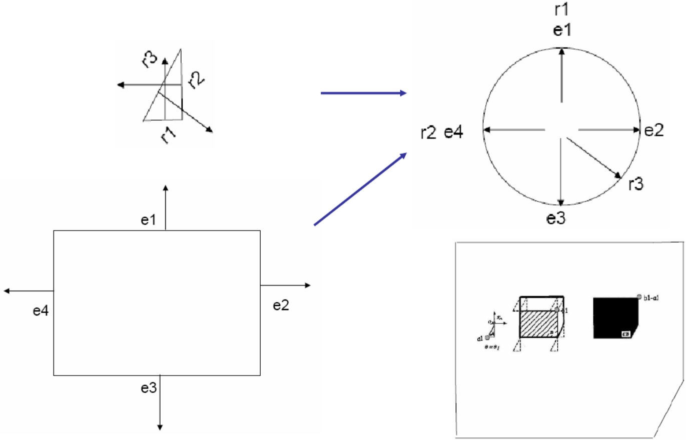
- A consistent reference point must be picked on the robot.
- What about non-circular robots?

When only translation is allowed

- For a fixed robot angle, we can build Q_{O_i}
- Choice of the reference point makes a difference.

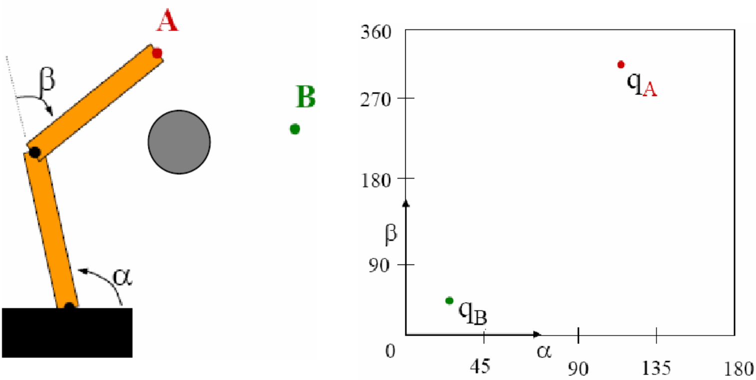


Star algorithm



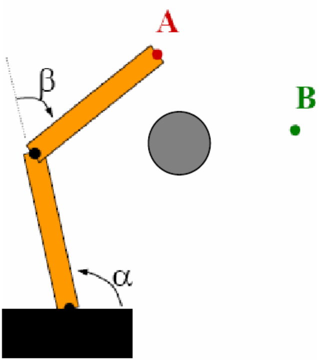
- Complexity: $O(m+n)$
- If one polygon is non-convex then complexity is $O(mn)$
- If both polygons are non-convex then complexity is $O((mn)^2)$

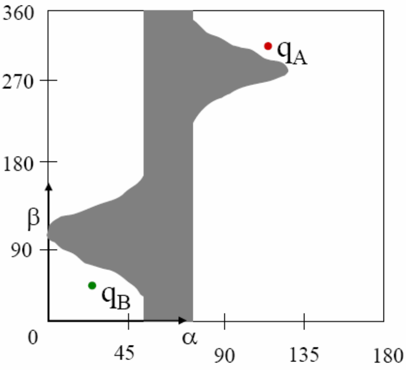
Obstacles for a manipulator arm



- Suppose and obstacle in the robot workspace.
- Where can we put q_B?

Configuration space obstacle

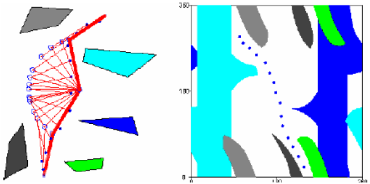


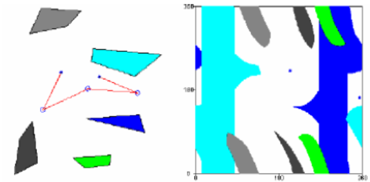


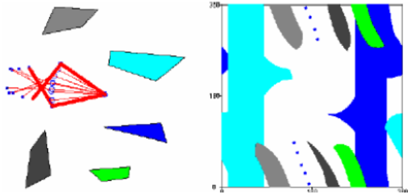
The C-space representation of this obstacle…

- Suppose and obstacle in the robot workspace.
- Where can we put q_B?

Two-link path

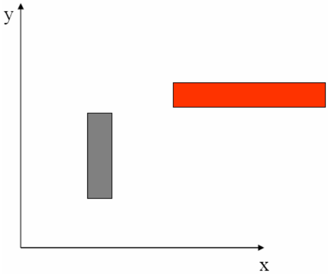


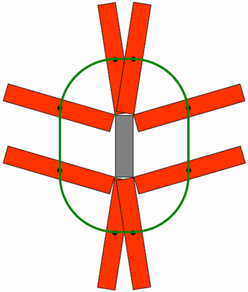




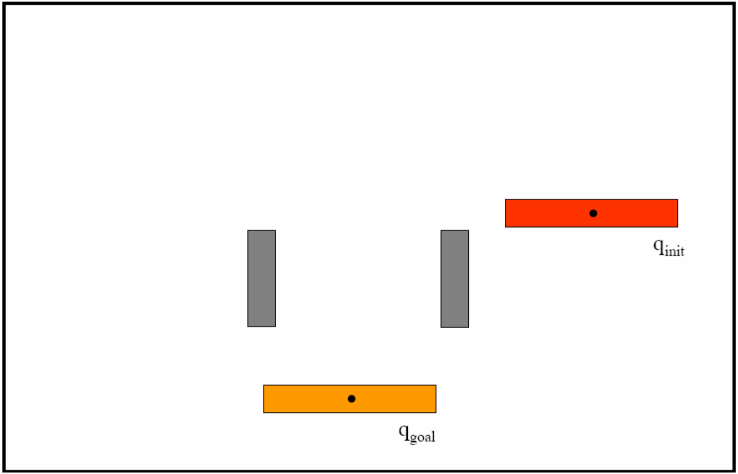
Additional dimensions

- If the robot can both translate and rotate.
- What would the configuration of the red rectangular robot look like?
- Naïve solution: $2D$

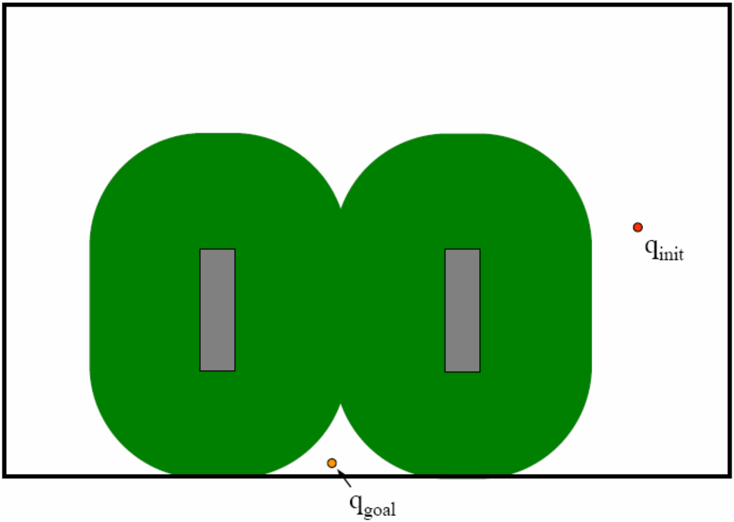




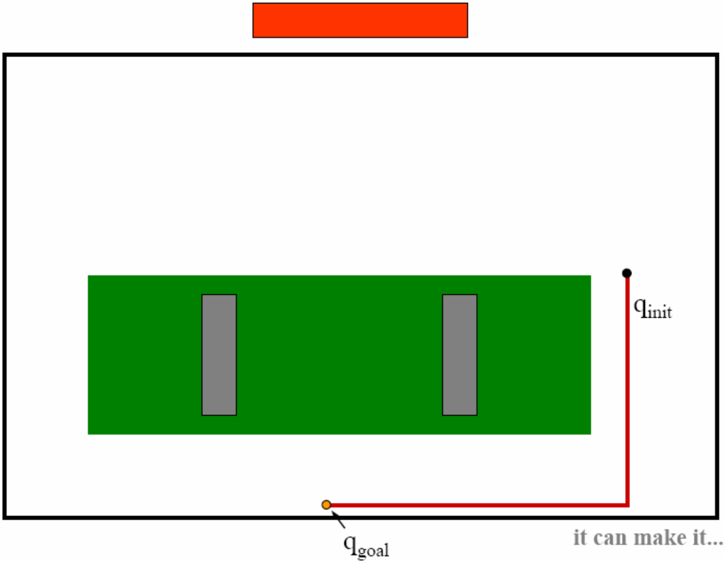
A serious problem?



A serious problem?

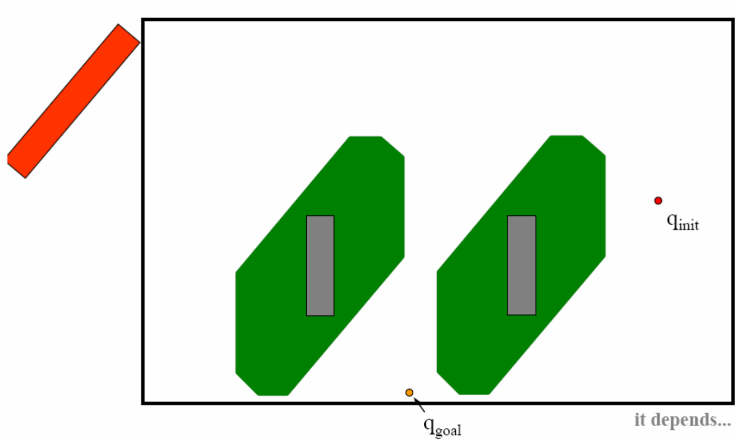


When the robot is at one orientation...



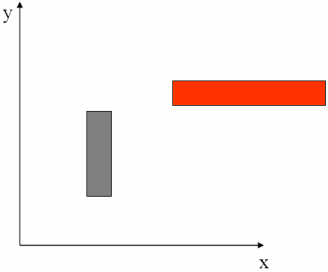
it can make it...

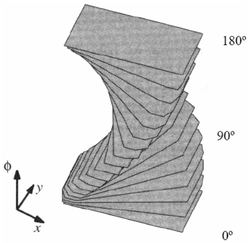
and the robot at another orientation...



Additional dimensions

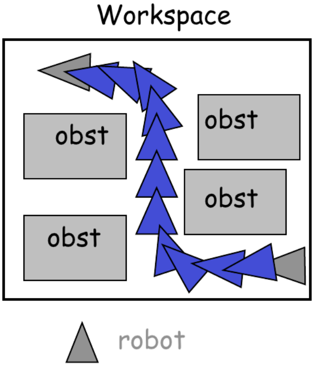
- If the robot can both translate and rotate.
- What woud the configuration of the red rectangular robot look like?
- Configuration space is 3D.



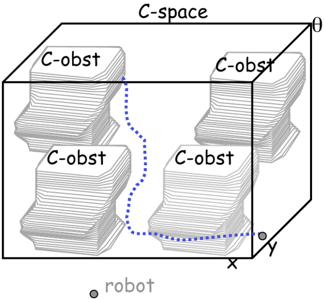


Motion planning in C-space

- Simple workspace obstacle transformed into complicated C-obstacle!



Plan is swept volume



Path is 1D curve

Configuration space - conclusion

- Reduction of path planning problem for n-dimensional robot in Euclidean space to path finding for a point robot in C-space.
- Unified approach to solving a large family of planning problems.
- Almost everyone use it ...
- Generally, the planning problem is hard:
    - Exponential time w.r.t. the number of C-space dimensions.
    - Polynomial time w.r.t. complexity of obstacles.
- Two theoretical methods:
    - Exact decomposition based on cylindrical decomposition (Schwartz, Sharir).
    - Roadmaps $O(2^d)$ (Canny)
- The complexities hold for ,,common" problems.
- Solution: simplification, approximation.