

Robotic Motion Planning: Configuration Space

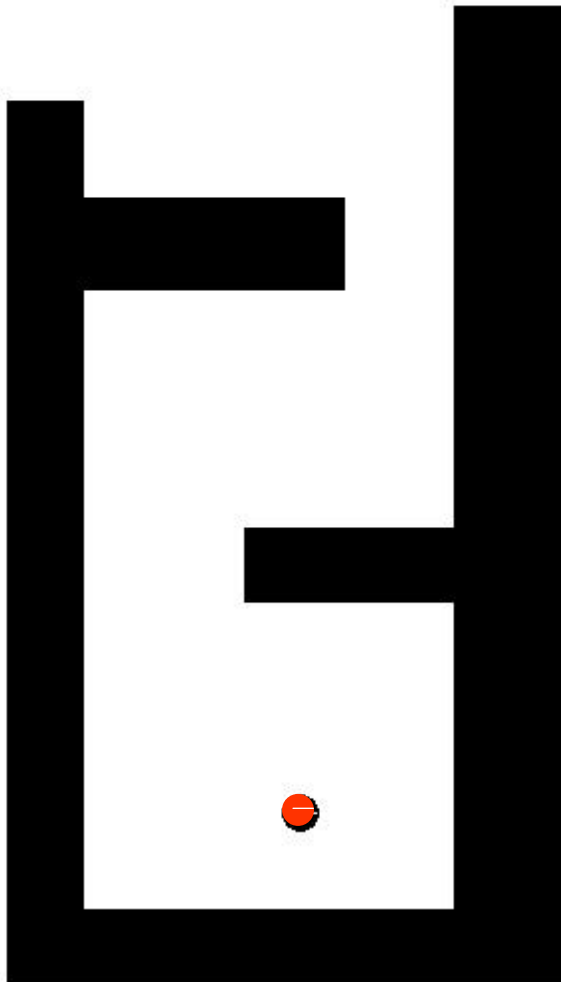
Robotics Institute 16-735

<http://www.cs.cmu.edu/~motionplanning>

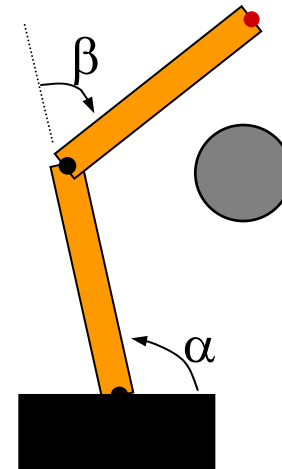
Howie Choset

<http://www.cs.cmu.edu/~choset>

What if the robot is not a point?

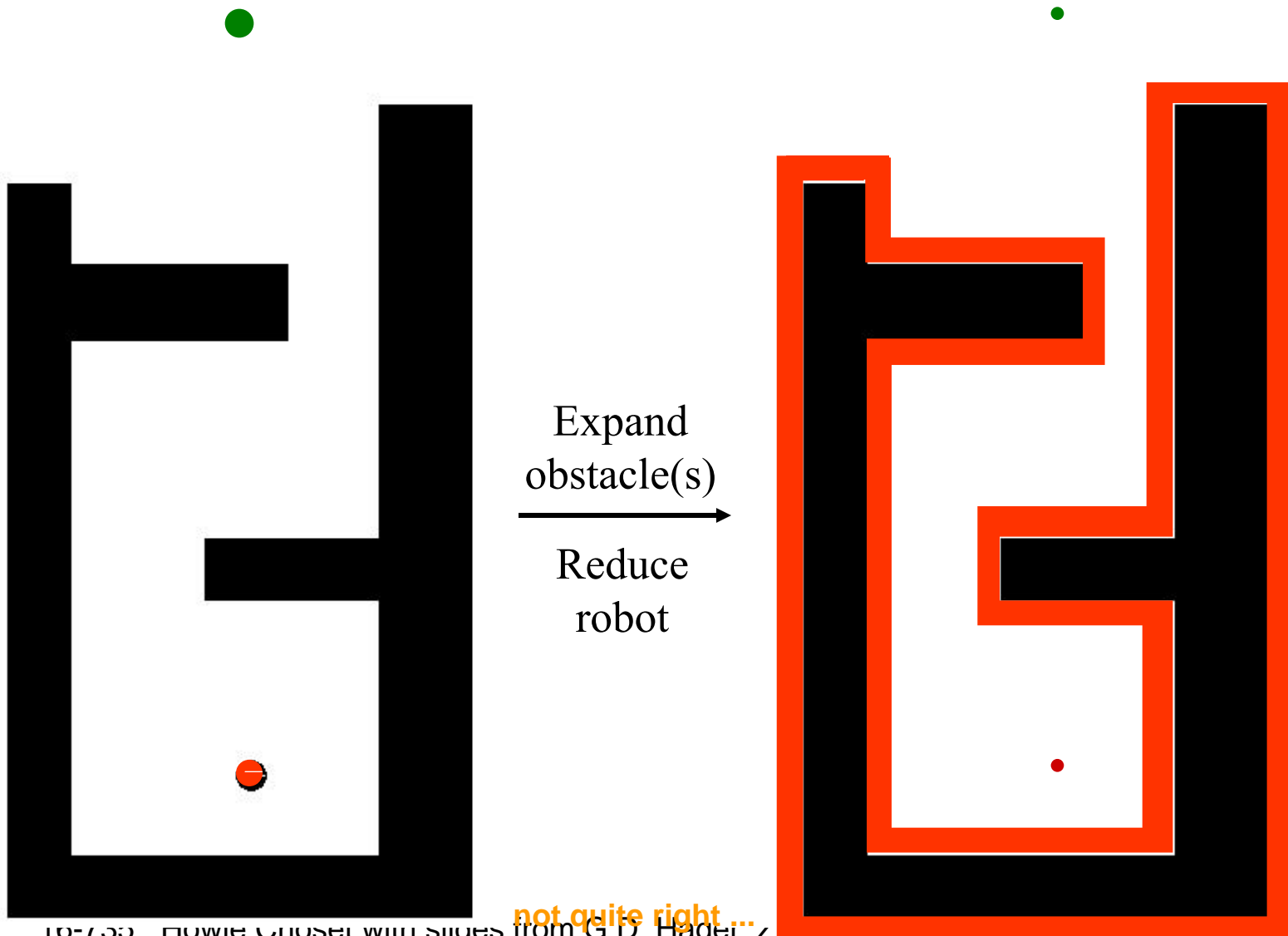


The Scout should probably not be modeled as a point...



Nor should robots with extended linkages that may contact obstacles...

What is the position of the robot?



Configuration Space

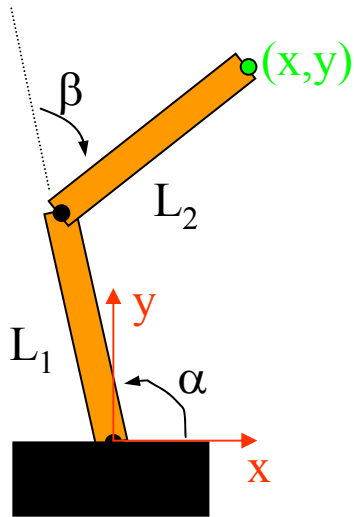
- A key concept for motion planning is a **configuration**:
 - *a complete specification of the position of every point in the system*
- A simple example: a robot that translates but does not rotate in the plane:
 - what is a sufficient representation of its configuration?
- The space of all configurations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

C-space formalism:
Lozano-Perez '79

Robot Manipulators

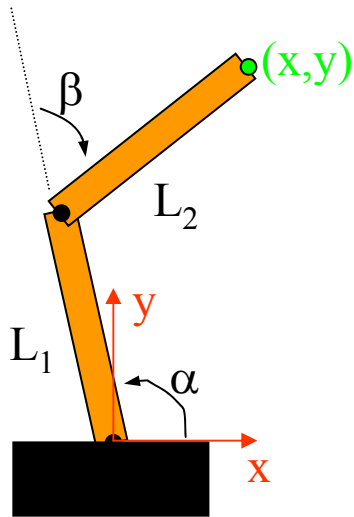
What are this arm's forward kinematics?

(How does its position depend on its joint angles?)



Robot Manipulators

What are this arm's forward kinematics?



Find (x, y) in terms of α and β ...

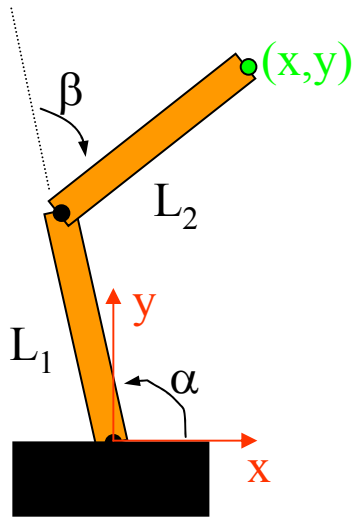
Keeping it “simple”

$$c_{\alpha} = \cos(\alpha) \quad , \quad s_{\alpha} = \sin(\alpha)$$

$$c_{\beta} = \cos(\beta) \quad , \quad s_{\beta} = \sin(\beta)$$

$$c_{+} = \cos(\alpha + \beta) \quad , \quad s_{+} = \sin(\alpha + \beta)$$

Manipulator kinematics



$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} L_1 c_\alpha \\ L_1 s_\alpha \end{pmatrix} + \begin{pmatrix} L_2 c_+ \\ L_2 s_+ \end{pmatrix} \quad \text{Position}$$

Keeping it “simple”

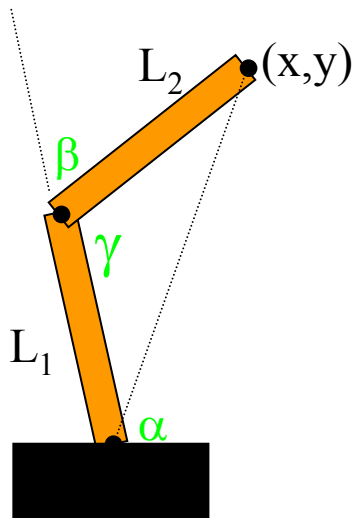
$$c_\alpha = \cos(\alpha) \quad , \quad s_\alpha = \sin(\alpha)$$

$$c_\beta = \cos(\beta) \quad , \quad s_\beta = \sin(\beta)$$

$$c_+ = \cos(\alpha + \beta) \quad , \quad s_+ = \sin(\alpha + \beta)$$

Inverse Kinematics

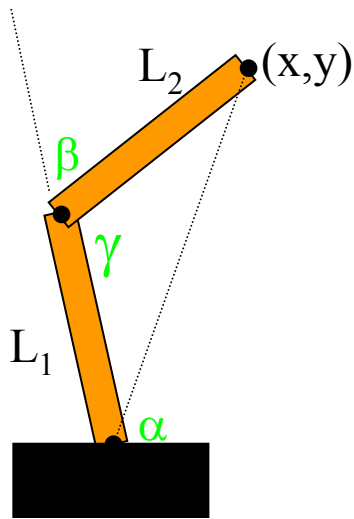
Inverse kinematics -- finding joint angles from Cartesian coordinates
via a geometric or algebraic approach...



Given (x,y) and L_1 and L_2 , what are the values of α , β , γ ?

Inverse Kinematics

Inverse kinematics -- finding joint angles from Cartesian coordinates via a geometric or algebraic approach...



$$\gamma = \cos^{-1} \left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} \right)$$

$$\beta = 180 - \gamma$$

$$\alpha = \sin^{-1} \left(\frac{L_2 \sin(\gamma)}{x^2 + y^2} \right) + \tan^{-1}(y/x)$$

↖ atan2(y,x)

$$(1,0) = 1.3183, -1.06$$

$$(-1,0) = 1.3183, 4.45$$

Puma

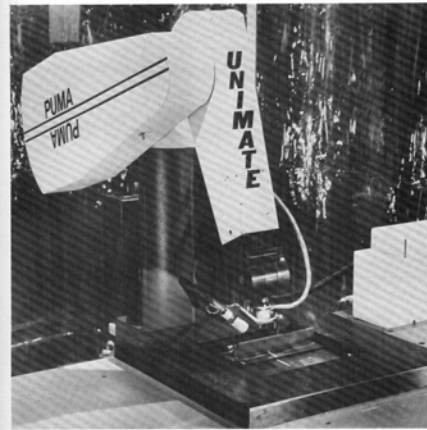


FIGURE 3.17 The Unimation PUMA 560.

```
%  
% Solve for theta(1)  
  
r=sqrt(Px^2 + Py^2);  
if (n1 == 1),  
    theta(1)= atan2(Py,Px) + asin(d3/r);  
else  
    theta(1)= atan2(Py,Px) + pi - asin(d3/r);  
end  
  
%  
% Solve for theta(2)  
  
V114= Px*cos(theta(1)) + Py*sin(theta(1));  
r=sqrt(V114^2 + Pz^2);  
Psi = acos((a2^2-d4^2-a3^2+V114^2+Pz^2)/  
           (2.0*a2*r));  
theta(2) = atan2(Pz,V114) + n2*Psi;  
  
%  
% Solve for theta(3)  
  
num = cos(theta(2))*V114+sin(theta(2))*Pz-a2;  
den = cos(theta(2))*Pz - sin(theta(2))*V114;  
theta(3) = atan2(a3,d4) - atan2(num, den);
```

Inv. Kinematics

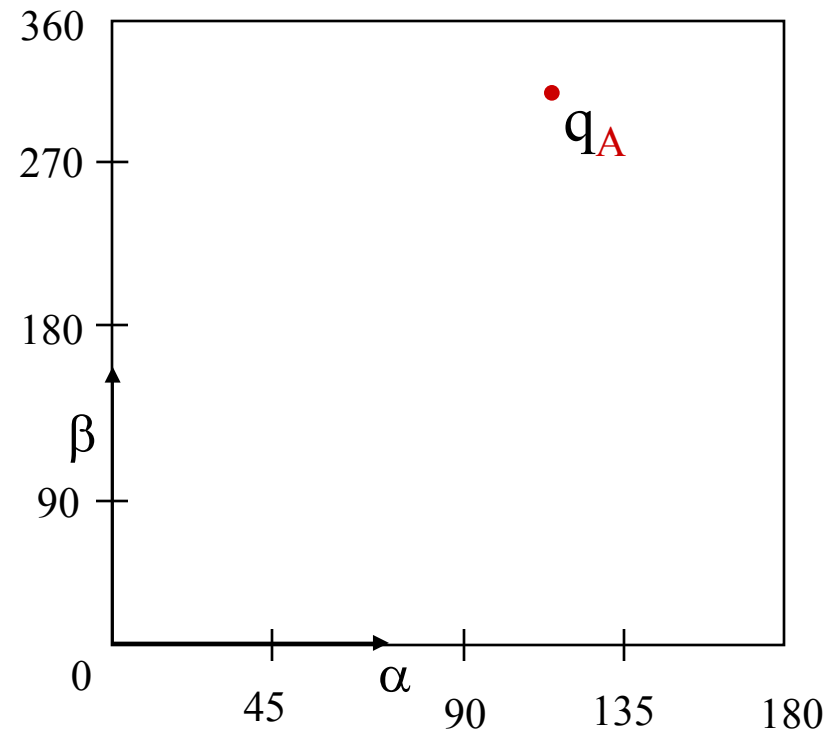
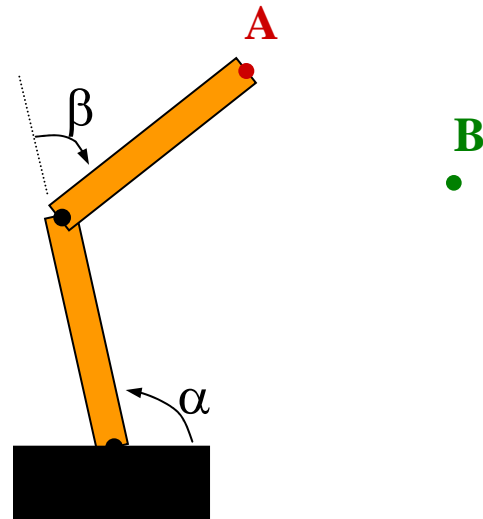
```
% Solve for theta(4)  
  
V113 = cos(theta(1))*Ax + sin(theta(1))*Ay;  
V323 = cos(theta(1))*Ay - sin(theta(1))*Ax;  
V313 = cos(theta(2)+theta(3))*V113 +  
        sin(theta(2)+theta(3))*Az;  
theta(4) = atan2((n4*V323),(n4*V313));  
  
% Solve for theta(5)  
  
num = -cos(theta(4))*V313 - V323*sin(theta(4));  
den = -V113*sin(theta(2)+theta(3)) +  
        Az*cos(theta(2)+theta(3));  
theta(5) = atan2(num,den);  
  
% Solve for theta(6)  
  
V112 = cos(theta(1))*Ox + sin(theta(1))*Oy;  
V132 = sin(theta(1))*Ox - cos(theta(1))*Oy;  
V312 = V112*cos(theta(2)+theta(3)) +  
        Oz*sin(theta(2)+theta(3));  
V332 = -V112*sin(theta(2)+theta(3)) +  
        Oz*cos(theta(2)+theta(3));  
V412 = V312*cos(theta(4)) - V132*sin(theta(4));  
V432 = V312*sin(theta(4)) + V132*cos(theta(4));  
num = -V412*cos(theta(5)) - V332*sin(theta(5));  
den = - V432;  
theta(6) = atan2(num,den);
```

Some Other Examples of C-Space

- A rotating bar fixed at a point
 - what is its C-space?
 - what is its workspace
- A rotating bar that translates along the rotation axis
 - what is its C-space?
 - what is its workspace
- A two-link manipulator
 - what is its C-space?
 - what is its workspace?
 - Suppose there are joint limits, does this change the C-space?
 - The workspace?

Configuration Space

Where can we put $\bullet q_B$?



An obstacle in the robot's workspace

Torus

(wraps horizontally and vertically)

Obstacles in C-Space

- Let q denote a point in a configuration space Q
- The path planning problem is to find a mapping $c:[0,1] \rightarrow Q$ s.t. no configuration along the path intersects an obstacle
- Recall a workspace obstacle is WO_i
- A *configuration space obstacle* QO_i is the set of configurations q at which the robot intersects WO_i , that is
 - $QO_i = \{q \in Q \mid R(q) \cap WO_i \neq \emptyset\}$.
- The *free configuration space* (or just *free space*) Q_{free} is

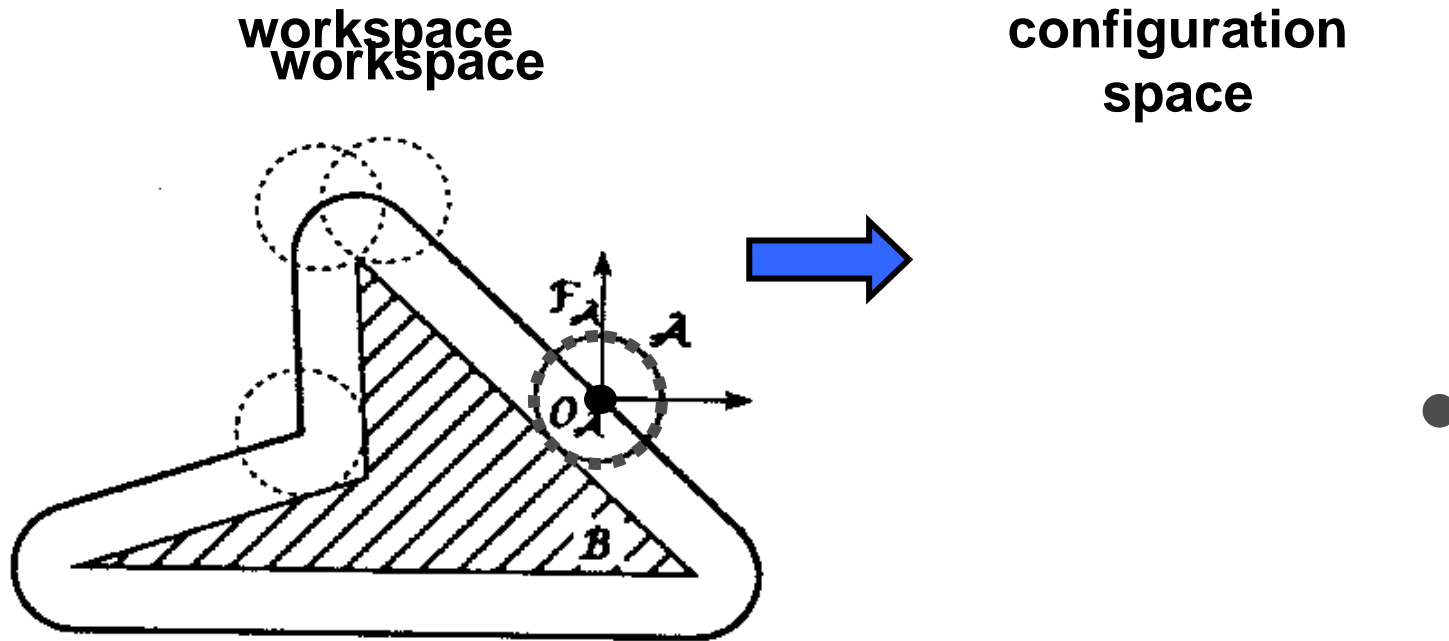
$$Q_{\text{free}} = Q \setminus \left(\bigcup QO_i \right).$$

The free space is generally an open set

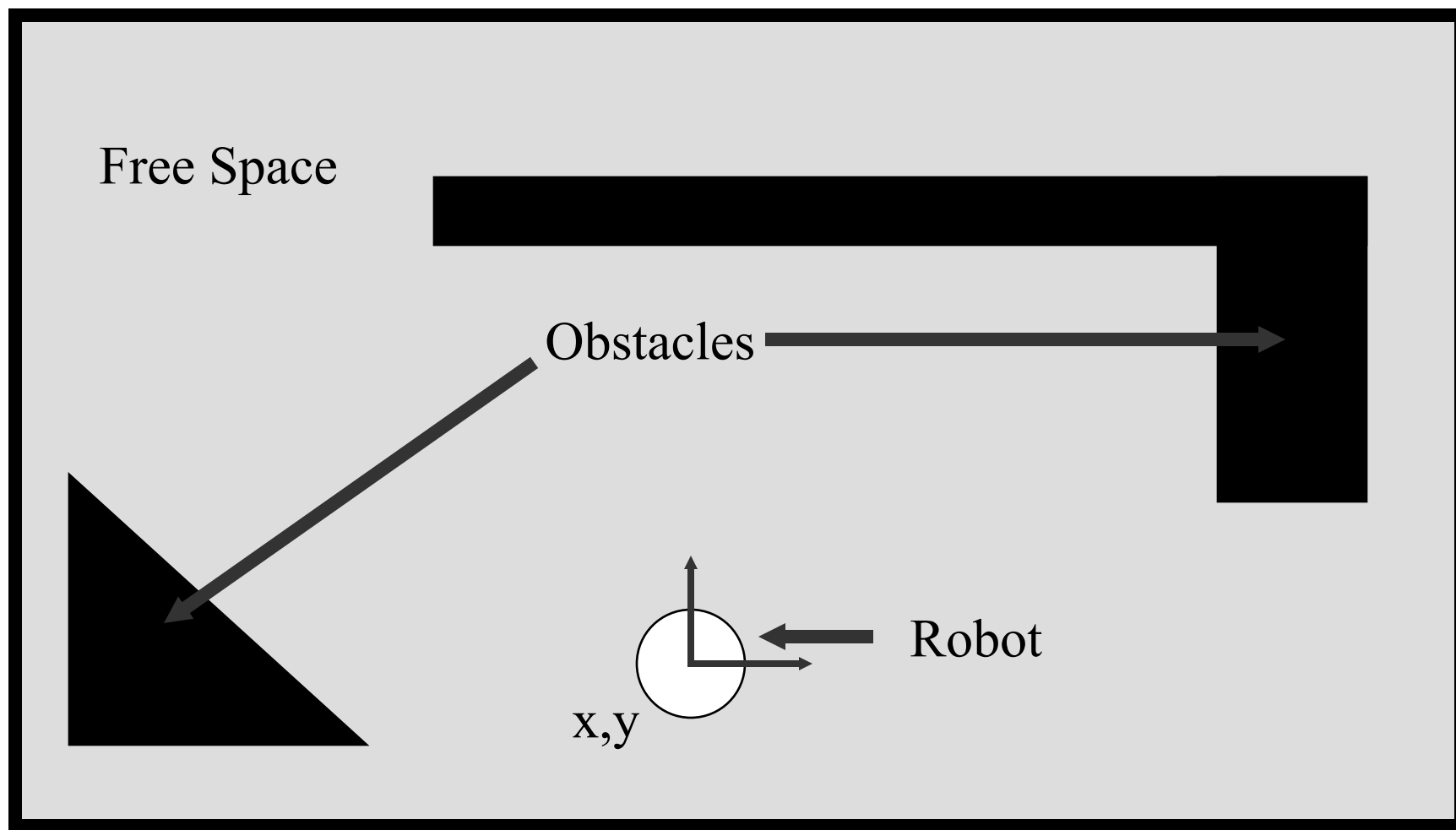
A *free path* is a mapping $c:[0,1] \rightarrow Q_{\text{free}}$

A *semifree path* is a mapping $c:[0,1] \rightarrow \text{cl}(Q_{\text{free}})$

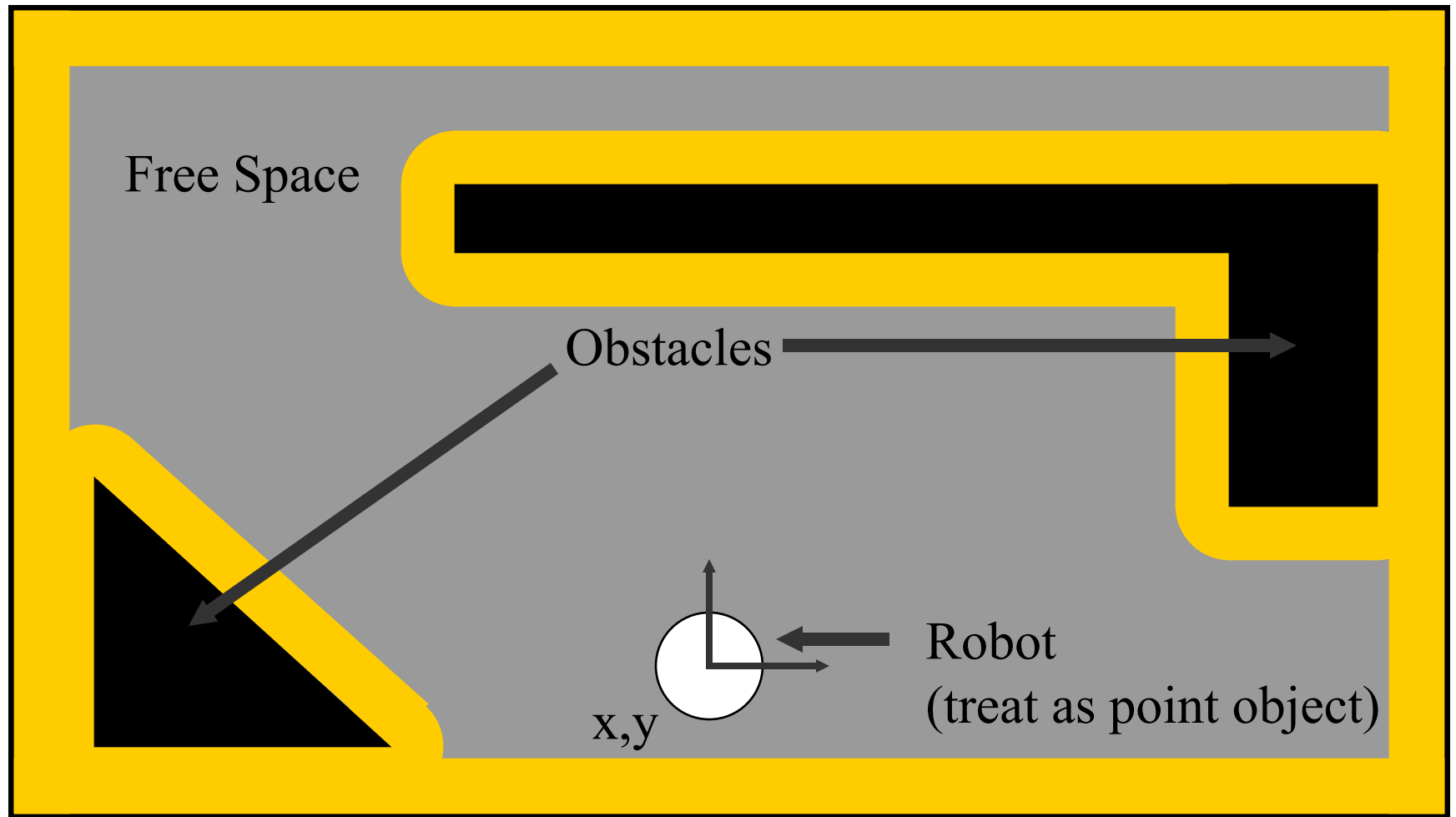
Disc in 2-D workspace



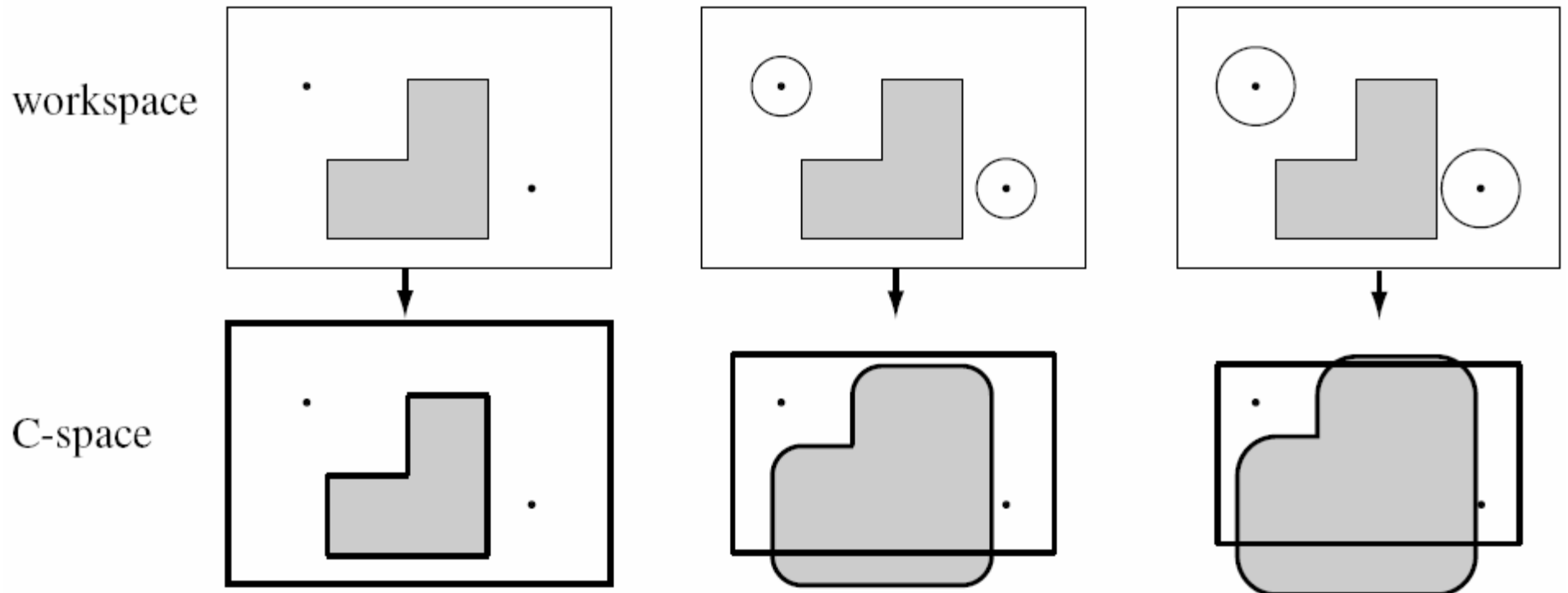
Example of a World (and Robot)



Configuration Space: Accommodate Robot Size



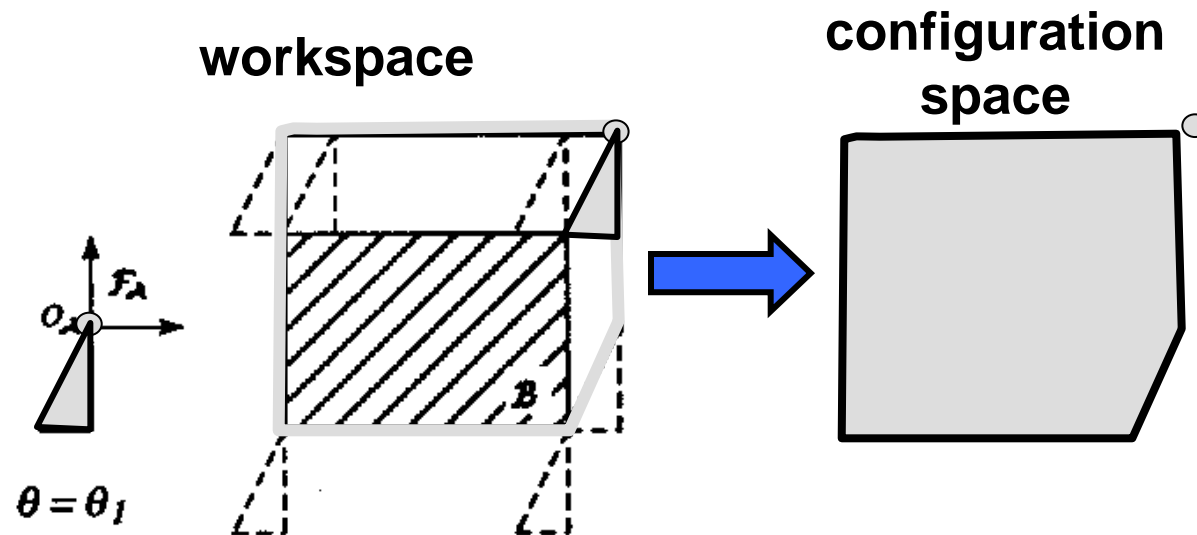
Trace Boundary of Workspace



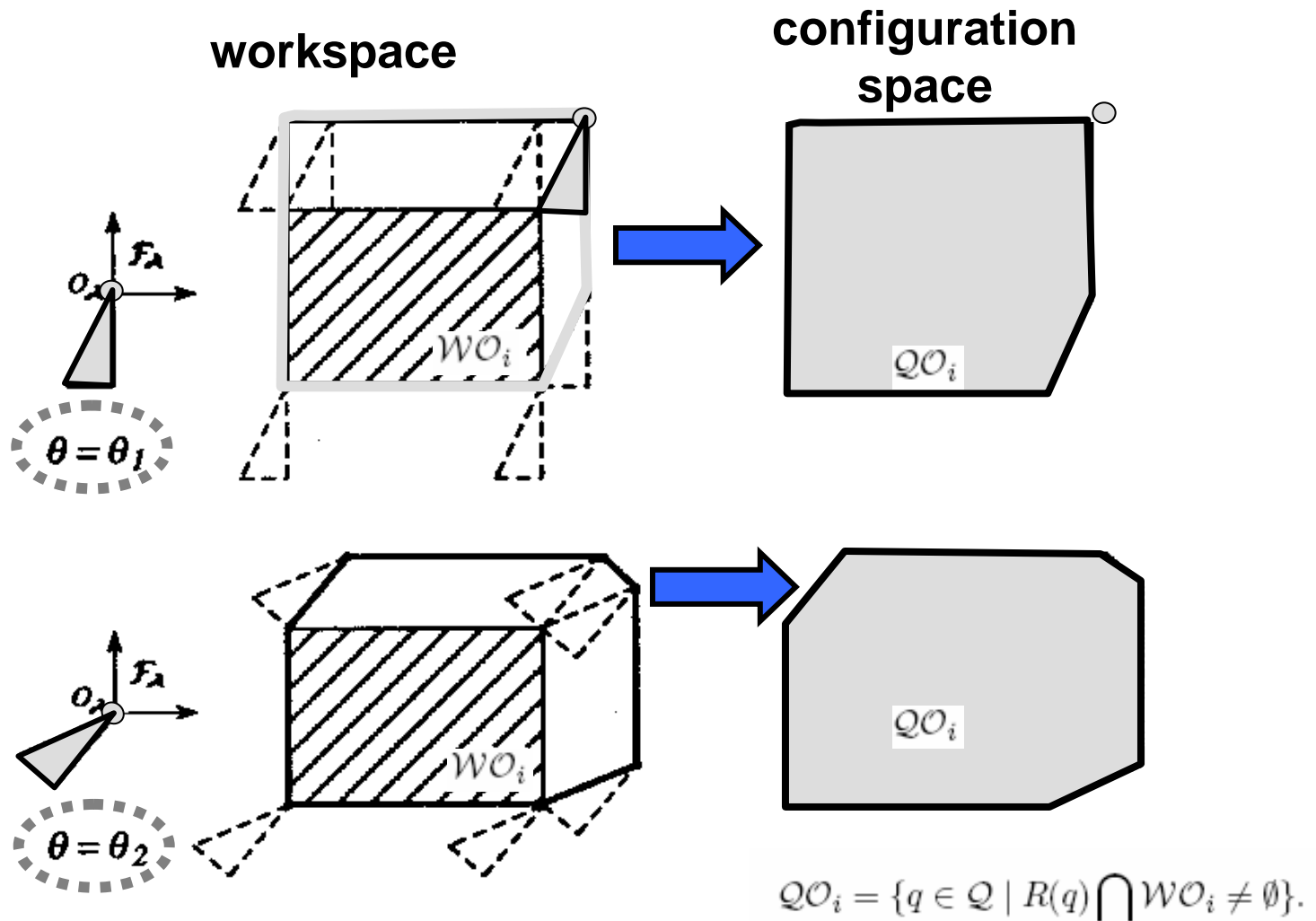
$$\mathcal{QO}_i = \{q \in \mathcal{Q} \mid R(q) \cap \mathcal{WO}_i \neq \emptyset\}.$$

Pick a reference point...

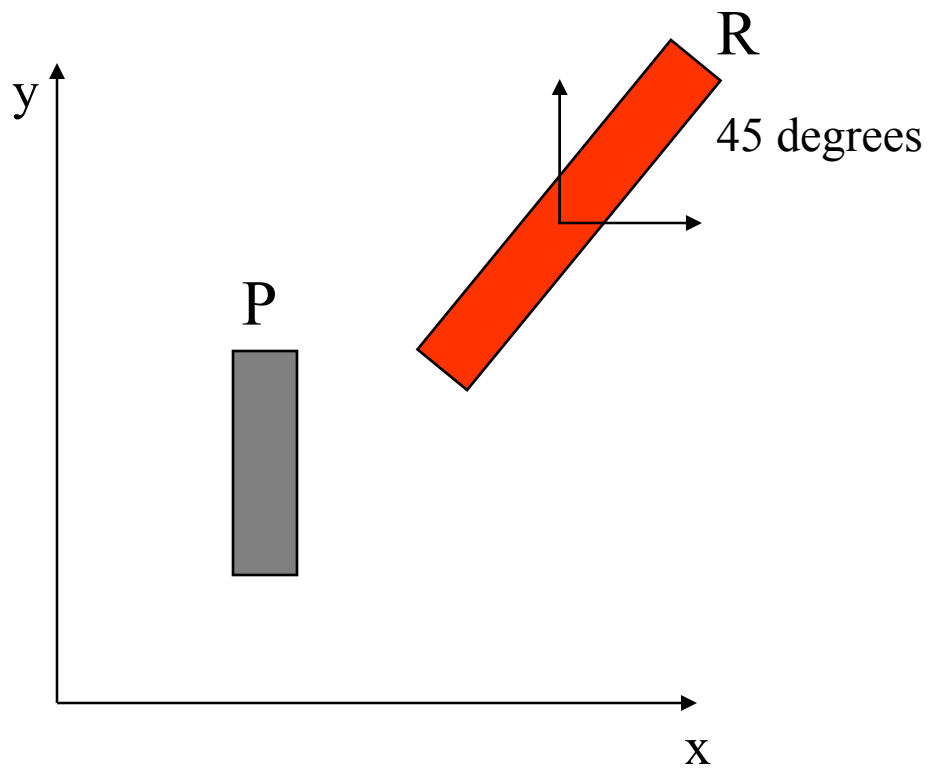
Polygonal robot translating in 2-D workspace



Polygonal robot translating & rotating in 2-D workspace

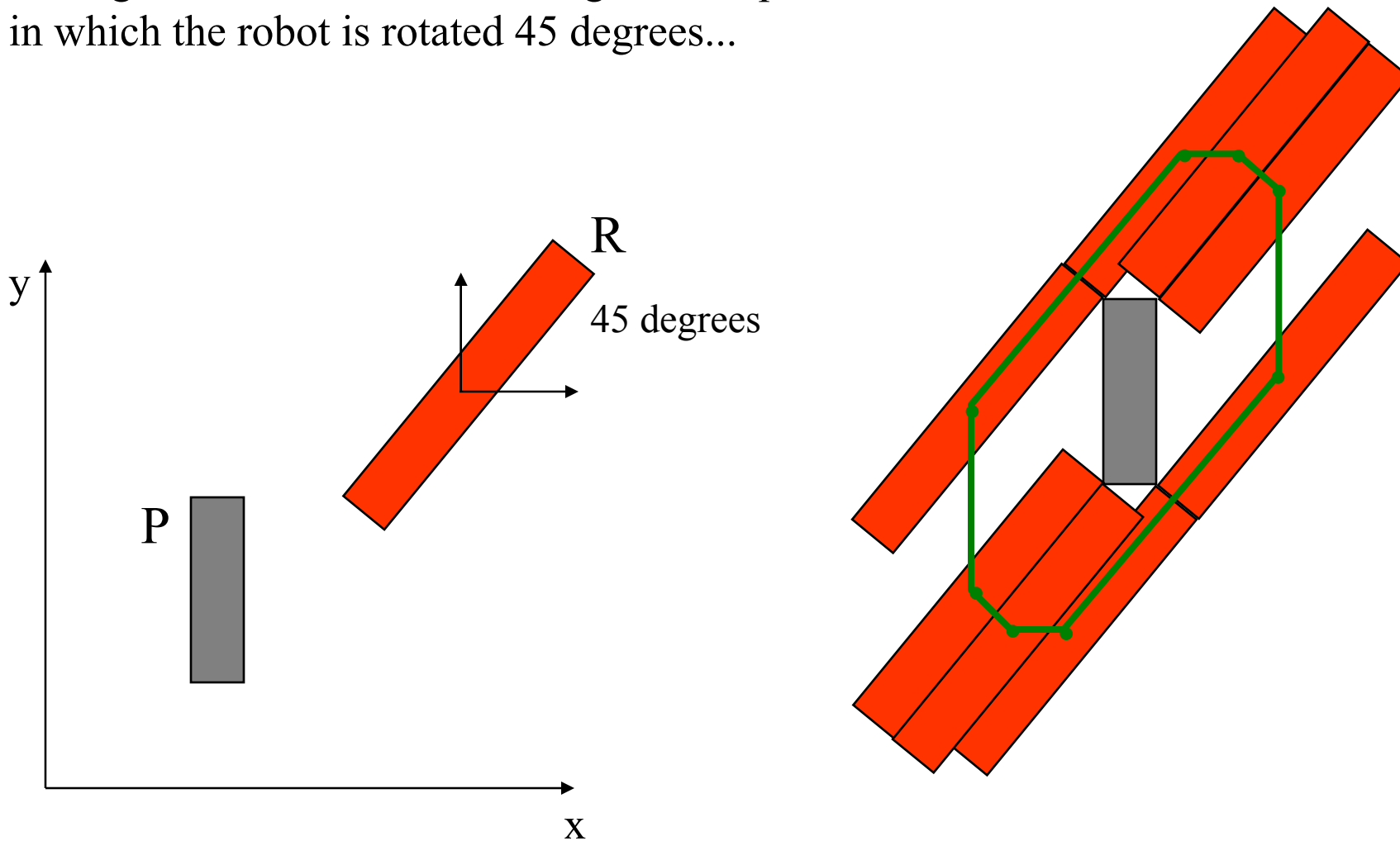


Any reference point



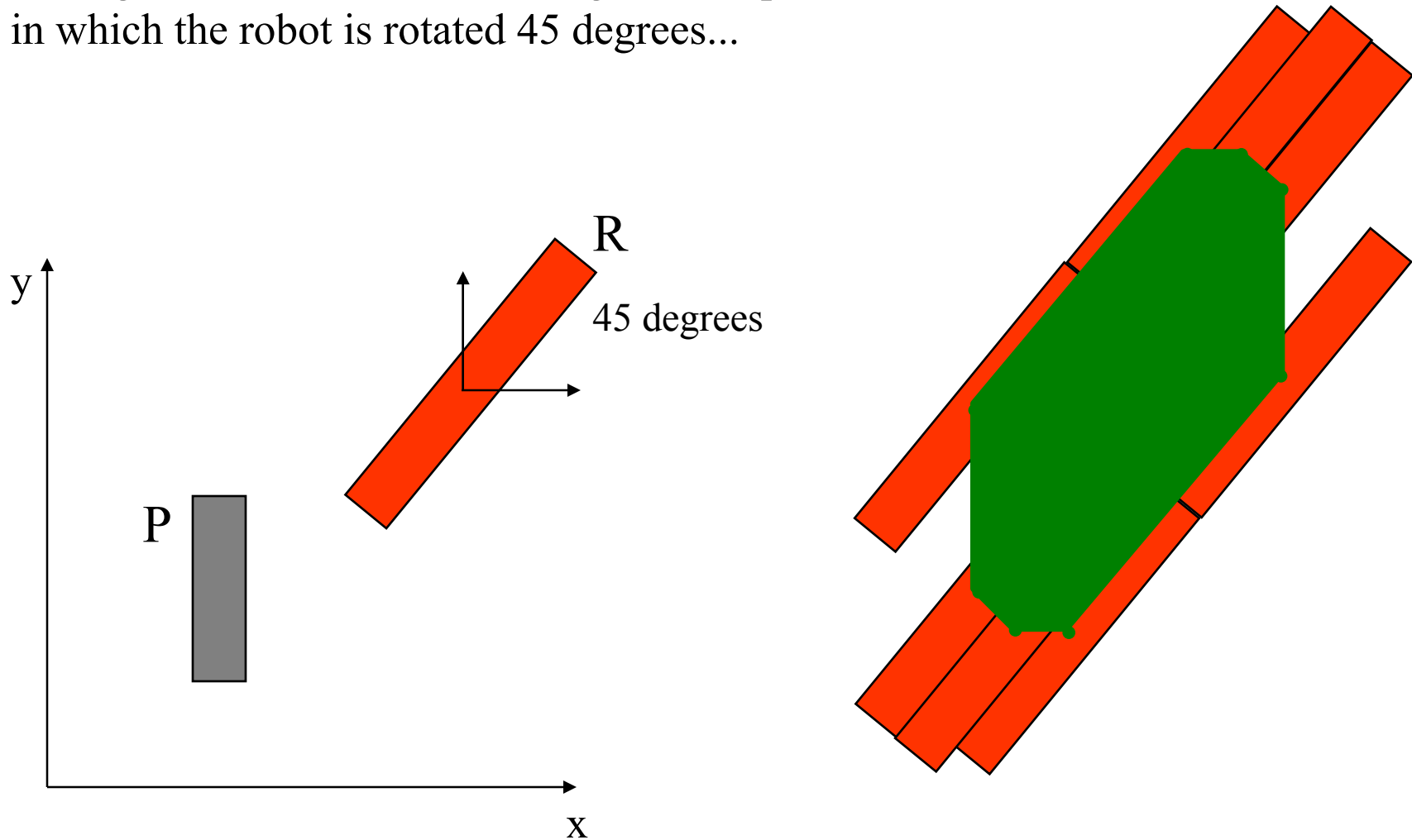
Any reference ~~point~~ configuration

Taking the cross section of configuration space
in which the robot is rotated 45 degrees...



Any reference point configuration

Taking the cross section of configuration space
in which the robot is rotated 45 degrees...



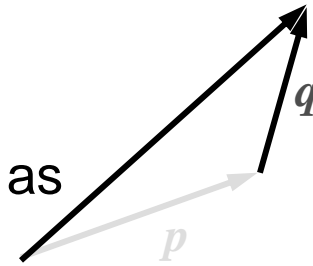
Minkowski sum

- The **Minkowski sum** of two sets P and Q , denoted by $P \oplus Q$, is defined as

$$P \oplus Q = \{ p+q \mid p \in P, q \in Q \}$$

- Similarly, the **Minkowski difference** is defined as

$$P \ominus Q = \{ p-q \mid p \in P, q \in Q \}$$

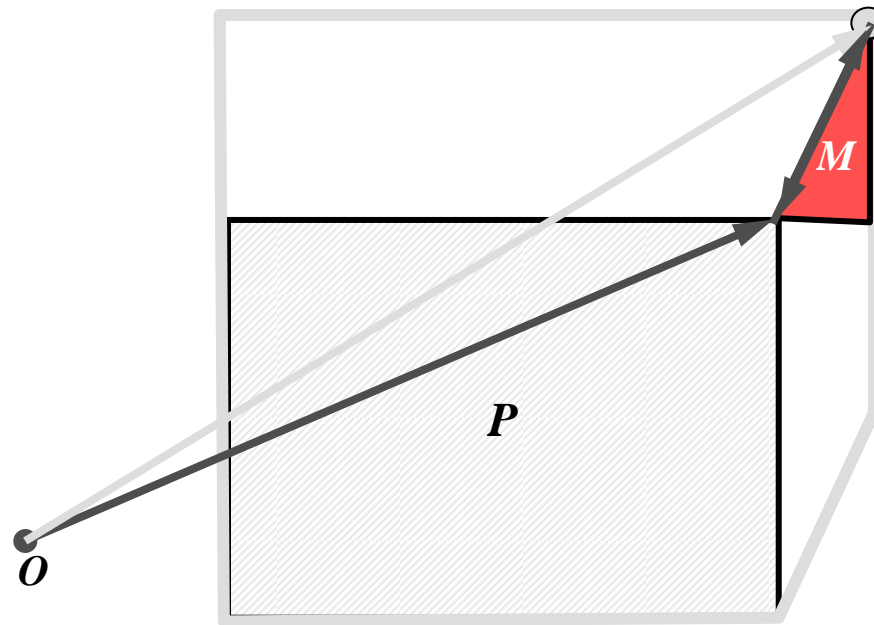


Minkowski sum of convex polygons

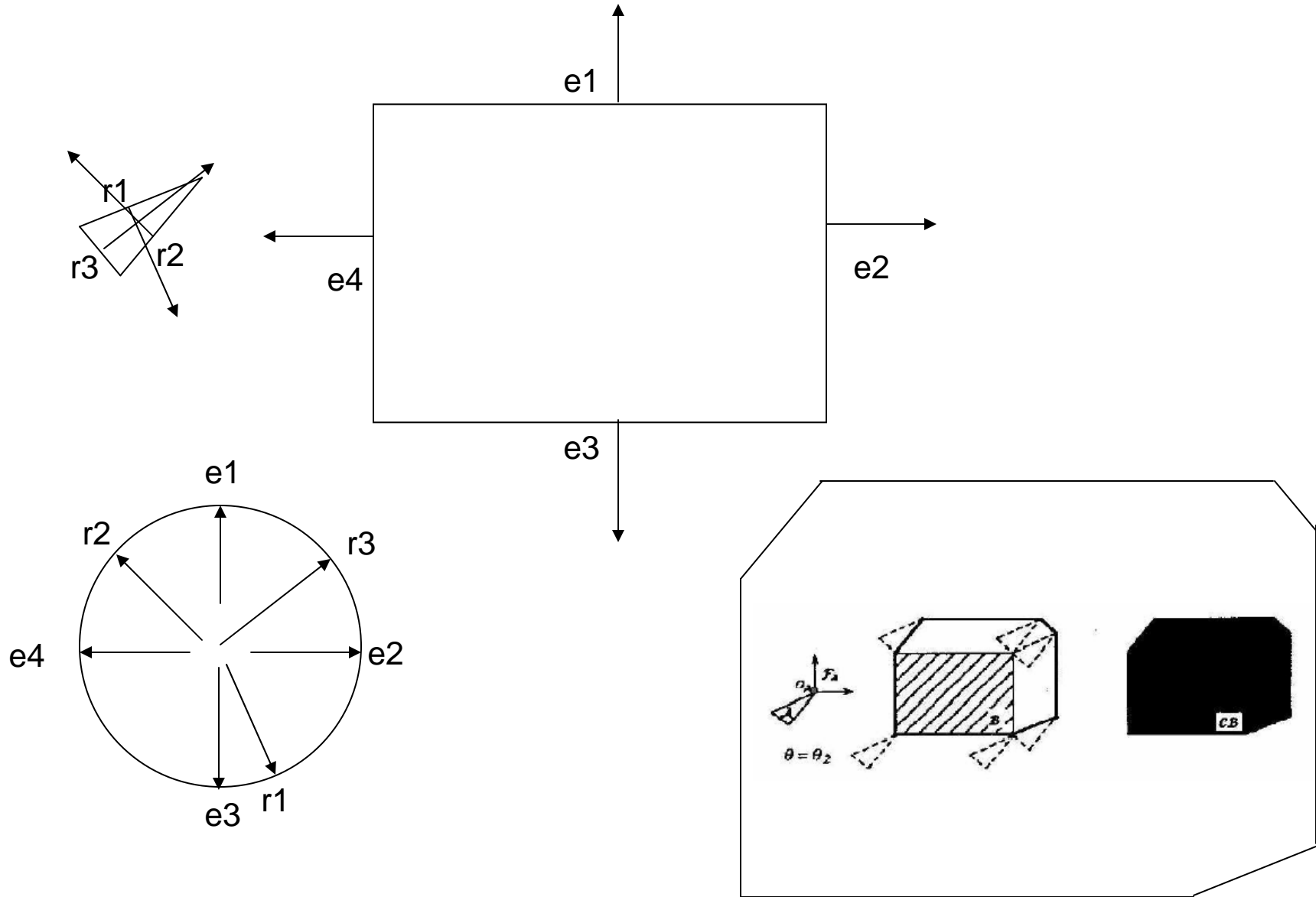
- The Minkowski sum of two convex polygons P and Q of m and n vertices respectively is a convex polygon $P \oplus Q$ of $m + n$ vertices.
 - The vertices of $P \oplus Q$ are the “sums” of vertices of P and Q .

Observation

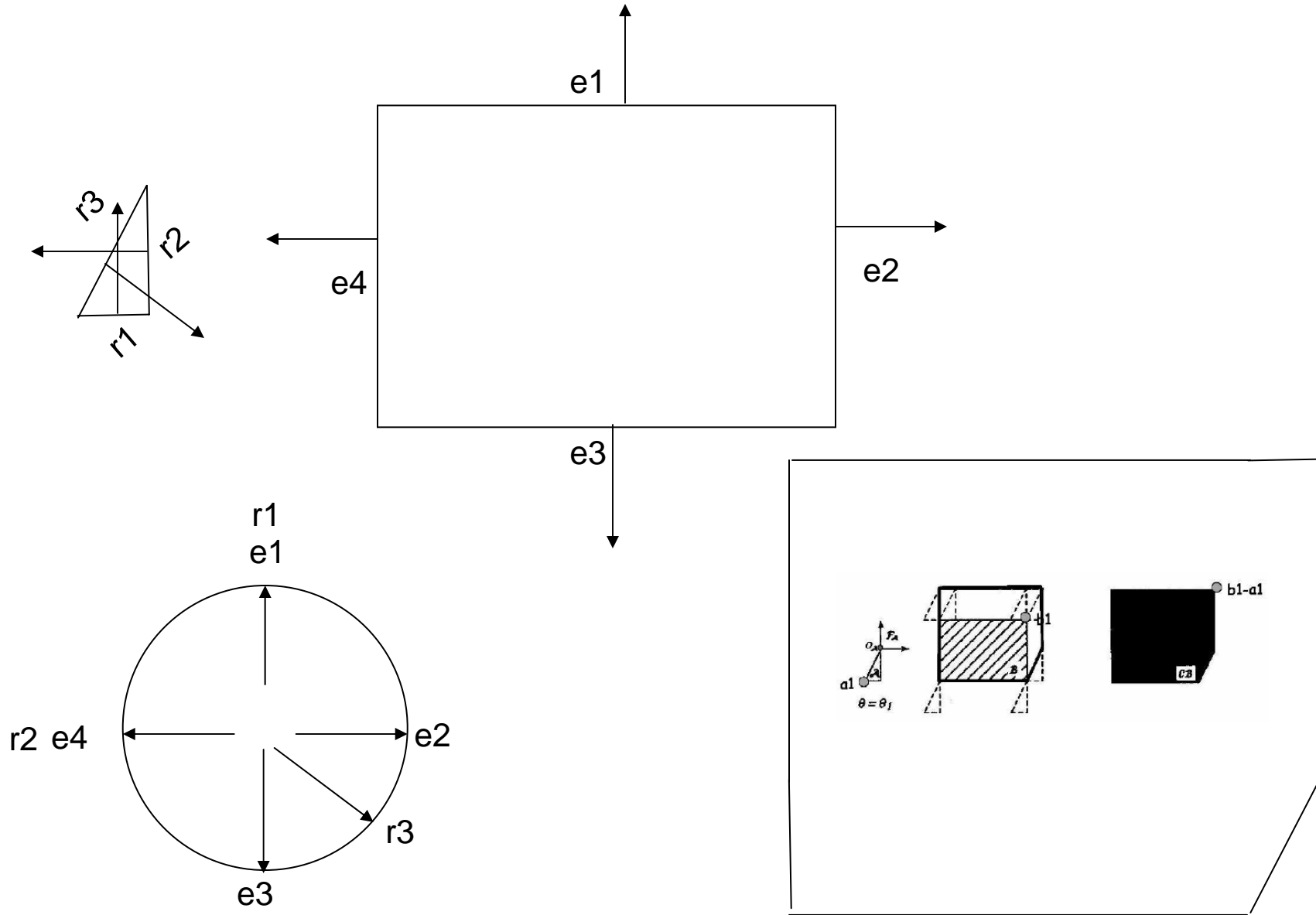
- If P is an obstacle in the workspace and M is a moving object. Then the C-space obstacle corresponding to P is $P \ominus M$.



Star Algorithm: Polygonal Obstacles



Star Algorithm

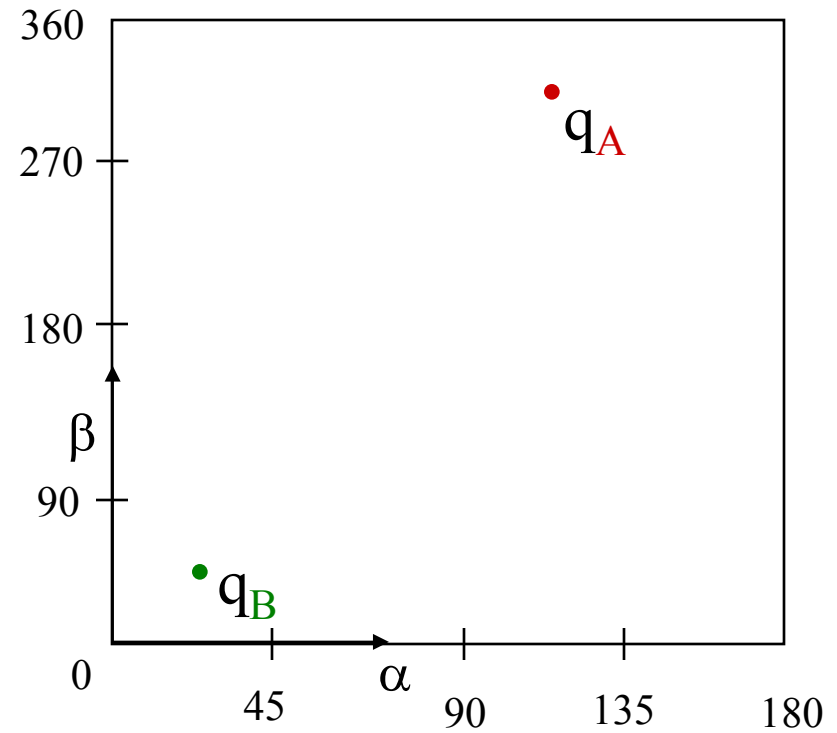
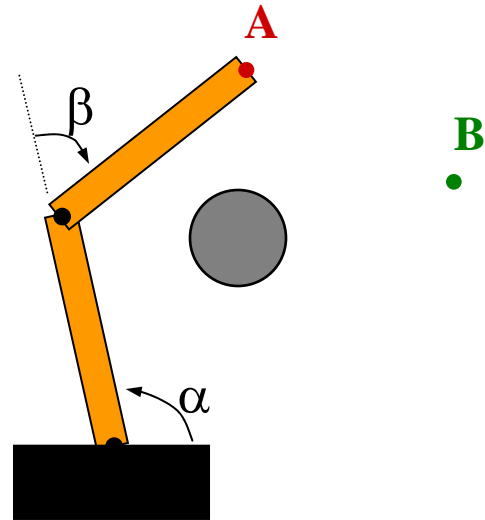


Start Point

- Leave that as an exercise for your homework.

Configuration Space “Quiz”

Where do we put  ?



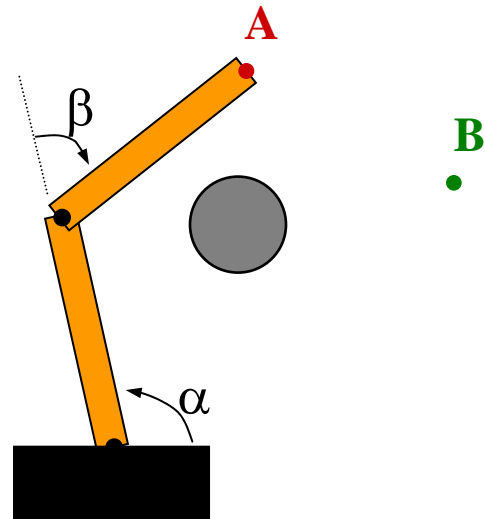
Torus

(wraps horizontally and vertically)

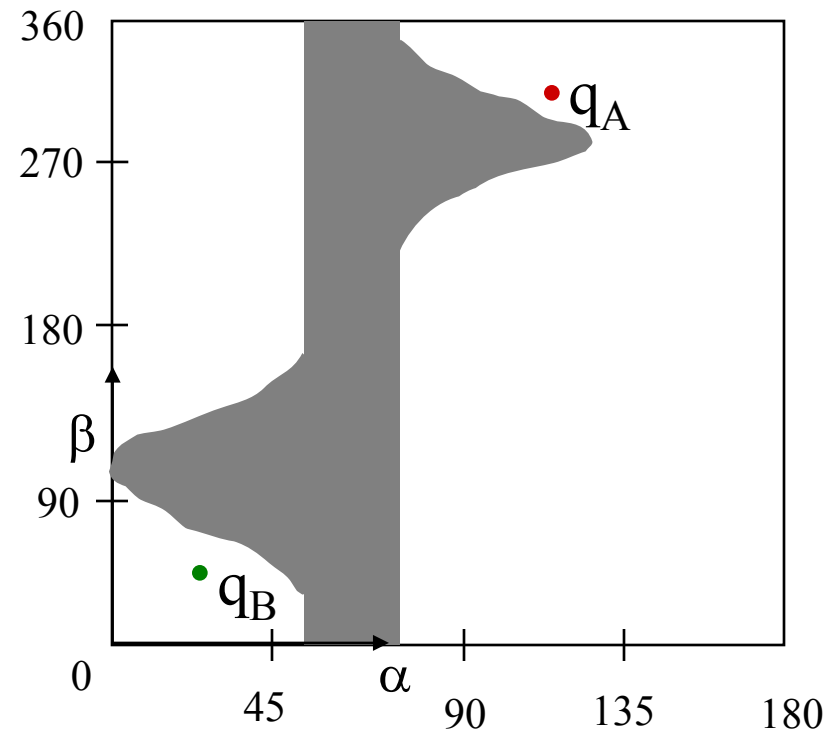
An obstacle in the robot’s workspace

Configuration Space Obstacle

Reference configuration



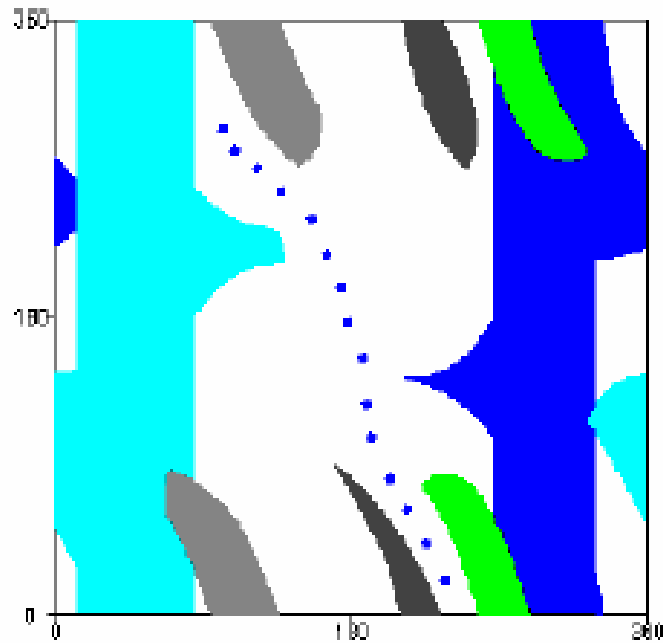
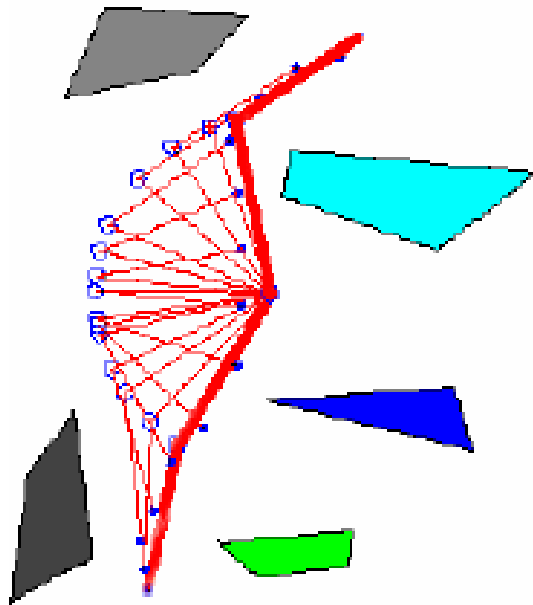
How do we get from **A** to **B** ?



An obstacle in the robot's workspace

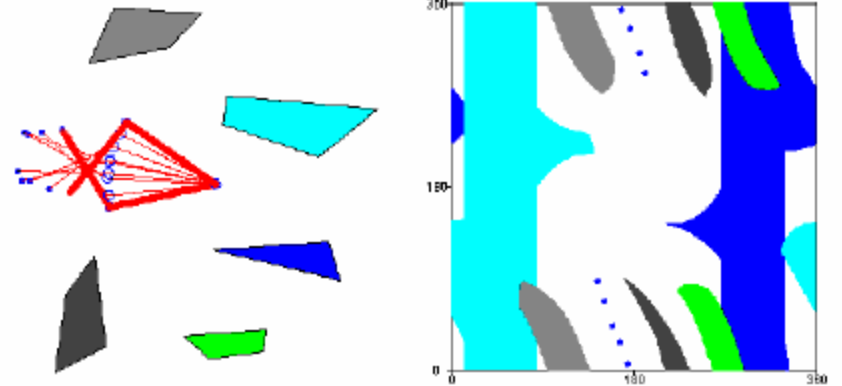
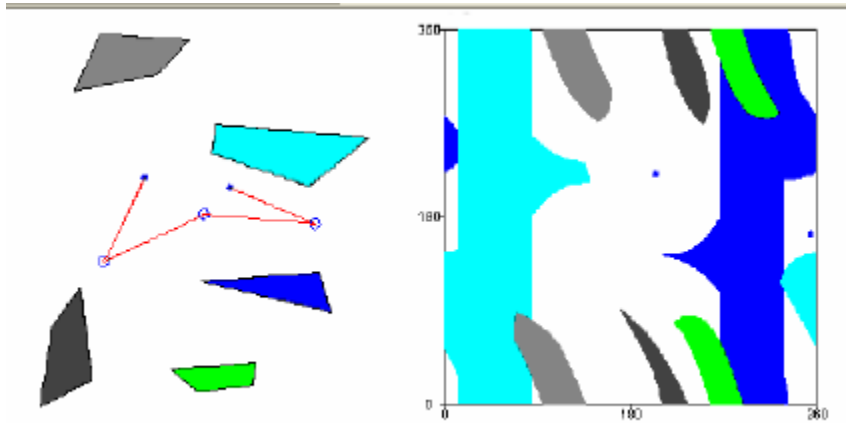
The C-space representation
of this obstacle...

Two Link Path



Thanks to Ken Goldberg

Two Link Path



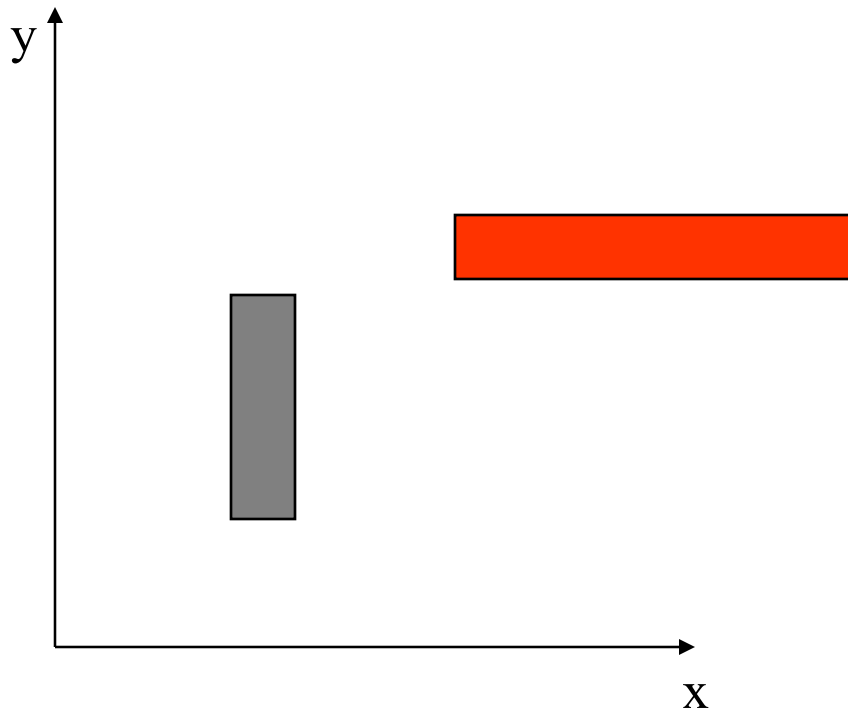
Properties of Obstacles in C-Space

- If the robot and WO_i are _____, then
 - *Convex* then QO_i is convex
 - *Closed* then QO_i is closed
 - *Compact* then QO_i is compact
 - *Algebraic* then QO_i is algebraic
 - *Connected* then QO_i is connected

Additional dimensions

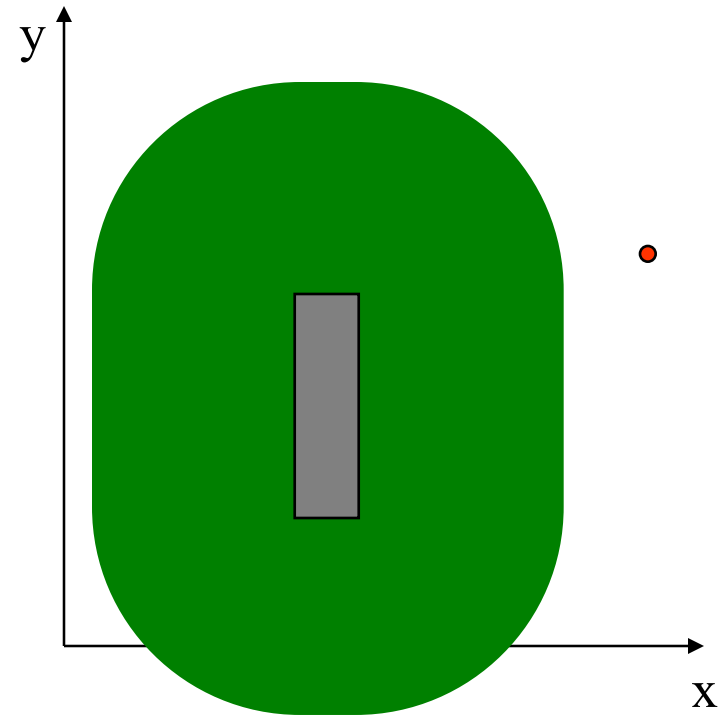
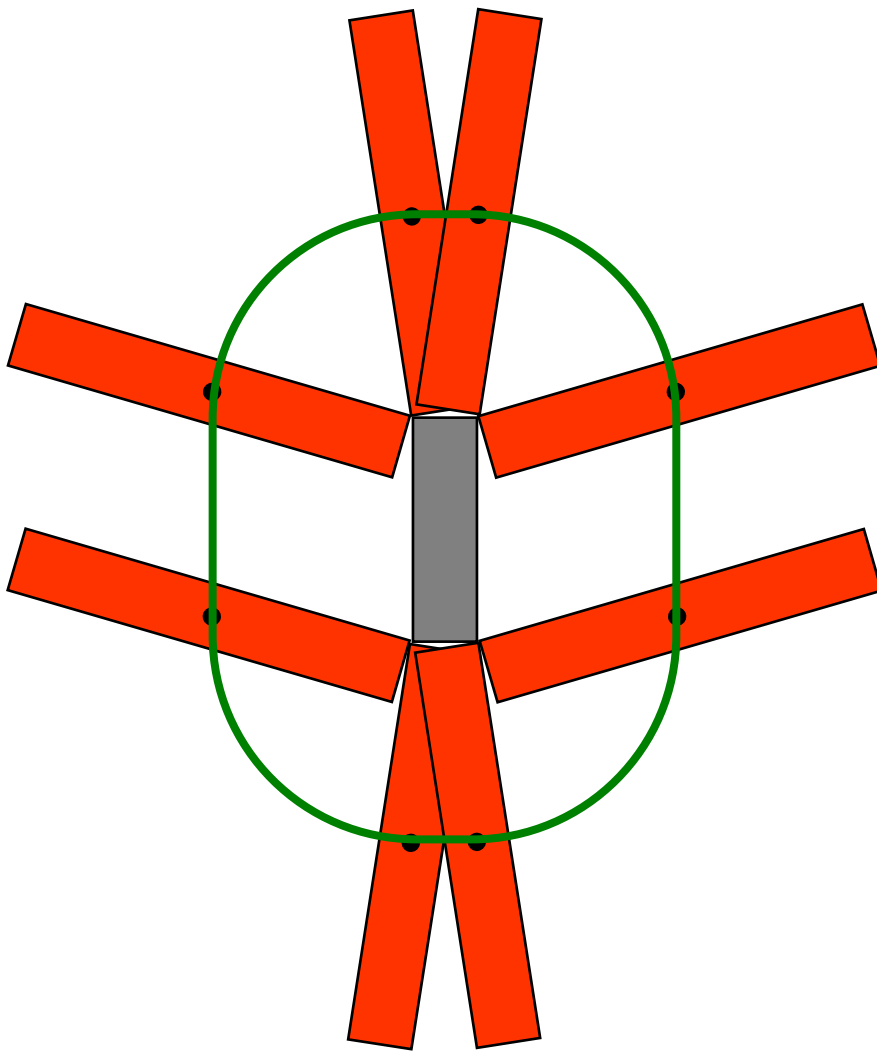
What would the configuration space of a rectangular robot (red) in this world look like?
Assume it can translate *and* rotate in the plane.

(The blue rectangle is an obstacle.)

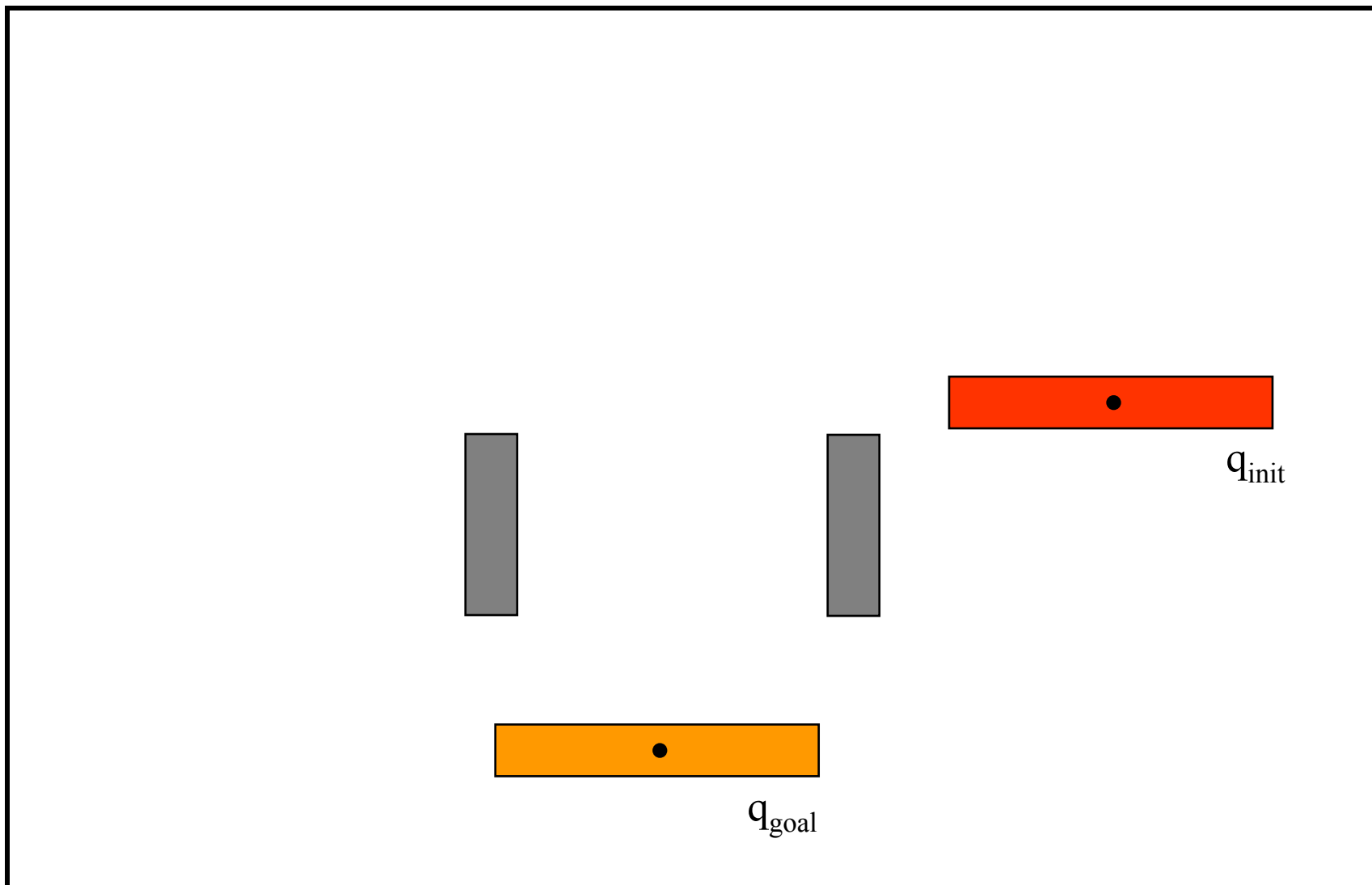


a 2d possibility

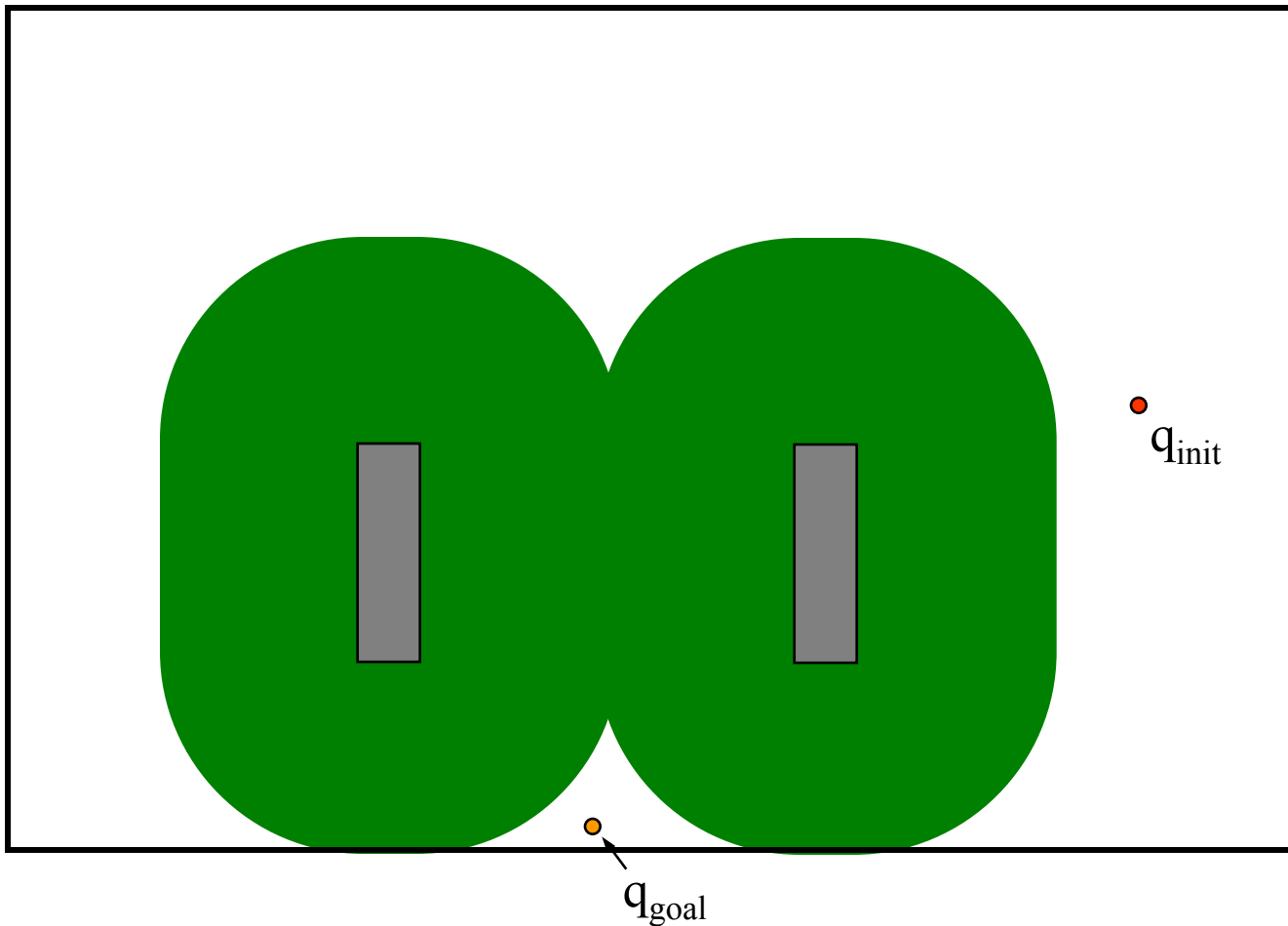
2d projection...



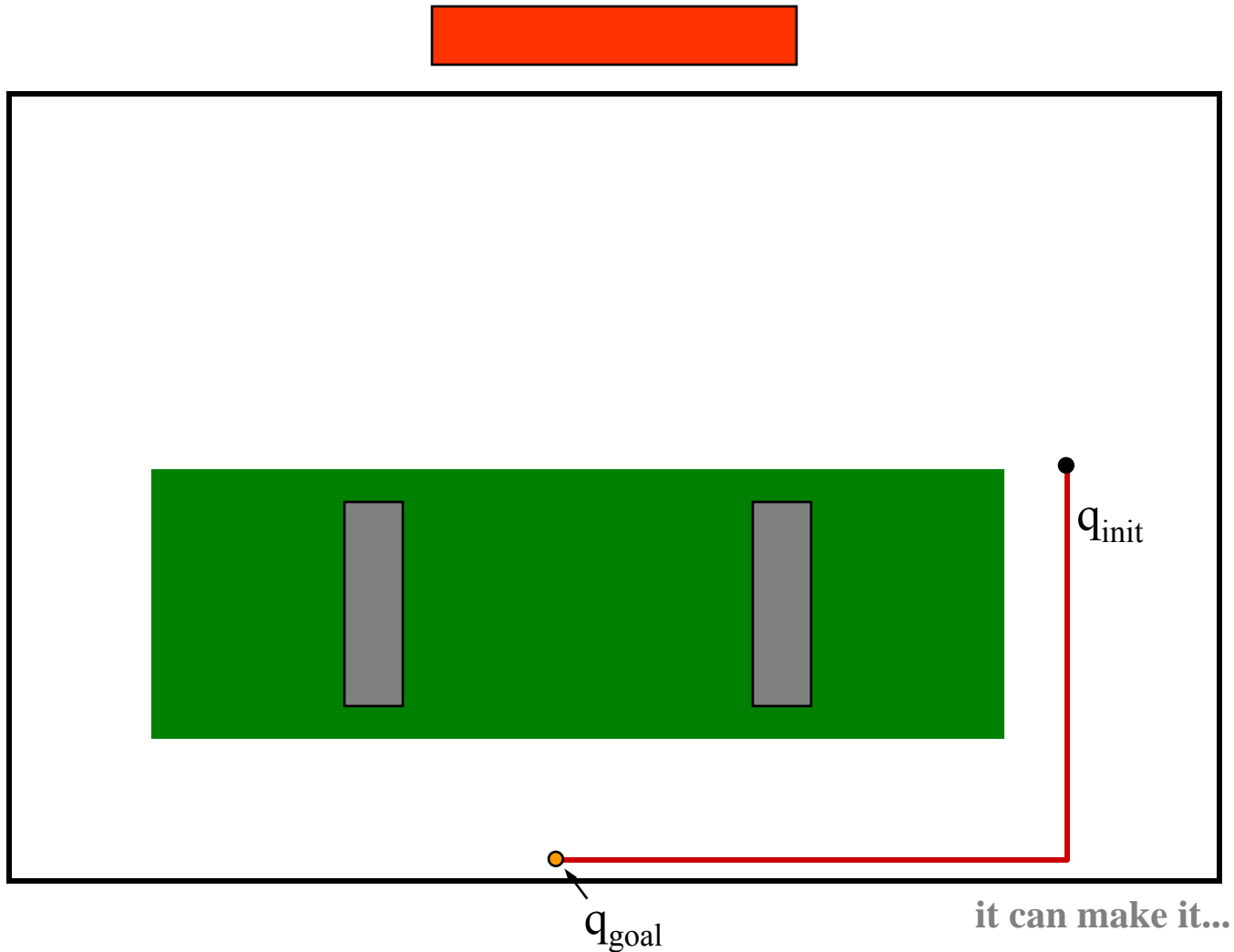
A problem?



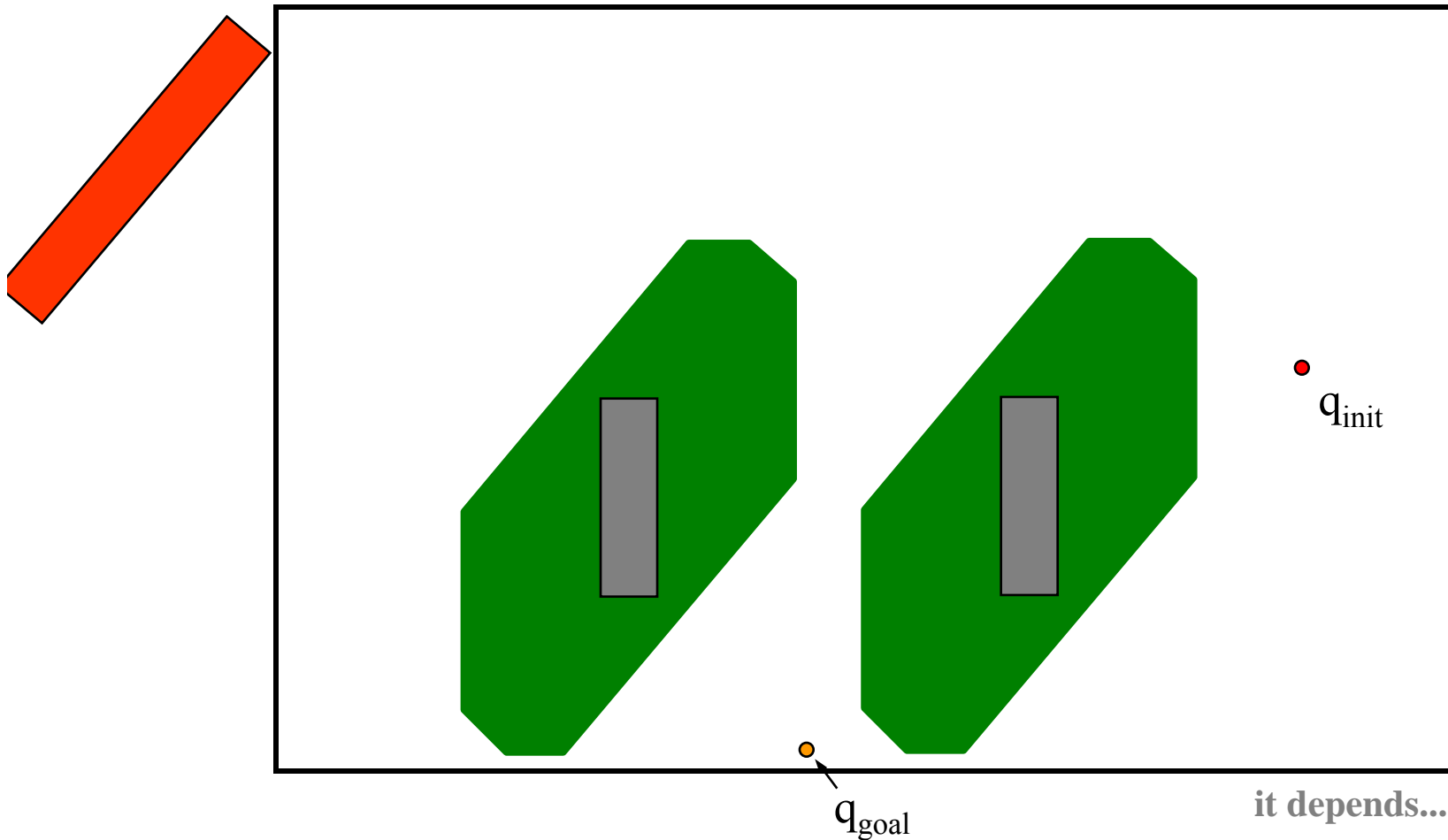
Requires one more d...



When the robot is at one orientation



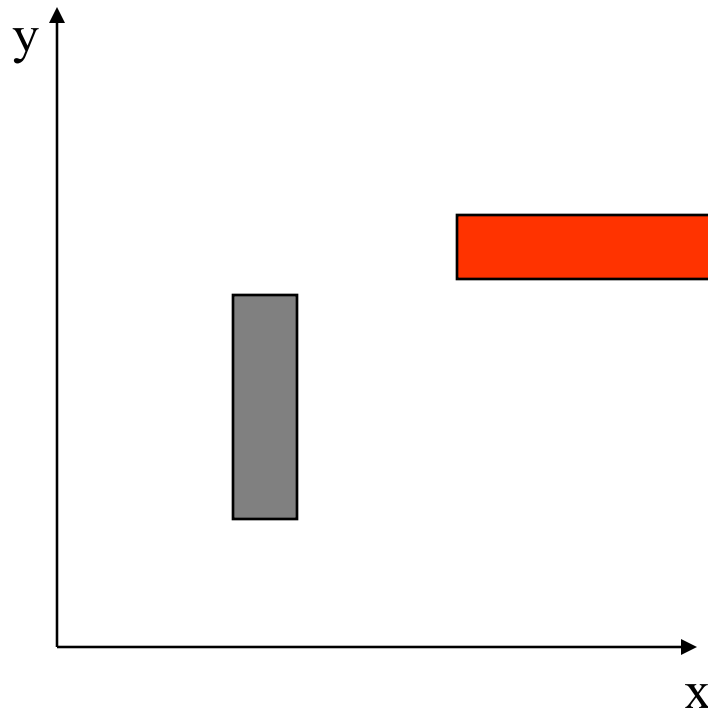
When the robot is at another orientation



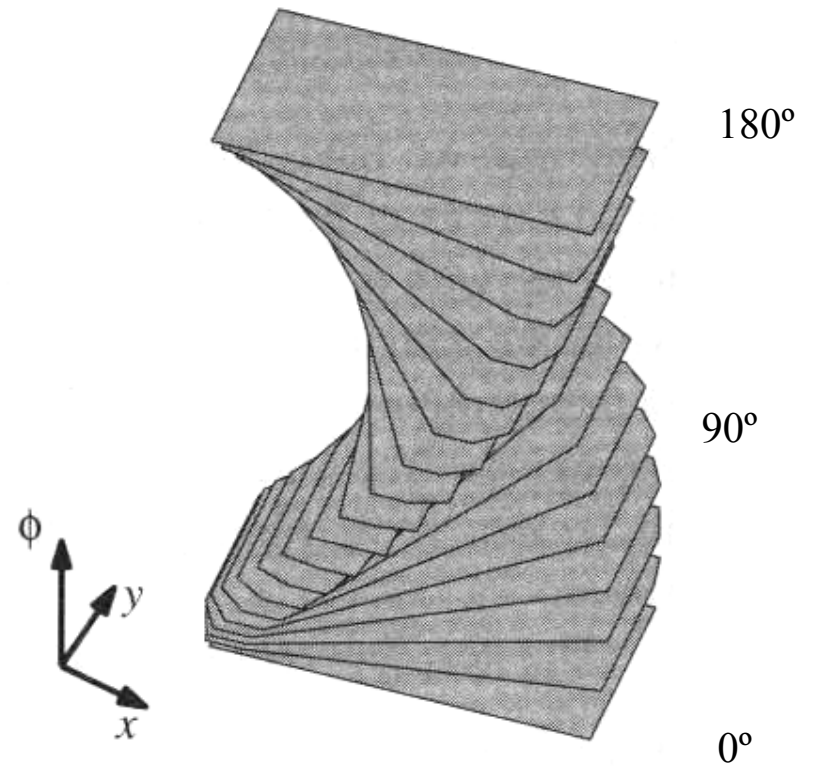
Additional dimensions

What would the configuration space of a rectangular robot (red) in this world look like?

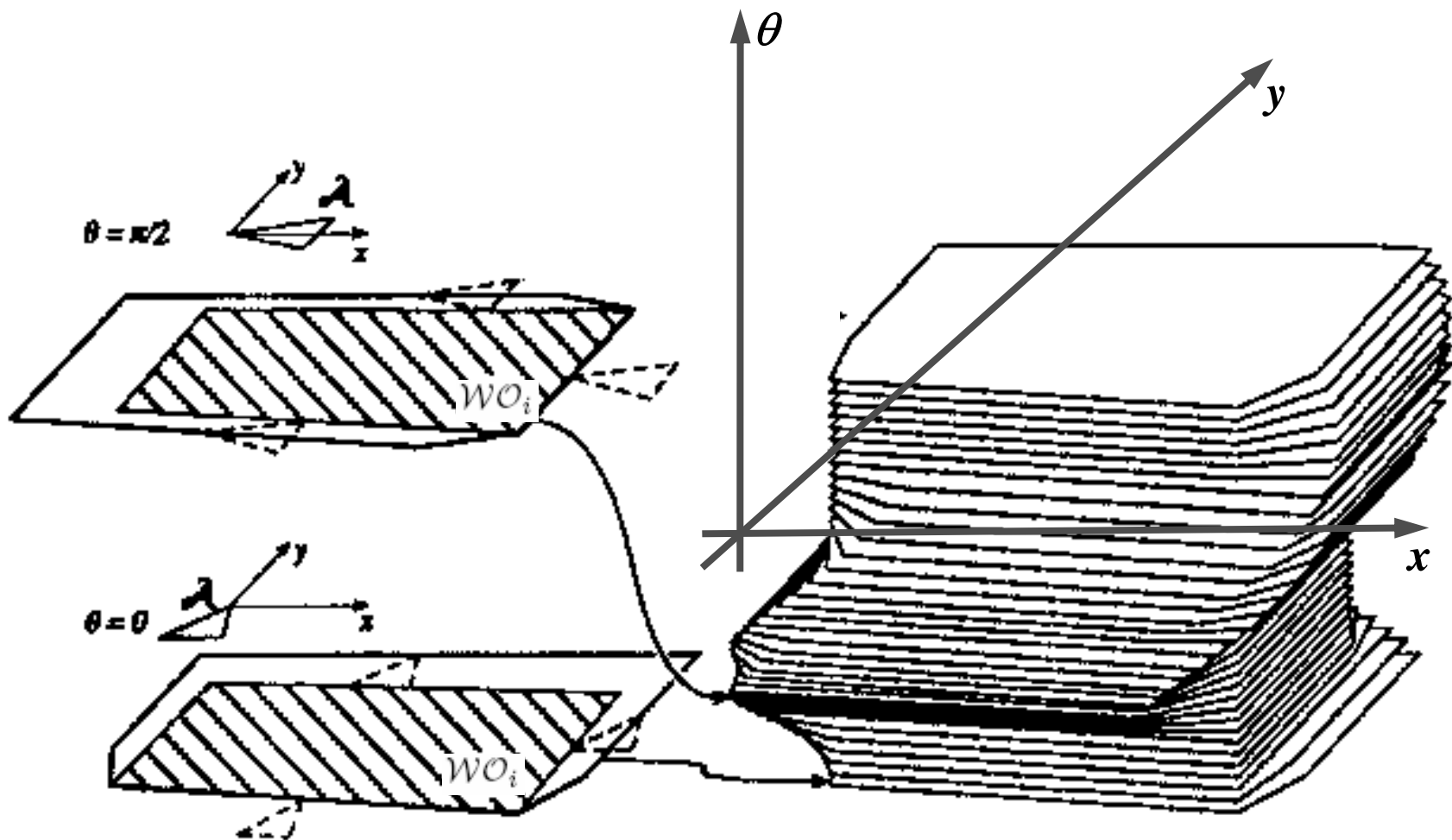
(The obstacle is blue.)



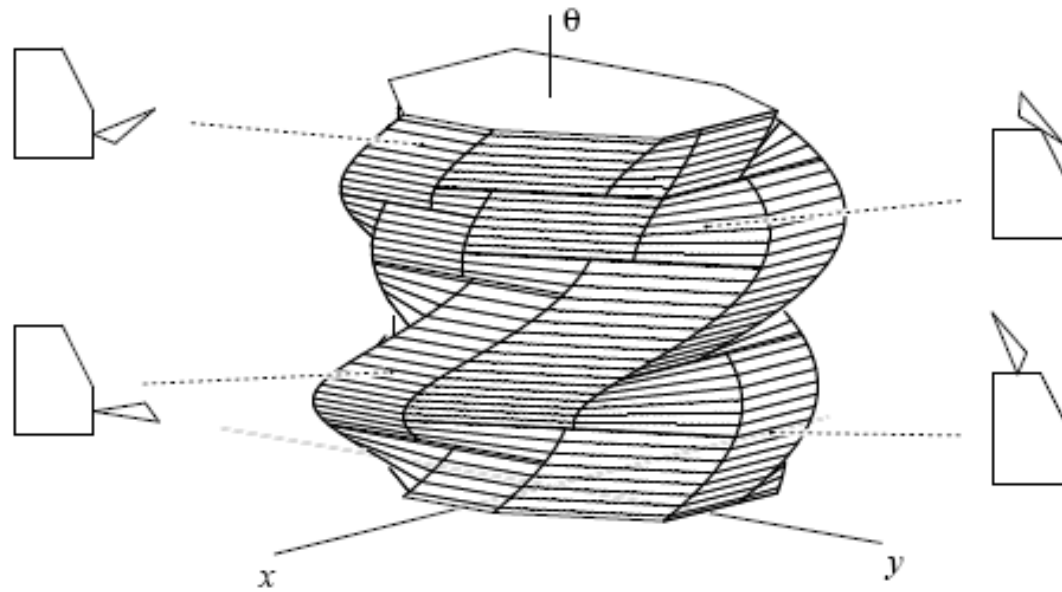
configuration space



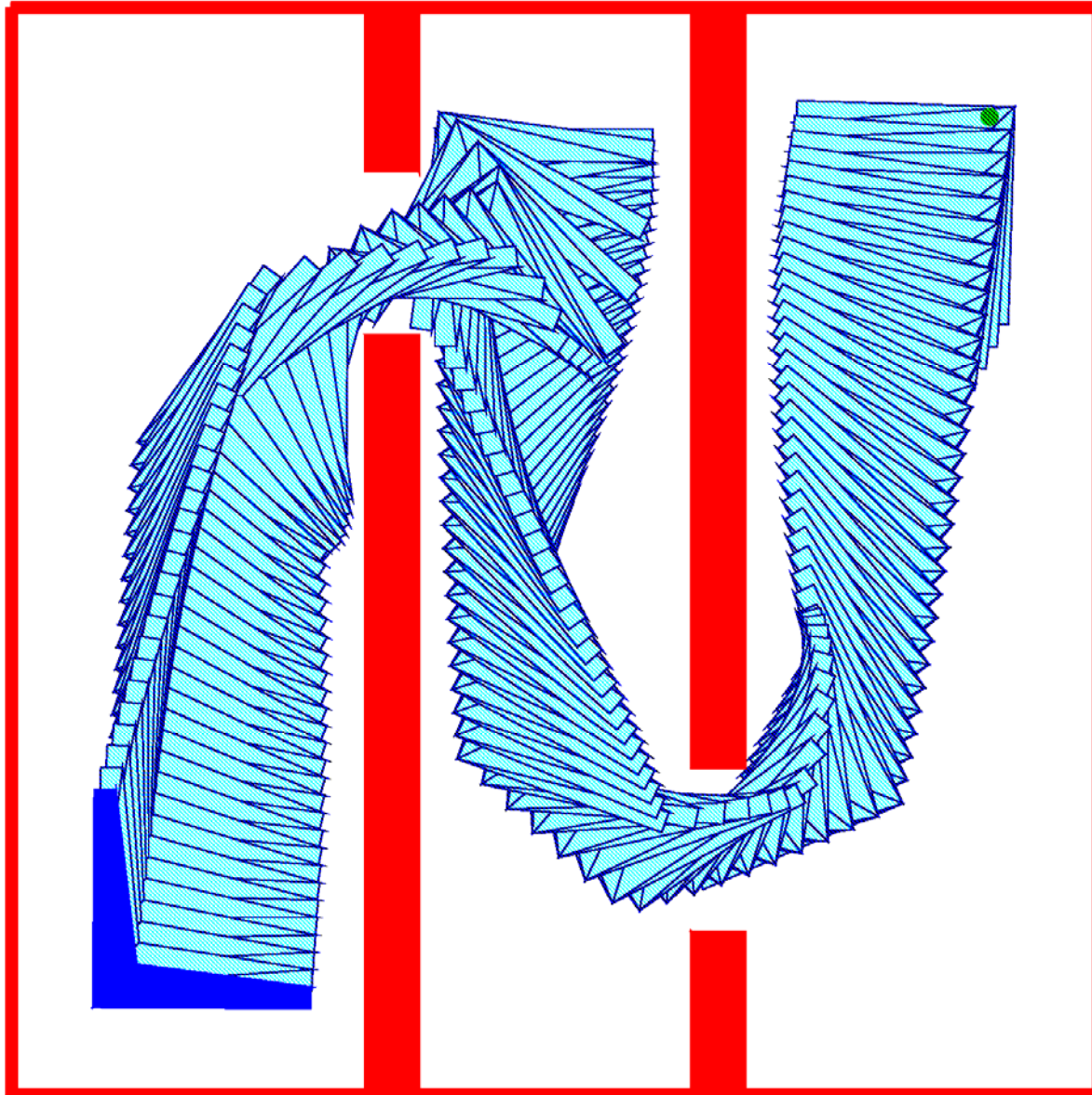
Polygonal robot translating & rotating in 2-D workspace



SE(2)

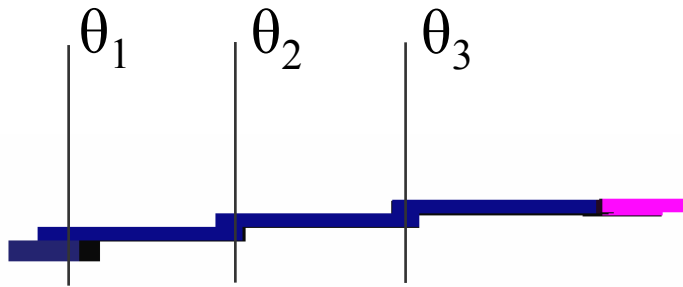


2D Rigid Object



16-735, Howie Choset with slides from G.D. Hager, Z. Dodds, and Dinesh Mocha

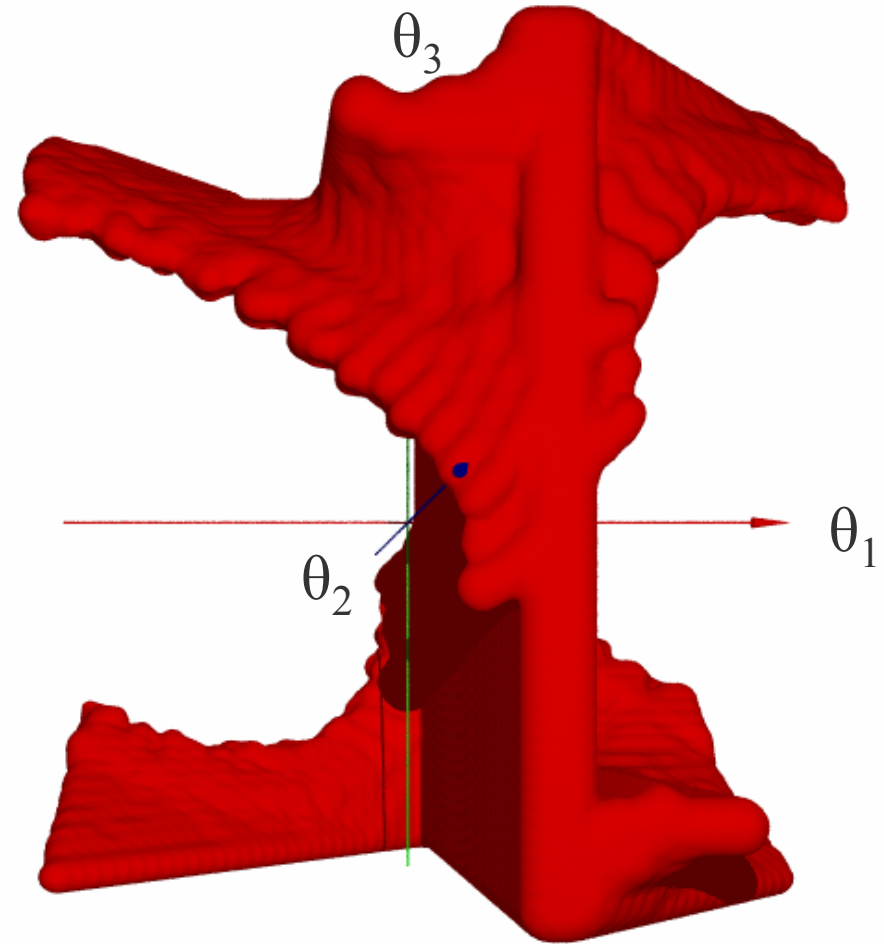
The Configuration Space (C-space)



**TOP
VIEW**

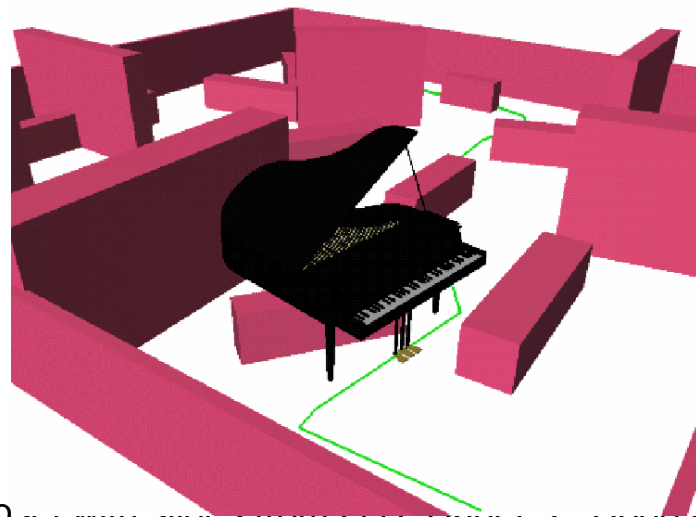
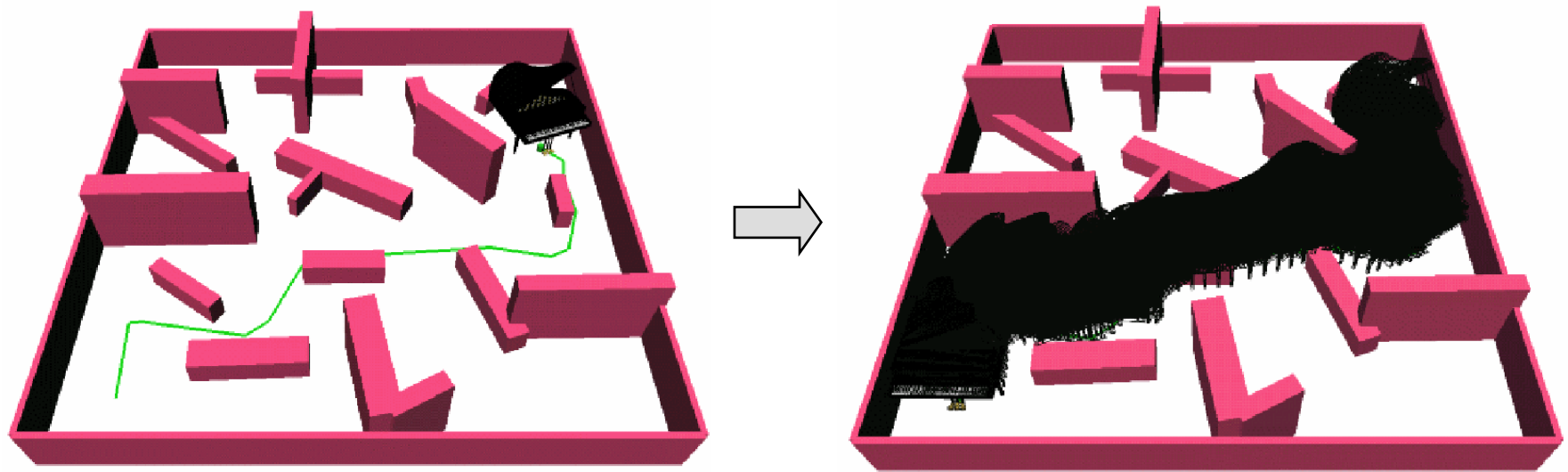


workspace



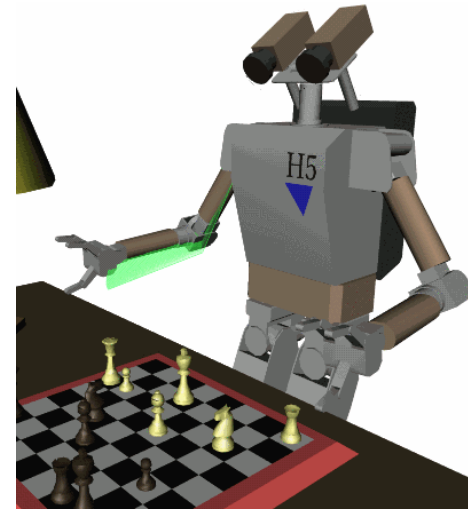
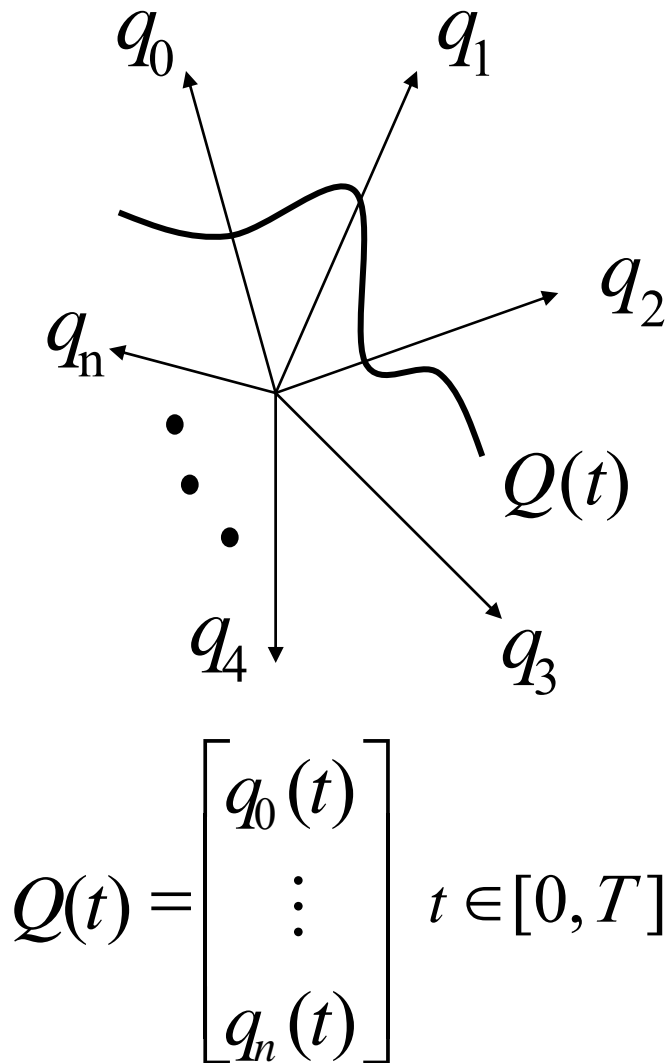
C-space

Moving a Piano

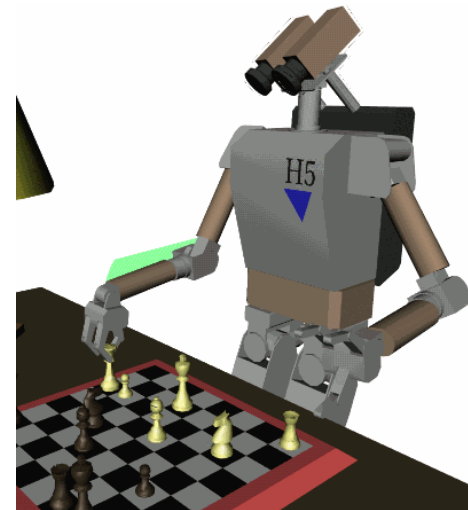


16-735, Howie Choset with slides from G.D. Hager, Z. Bouas, and Dinesh Mocha

Configuration Space (C-space)

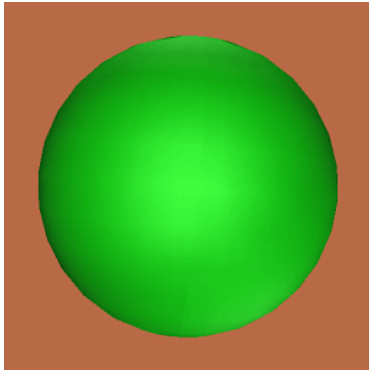


INIT:
 $Q(0)$



GOAL:
 $Q(T)$

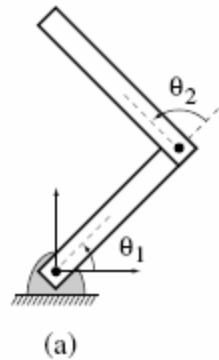
Topology?



Sphere?



Torus?



2R manipulator

Configuration space

Why study the Topology

- Extend results from one space to another: spheres to stars
- Impact the representation
- Know where you are
- Others?

The Topology of Configuration Space

- Topology is the “intrinsic character” of a space
- Two space have a different topology if cutting and pasting is required to make them the same (e.g. a sheet of paper vs. a mobius strip)
 - think of rubber figures --- if we can stretch and reshape “continuously” without tearing, one into the other, they have the same topology
- A basic mathematical mechanism for talking about topology is the homeomorphism.

Homeo- and Diffeomorphisms

- Recall mappings:
 - $\phi: S \rightarrow T$
 - If each element of S goes to a unique T , ϕ is *injective* (or 1-1)
 - If each element of T has a corresponding preimage in S , then ϕ is *surjective* (or onto).
 - If ϕ is surjective and injective, then it is bijective (in which case an inverse, ϕ^{-1} exists).
 - ϕ is *smooth* if derivatives of all orders exist (we say ϕ is C^∞)
- If $\phi: S \rightarrow T$ is a bijection, and both ϕ and ϕ^{-1} are continuous, ϕ is a *homeomorphism*; if such a ϕ exists, S and T are *homeomorphic*.
- If homeomorphism where both ϕ and ϕ^{-1} are smooth is a *diffeomorphism*.

Some Examples

- How would you show a square and a rectangle are diffeomorphic?
- How would you show that a circle and an ellipse are diffeomorphic (implies both are topologically S^1)
- Interestingly, a “racetrack” is not diffeomorphic to a circle
 - composed of two straight segments and two circular segments
 - at the junctions, there is a discontinuity; it is therefore not possible to construct a smooth map!
 - How would you show this (hint, do this for a function on \mathbb{R}^1 and think about the chain rule)
 - Is it homeomorphic?

Local Properties

$B_\epsilon(p) = \{p' \in \mathcal{M} \mid d(p, p') < \epsilon\}$ Ball

$p \in \mathcal{M}$ $\mathcal{U} \subseteq \mathcal{M}$ with $p \in \mathcal{U}$ such that for every $p' \in \mathcal{U}$, $B_\epsilon(\overline{p'}) \subset \mathcal{U}$. Neighborhood

Manifolds

- A space S *locally diffeomorphic* (homeomorphic) to a space T if each $p \in S$ there is a neighborhood containing it for which a diffeomorphism (homeomorphism) to some neighborhood of T exists.
- S^1 is locally diffeomorphic to \mathcal{R}^1
- The sphere is locally diffeomorphic to the plane (as is the torus)
- A set S is a *k -dimensional manifold* if it is locally **homeomorphic** to \mathcal{R}^k

Charts and Differentiable Manifolds

- A Chart is a pair (U, ϕ) such that U is an open set in a k -dimensional manifold and ϕ is a diffeomorphism from U to some open set in \mathcal{R}^k
 - think of this as a “coordinate system” for U (e.g. lines of latitude and longitude away from the poles).
 - The inverse map is a parameterization of the manifold
- Many manifolds require more than one chart to cover (e.g. the circle requires at least 2)
- An *atlas* is a set of charts that
 - cover a manifold
 - are smooth where they overlap (the book defines the notion of C^∞ related for this; we will take this for granted).
- A set S is a *differentiable manifold of dimension n* if there exists an atlas from S to \mathcal{R}^n
 - For example, this is what allows us (locally) to view the (spherical) earth as flat and talk about translational velocities upon it.

Some Minor Notational Points

- $\mathbb{R}^1 \times \mathbb{R}^1 \times \dots \times \mathbb{R}^1 = \mathbb{R}^n$
- $S^1 \times S^1 \times \dots \times S^1 \neq S^n$ ($= T^n$, the n-dimensional torus)
- S^n is the n-dimensional sphere

- Although S^n is an n-dimensional manifold, it is not a manifold of a single chart --- there is no single, smooth, invertible mapping from S^n to \mathbb{R}^n ---
 - they are not ??morphic?

Examples

Type of robot	Representation of Q
Mobile robot translating in the plane	\mathbb{R}^2
Mobile robot translating and rotating in the plane	$SE(2)$ or $\mathbb{R}^2 \times S^1$
Rigid body translating in the three-space	\mathbb{R}^3
A spacecraft	$SE(3)$ or $\mathbb{R}^3 \times SO(3)$
An n -joint revolute arm	T^n
A planar mobile robot with an attached n -joint arm	$SE(2) \times T^n$

$S^1 \times S^1 \times \dots \times S^1$ (n times) = T^n , the n -dimensional torus

$S^1 \times S^1 \times \dots \times S^1$ (n times) $\neq S^n$, the n -dimensional sphere in \mathbb{R}^{n+1}

$S^1 \times S^1 \times S^1 \neq SO(3)$

$SE(2) \neq \mathbb{R}^3$

$SE(3) \neq \mathbb{R}^6$

What is the Dimension of Configuration Space?

- The dimension is the number of parameter necessary to uniquely specify configuration
- One way to do this is to explicitly generate a parameterization (e.g with our 2-bar linkage)
- Another is to start with too many parameters and add (independent) constraints
 - suppose I start with 4 points in the plane (= 8 parameters), A, B, C, D
 - Rigidity requires $d(A,B) = c_1$ (1 constraints)
 - Rigidity requires $d(A,C) = c_2$ and $d(B,C) = c_3$ (2 constraints)
 - Rigidity requires $d(A,D) = c_4$ and $d(B,D) = c_5$ and ??? (?? constraints)

 - HOW MANY D.O.F?
- QUIZ:
 - HOW MANY DOF DO YOU NEED TO MOVE FREELY IN 3-space?

What is the Dimension of Configuration Space?

- The dimension is the number of parameter necessary to uniquely specify configuration
- One way to do this is to explicitly generate a parameterization (e.g with our 2-bar linkage)
- Another is to start with too many parameters and add (independent) constraints
 - suppose I start with 4 points in the plane (= 8 parameters), A, B, C, D
 - Now, require $\|A-B\| = c_1$ and $\|C-D\| = c_2$ (2 constraints)
 - Now, require $B = C$ (? constraints)
 - Now, fix $A = 0$ (? constraints)
 - HOW MANY D.O.F?
- QUIZ:
 - HOW MANY DOF DO YOU NEED TO MOVE FREELY IN 3-space?
 - 3+3
 - HOW MANY in 4-space?

More on dimension

\mathbb{R}^1 and $SO(2)$ are one-dimensional manifolds;

\mathbb{R}^2 , S^2 and T^2 are two-dimensional manifolds;

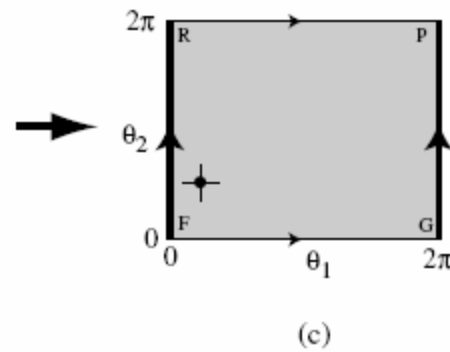
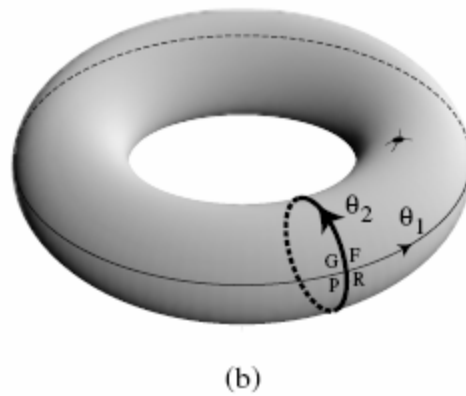
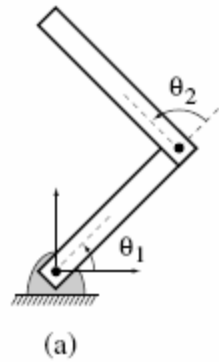
\mathbb{R}^3 , $SE(2)$ and $SO(3)$ are three-dimensional manifolds;

\mathbb{R}^6 , T^6 and $SE(3)$ are six-dimensional manifolds.

More Example Configuration Spaces (contrasted with workspace)

- Holonomic robot in plane:
 - workspace \mathbb{R}^2
 - configuration space \mathbb{R}^2
- 3-joint revolute arm in the plane
 - Workspace, a torus of outer radius $L1 + L2 + L3$
 - configuration space T^3
- 2-joint revolute arm with a prismatic joint in the plane
 - workspace disc of radius $L1 + L2 + L3$
 - configuration space $T^2 \times \mathbb{R}$
- 3-joint revolute arm mounted on a mobile robot (holonomic)
 - workspace is a “sandwich” of radius $L1 + L2 + L3$
 - $\mathbb{R}^2 \times T^3$
- 3-joint revolute arm floating in space
 - workspace is \mathbb{R}^3
 - configuration space is T^3

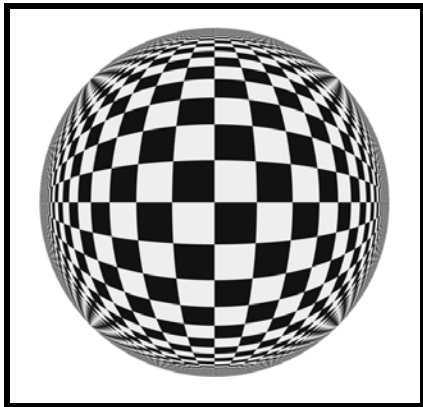
Parameterization of Torus



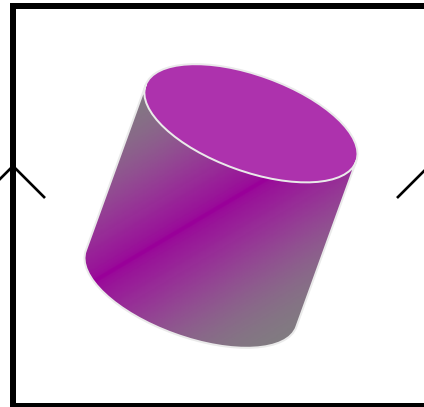
$$(\theta_1, \theta_2) \in \mathbb{R}^2,$$

problems at $\theta_i = \{0, 2\pi\}$

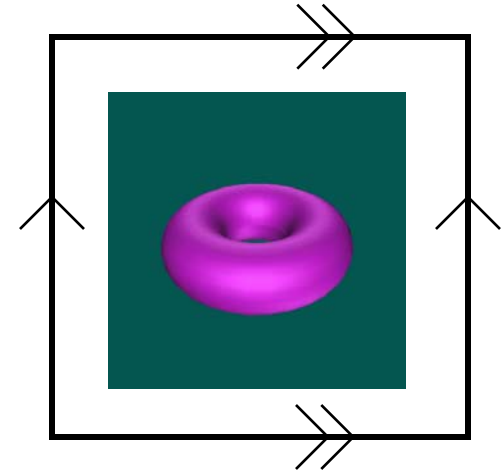
2d Manifolds



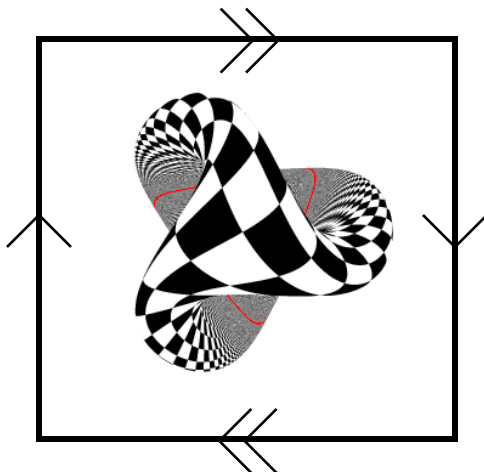
real plane



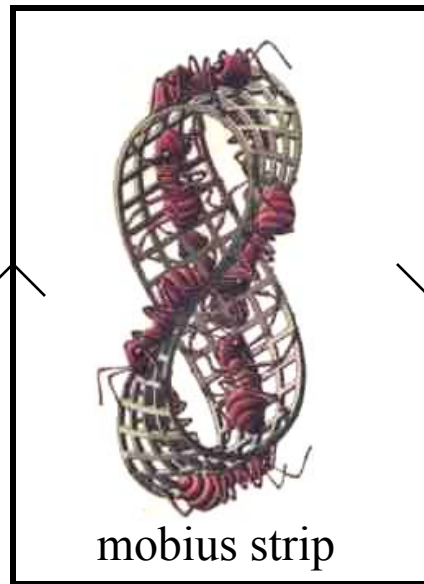
cylinder



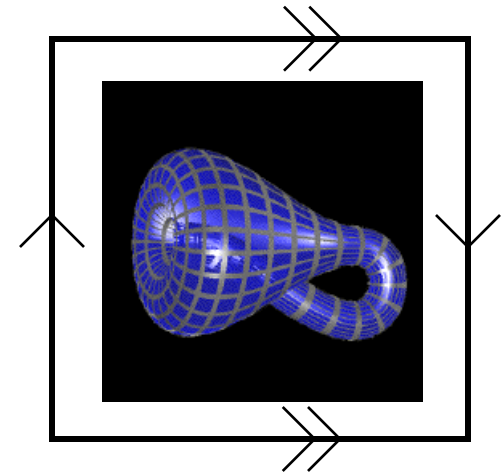
torus



projective plane



mobius strip



klein bottle

Representing Rotations

- Consider S^1 --- rotation in the plane
- The action of a rotation is to, well, rotate $\rightarrow R_\theta: \mathbb{R}^2 \rightarrow \mathbb{R}^2$
- We can represent this action by a matrix R that is applied (through matrix multiplication) to points in \mathbb{R}^2

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

- Note, we can either think of rotating a point through an angle, or rotate the **coordinate system (or frame)** of the point.

Geometric Transforms

Now, using the idea of homogeneous transforms,
we can write:

$$p' = \begin{pmatrix} R & T \\ \mathbf{0} & 1 \end{pmatrix} p$$

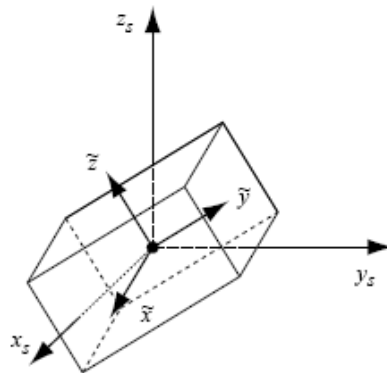
The group of rigid body rotations $SO(2) \times \mathfrak{R}(2)$ is
denoted $SE(2)$ (for special Euclidean group)

$$R = \begin{bmatrix} \tilde{x}_1 & \tilde{y}_1 \\ \tilde{x}_2 & \tilde{y}_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \in SO(2)$$

This space is a type of torus

From 2D to 3D Rotation

- I can think of a 3D rotation as a rotation about different axes:
 - $\text{rot}(x,\theta) \text{rot}(y,\theta) \text{rot}(z,\theta)$
 - there are many conventions for these (see Appendix E)
 - Euler angles (ZYX) --- where is the singularity (see eqn 3.8)
 - Roll Pitch Yaw (ZYX)
 - Angle axis
 - Quaternion
- The space of rotation matrices has its own special name: $SO(n)$ (for special orthogonal group of dimension n). It is a manifold of dimension n



$$R = \begin{bmatrix} \tilde{x}_1 & \tilde{y}_1 & \tilde{z}_1 \\ \tilde{x}_2 & \tilde{y}_2 & \tilde{z}_2 \\ \tilde{x}_3 & \tilde{y}_3 & \tilde{z}_3 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \in SO(3)$$

- What is the derivative of a rotation matrix?

Geometric Transforms

Now, using the idea of homogeneous transforms,
we can write:

$$p' = \begin{pmatrix} R & T \\ \mathbf{0} & 1 \end{pmatrix} p$$

The group of rigid body rotations $SO(3) \times \mathcal{R}(3)$ is
denoted $SE(3)$ (for special Euclidean group)

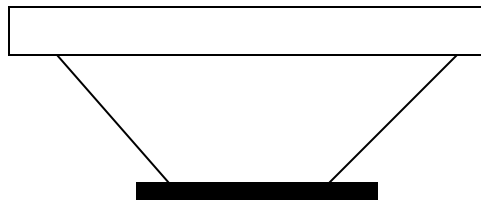
$$SE(n) \equiv \begin{bmatrix} SO(n) & \mathbb{R}^n \\ \mathbf{0} & 1 \end{bmatrix}$$

What does the inverse transformation look like?

Open vs. Closed Chains

- Serial (or open) chain mechanisms can usually be understood simply by looking at how they are put together (like our 2-link manipulator)
- Closed chain mechanisms have additional internal constraints --- the links form closed loops, e.g.

Suppose 4 revolute, 2 prismatic, 6 links



Gruebler's formula: $N(k-n-1) + \sum f_i$

N = DOF of space (here 3) f = dof of joints (here 1) n = # of joints; k # of links

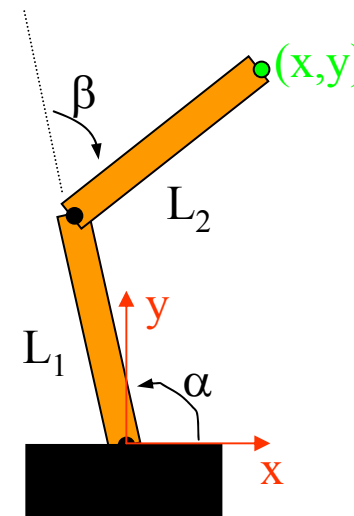
Transforming Velocity

- Recall forward kinematics $K: Q \rightarrow W$
- The *Jacobian* of K is the $n \times m$ matrix with entries
 - $J_{i,j} = d K_i / d q_j$

- The Jacobian transforms velocities:
 - $dw/dt = J dq/dt$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 c_\alpha \\ L_1 s_\alpha \end{bmatrix} + \begin{bmatrix} L_2 c_{\alpha+\beta} \\ L_2 s_{\alpha+\beta} \end{bmatrix}$$

- If square and invertible, then
 - $dq/dt = J^{-1} dw/dt$
- Example: our favorite two-link arm...



A Useful Observation

- The Jacobian maps configuration velocities to workspace velocities
- Suppose we wish to move from a point A to a point B in the workspace along a path $p(t)$ (a mapping from some time index to a location in the workspace)
 - dp/dt gives us a velocity profile --- how do we get the configuration profile?
 - Are the paths the same if choose the shortest paths in workspace and configuration space?

Summary

- Configuration spaces, workspaces, and some basic ideas about topology
- Types of robots: holonomic/nonholonomic, serial, parallel
- Kinematics and inverse kinematics
- Coordinate frames and coordinate transformations
- Jacobians and velocity relationships

T. Lozano-Pérez.
Spatial planning: A configuration space approach.
IEEE Transactions on Computing, C-32(2):108-120, 1983.

A Few Final Definitions

- A manifold is *path-connected* if there is a path between any two points.
- A space is *compact* if it is closed and bounded
 - configuration space might be either depending on how we model things
 - compact and non-compact spaces cannot be diffeomorphic!
- With this, we see that for manifolds, we can
 - live with “global” parameterizations that introduce odd singularities (e.g. angle/elevation on a sphere)
 - use atlases
 - embed in a higher-dimensional space using constraints
- Some prefer the later as it often avoids the complexities associated with singularities and/or multiple overlapping maps