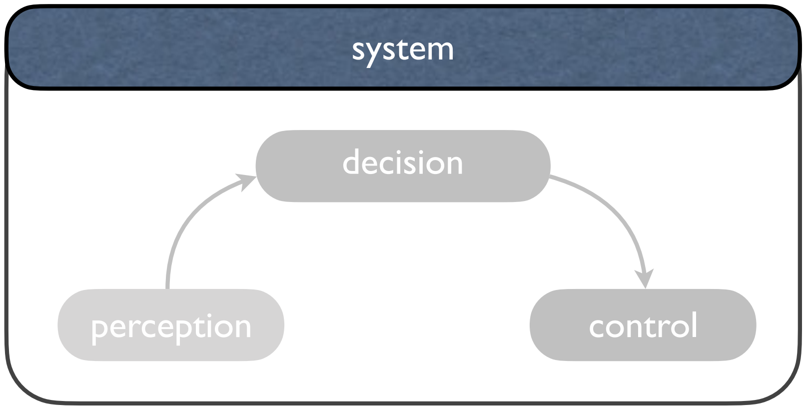
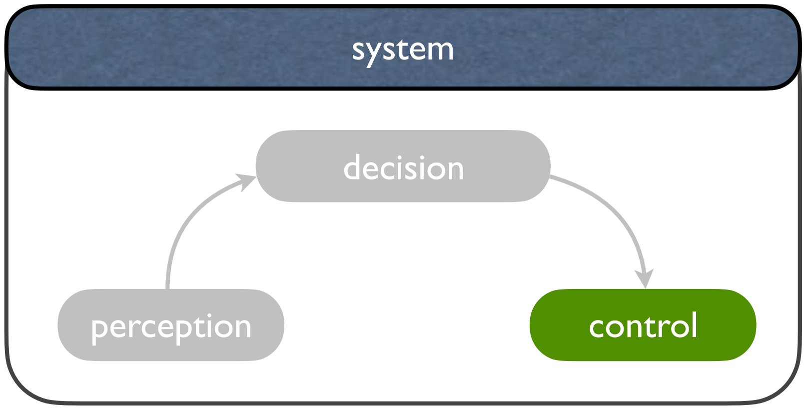
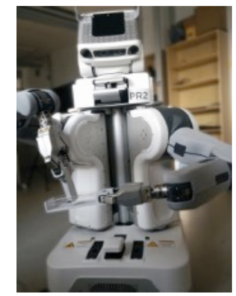
Last lecture ...



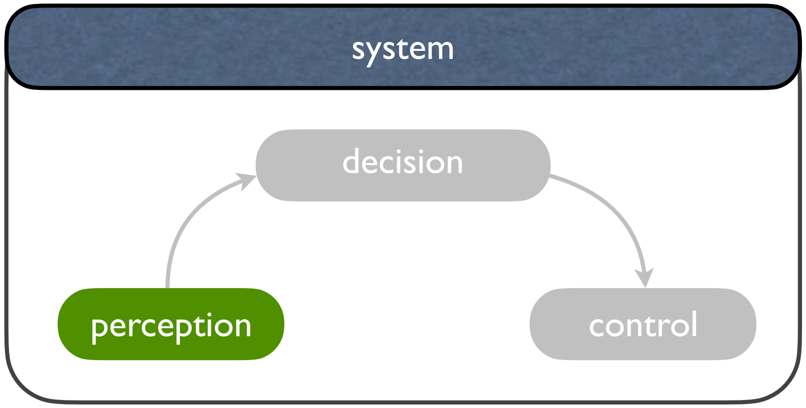
A classic robot architecture.

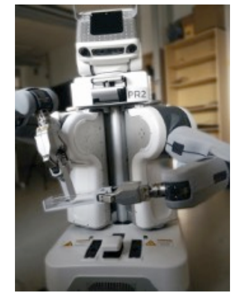




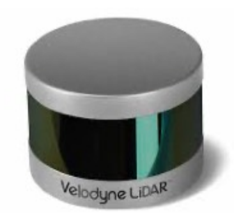
actuator

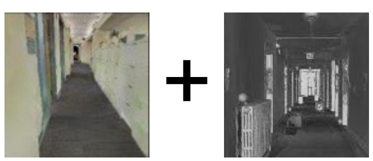
Last lecture …



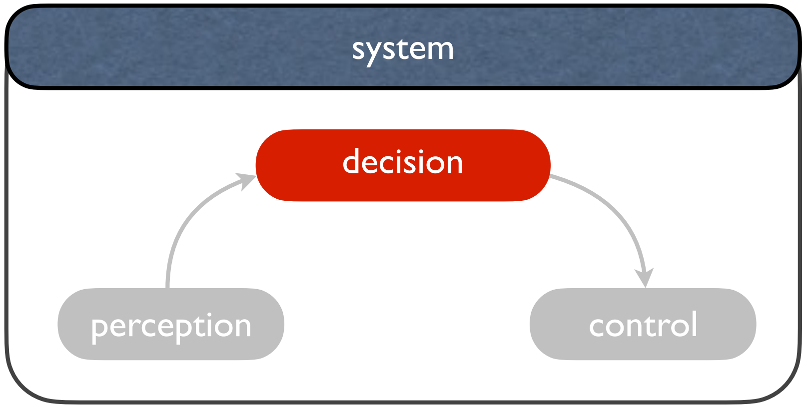


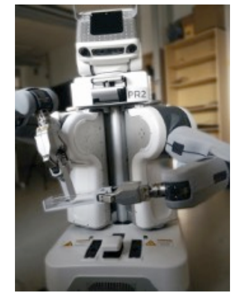
sensor





A classic robot architecture.



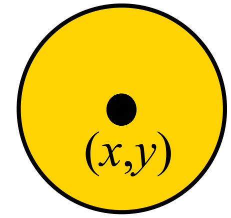


Question. Do we need to build/rebuild a "new" decision making system for every robot, for every combination of sensors and actuators, for …?

Configuration space is the key unifying concept that enables us to think about a "single" abstract robot for decision making.

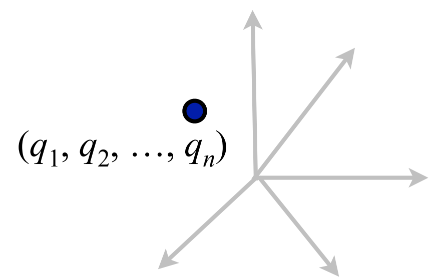
Tomas Lozano-Perez

Configuration space. The configuration of a moving object is a set of parameters $(q_1, q_2, \ldots, q_n)$ that completely specifies of the position of every point on the object.

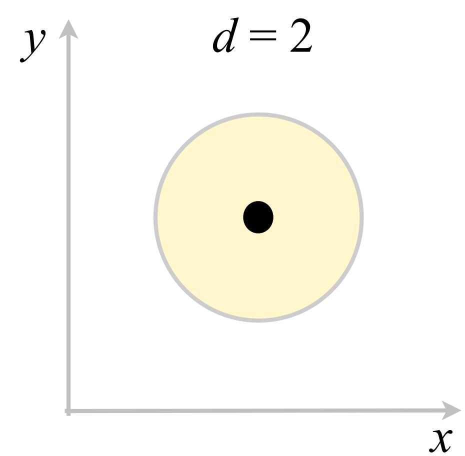


The coordinates for the disc center specify the configuration of a translation-only disc.

The configuration space (C-space) C is the set of all possible configurations. A configuration is a point in C.



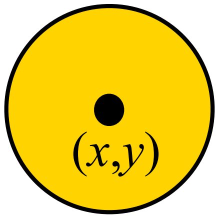
The dimension d of a configuration space is the minimum number of parameters required to specify the configuration of an object. It is also called the number of degrees of freedom (dofs) of a moving object.

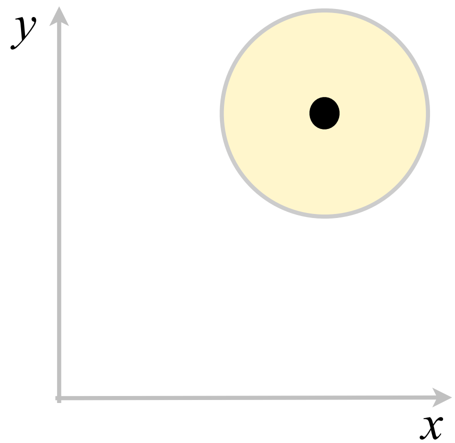


C-space topology.

Topology is the geometry of the space subject to continuous deformations. Intuitively, it is rubber-band geometry.

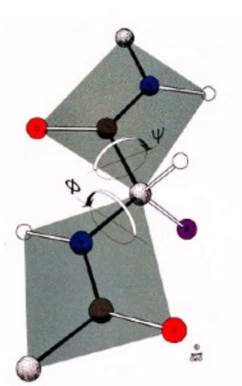
Example. R^2

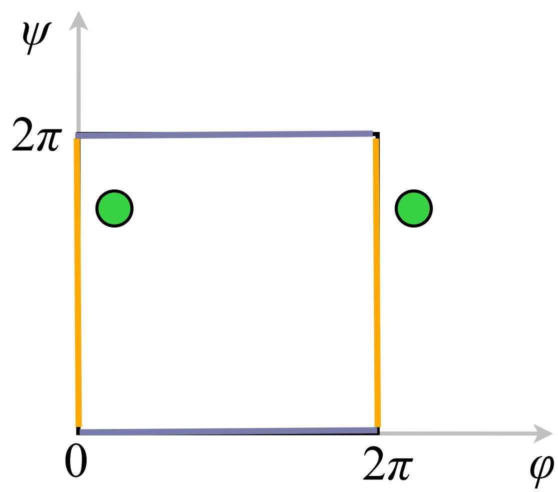


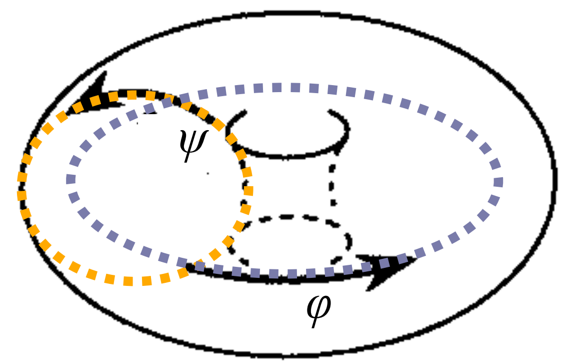


The center of the disc may take any position in the plane.

Example. $S^1 \times S^1$

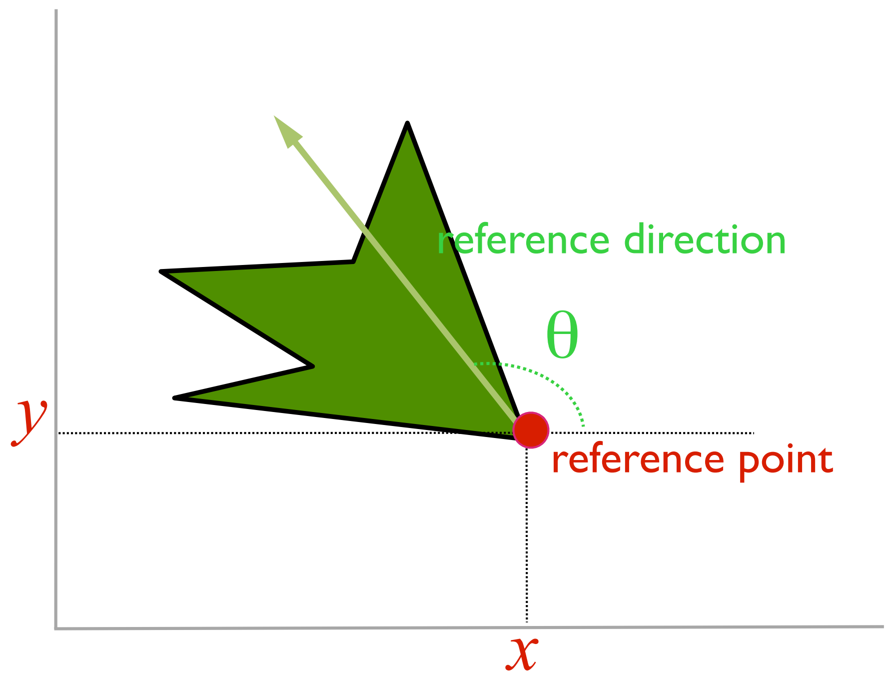






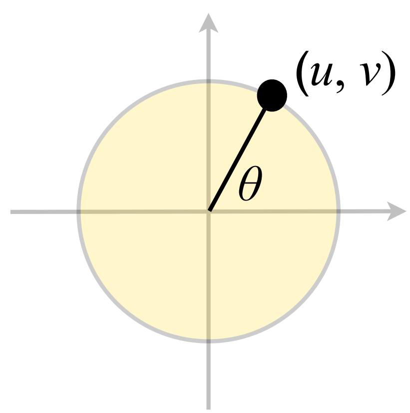
The opposing edges are identified, because 0 and 2π represents the same angle. The space thus "wraps around" to form a torus.

The configuration space of a rigid body translating and rotating in 2D space.

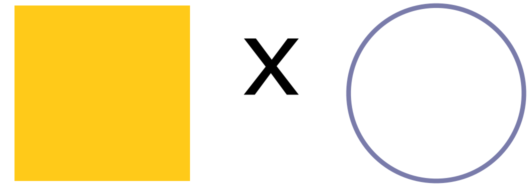


The configuration is specified by 3 parameters: $q = (x, y, \theta)$ with $\theta \in [0, 2\pi)$.

The configuration is specified by 4 parameters: $q = (x, y, u, v)$ with $u^2+v^2 = 1$. Note that $u = \cos\theta$ and $v = \sin\theta$.



The topology is a 3-D cylinder $C = R^2 \times S^1$.



Question. What is the dimension d of the configuration space for a moving object translating and rotating in a 2-D environment?

The dimension of the configuration space d = 3, as we have a 3-parameter specification (x,y, θ). Alternatively, using the 4-parameter specification, we have a constraint. So d = 4-1 = 3.

Question. Does the dimension of the configuration space depend on the parametrization? Does the topology depend on the parametrization?

Both the dimension and the topology are intrinsic properties of a configuration and are invariant with respect to parametrization.

The configuration space of a rigid body translating and rotating in 3D space.

$$q = (\underbrace{x, y, z,}_{\text{translation}}\ \underbrace{?}_{\text{rotation}})$$

In 3D space, specifying the translation is straightforward by additional one coordinate. How do we specify the rotation?

Rotation matrix. An orthonormal matrix contains 9 parameters:

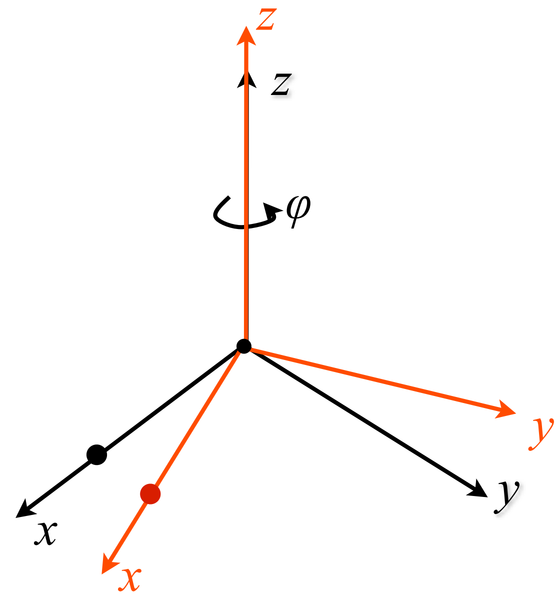
$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

with

- $r_{1i}^2 + r_{2i}^2 + r_{3i}^2 = 1$ for all i,
- $r_{1i} r_{1j} + r_{2i} r_{2j} + r_{3i} r_{3j} = 0$ for all $i \neq j$,
- $\det(R) = +1$

The matrix for rotating about the z-axis is

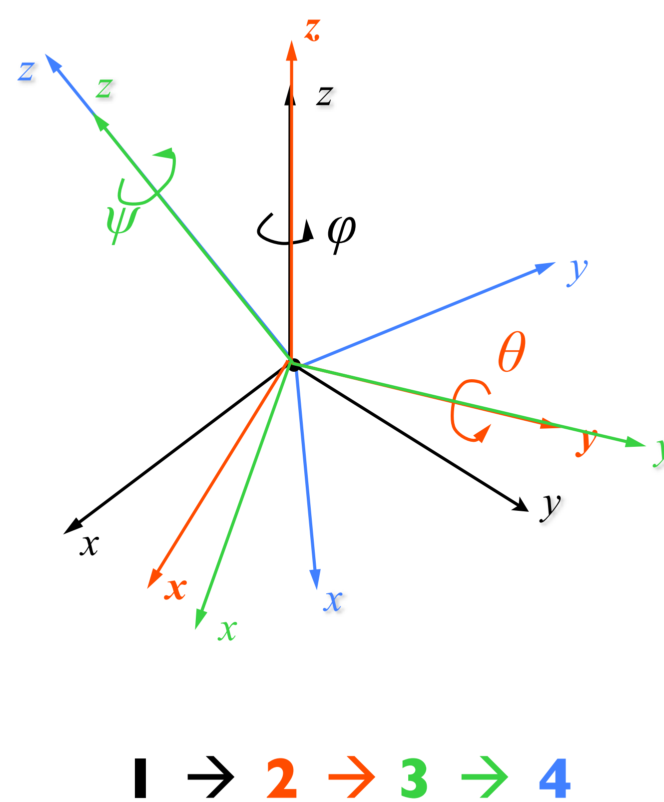
$$R_{z,\varphi} = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



To confirm, we apply the matrix to the unit vector $(1{,}0{,}0)$ and check its effect:

$$\begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{pmatrix}$$

Euler angles. Euler angles specify the rotation with 3 angles, one for each chosen axis. In our example, we have chosen z, y, and then z again. There are many valid combinations of axis choices. However, not all combinations are valid. For example, y-y-y is not valid, as it cannot represent rotation about the x-axis.



Euler angles are historically important, but have various undesirable limitations. Suppose that we perform two successive rotations, both represented as Euler angles. It is difficult to calculate the Euler angles for the cumulative final outcome.

Euler angles and rotation matrices are related. In our example, $R = R_{z,\varphi} R_{y,\theta} R_{z,\psi}$

Axis-angle. This specification contains 4 parameters: $q = (n_x, n_y, n_z, \theta)$ with $n_x^2 + n_y^2 + n_z^2 = 1$. (n_x, n_y, n_z) is a unit vector representing the axis of rotation, and θ is the angle of rotation about the axis. This is intuitive. However, it is not easy to compose rotations with the axis-angle specification, just like Euler angles.

Unit quaternion. This specification also contains 4 parameters: $u = (u_1, u_2, u_3, u_4)$ with $u_1^2 + u_2^2 + u_3^2 + u_4^2 = 1$. The intuition is similar to that of the axis-angle specification. It is connected to the axis-angle specification through the following relationship:

$$(u_1, u_2, u_3, u_4) = (\cos\frac{\theta}{2}, n_x \sin\frac{\theta}{2}, n_y \sin\frac{\theta}{2}, n_z \sin\frac{\theta}{2})$$

with $n_x^2 + n_y^2 + n_z^2 = 1$. Basically, it is the axis-angle specification re-coded into a different set of 4 numbers.

Compare unit quaternion specification of 3D rotations with the (u,v)-representation of 2-D rotations: (u,v) with $u^2+v^2 = 1$.

The unit quaternion representation offers many advantages over the alternatives:

- Compact
- Naturally capture the topology of 3-D rotation space
- No singularity

It is also easy to compose rotations directly through the quaternion algebra.

The unit quaternion representation is usually the preferred choice for representing rotations.

Question. What is the topology of the configuration space for a moving object translating and rotating in a 3-D environment? What is the dimension of this configuration space?

C = R3 x SO(3).
Using the 3-parameter Euler-angle representation, we see that the dimension of the 3-D rotation space is d=3.

The same can be derived from the 4-parameter axis-angle or the unit quaternion representation. There are 4 parameters and 1 constraint. So d = 4-1 = 3.

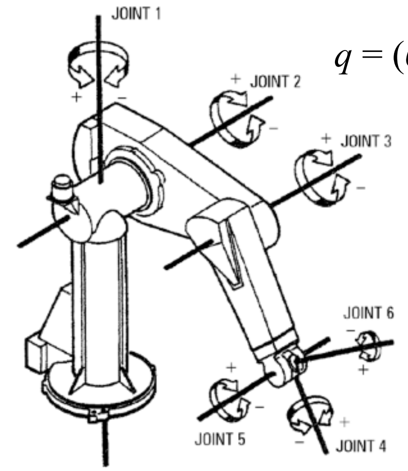
It is more difficult to derive the answer from the orthonormal matrix representation. There are 9 parameters. How many constraints are there?

Articulated robots. The configuration is specified by $q = (\theta_1, \theta_2 \ldots, \theta_n)$.

The dimension of the configuration space C is n.

The topology of C is

- a hyper-cube in R^n if there are mechanical stops,
- $S^1 \times S^1 \times \ldots \times S^1$ if there are no mechanical stops.



$q = (\theta_1, \theta_2 \ldots, \theta_6)$

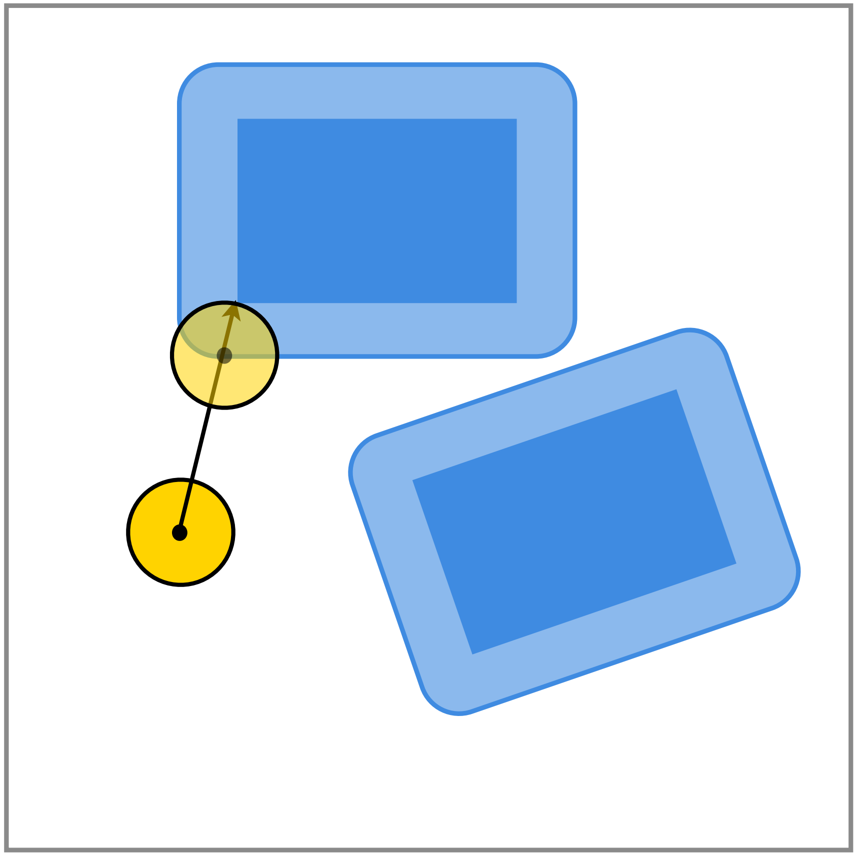
A mechanical stop sets the maximum angle by which a rotational joint can turn

C-space obstacles. A configuration q is collision-free, or free, if a moving object placed at q does not intersect any obstacles in the workspace.

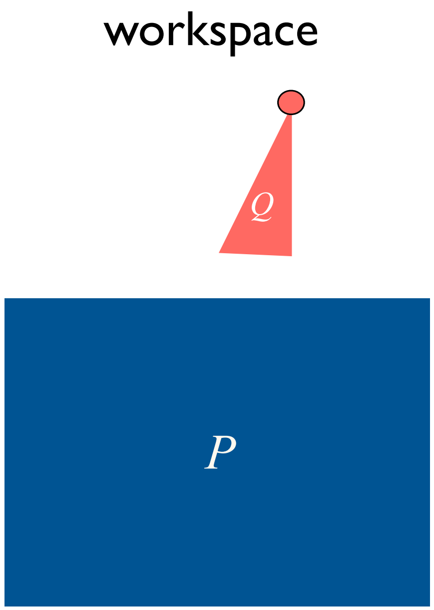
The free space F is the set of all free configurations.

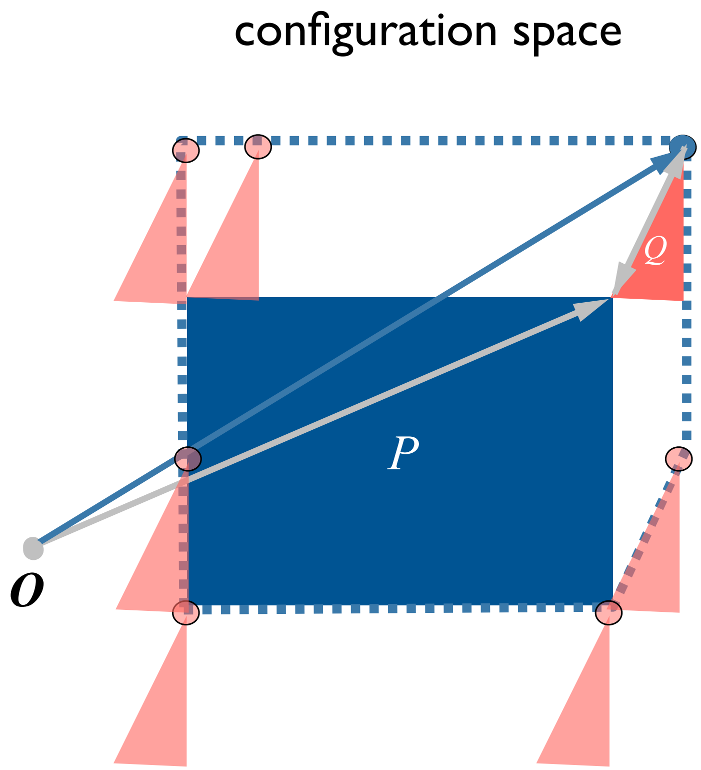
A configuration-space obstacle (C-obstacle) is the set of configurations where the moving object collides with the workspace obstacles or with itself.

In this example, the “expanded” obstacles are in fact C-space obstacles for a disc robot.

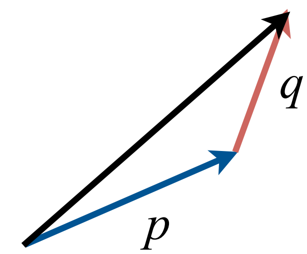


For a polygonal robot, how do we expand the obstacles and compute the C-space obstacles?





We can formalize this process of tracing the locus of a reference point on the robot using Minkowski sum.



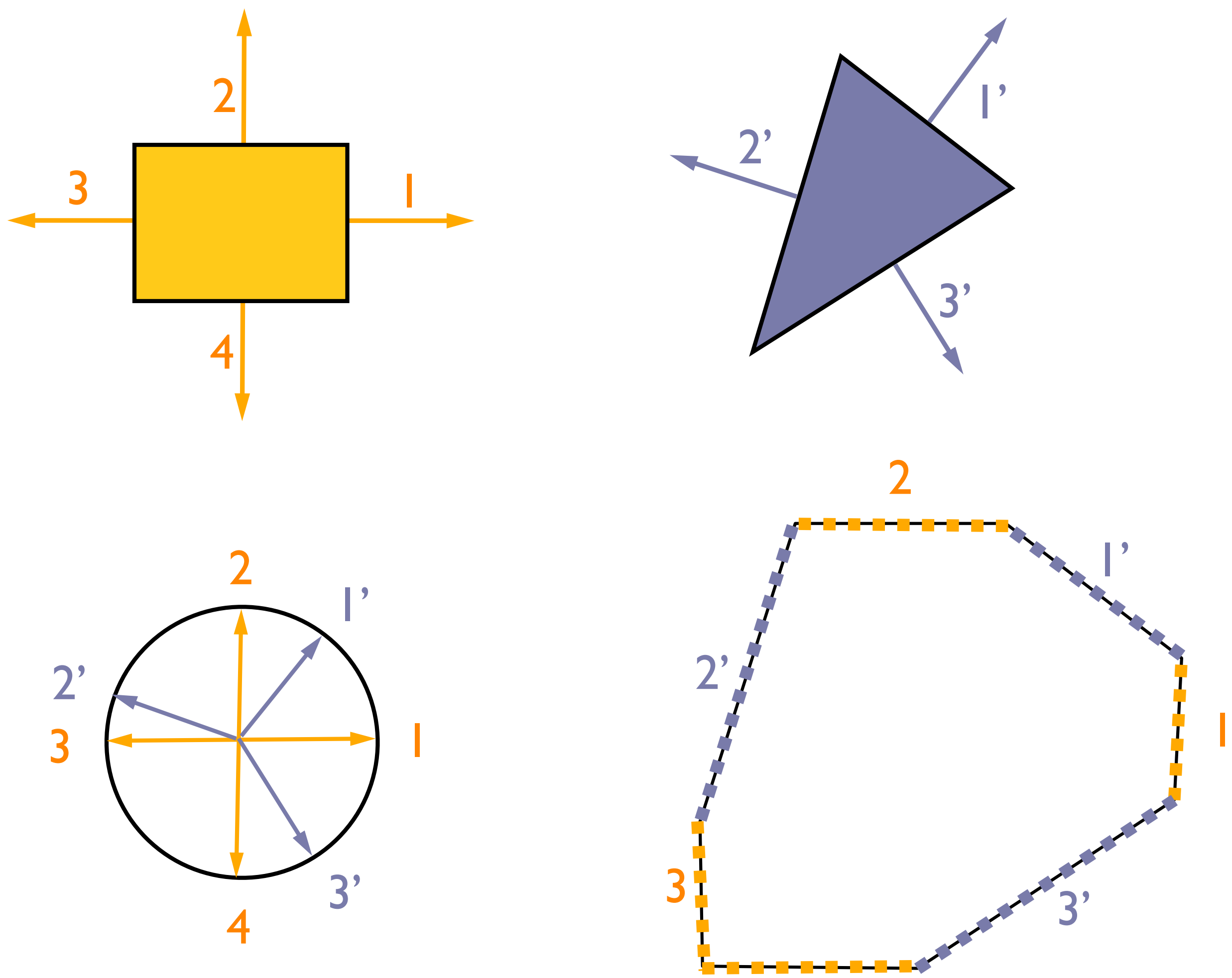
Minkowski sum. The Minkowski sum of two sets P and Q is defined as $P \oplus Q = \{ p + q \mid p \in P, q \in Q \}$.

The Minkowski difference of two sets P and Q is defined as $P \ominus Q = \{ p - q \mid p \in P, q \in Q \}$.

Computing the Minkowski sum of two convex polygons.

Computing the Minkowski sum of two convex polygons.

- Sort the edges of two convex polygons according to the orientation of their outward normals.
- Retrieve the edges in this order according to their corresponding normals, and link them together.



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, each of which is the "sum" of a vertex of P and a vertex of Q. The worst-case running time is $O(n+m)$.

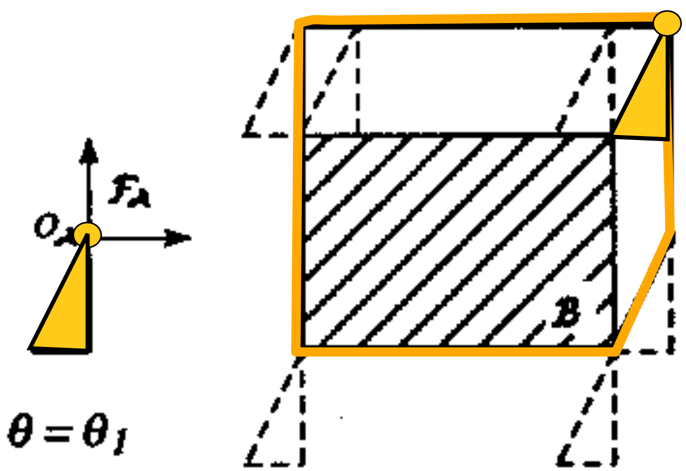
Computing the Minkowski sum of two non-convex polygons.

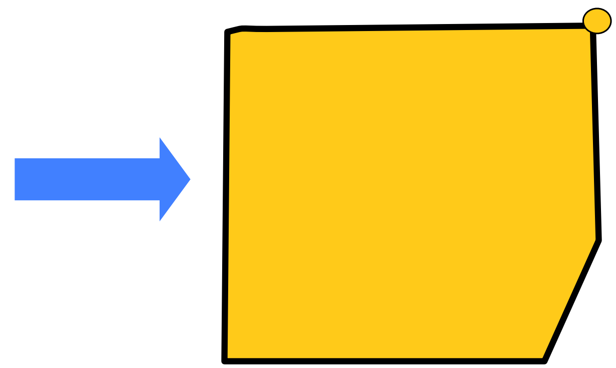
- Decompose each non-convex polygons (P and Q) into a union of convex polygons, e.g., triangles.
- Compute the Minkowski sum $P_i \oplus Q_j$ for all pairs P_i and Q_j with $P_i \subset P$ and $Q_i \subset Q$.
- Take the union of all computed Minkowski sums.

Compute the C-space obstacles for a polygonal robot translating in 2-D workspace. Compute the Minkowski difference of the obstacle and the robot.

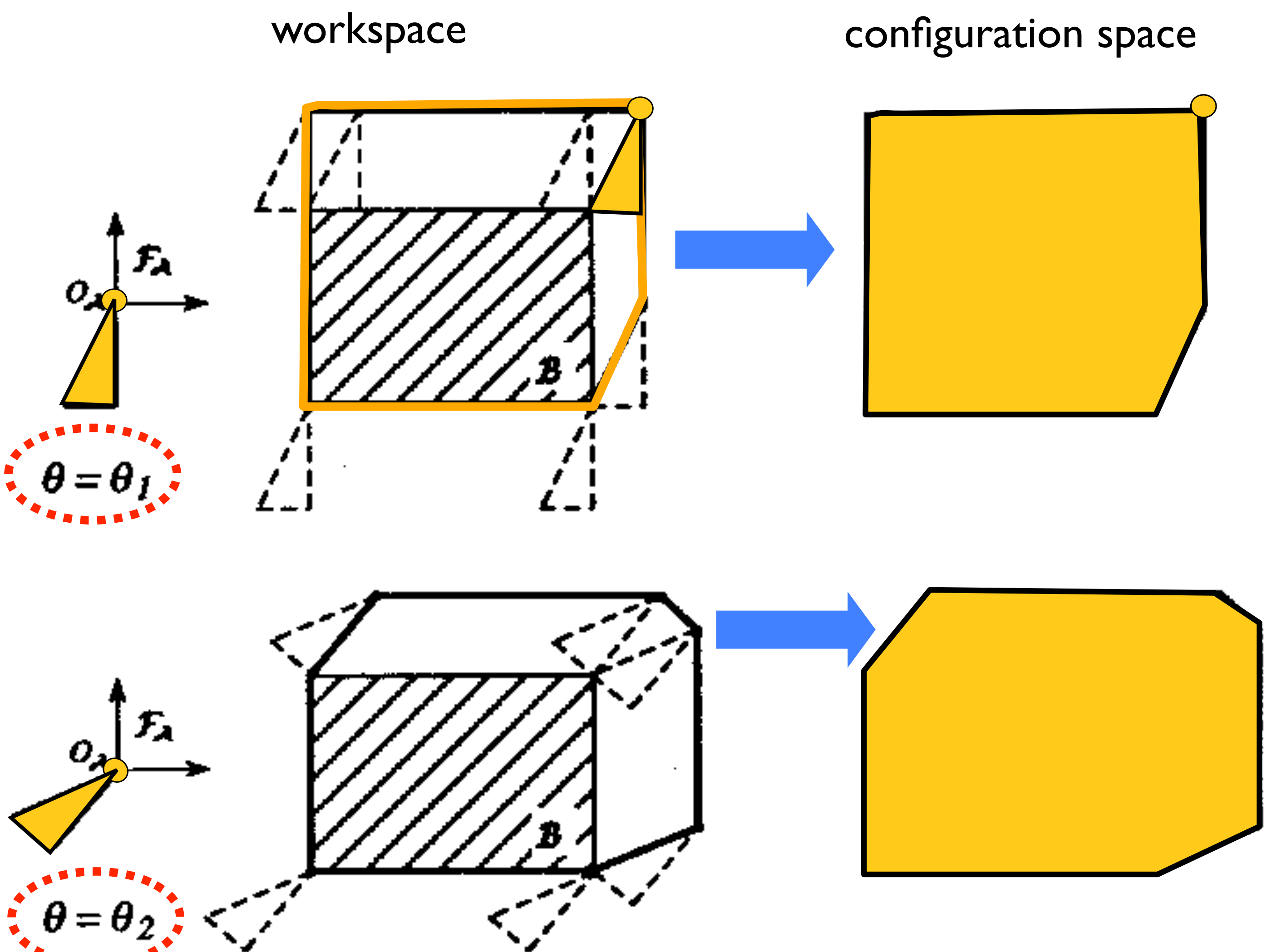
workspace

configuration space

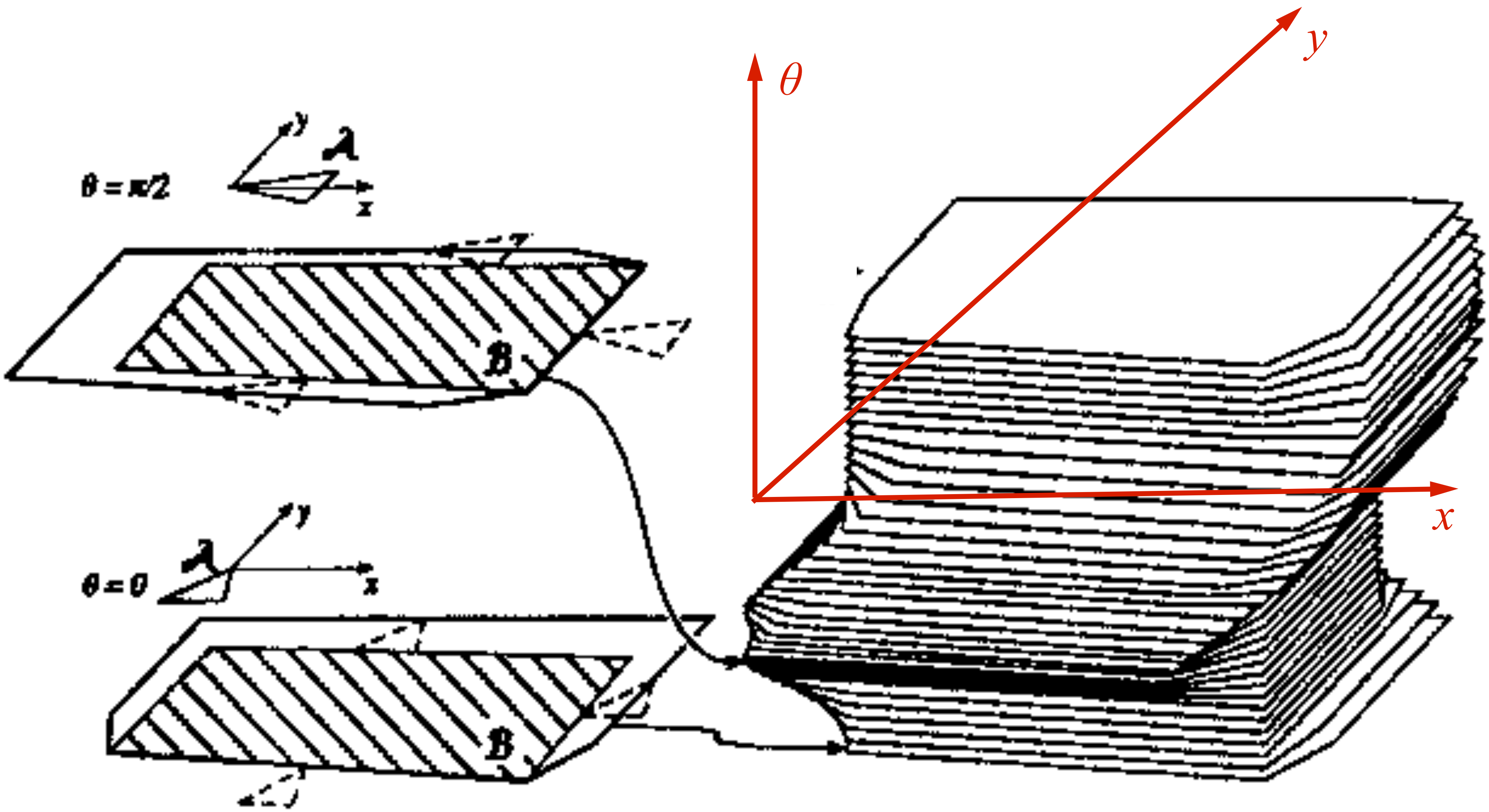




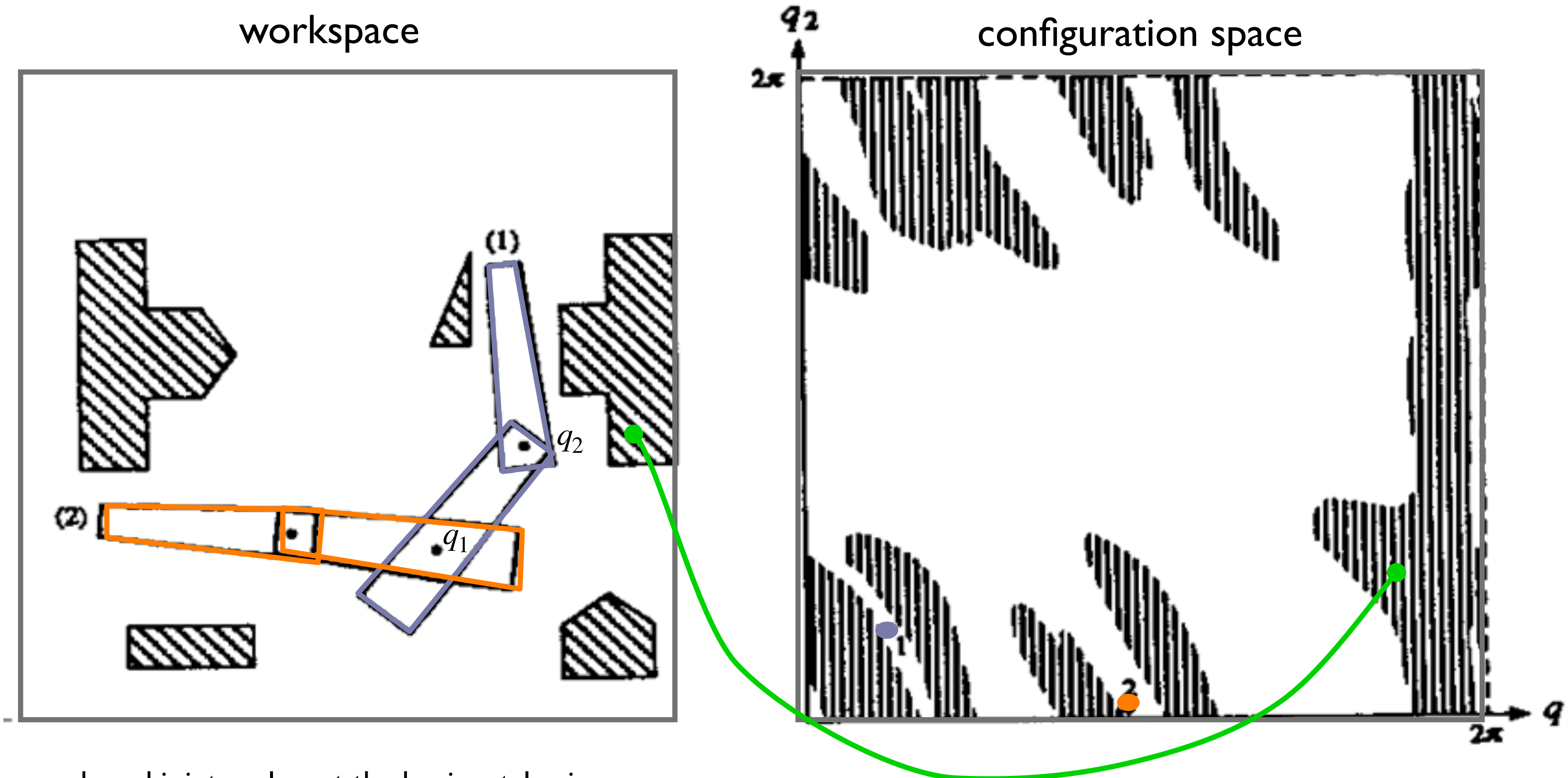
Compute the C-space obstacles for a polygonal robot translating and rotating in 2-D workspace. Compute the Minkowski difference of the obstacle and the robot at each orientation, and "stack" them together.



Computing the C-space obstacles for a polygonal robot translating and rotating in 2-D workspace. Compute the Minkowski difference of the obstacle and the robot at each orientation, and "stack" them together.



Compute the C-space obstacles for an articulate robot.



q_1: based joint angle wrt the horizontal axis

q_2: joint angle of link 2 wrt to the link 1

C-space metrics. A metric or distance function d in a configuration space C satisfies

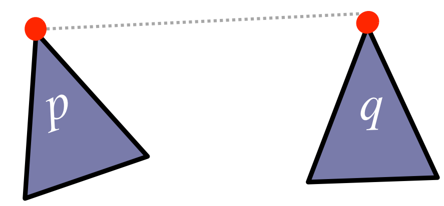
- $d(p,q) = 0$ if and only if $p = q$,
- $d(p,q) = d(q,p)$,
- $d(p,q) \leq d(p,r) + d(r,q)$,

for all $p,q \in C$. A familiar example is the Euclidean metric in the plane.

Example. Consider a robot A and a point $a \in A$. Let $a(q)$ be position of a in the workspace when A is at configuration q. A distance function d in C can be defined as

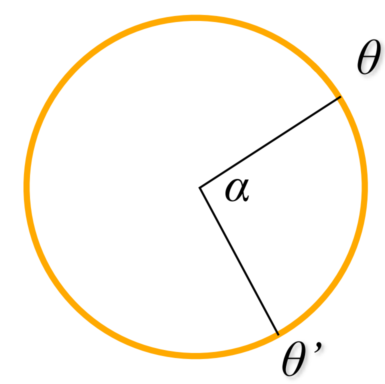
$$d(p,q) = \max_{a \in A} \|a(p) - a(q)\|$$

where $\|\cdot\|$ denotes the Euclidean metric in the workspace.



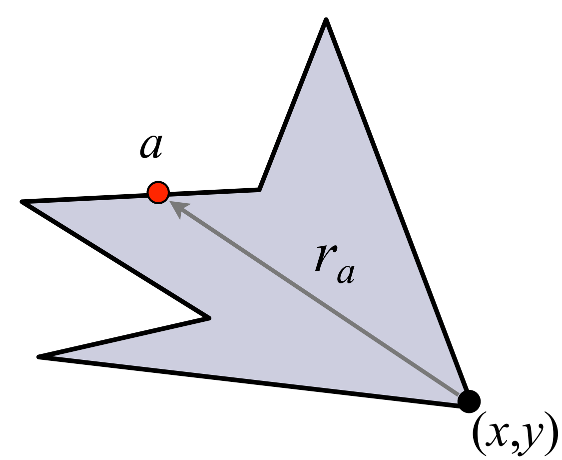
Computing this distance exactly is not easy, but we can approximate it. Let

- $q = (x, y, \theta)$, $q' = (x', y', \theta')$ with $\theta, \theta' \in [0, 2\pi)$
- $\alpha = \min \{ |\theta - \theta'| , 2\pi - |\theta - \theta'| \}$

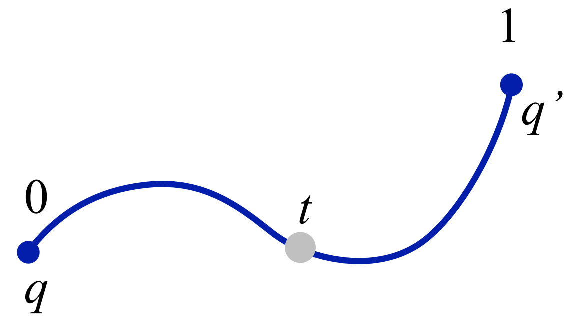


We then have

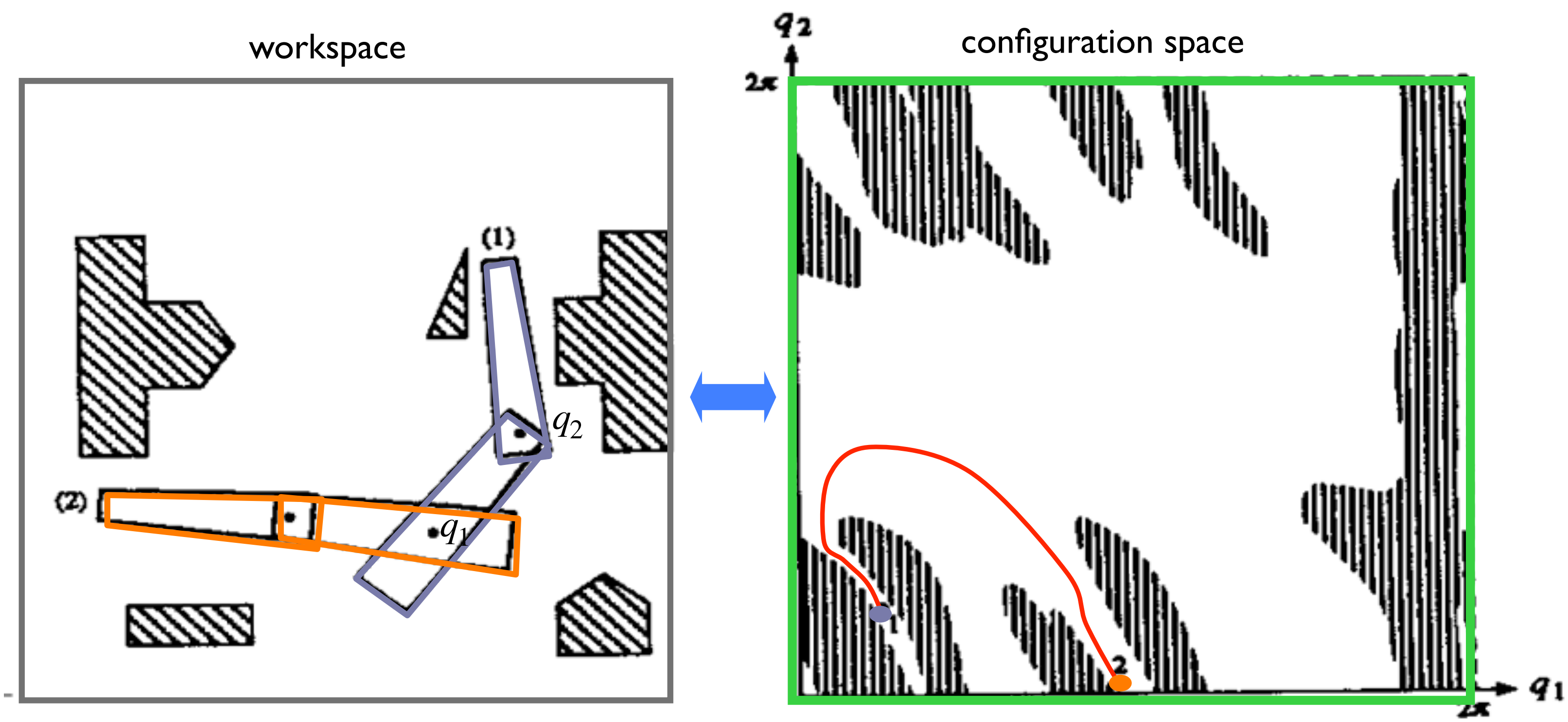
$$\begin{aligned} d(q,q') &= \max_{a \in A} \|a(q) - a(q')\| \\ &\leq \max_{a \in A} \sqrt{(x-x')^2 + (y-y')^2 + (\alpha r_a)^2} \\ &= \sqrt{(x-x')^2 + (y-y')^2 + (\alpha \max_{a \in A} r_a)^2} \\ &= \sqrt{(x-x')^2 + (y-y')^2 + (\alpha r_{\max})^2} \end{aligned}$$



C-space paths. A path in C is a continuous function $\tau(t)$ connecting two configurations q and q' such that $\tau(0) = q$ and $\tau(1)=q'$.



Example. The path in the C-space corresponds to the motion in the workspace between the two marked configurations. Can you imagine the motion in between?



Path constraints. We can place various constraints on C-space paths to induce desirable motions.

- Bounded curvature
  Example. A car has a minimum turning radius.
- Holonomic constraints
  Example. In the 2-parameter specification of 2-D rotations, we have $u^2+v^2 = 1$.
- Nonholonomic constraints
  Example. A car cannot move sidewise.
- ...

Summary.

- The workspace is the physical space that contains the robot and the obstacles. The configuration space represents the robot posture as a point along with transformed obstacles. The configuration space provides a general and conceptually elegant representation framework.
  - The posture of a robot corresponds to a point in the configuration space.
  - The motion of a robot corresponds to a path in the configuration space.
- Both spaces contain exactly the same information.
  - Given the geometry of a robot and obstacles, we can construct the configuration space (in principle).
  - Given the configuration of a robot, we can determine the position of every point on the robot.

Required readings.

- [CSpace-Cho01] Sect 3.1-3.3, 3.5-3.7

Supplementary readings.

- [CSpace-Cho01] Appendix E

Key concepts.

- Configuration and its parametrization
- Configuration space
    - Dimension
    - Metric (distance)
    - Path
- Configuration space obstacles
    - Minkowski sum/difference
- The relationship between workspace and configuration space