
CS686: Configuration Space I

Sung-Eui Yoon
(윤성익)

Course URL:
<http://sglab.kaist.ac.kr/~sungeui/MPA>

KAIST



Announcements

- **Make a project team of 2 or 3 persons for your final project**
 - Each student has a clear role
 - Declare the team at the noah board by Oct-5; you don't need to define the topic by then
- **Each student**
 - Present two papers related to the project
 - 20 min for each talk
- **Each team**
 - Give a mid-term review presentation for the project
 - Give the final project presentation

Tentative schedule

10월

27일 Student Presentation 1
29일 Student Presentation 2

11월

3 Student Presentation 3
5 Student Presentation 4

12 Student Presentation 5

17 Mid. Project Presentation 1
19 Mid. Project Presentation 2

24 Student Presentation 1

12월

1 Student Presentation 2
3 Student Presentation 3

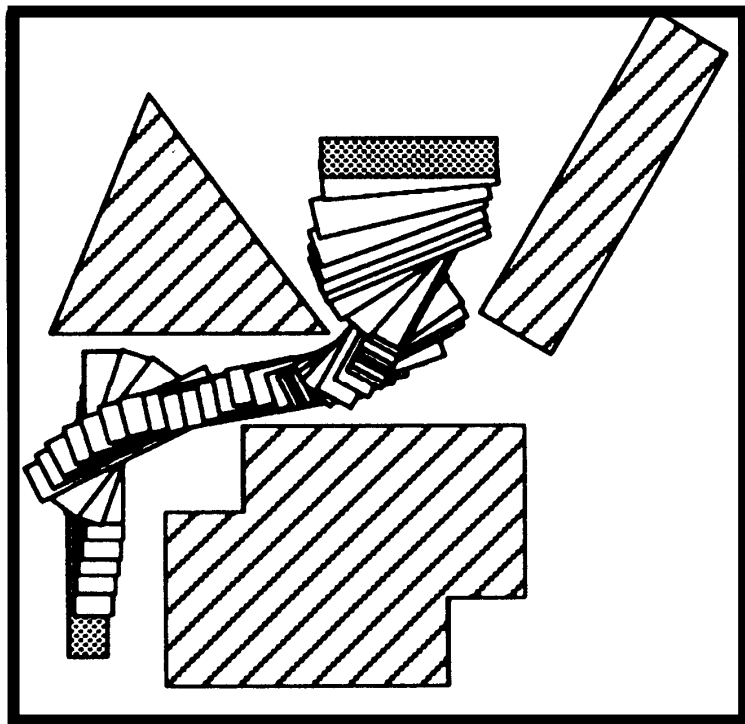
8 Student Presentation 4
10 Student Presentation 5

3 15 Final Presentation 1
17 Final Presentation 2

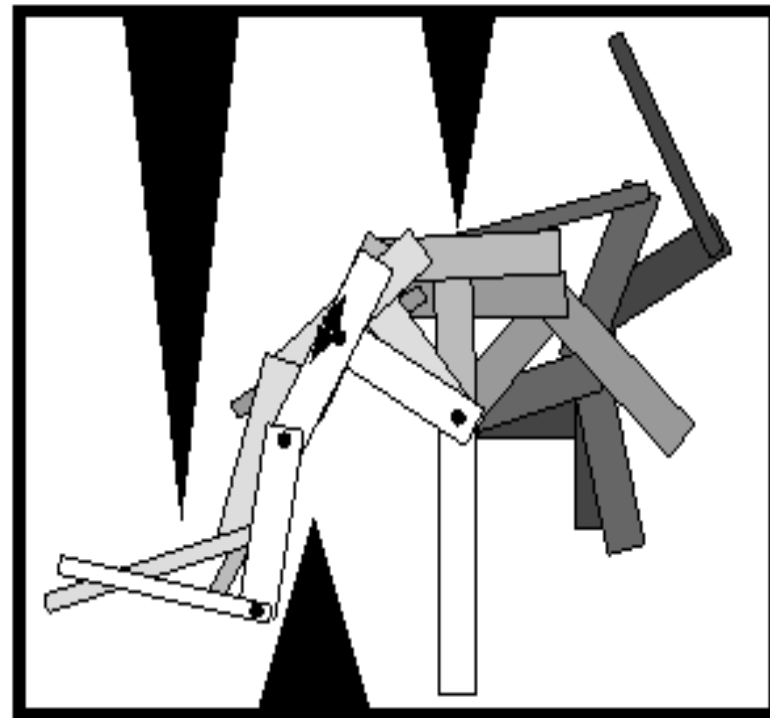
Class Objectives

- **Configuration space**
 - **Definitions and examples**
 - **Obstacles**
 - **Paths**
 - **Metrics**

What is a Path?



A box robot

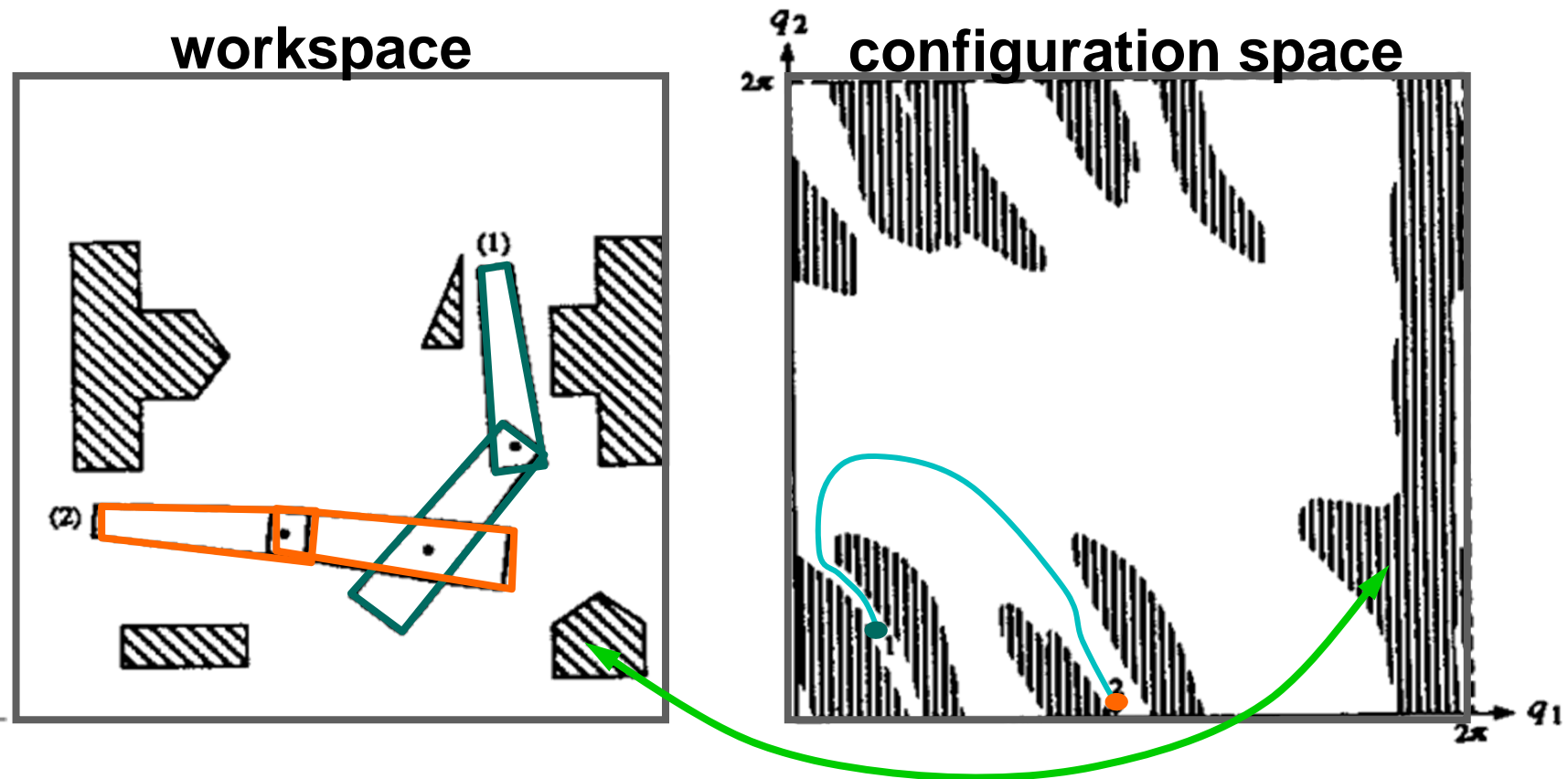


Linked robot

Rough Idea of C-Space

- Convert rigid robots, articulated robots, *etc.* into points
- Apply algorithms in that space, in addition to the work space

Mapping from the Workspace to the Configuration Space



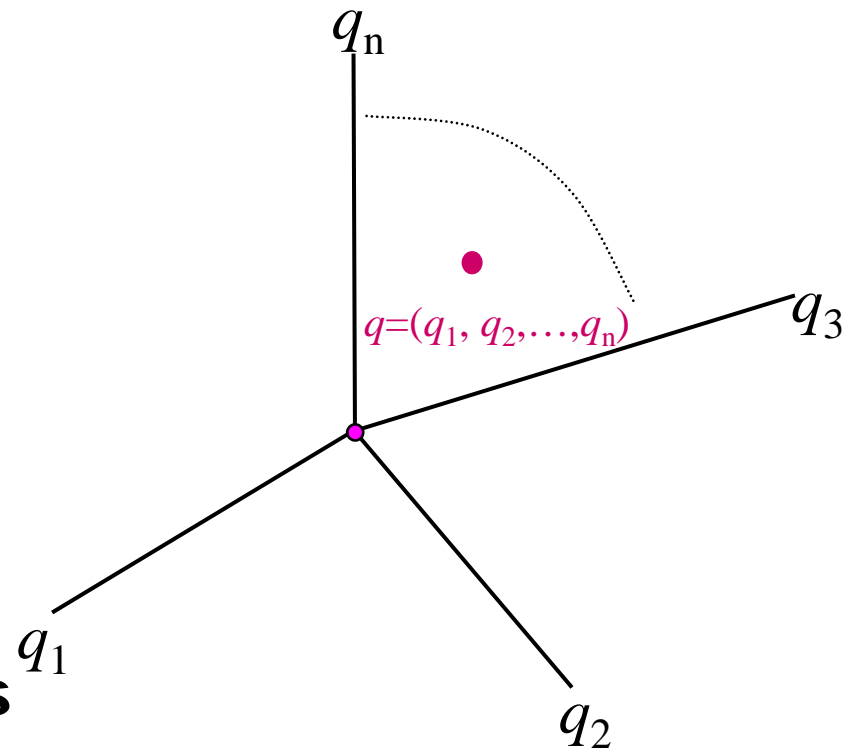
Configuration Space

- **Definitions and examples**
- Obstacles
- Paths
- Metrics

Configuration Space (C-space)

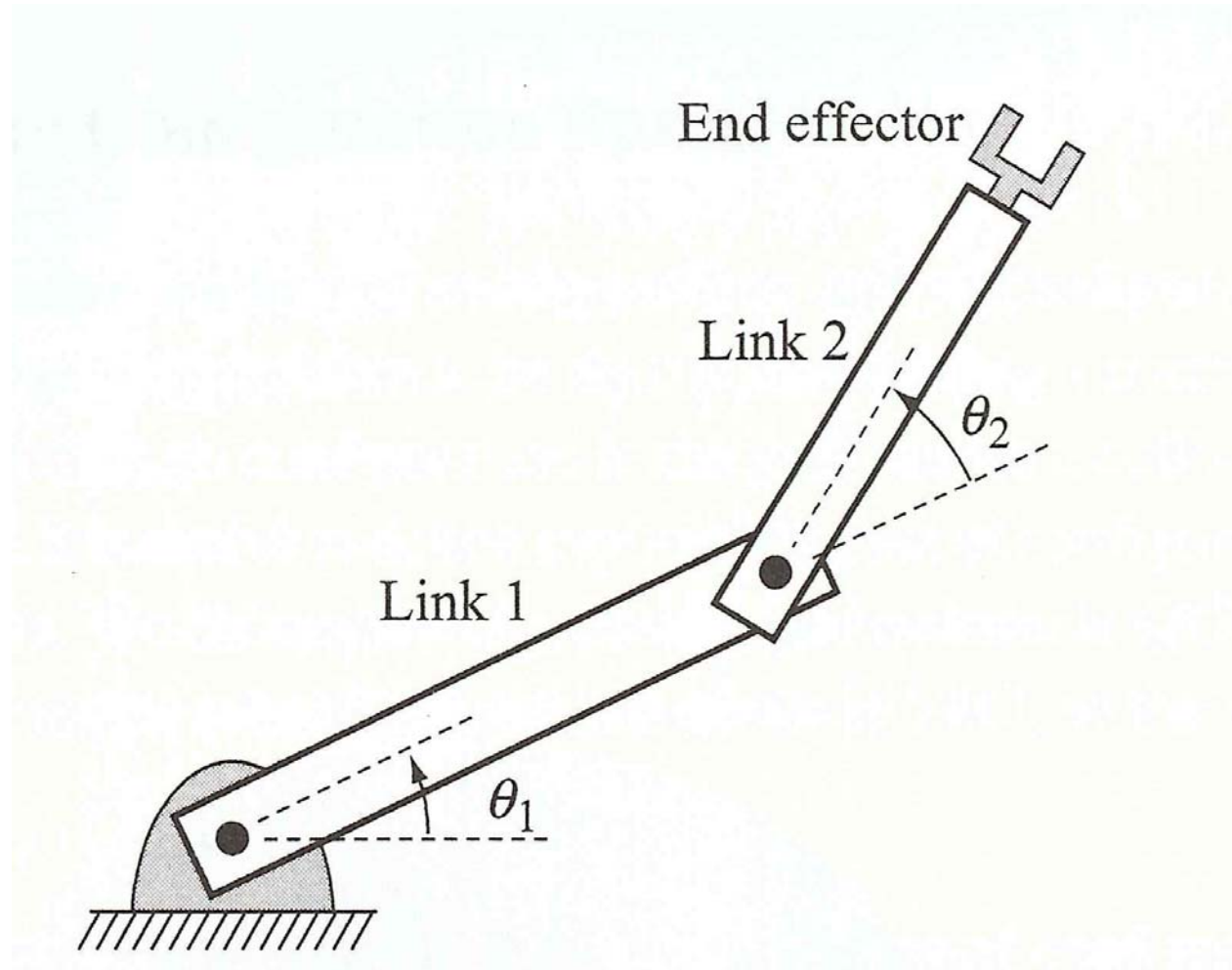
- The **configuration** of an object is a complete specification of the position of **every** point on the object
 - Usually a configuration is expressed as a vector of position & orientation parameters: $q = (q_1, q_2, \dots, q_n)$

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

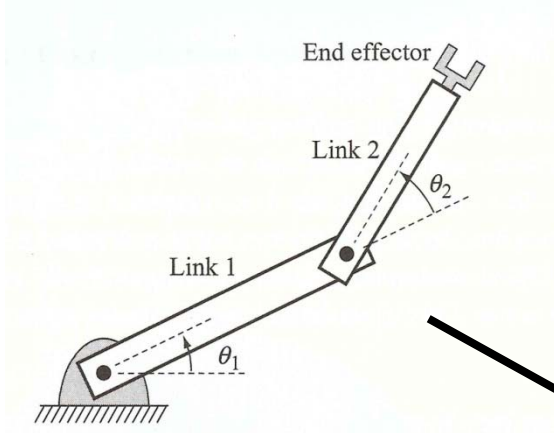


**C-space formalism:
Lozano-Perez '79**

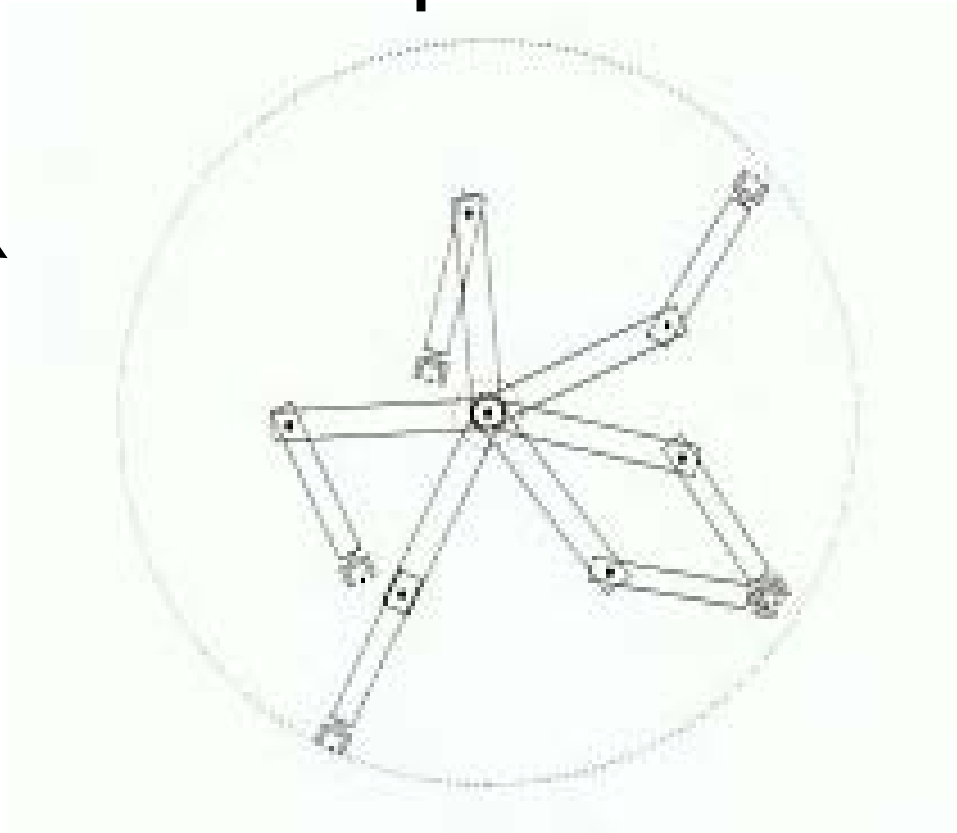
Examples of Configuration Spaces



Examples of Configuration Spaces

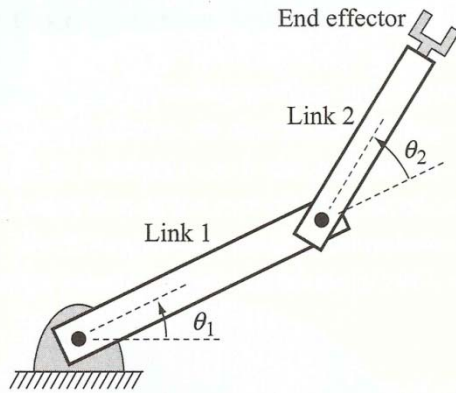


Consider the end-effector in the workspace?

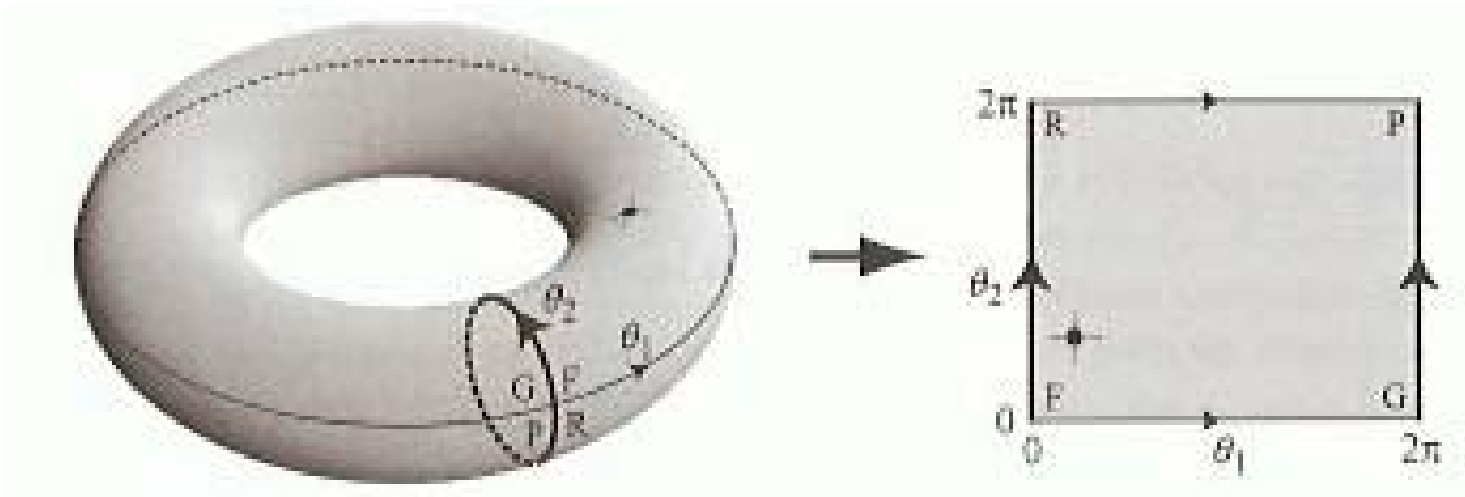


This is not a valid C-space!

Examples of Configuration Spaces

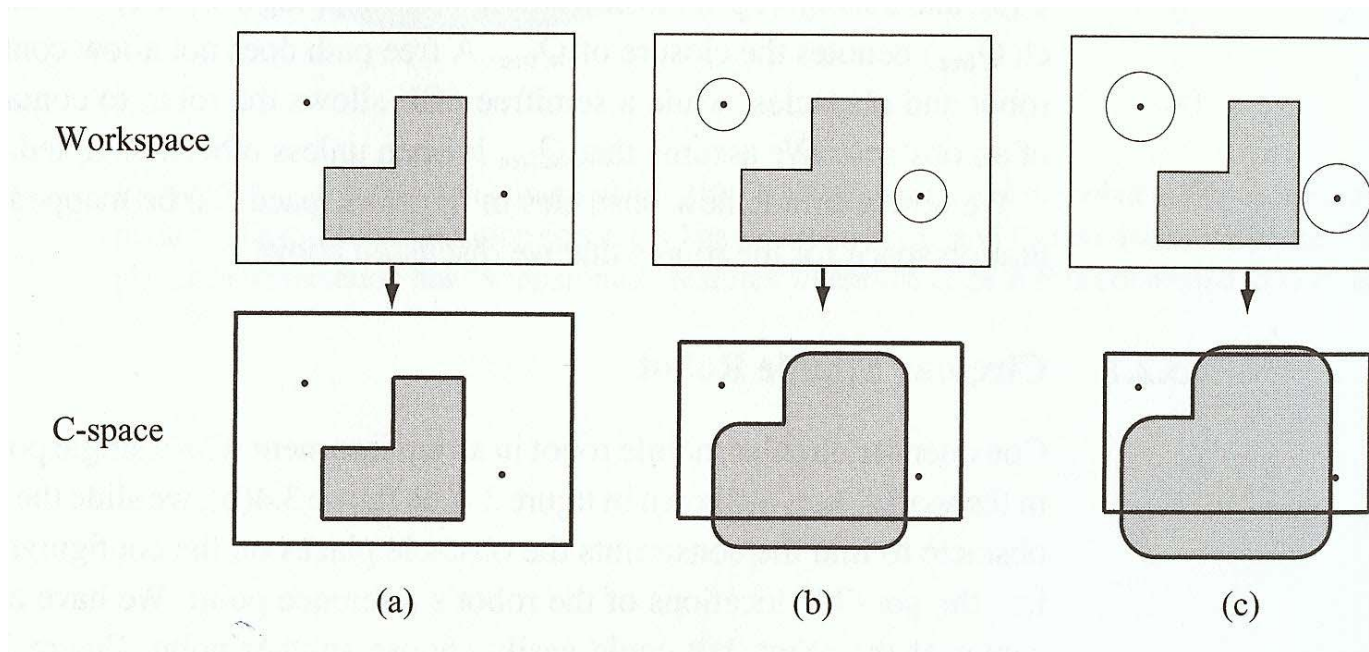
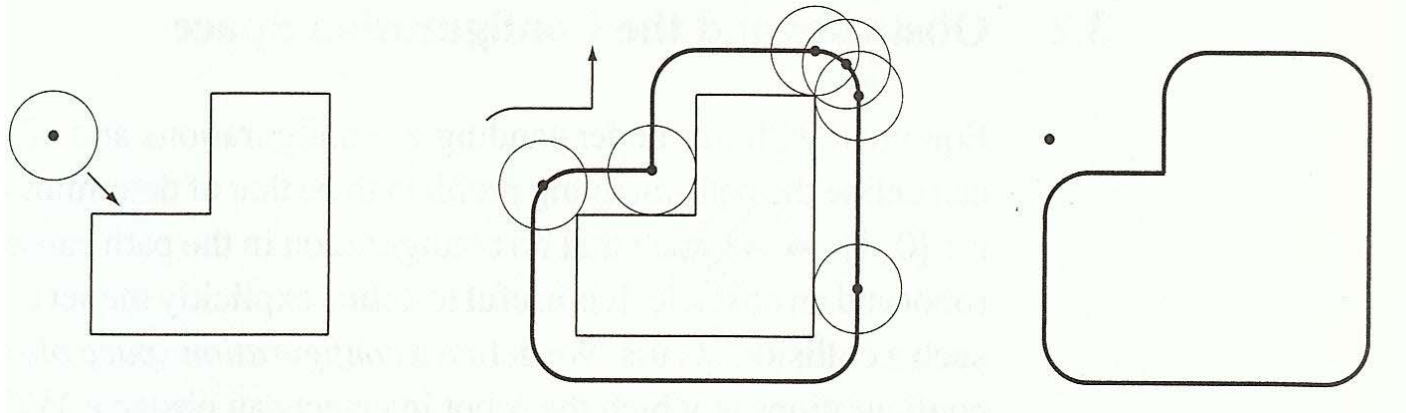


The topology of C is usually **not** that of a Cartesian space R^n .



$$S^1 \times S^1 = T^2$$

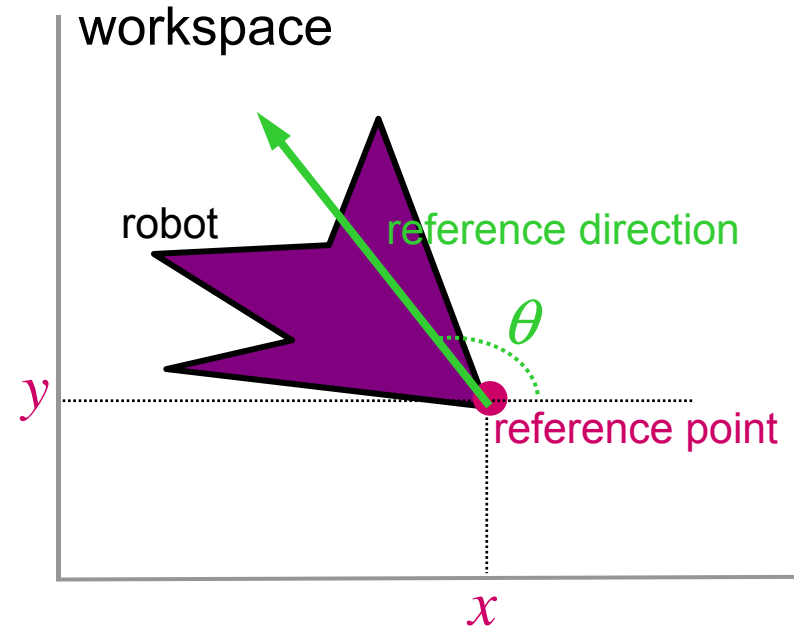
Examples of Circular Robot



Dimension of Configuration Space

- The **dimension of the configuration space** is the **minimum** number of parameters needed to specify the configuration of the object completely
- It is also called the number of **degrees of freedom** (dofs) of a moving object

Example: Rigid Robot in 2-D Workspace



- **3-parameter specification: $q = (x, y, \theta)$ with $\theta \in [0, 2\pi)$.**
 - 3-D configuration space

Example: Rigid Robot in 2-D workspace

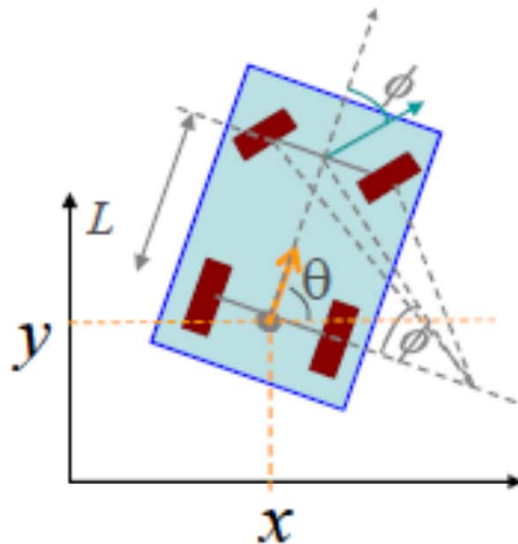
- 4-parameter specification: $q = (x, y, u, v)$ with $u^2 + v^2 = 1$. Note $u = \cos \theta$ and $v = \sin \theta$
- dim of configuration space = **3**
 - Does the dimension of the configuration space (number of dofs) depend on the parametrization?

Holonomic and Non-Holonomic Constraints

- **Holonomic constraints**
 - $g(q, t) = 0$
- **Non-holonomic constraints**
 - $g(q, q', t) = 0$
 - This is related to the kinematics of robots
 - To accommodate this, the C-space is extended to include the position and its velocity
- **Dynamic constraints**
 - Dynamic equations are represented as $G(q, q', q'') = 0$
 - These constraints are reduced to non-holonomic ones when we use the extended C-space such as the state space

Example of Non-Holonomic Constraints

The path of the car is a curve tangent to its main rotation axis



$dx \sin \theta - dy \cos \theta = 0$

Computation of Dimension of C-Space

- Suppose that we have a rigid body that can translate and rotate in 2D workspace
 - Start with three points: A, B, C (6D space)
- We have the following (holonomic) constraints
 - Given A, we know the dist to B: $d(A,B) = |A-B|$
 - Given A and B, we have similar equations:
 $d(A,C) = |A-C|$, $d(B,C) = |B-C|$
- Each holonomic constraint reduces one dim.
 - Not for non-holonomic constraint

Example: Rigid Robot in 3-D Workspace

- We can represent the positions and orientations of such robots with matrices (i.e., $SO(3)$ and $SE(3)$)

SO (n) and SE (n)

- **Special orthogonal group, SO(n)**, of n x n matrices R ,

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

that satisfy:

$$r_{1i}^2 + r_{2i}^2 + r_{3i}^2 = 1 \text{ for all } i,$$

$$r_{1i}r_{1j} + r_{2i}r_{2j} + r_{3i}r_{3j} = 0 \text{ for all } i \neq j,$$

$$\det(R) = +1$$

Refer to the 3D Transformation at the undergraduate computer graphics.

- Given the orientation matrix R of SO (n) and the position vector p , **special Euclidean group, SE (n)**, is defined as:

$$\begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$

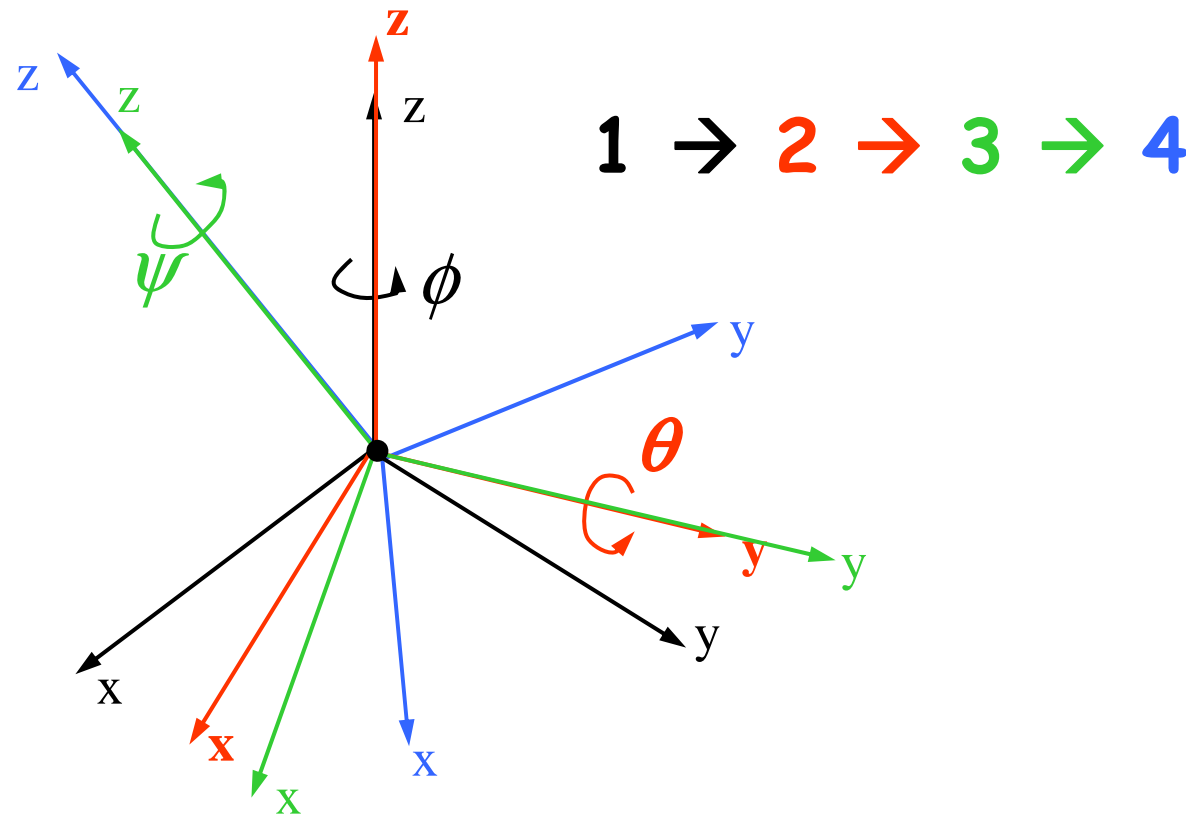
Example: Rigid Robot in 3-D Workspace

- $q = (\text{position, orientation}) = (x, y, z, ???)$
- Parametrization of orientations by matrix:
 $q = (r_{11}, r_{12}, \dots, r_{33}, r_{33})$ where $r_{11}, r_{12}, \dots, r_{33}$ are the elements of rotation matrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \in SO(3)$$

Example: Rigid Robot in 3-D Workspace

- Parametrization of orientations by Euler angles:
 (ϕ, θ, ψ)

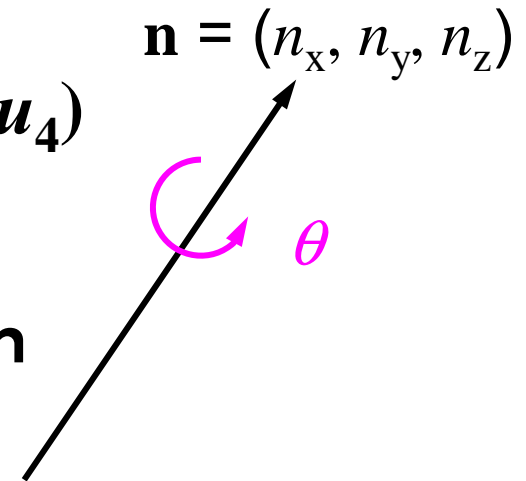


Example: Rigid Robot in 3-D Workspace

- Parametrization of orientations by **unit quaternion**: $u = (u_1, u_2, u_3, u_4)$ with $u_1^2 + u_2^2 + u_3^2 + u_4^2 = 1$.

- Note $(u_1, u_2, u_3, u_4) = (\cos\theta/2, n_x \sin\theta/2, n_y \sin\theta/2, n_z \sin\theta/2)$ with $n_x^2 + n_y^2 + n_z^2 = 1$

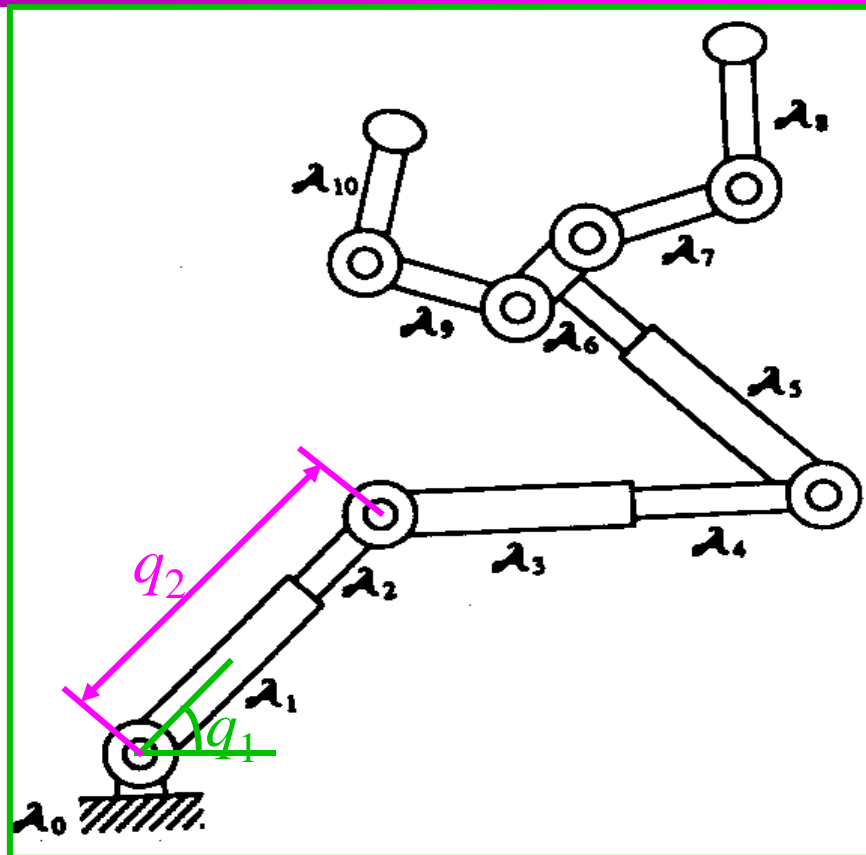
- Compare with representation of orientation in 2-D:
 $(u_1, u_2) = (\cos\theta, \sin\theta)$



Example: Rigid Robot in 3-D Workspace

- Advantage of unit quaternion representation
 - Compact
 - No singularity
 - Naturally reflect the topology of the space of orientations
- Number of dofs = 6
- Topology: $\mathbb{R}^3 \times \text{SO}(3)$

Example: Articulated Robot



- $q = (q_1, q_2, \dots, q_{2n})$
- Number of dofs = $2n$
- What is the topology?

An articulated object is a set of rigid bodies connected at the joints.

Class Objectives were:

- **Configuration space**
 - **Definitions and examples**
 - **Obstacles**
 - **Paths**
 - **Metrics**

Next Time....

- **Configuration space**
 - Definitions and examples
 - **Obstacles**
 - **Paths**
 - **Metrics**

Homework

- **Browse 2 ICRA/IROS/RSS/WAFR/TRO/IJRR papers**
 - Prepare two summaries and submit at the beginning of every Tue. class, or
 - Submit it online before the Tue. Class
- **Example of a summary (just a paragraph)**

Title: XXX XXXX XXXX
Conf./Journal Name: ICRA, 2015
Summary: this paper is about accelerating the performance of collision detection. To achieve its goal, they design a new technique for reordering nodes, since by doing so, they can improve the coherence and thus improve the overall performance.

Homework for Every Class

- **Go over the next lecture slides**
- **Come up with one question on what we have discussed today and submit at the end of the class**
 - 1 for typical questions
 - 2 for questions with thoughts or that surprised me
- **Write a question more than 4 times on Sep./Oct.**