# CS686: Configuration Space I


Sung-Eui Yoon
(윤성의)


Course URL:
http://sgvr.kaist.ac.kr/~sungeui/MPA

KAIST

# Announcements

- Make a project team of 1 to 3 persons for your project
  - Each student needs a clear role
  - Declare team members at KLMS by Sep-28; you don't need to define the topic by then
- Each student
  - Present two papers related to the project; 15 min for each talk
  - Declare your papers at KLMS by Oct-12
- Each team
  - Give a mid-term presentation for the project
  - Give the final project presentation

# Tentative schedule

- Oct. 27, 29: Student Presentation 1 & 2
  - 2 to 3 talks per each class; 10 students in the class
- Nov. 3:   SP3   (Nov. 5: no class due to interview process)
- Nov. 10: SP 4
- Nov 12 (Th) , Nov 17 (Tu): Mid-term project presentation
- Nov. 19 (Th) : SP 5
- Nov. 24, 26: SP 6 & 7
- Dec 1, 3: SP 8 & 9
- Dec. 8, 10: Final-term project presentation
- Dec. 15, 17: reservation for now (exam period, no class for now)

# Class Objectives

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

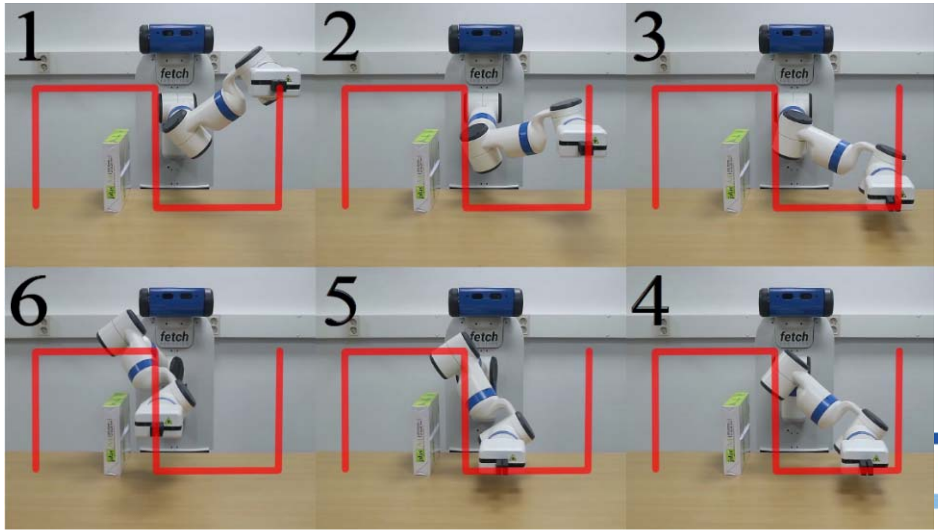
- **Last time:**
  - **Classic motion planning approaches including roadmap, cell decomposition and potential field**

# Questions

- **Are all path planning problems solved by graph navigation problems?**
  - **Trajectory optimization is also useful**

**TORM: Fast and Accurate Trajectory Optimization of Redundant Manipulator given an End-Effector Path, by Mincheul Kang, Heechan Shin, Donghyuk Kim, and Sung-Eui Yoon**

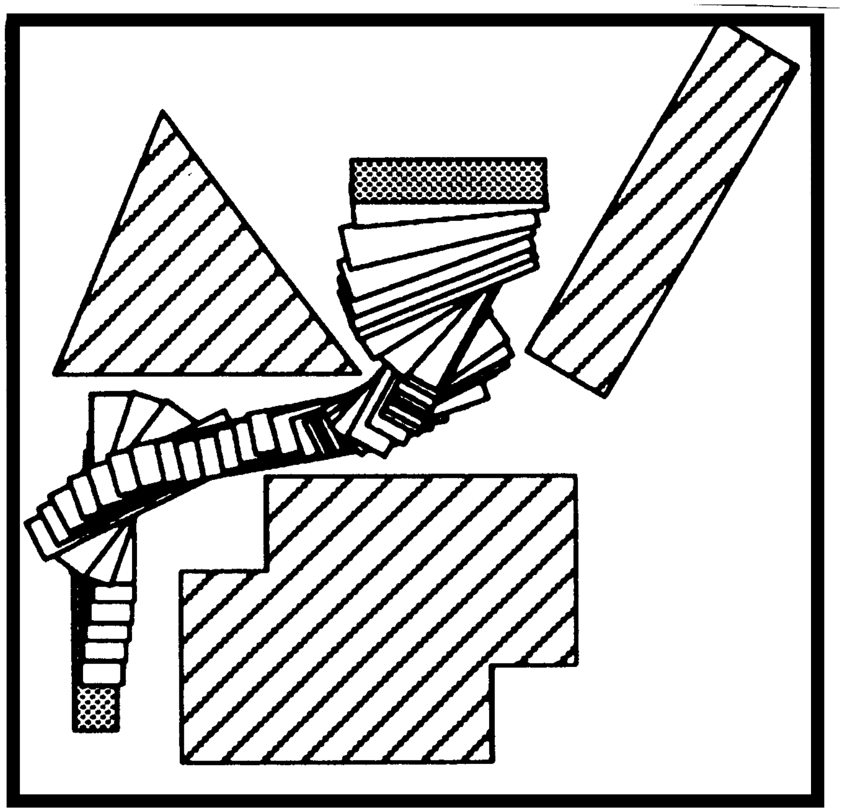
**http://sglab.kaist.ac.kr/TORM/**



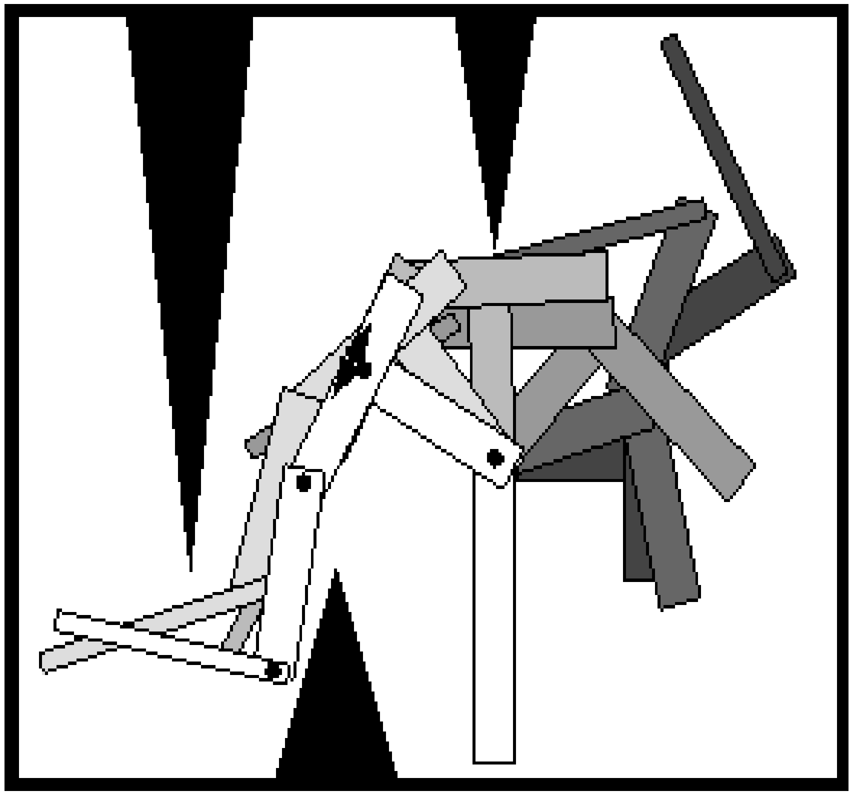
# Paper Summary

- **Submodular Trajectory Optimization for Aerial 3D Scanning, Conference name: ICCV, 2017**
- **Applying Asynchronous Deep Classification Networks and Gaming Reinforcement Learning-Based Motion Planners to Mobile Robots, ICRA 2018**
- **RL-RRT: Kinodynamic Motion Planning via Learning Reachability Estimators from RL Policies**
- **Learning Patch Reconstructability for Accelerating Multi-View Stereo, Conference name: CVPR, 2018**
  - **Relation to robotics should be clearly mentioned in the summary**

# What is a Path?



A box robot



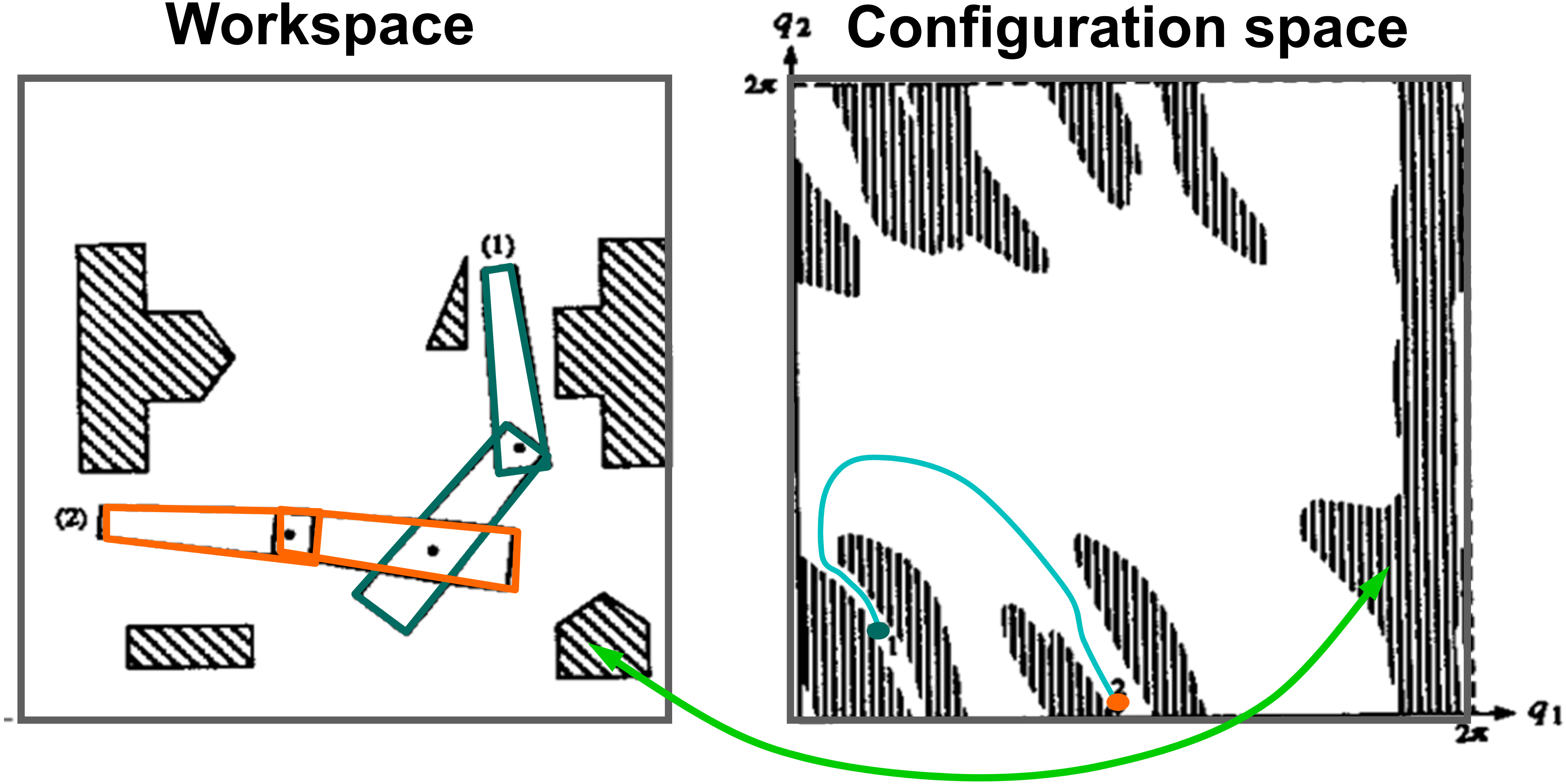
Linked robot

# Rough Idea of C-Space

- **Represent degrees-of-freedom (DoFs) of rigid robots, articulated robots, *etc.* into points**
- **Apply algorithms in that space, in addition to the workspace**

# Mapping from the Workspace to the Configuration Space

**Workspace** **Configuration space**

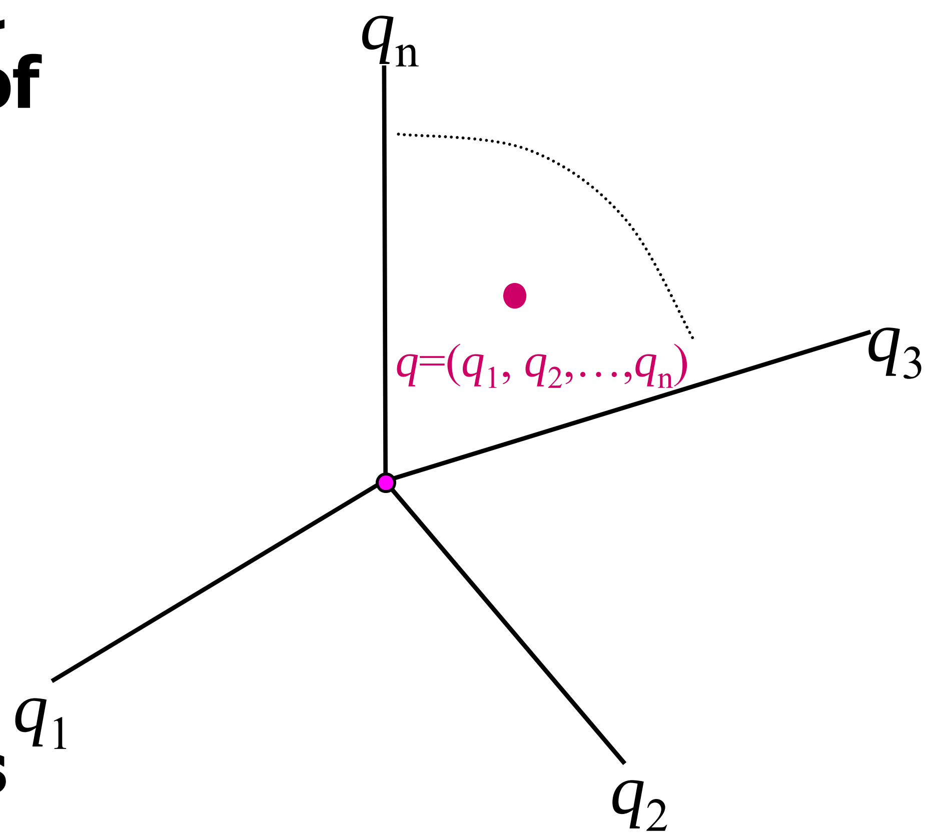


# Configuration Space

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

# Configuration Space (C-space)

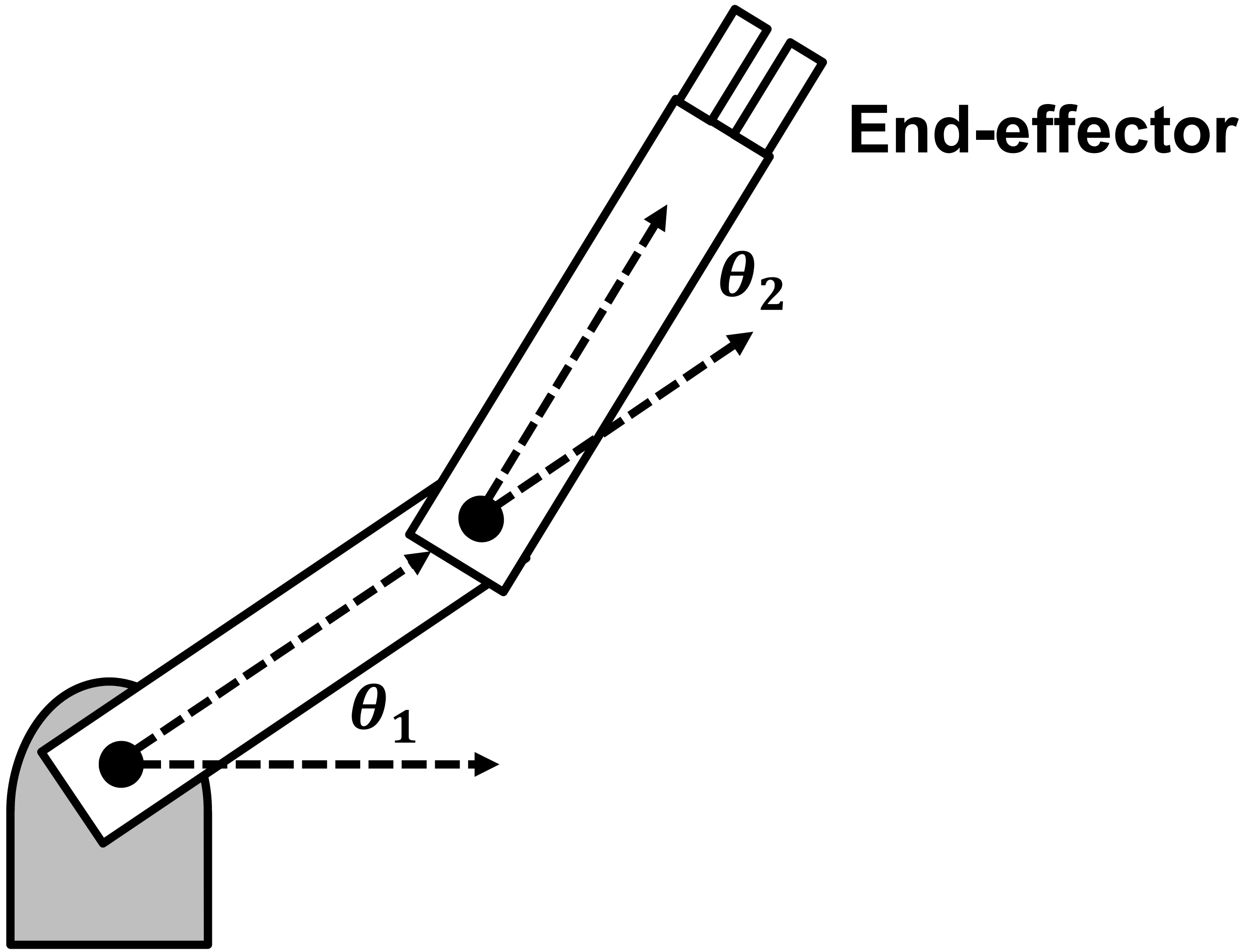
- **The configuration of a robot is a complete specification of the position of every point on the robot**
  - **Usually a configuration is expressed as a vector of position & orientation parameters:** $q = (q_1, q_2, \ldots, q_n)$




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

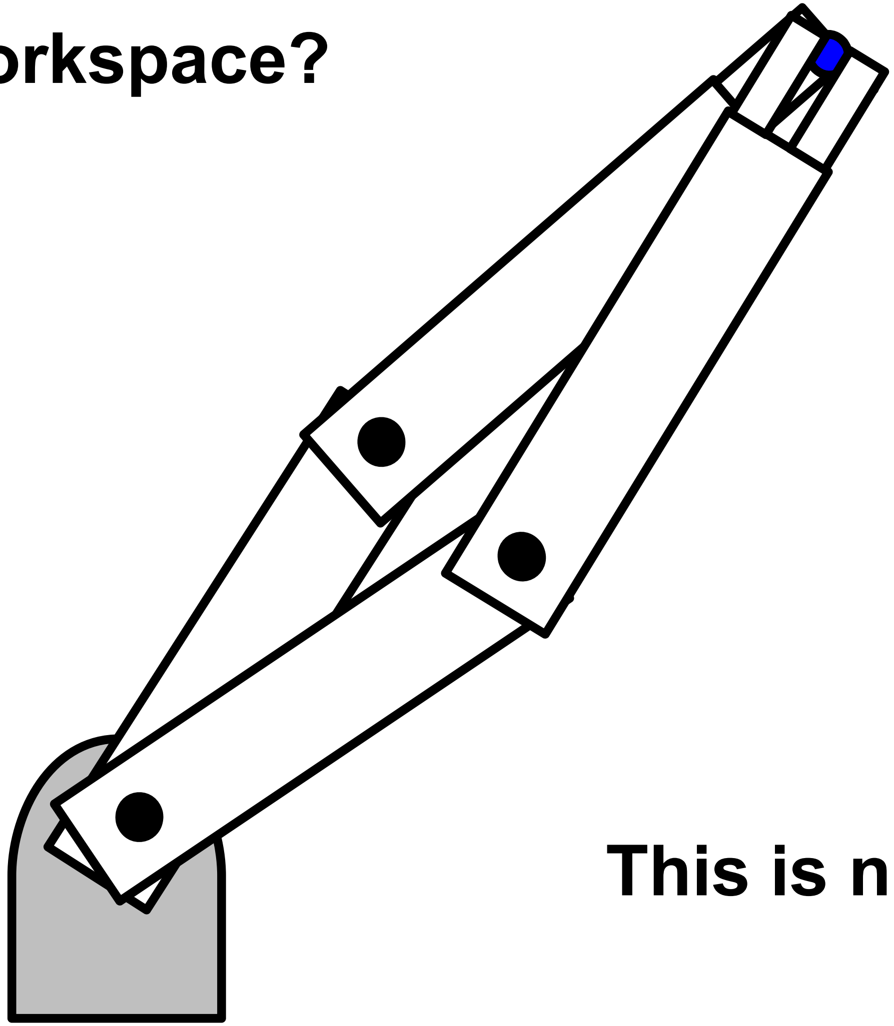
**C-space formalism: Lozano-Perez '79**

# Examples of Configuration Spaces



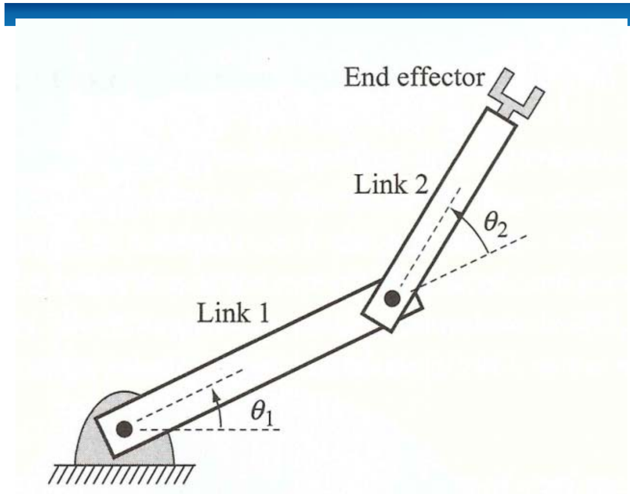
# Examples of Configuration Spaces

**Consider the end-effector in the workspace?**

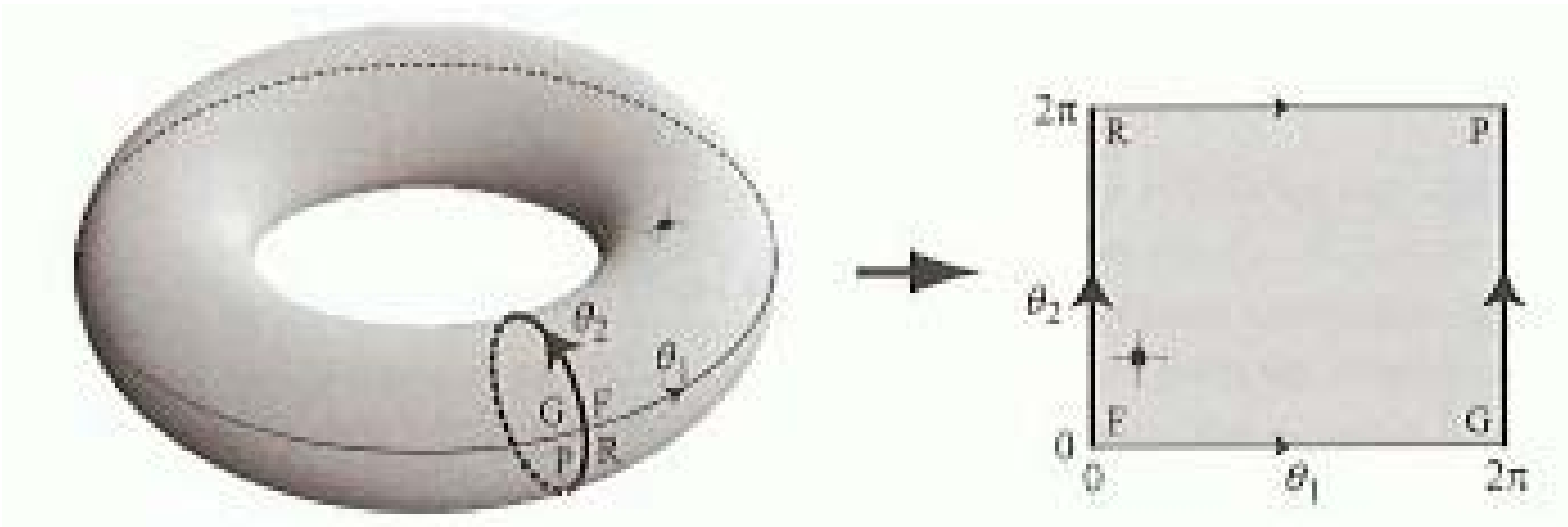


**This is not a valid C-space!**

# Examples of Configuration Spaces

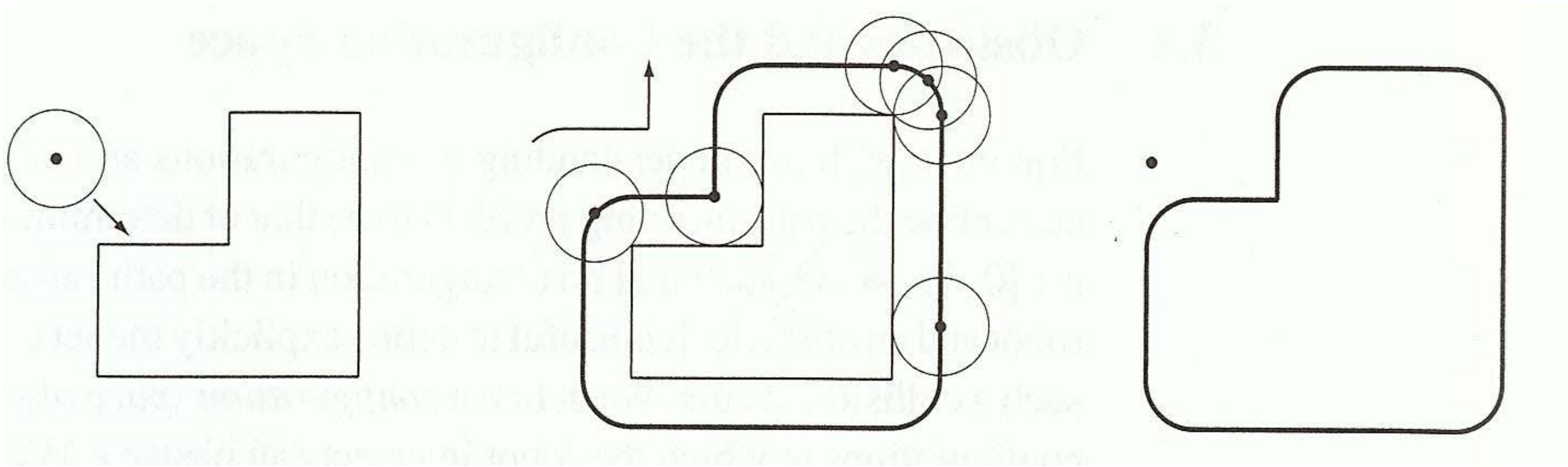


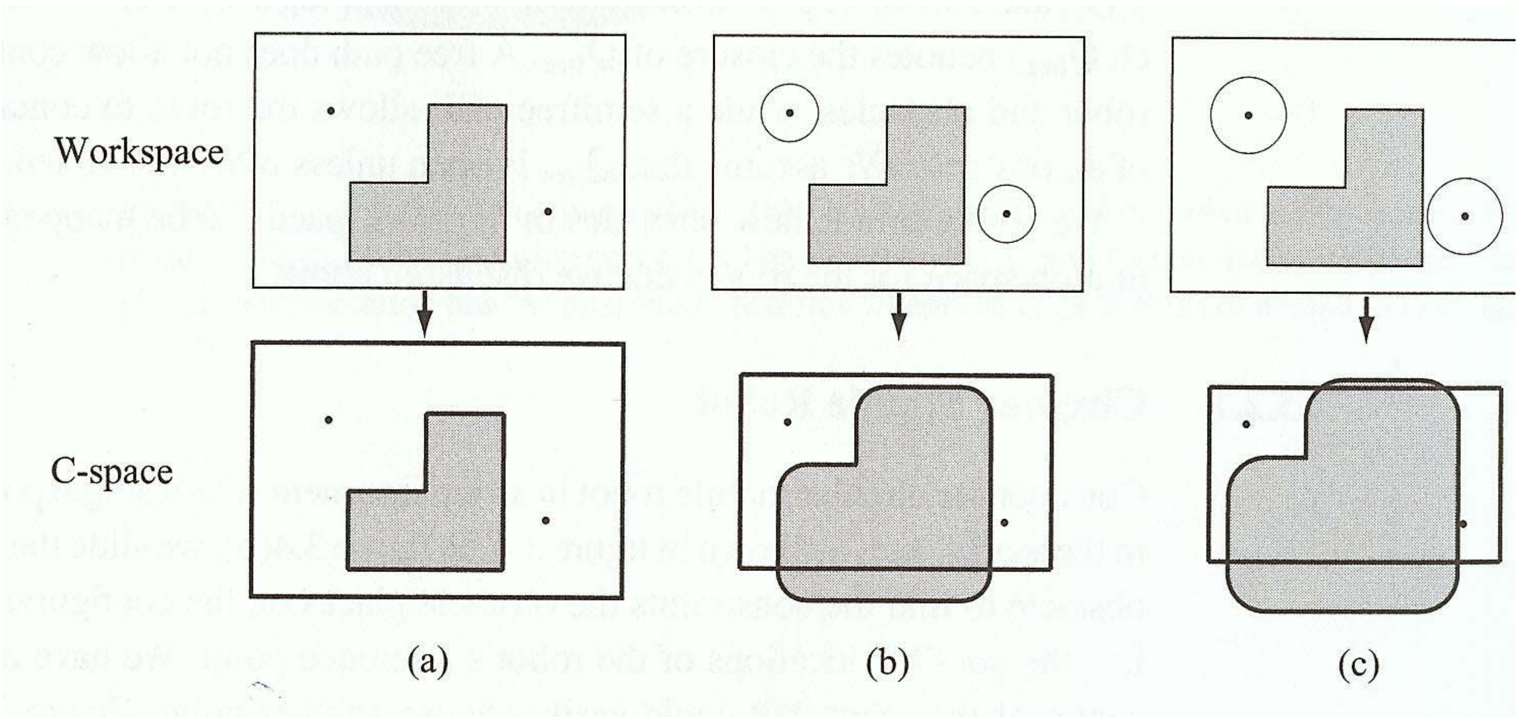

The topology of $C$ is usually **not** that of a Cartesian space $R^n$.




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

# Examples of Circular Robot



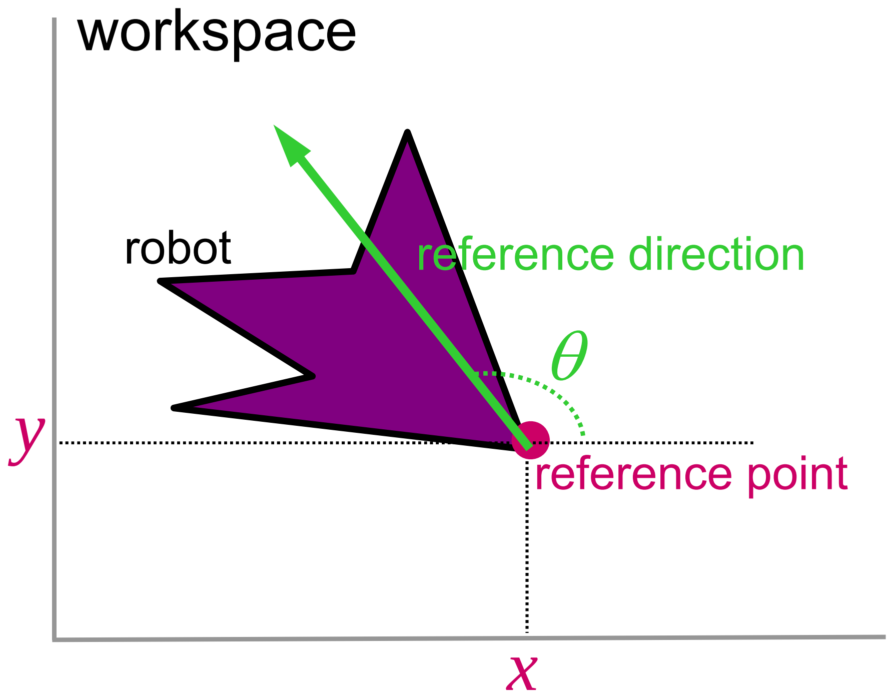


# Dimension of Configuration Space

- 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**

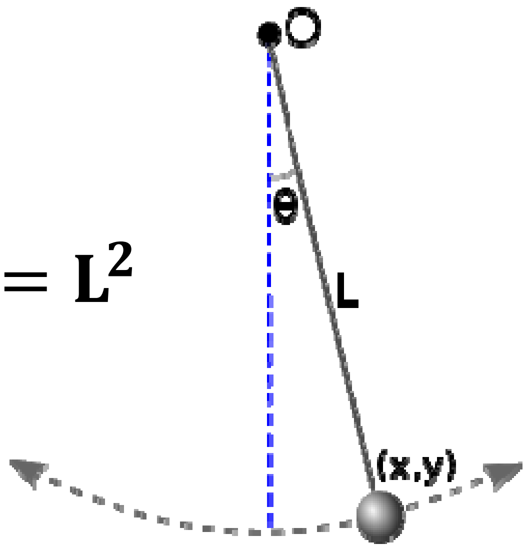
# Example: Rigid Robot in 2-D workspace

- **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**
  - **E.g., pendulum motion:** $x^2 + y^2 = L^2$

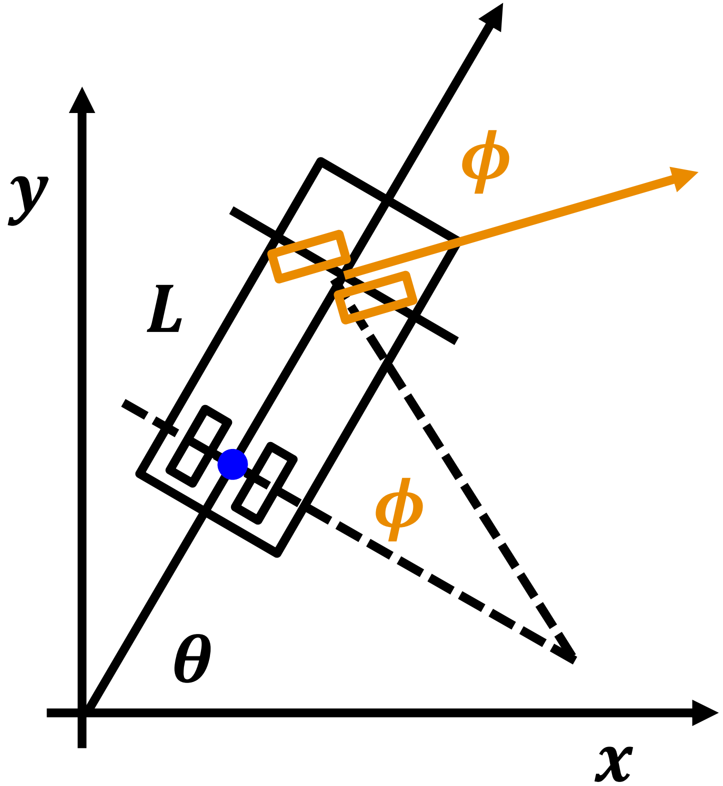



wiki

- **Non-holonomic constraints**
  - **g (q, q', t) = 0 (or q' = f(q, u), where u is an action parameter)**
  - **This is related to the kinematics of robots**
  - **To accommodate this, the C-space is extended to include the position and its velocity**

# Example of Non-Holonomic Constraints

**See Kinematic Car Model of my draft**




$$\tan(\boldsymbol{\theta}) = \frac{\boldsymbol{sin(\theta)}}{\boldsymbol{cos(\theta)}} = \frac{\boldsymbol{dy}}{\boldsymbol{dx}}$$

$$\boldsymbol{sin(\theta)dx - cos(\theta)dy = 0}$$

$$\frac{\boldsymbol{dx}}{\boldsymbol{dt}} = \boldsymbol{v \cdot cos(\theta)}, \quad \frac{\boldsymbol{dy}}{\boldsymbol{dt}} = \boldsymbol{v \cdot sin(\theta)},$$

$$\frac{\boldsymbol{d\theta}}{\boldsymbol{dt}} = \frac{\boldsymbol{v}}{\boldsymbol{L}} \boldsymbol{tan(\phi)}$$

**Note that $v, \phi$ are action parameters**

# Holonomic and Non-Holonomic Constraints

- **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:**

    **S= (X, X′), where X=(q, q′)**

# 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.
http://sgvr.kaist.ac.kr/~sungeui/render/raster/transformation.pdf

- 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

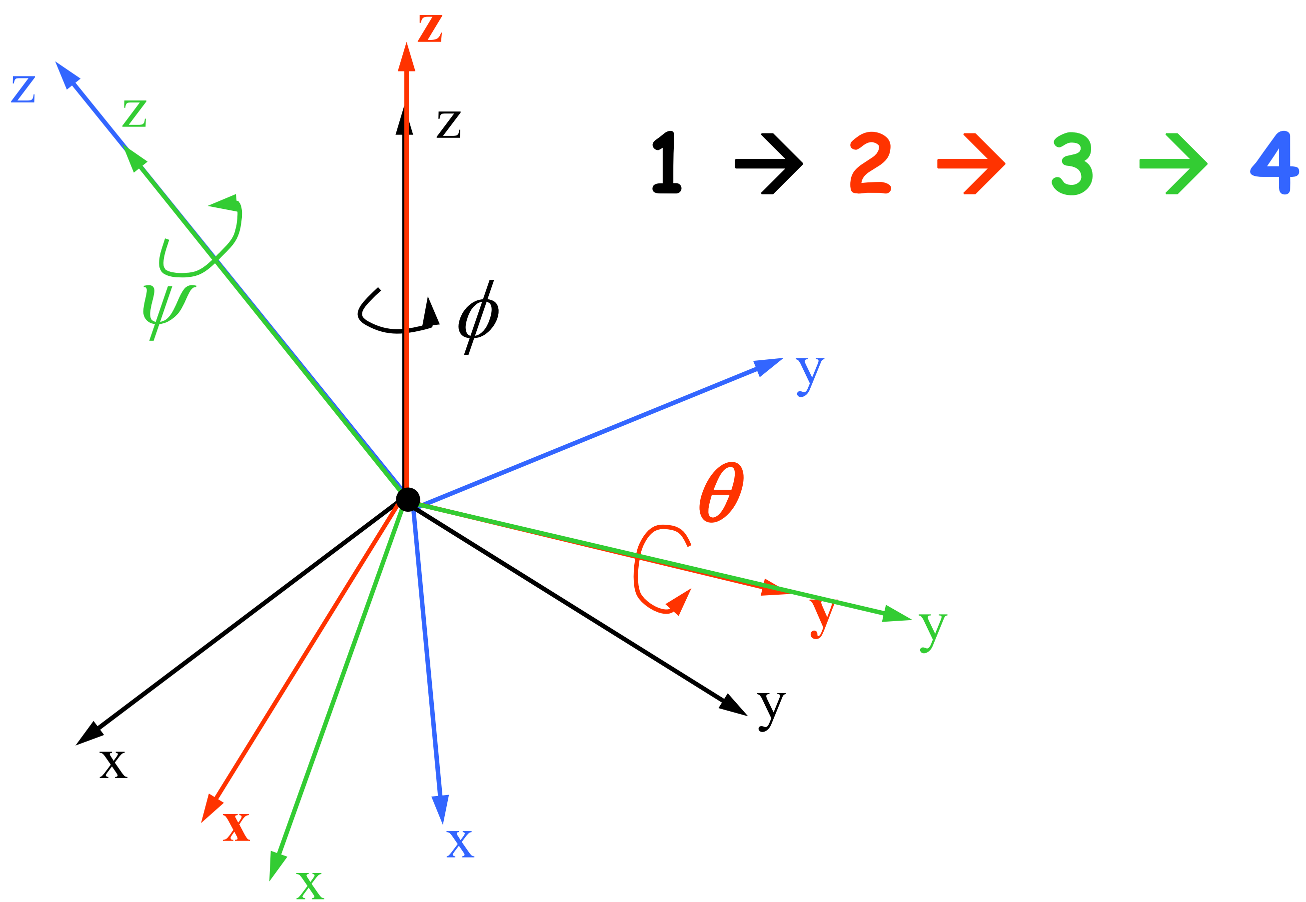
- $q$ = (**position, orientation**) = $(x, y, z, ???)$

- **Parametrization of orientations by matrix:** $q = (r_{11}, r_{12}, \ldots, r_{33}, r_{33})$ **where** $r_{11}, r_{12}, \ldots, r_{33}$ **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

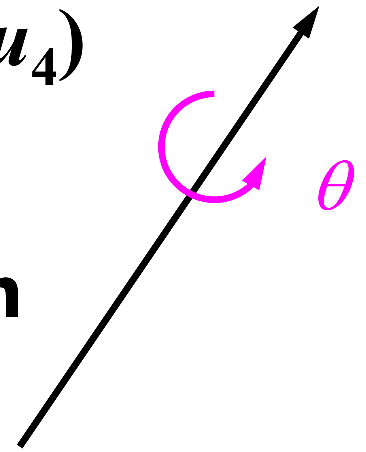
- **Parametrization of orientations by Euler angles:** $(\phi,\theta,\psi)$



# Example: Rigid Robot in 3-D Workspace

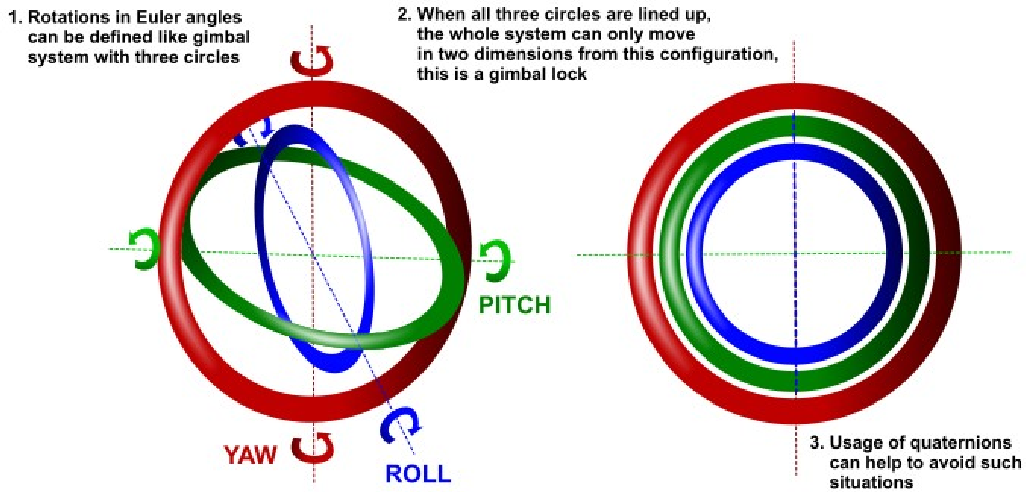
- **Parametrization of orientations by unit quaternion:** $u = (u_1, u_2, u_3, u_4)$ **with** $u_1^2 + u_2^2 + u_3^2 + u_4^2 = 1.$
  - **Note** $(u_1, u_2, u_3, u_4) = (\cos\theta/2, n_x\sin\theta/2, n_y\sin\theta/2, n_z\sin\theta/2)$ **with** $n_x^2 + n_y^2 + n_z^2 = 1$
  - **Compare with representation of orientation in 2-D:** $(u_1,u_2) = (\cos\theta, \sin\theta)$

$\mathbf{n} = (n_x, n_y, n_z)$



# Example: Rigid Robot in 3-D Workspace

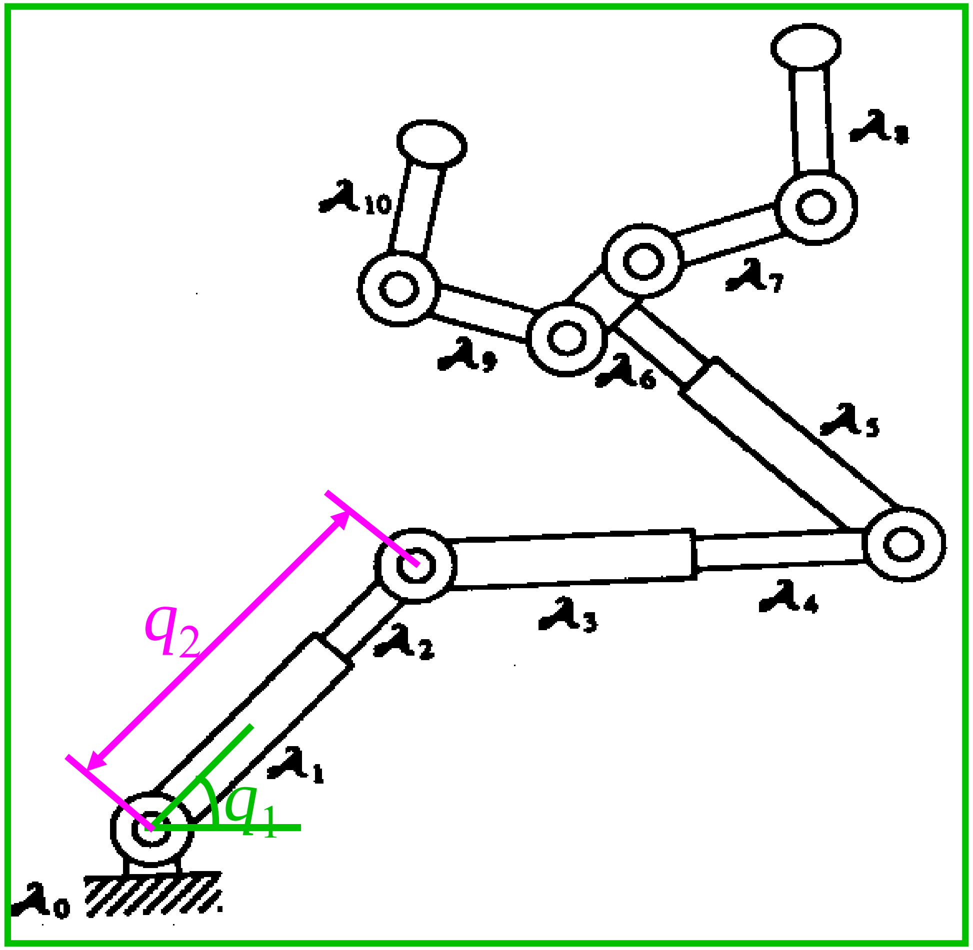
- **Advantage of unit quaternion representation**
  - **Compact**
  - **No singularity (no gimbal lock indicating two axes are aligned)**
  - **Naturally reflect the topology of the space of orientations**




Cyrille Fauvel

- **Number of dofs = 6**
- **Topology:** $R^3 \times SO(3)$

# Example: Articulated Robot




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

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

# Class Objectives were:

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

# Next Time….

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

# Homework

- **Come up with one question on what we have discussed today**
  - **https://forms.gle/R2ZcS9pZ9me9RzmKA**
  - **Write a question two times before the mid-term exam**

- **Browse two papers**
  - **Submit their summaries online before the Tue. Class**
  - **https://forms.gle/2jdXkgYu5snyAb3s8**