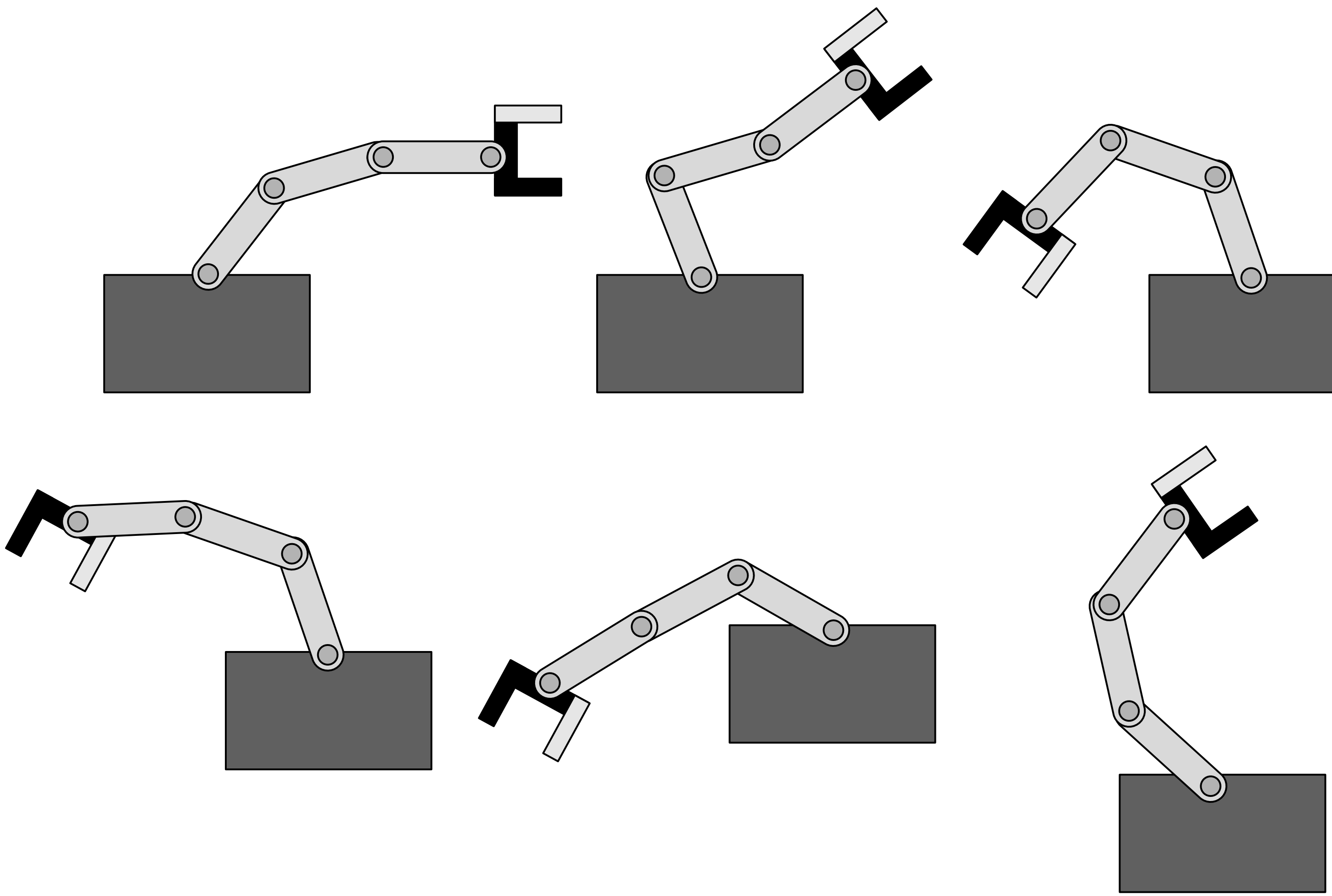
Configuration Space

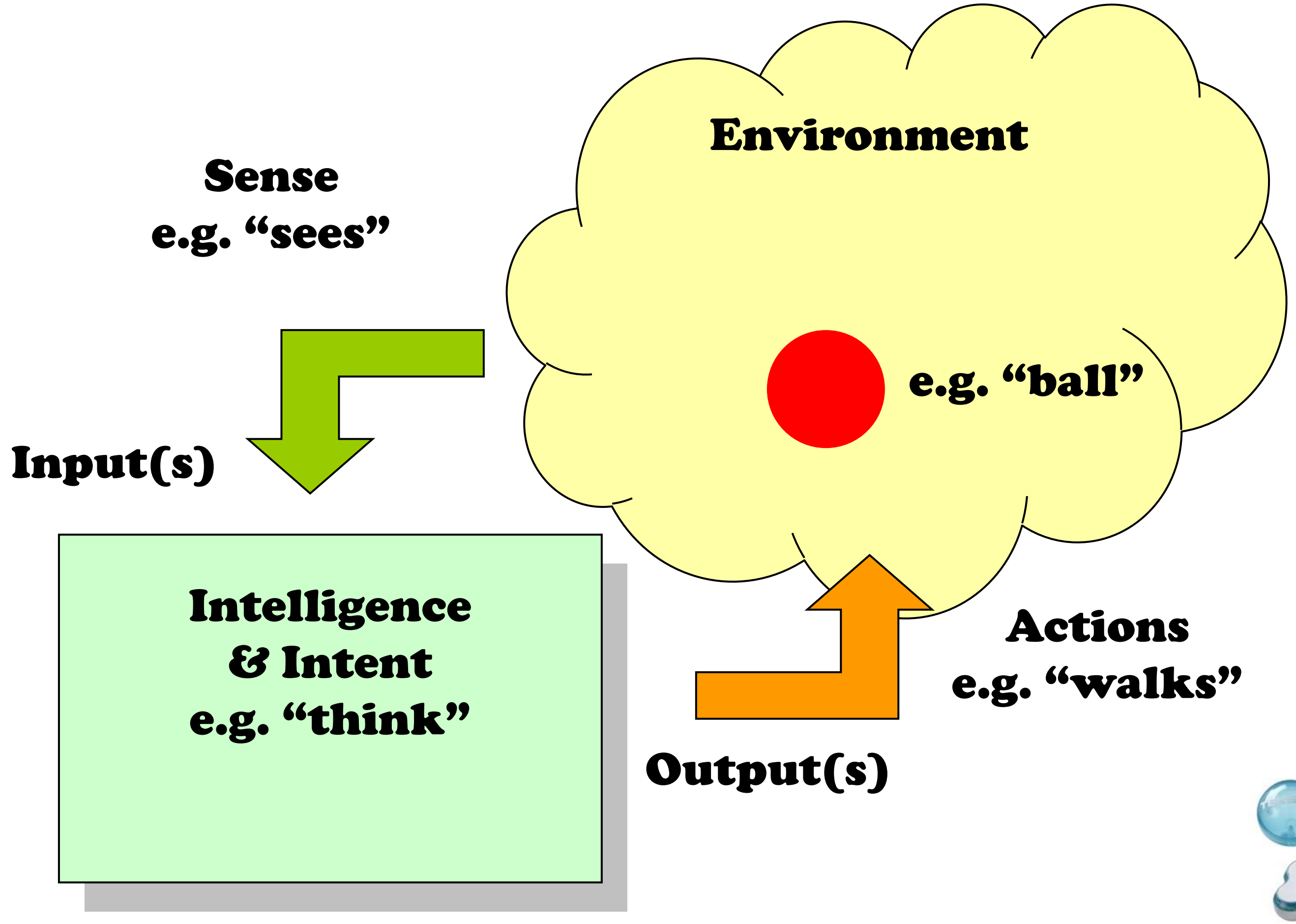
ZA-2203 Robotic Systems

Topics

- Actions
- Describing motion
- Configuration of a robotic system
- Configuration space (CS)
  - CS Dimension: Degree of freedom (DoF)
  - CS Topology
  - CS Representation: explicit, implicit
  - Holonomic, nonholonomic constraints
- Workspace: dextrous, reachable
- Task space

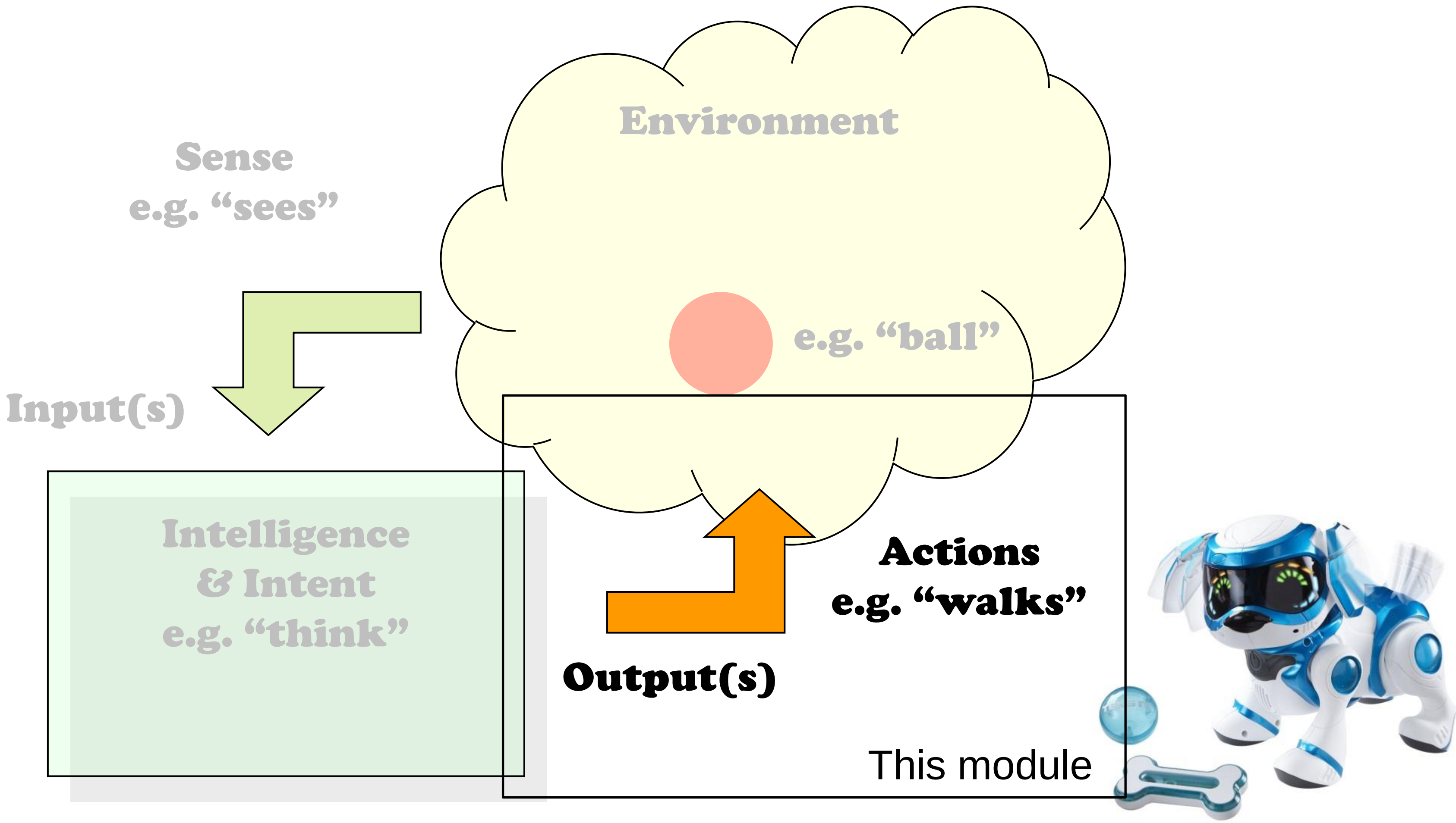


Properties of a robot



Sense → Plan → Act

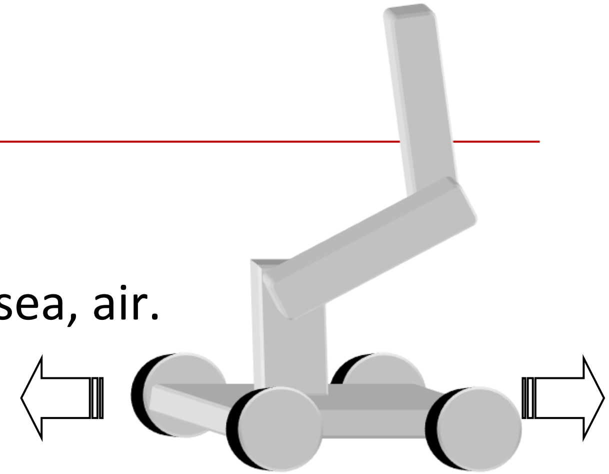
Properties of a robot



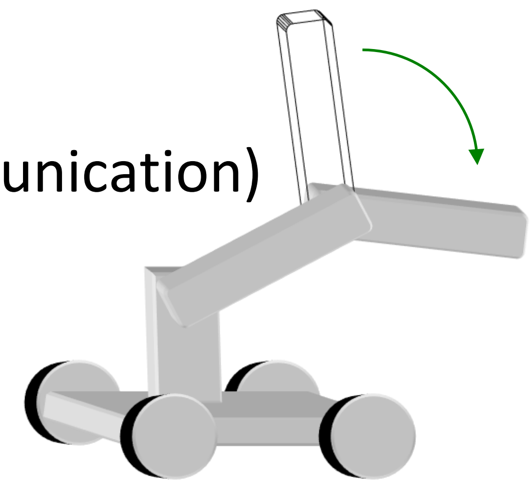
Sense → Plan → Act

Types of actions in robots

- **Locomotion** (interact with own body)
  - Going from one place to another, e.g. ground, sea, air.



- **Manipulation** (interact with environment)
  - Changing the environment, e.g. handling objects.

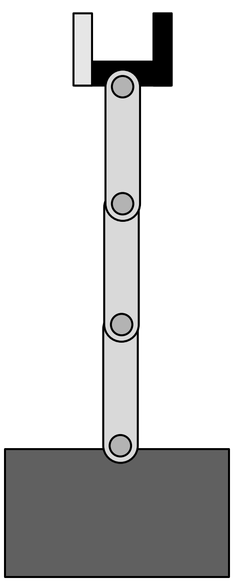
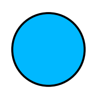
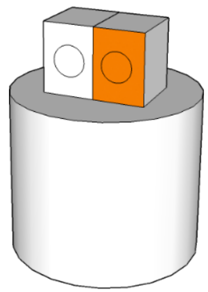


- **Information Presentation** (perceptual, communication)
  - Non-physical changes to the environment, e.g. sound, display.

Actions are delivered through effectors with actuators

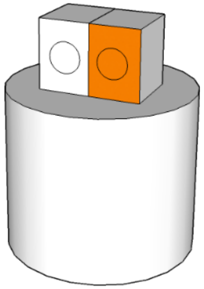
Describing motion

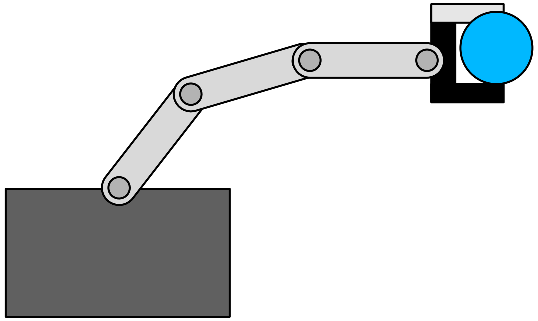
- Robot actions (locomotion and manipulation) involve **motion**, e.g. moving from one place to another, moving the robotic hand to reach an object.
- Involves two properties of the object (as well as the robot) being moved: changing the **position** and **orientation**.

Describing motion

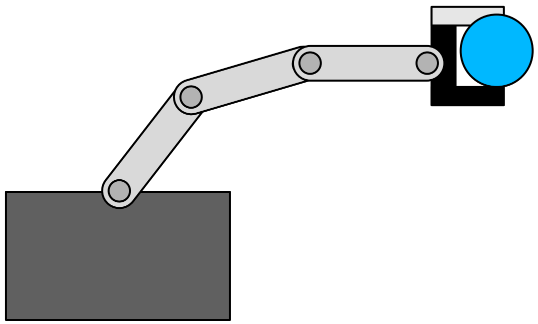
- Robot actions (locomotion and manipulation) involve **motion**, e.g. moving from one place to another, moving the robotic hand to reach an object.
- Involves two properties of the object (as well as the robot) being moved: changing the **position** and **orientation**.

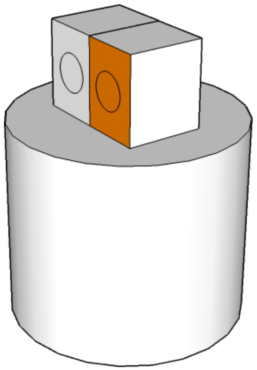




Describing motion

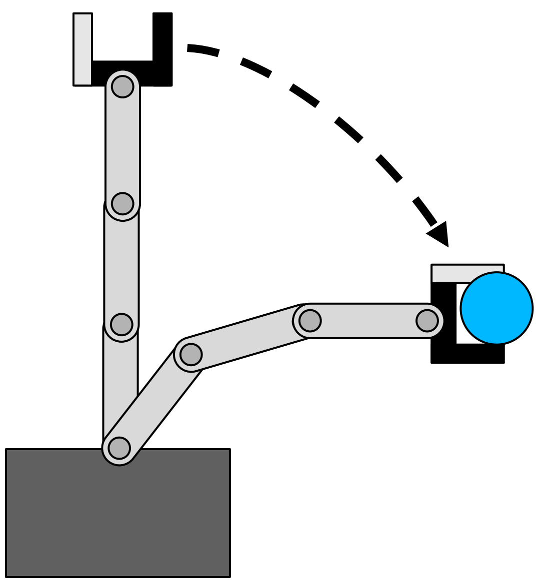
- Robot actions (locomotion and manipulation) involve **motion**, e.g. moving from one place to another, moving the robotic hand to reach an object.
- Involves two properties of the object (as well as the robot) being moved: changing the **position** and **orientation**.

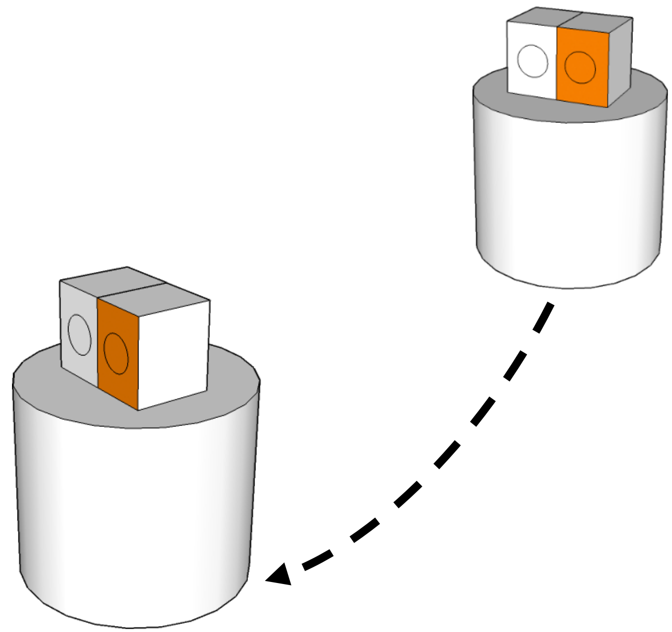




Describing motion

- Robot actions (locomotion and manipulation) involve **motion**, e.g. moving from one place to another, moving the robotic hand to reach an object.
- Involves two properties of the object (as well as the robot) being moved: changing the **position** and **orientation**.

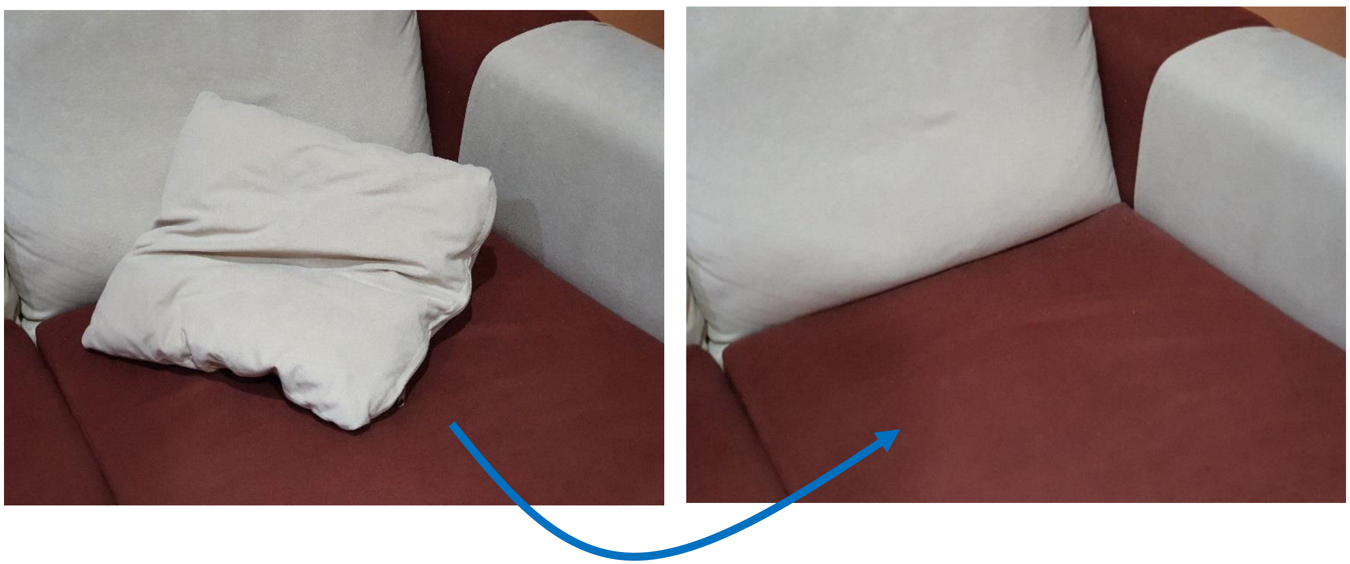




How do we write the position and orientation on paper and in computer?

Configuration of a body

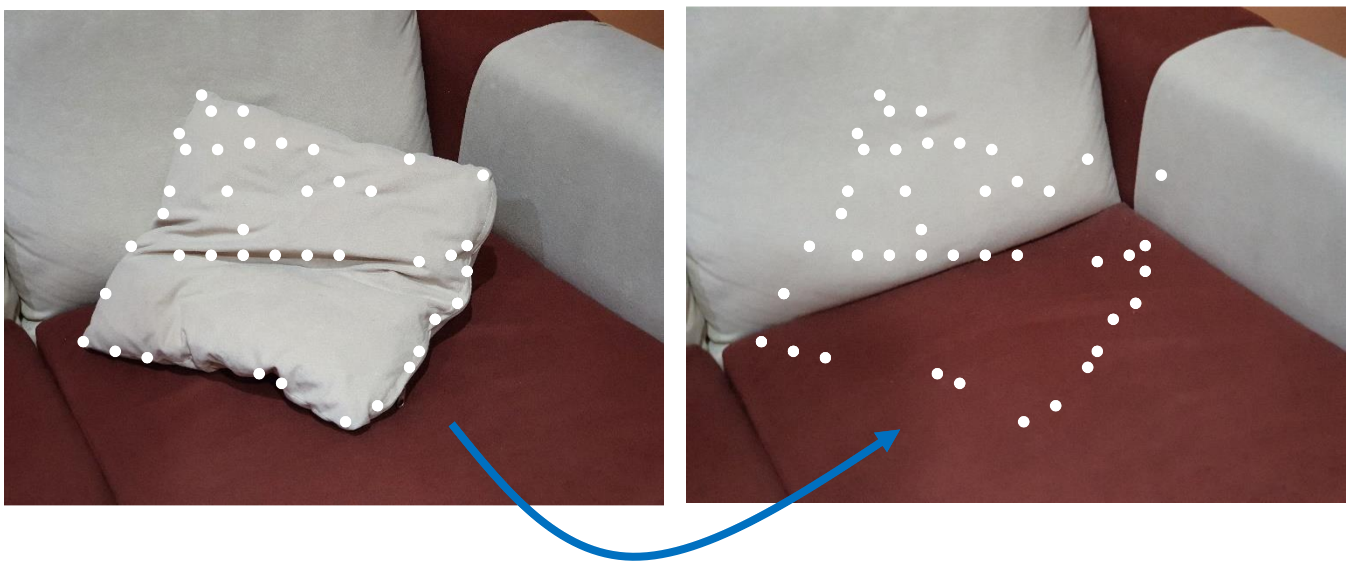
- **Configuration**: a specification of the positions of **all points** of the body.
  - Configuration is a terminology from **classical mechanics**.



How can we describe the pillow such that it can be accurately reproduced, i.e. put back at exact same position and shape?

Configuration of a body

- **Configuration**: a specification of the positions of **all points** of the body.
  - Configuration is a terminology from **classical mechanics**.

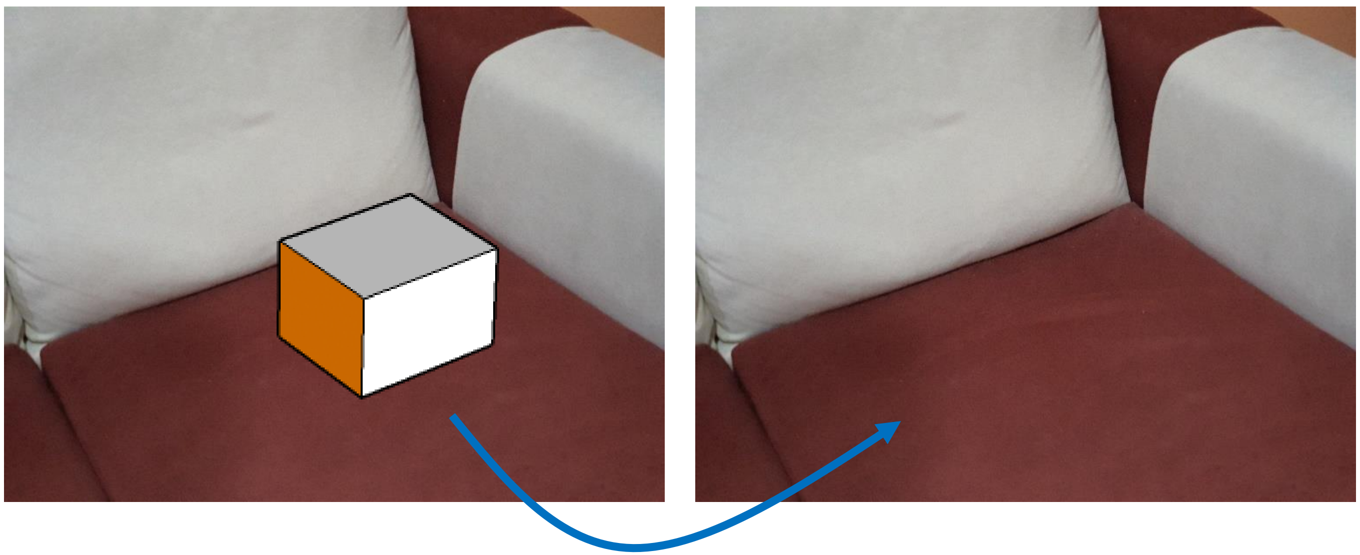


The pillow can be accurately described by its **configuration**, i.e. specifying the position of every point on the pillow in 3D space.

That’s a lot of numbers to specify since all points can change differently with relative to other points. The pillow has no fixed shape. This is one of its many possible configurations.

Configuration of a rigid body

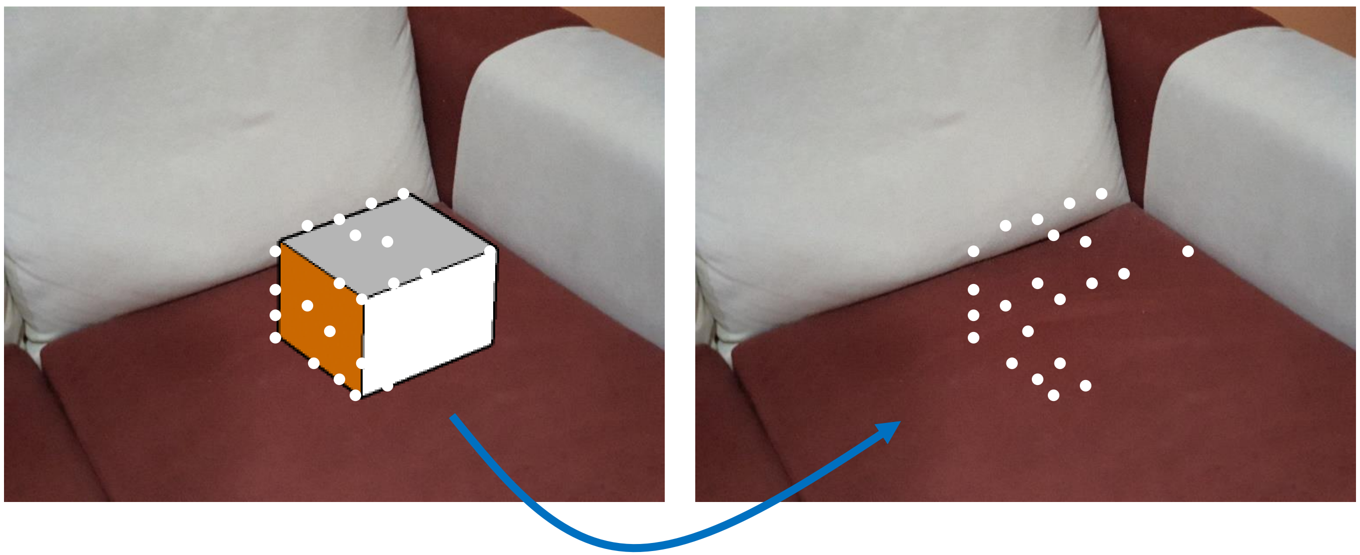
- For a **rigid body**, all points of the body have fix relationship in their positions, hence the configurations of the rigid body can be specified by a **small set of numbers**.
  - This module assumes **robots** are made of **rigid bodies**.



How can we describe the box such that it can be accurately reproduced, i.e. drawn at exact same position and shape?

Configuration of a rigid body

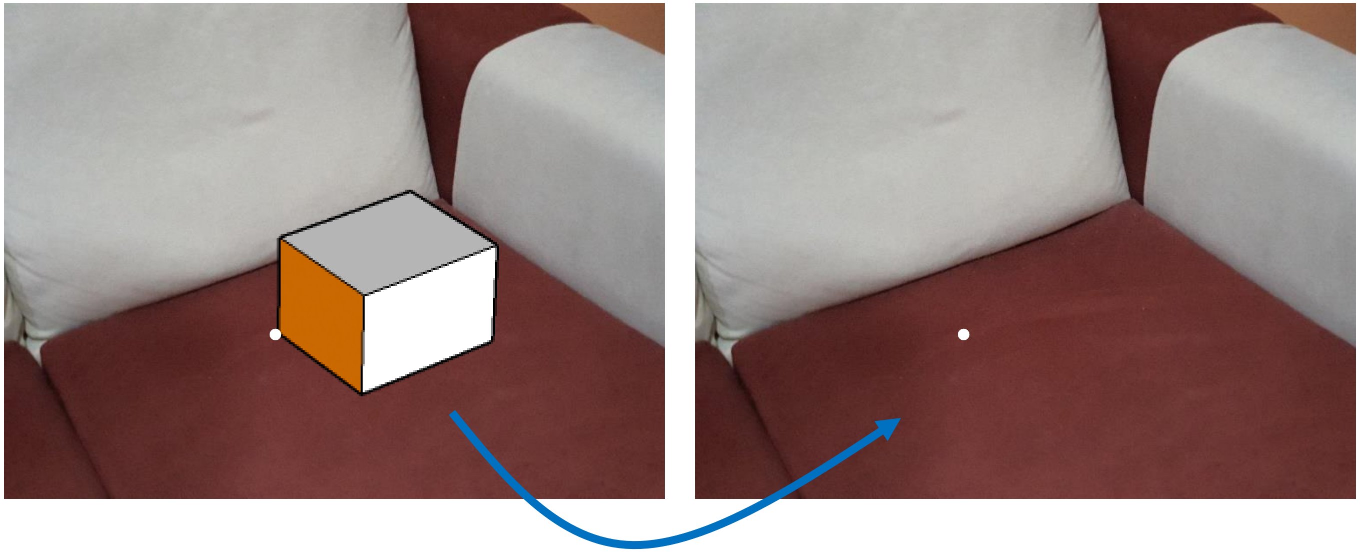
- For a **rigid body**, all points of the body have fix relationship in their positions, hence the configurations of the rigid body can be specified by a **small set of numbers**.
  - This module assumes **robots** are made of **rigid bodies**.



The box can be accurately described by its **configuration**, i.e. specifying the position of every point on the pillow in 3D space.

Configuration of a rigid body

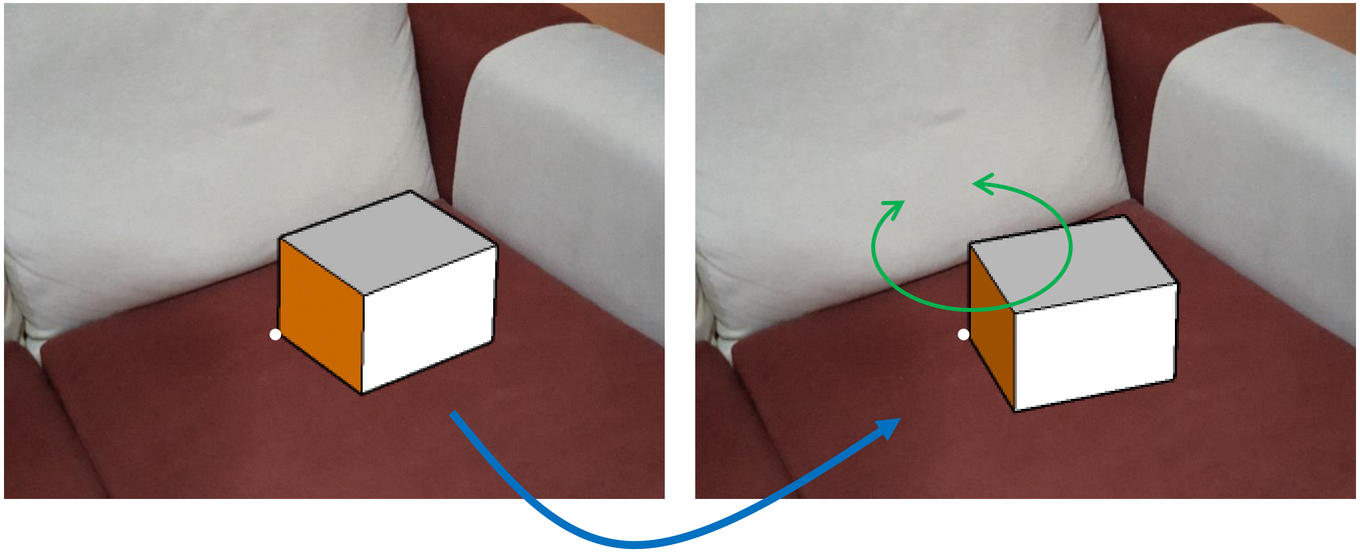
- For a **rigid body**, all points of the body have fix relationship in their positions, hence the configurations of the rigid body can be specified by a **small set of numbers**.
  - This module assumes **robots** are made of **rigid bodies**.



However, given that we know the shape, which is fixed, we need less points

Configuration of a rigid body

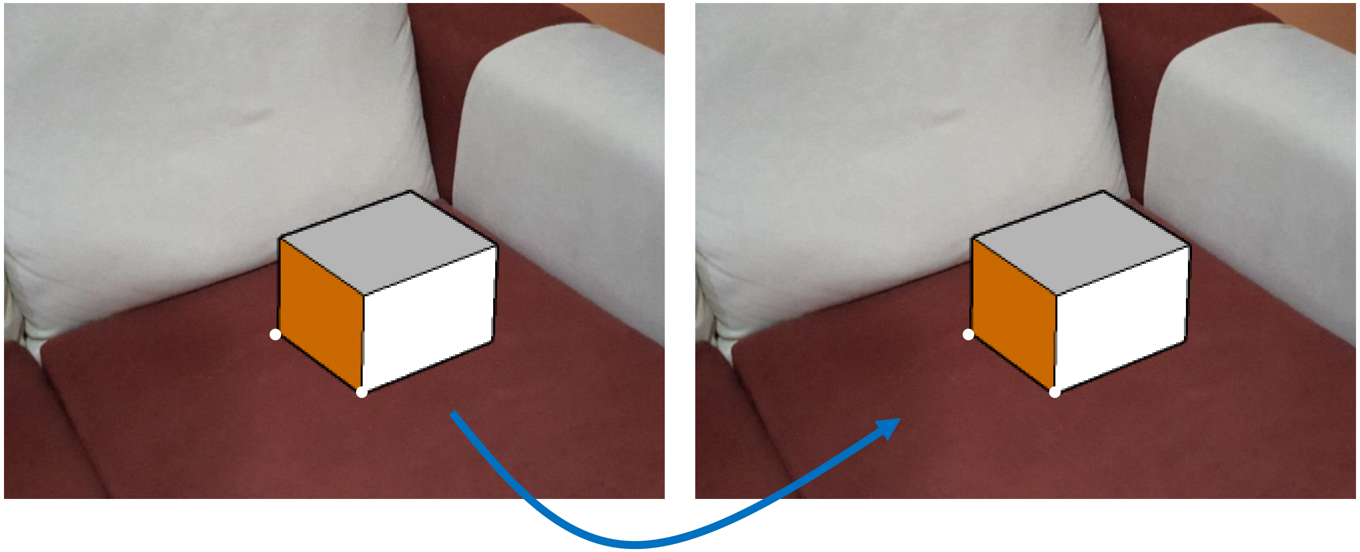
- For a **rigid body**, all points of the body have fix relationship in their positions, hence the configurations of the rigid body can be specified by a **small set of numbers**.
  - This module assumes **robots** are made of **rigid bodies**.



One point is not enough

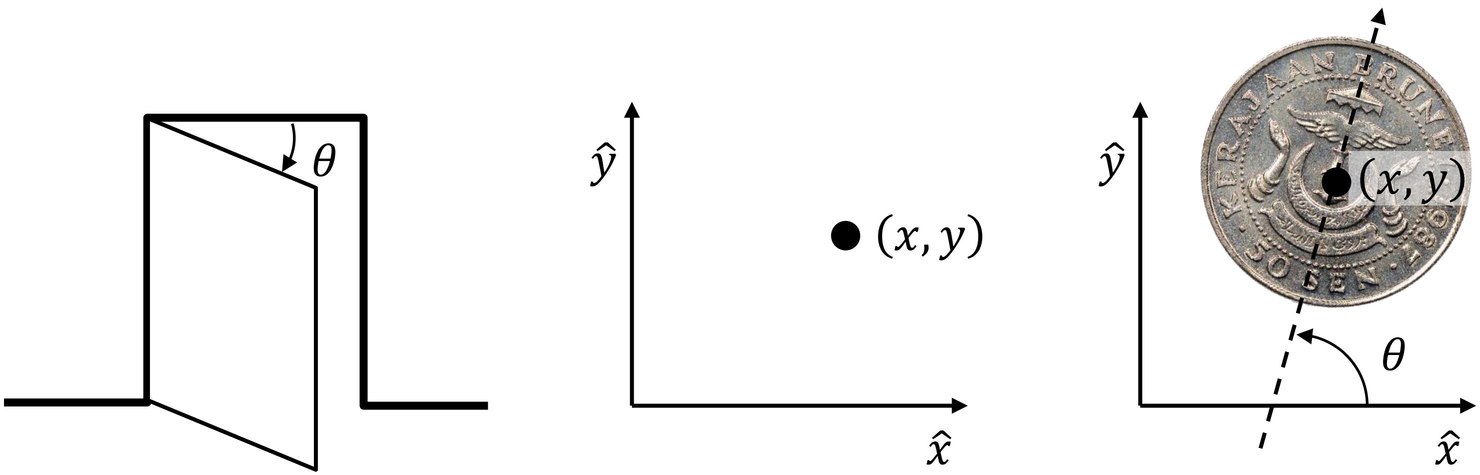
Configuration of a rigid body

- For a **rigid body**, all points of the body have fix relationship in their positions, hence the configurations of the rigid body can be specified by a **small set of numbers**.
  - This module assumes **robots** are made of **rigid bodies**.



Two points are sufficient to specify the configuration of the box

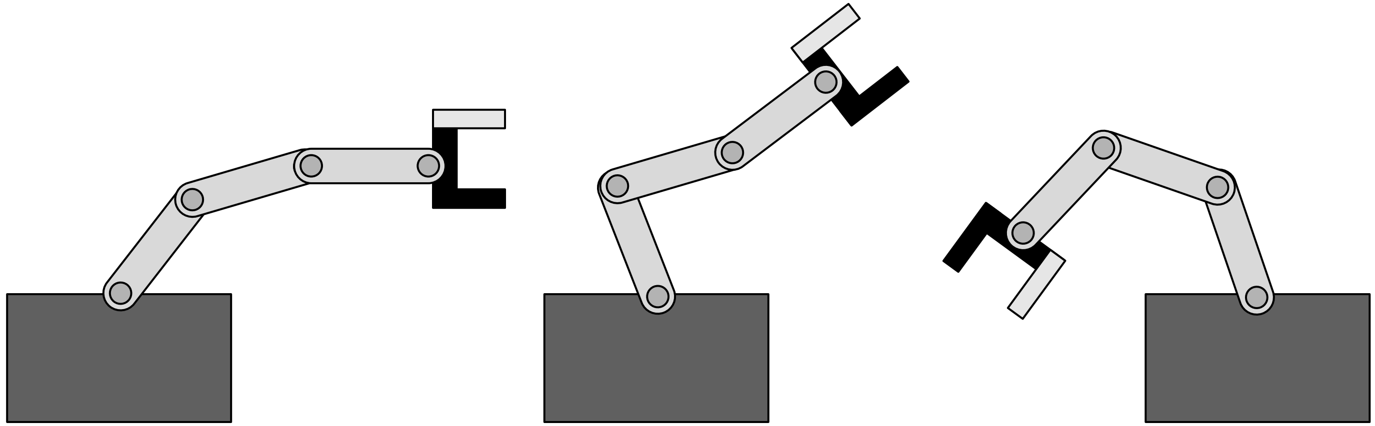
Example configuration of rigid bodies



"The parameters that define the configuration of a system are called **generalized coordinates**, and the space defined by these coordinates is called the **configuration space** of the physical system." - Wikipedia

Configurations of a robotic system

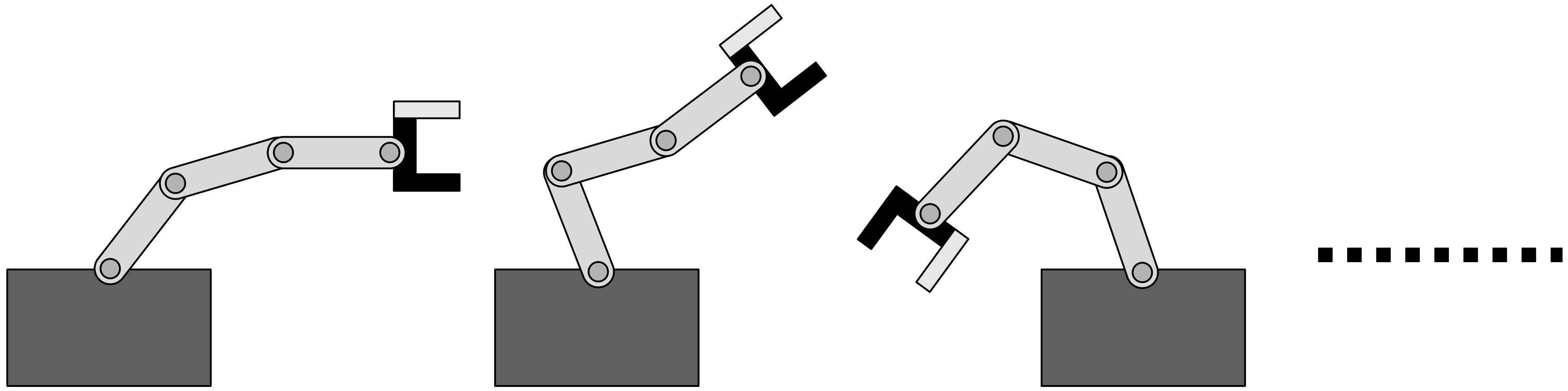
- Robot's **configuration**: a specification of the positions of **all points** of the robot.



Different configurations of a robotic system

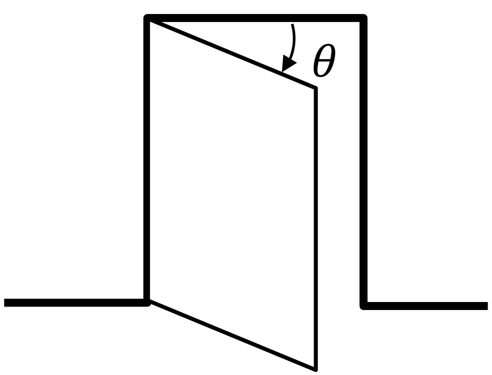
Configuration space: where is the robot?

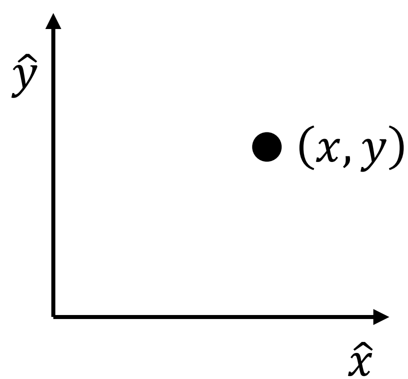
- **Configuration space** (C-space) of a robot refers to the set (space) of all possible configurations of the robot.
- A configuration of a robot is represented by a point in its C-space.
- C-space has two properties:
  - **Degree of freedom** (dof): dimension of the C-space
  - **Topology** (shape)

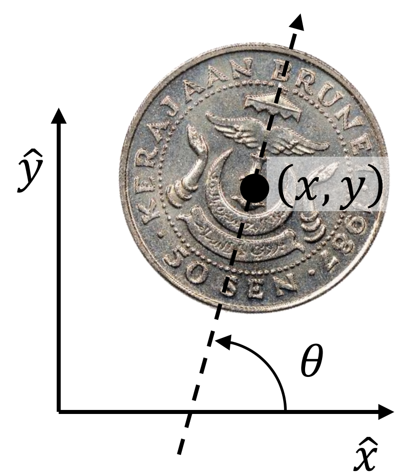


Degree of freedom (dof)

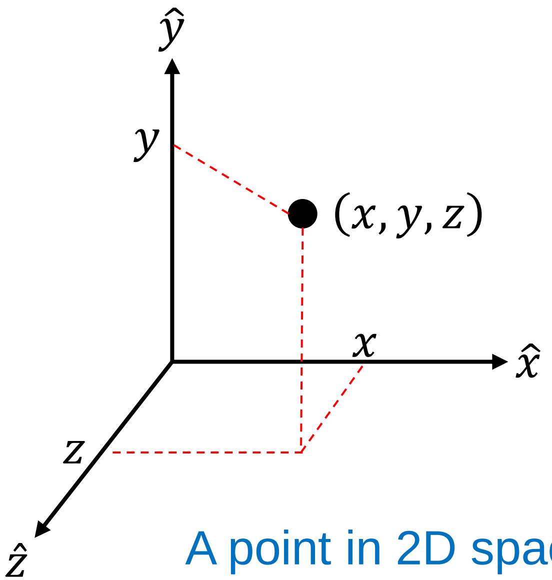
- **Degree of freedom** (dof) can be defined as, depending on the context:
  - **Robotic systems**: number of independent directions of movement. In this context, dof is also called **mobility**.
  - **Configuration**: minimum number of continuous real numbers to specify a configuration.
  - **C-space**: dimension of C-space.



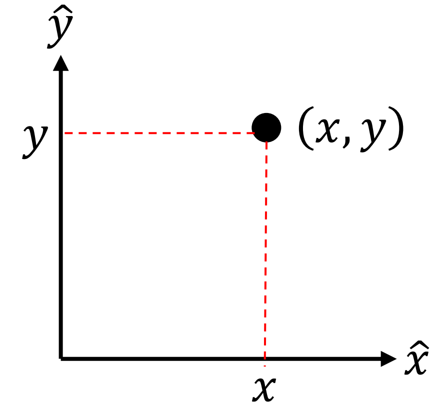




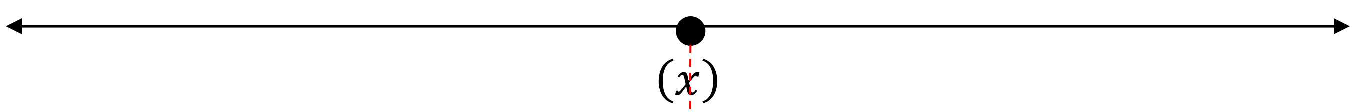
Degree of freedom of a point



A point in 3-dimensional (3D) space can be represented by three numbers or coordinates (x, y, z). If it is free to move in all 3 dimensions, it has 3 dof.



A point in 2D space (plane) lost one dof in one of the dimensions, i.e. one dimension is constrained. It can be represented by two numbers or coordinates (x, y). it has 2 dof.

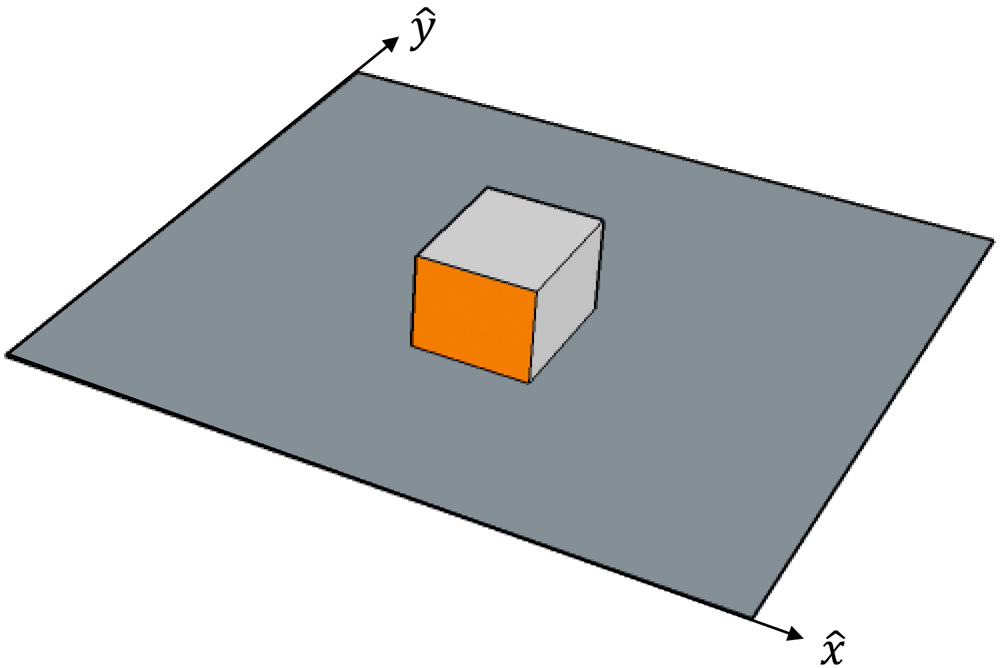


A point in 1D space (line) lost two dof in two of the dimensions, i.e. two dimensions are constrained. It can be represented by one number or coordinate (x). it has 1 dof.

Degree of freedom of a rigid body on a plane

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$
\begin{aligned}
&dof \\
&= (sum\ of\ dof\ of\ the\ points) \\
&- (number\ of\ independent\ constraints)
\end{aligned}
$$

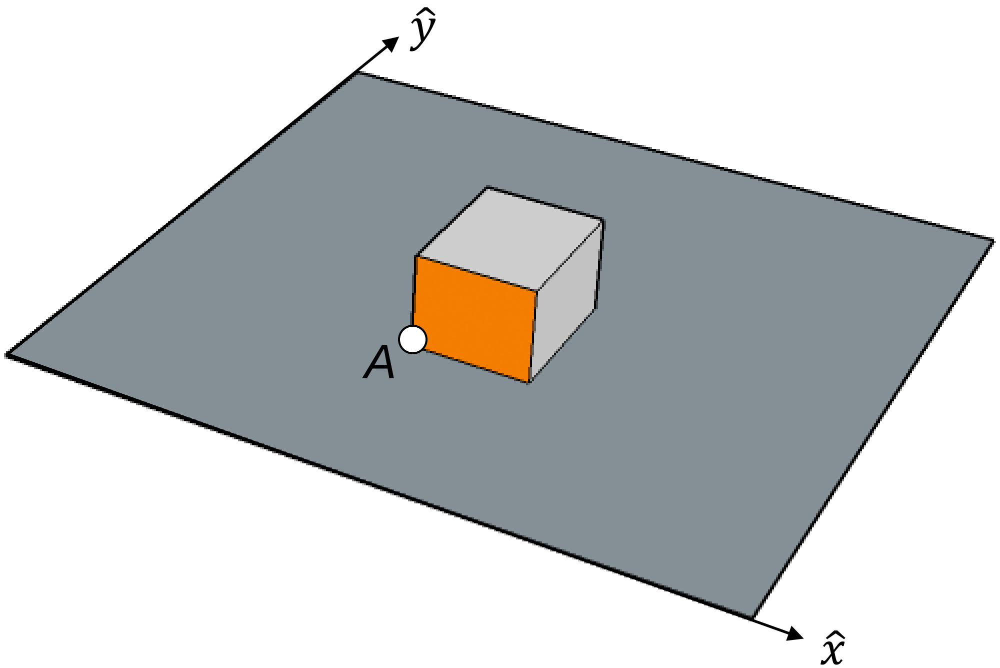


A rigid body on a plane (2D).
Think an autonomous car.

Degree of freedom of a rigid body on a plane

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$
\begin{aligned}
&dof\\
&= (sum\ of\ dof\ of\ the\ points)\\
&- (number\ of\ independent\ constraints)
\end{aligned}
$$



Start at 1 point on the body, say *A*

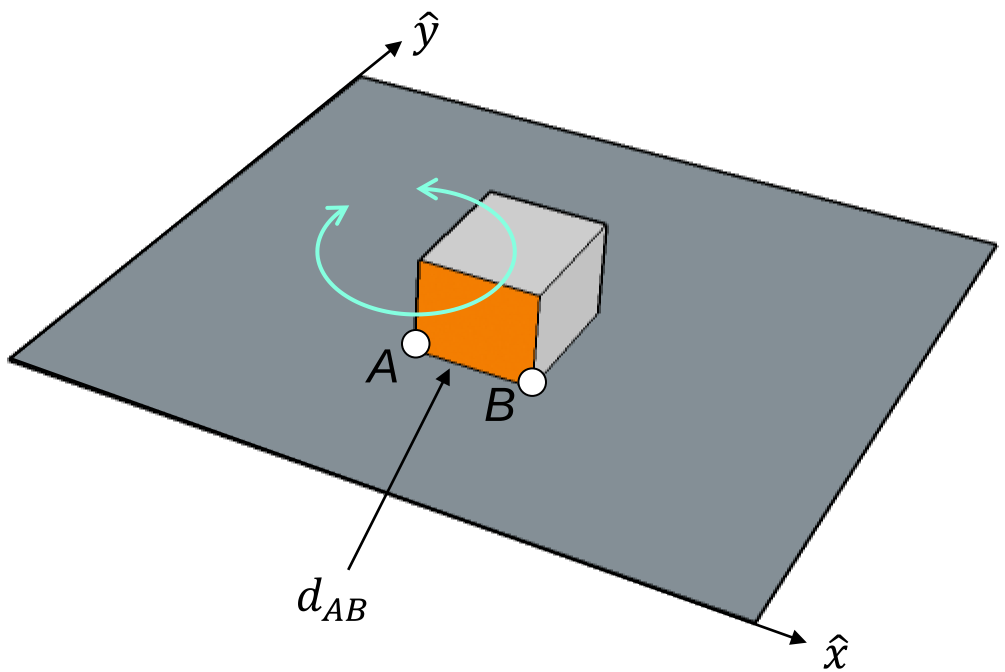
$$A = (x_A, y_A)$$

A point has 2 dof on a plane. Point *A* has 2 dof and is free to move.

Degree of freedom of a rigid body on a plane

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$dof \\ = (sum\ of\ dof\ of\ the\ points) \\ - (number\ of\ independent\ constraints)$$



Select a second point on the body, say *B*

$$B = (x_B, y_B)$$

B can move however constrained on the circle at a fixed distance to *A*.

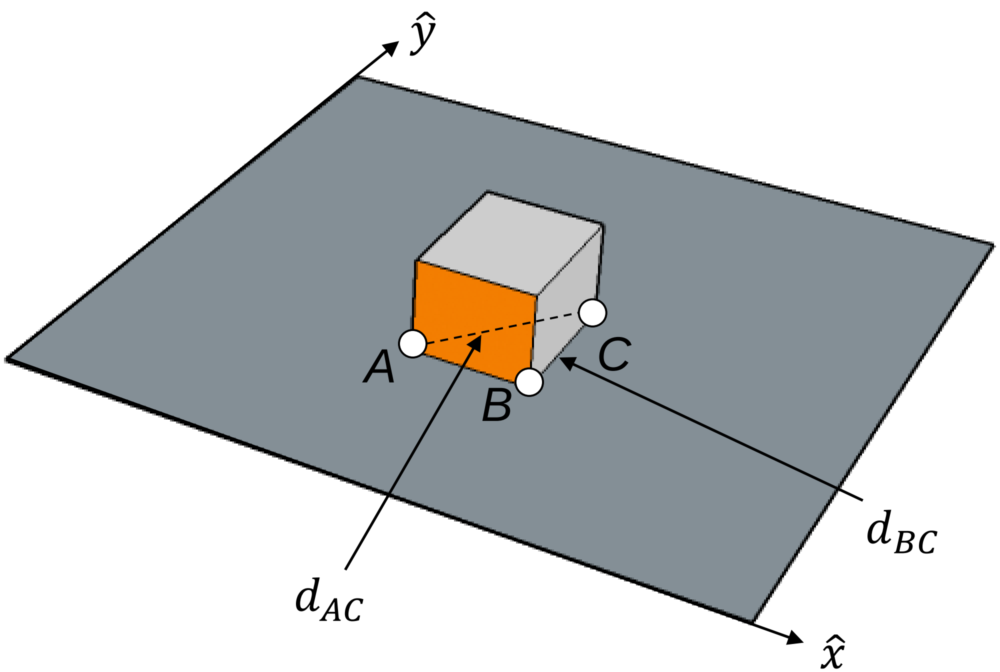
$$(x_B - x_A)^2 + (y_B - y_A)^2 = d_{AB}{}^2$$

(Circle equation)

Degree of freedom of a rigid body on a plane

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$
\begin{aligned}
&dof \\
&= (sum\ of\ dof\ of\ the\ points) \\
&- (number\ of\ independent\ constraints)
\end{aligned}
$$



Select a third point on the body, say *C*

$$C = (x_C, y_C)$$

C cannot move independently due to two constraints. If *A* and *B* are positioned, *C* will be fixed at fixed distances to *A* and *B*.

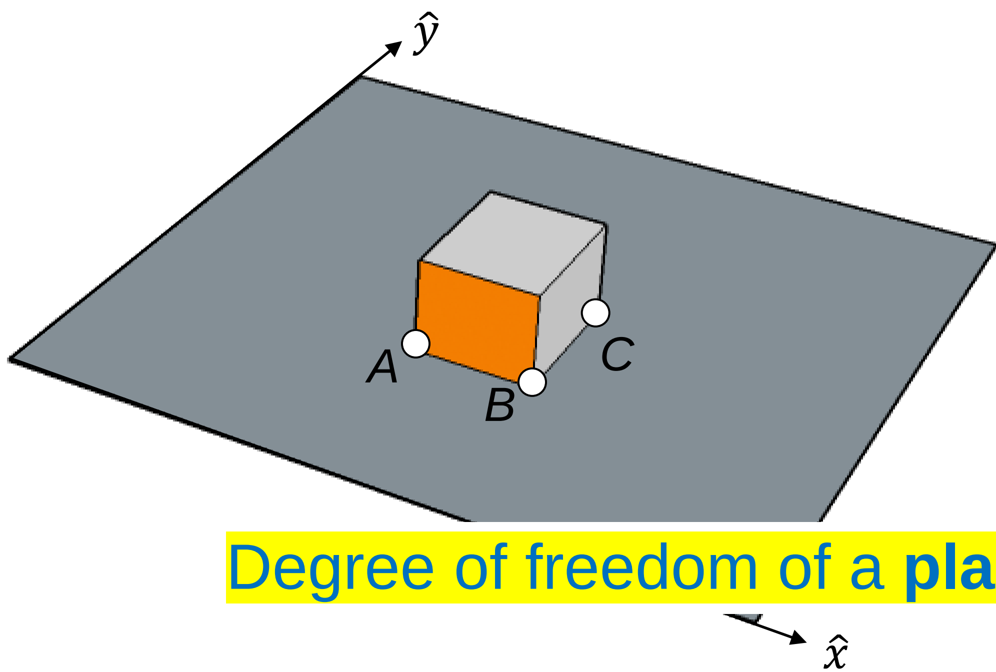
$$(x_C - x_A)^2 + (y_C - y_A)^2 = {d_{AC}}^2$$
$$(x_C - x_B)^2 + (y_C - y_B)^2 = {d_{BC}}^2$$

Any further points will be likewise constrained.

Degree of freedom of a rigid body on a plane

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$dof = (sum\ of\ dof\ of\ the\ points) - (number\ of\ independent\ constraints)$$



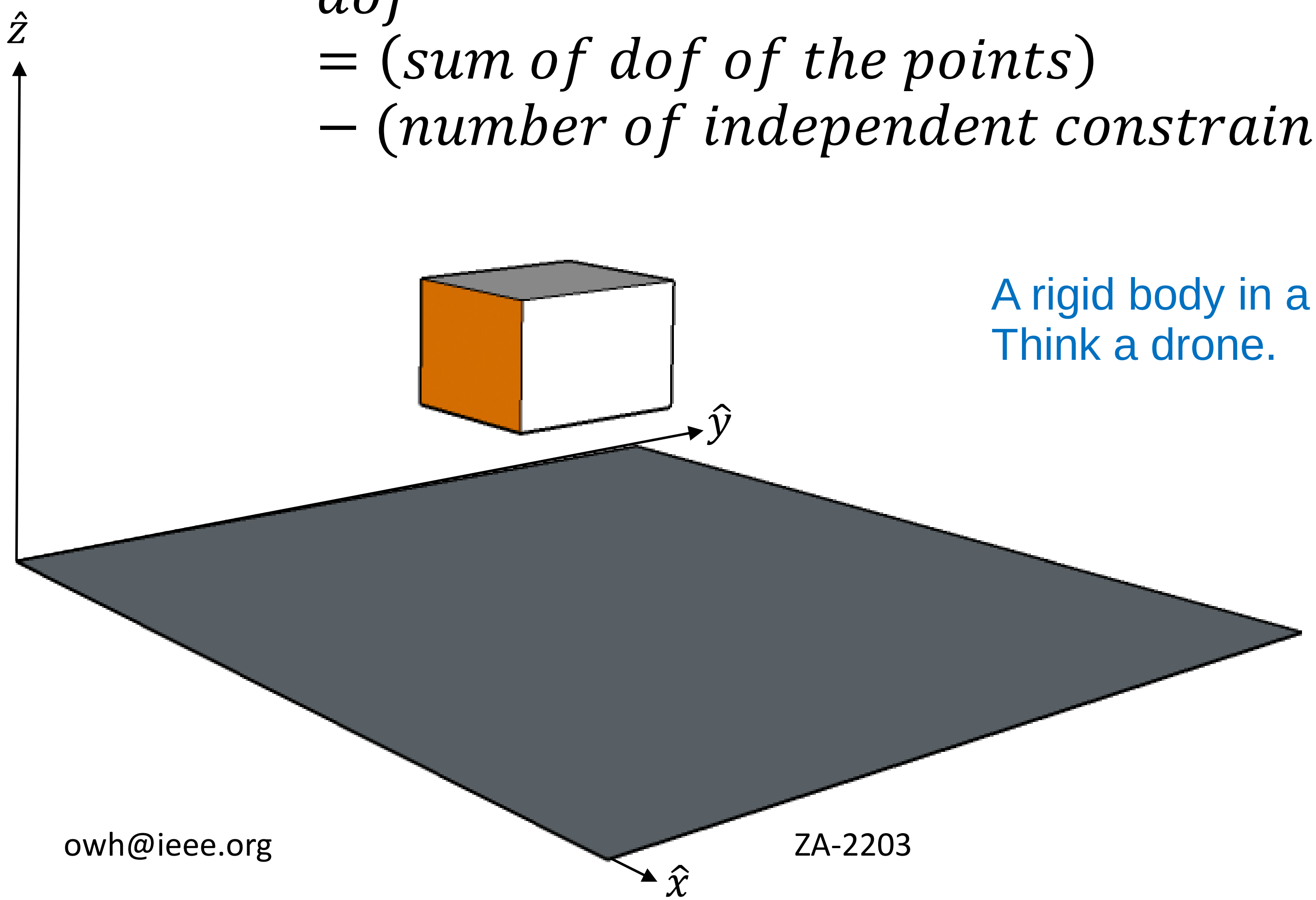
$$dof = (dof_A + dof_B + dof_C + \cdots) - (c_A + c_B + c_C + \cdots) = (2 + 2 + 2 + \cdots) - (0 + 1 + 2 + \cdots) = 3$$

Degree of freedom of a **planar rigid body** is 3

Degree of freedom of a rigid body in space

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$\begin{aligned} &dof \\ &= (sum\ of\ dof\ of\ the\ points) \\ &- (number\ of\ independent\ constraints) \end{aligned}$$

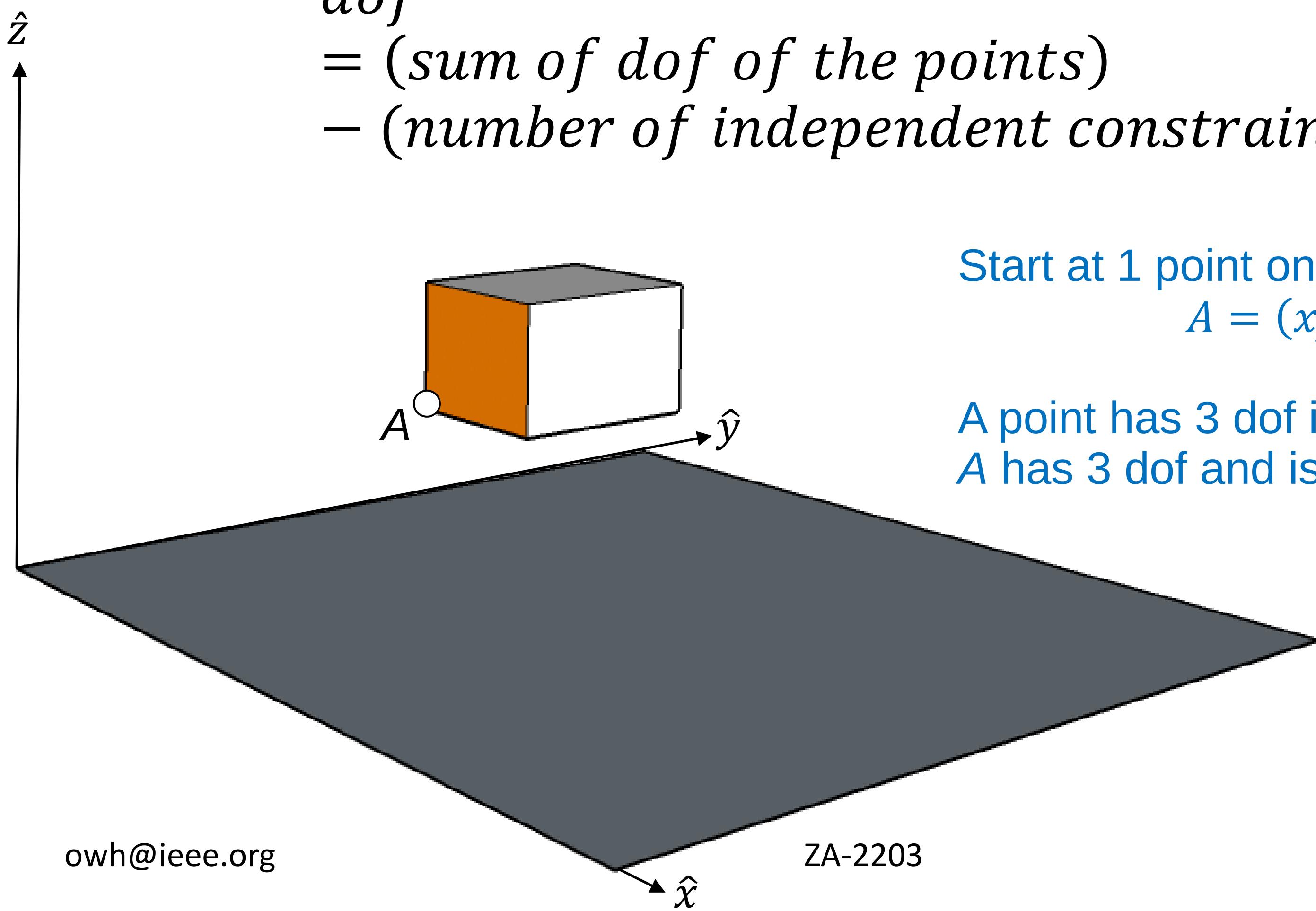


A rigid body in a space (3D).
Think a drone.

Degree of freedom of a rigid body in space

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$
\begin{aligned}
&dof \\
&= (sum\ of\ dof\ of\ the\ points) \\
&- (number\ of\ independent\ constraints)
\end{aligned}
$$



Start at 1 point on the body, say *A*

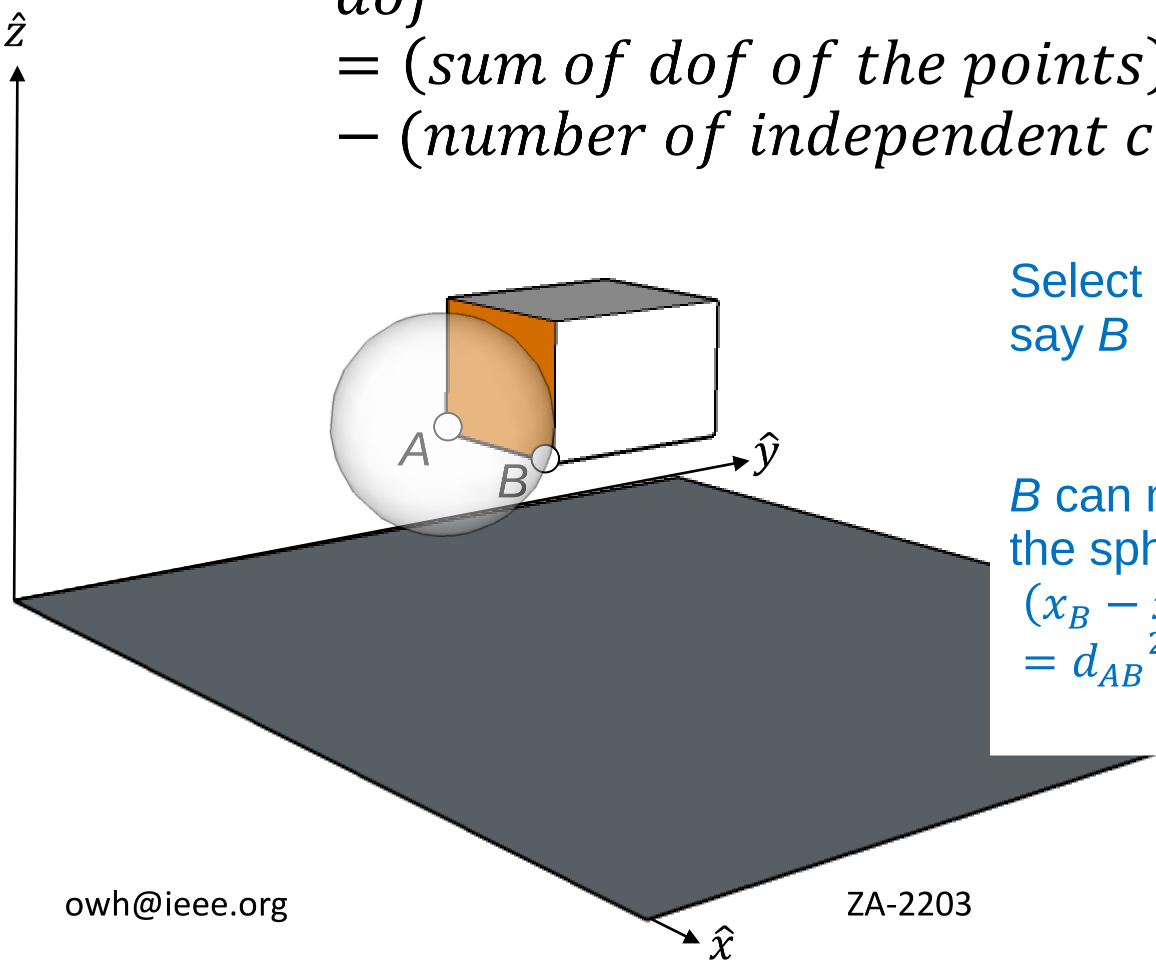
$$A = (x_A, y_A, z_A)$$

A point has 3 dof in the space. Point *A* has 3 dof and is free to move.

Degree of freedom of a rigid body in space

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$dof = (sum\ of\ dof\ of\ the\ points) - (number\ of\ independent\ constraints)$$



Select a second point on the body, say *B*

$$B = (x_B, y_B, z_B)$$

B can move however constrained on the sphere at a fixed distance to *A*.

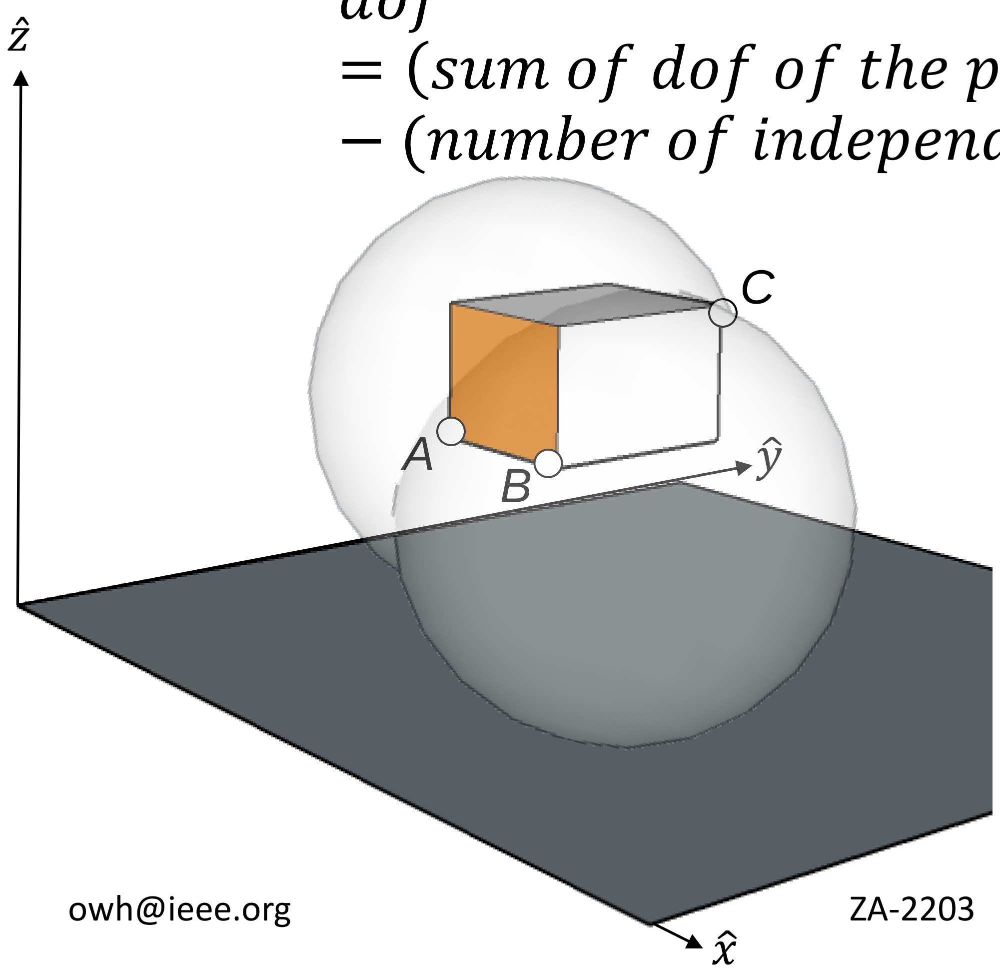
$$(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2 = {d_{AB}}^2$$

(Sphere equation)

Degree of freedom of a rigid body in space

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$dof = (sum\ of\ dof\ of\ the\ points) - (number\ of\ independent\ constraints)$$



Select a third point on the body, say *C*

$$C = (x_C, y_C, z_C)$$

C can move however constrained on the intersection of the spheres centered at *A* and *B*.

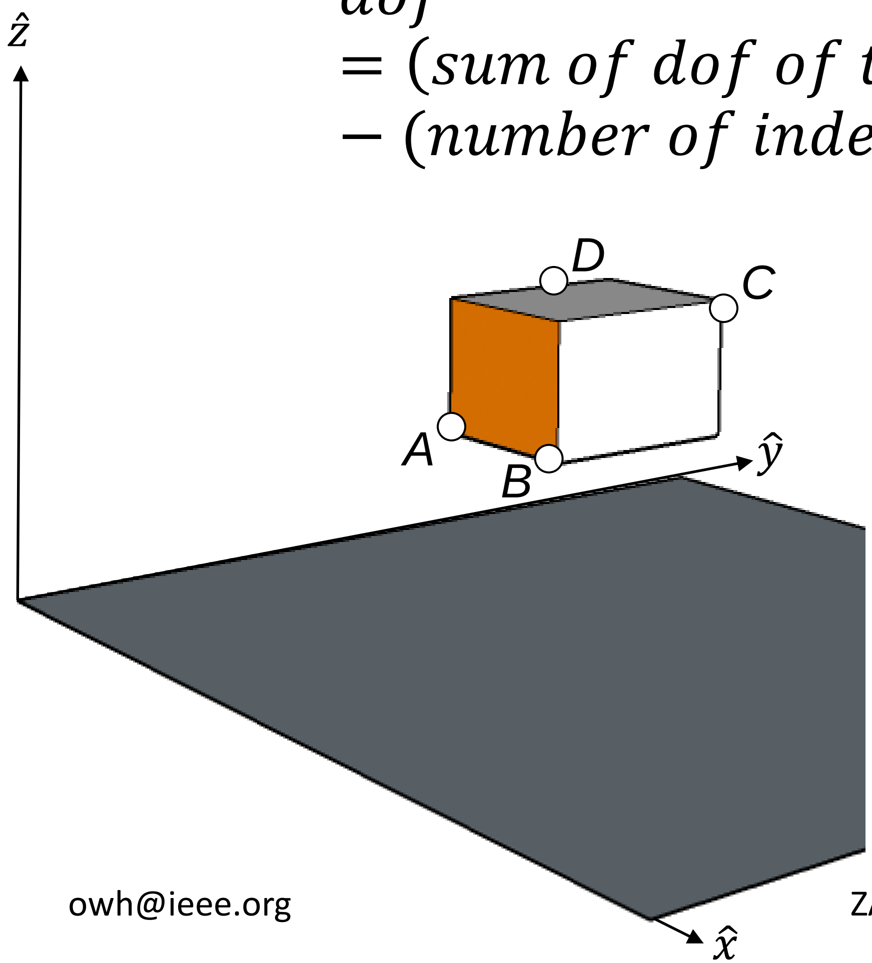
$$(x_C - x_A)^2 + (y_C - y_A)^2 + (z_C - z_A)^2 = {d_{AC}}^2$$

$$(x_C - x_B)^2 + (y_C - y_B)^2 + (z_C - z_B)^2 = {d_{BC}}^2$$

Degree of freedom of a rigid body in space

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$
\begin{aligned}
&dof \\
&= (sum\ of\ dof\ of\ the\ points) \\
&- (number\ of\ independent\ constraints)
\end{aligned}
$$



Select a fourth point on the body, say *D*

$$D = (x_D, y_D, z_D)$$

D cannot move independently if *A*, *B* and *C* have been positioned. It is subject to three fixed distance constraints to *A*, *B* and *C*.

$$(x_D - x_A)^2 + (y_D - y_A)^2 + (z_D - z_A)^2 = {d_{AD}}^2$$

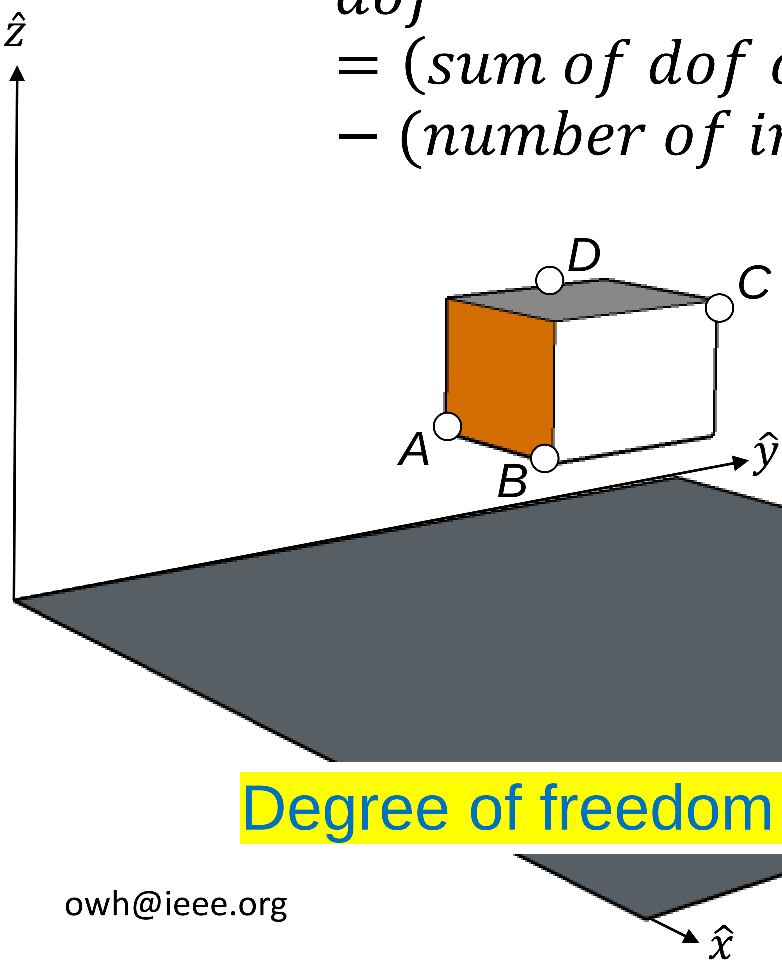
$$(x_D - x_B)^2 + (y_D - y_B)^2 + (z_D - z_B)^2 = {d_{BD}}^2$$

$$(x_D - x_C)^2 + (y_D - y_C)^2 + (z_D - z_C)^2 = {d_{CD}}^2$$

Degree of freedom of a rigid body in space

- A **rigid body** can be considered as composite of points (particles). Its dof can be determined by

$$
\begin{aligned}
&dof \\
&= (sum\ of\ dof\ of\ the\ points) \\
&- (number\ of\ independent\ constraints)
\end{aligned}
$$

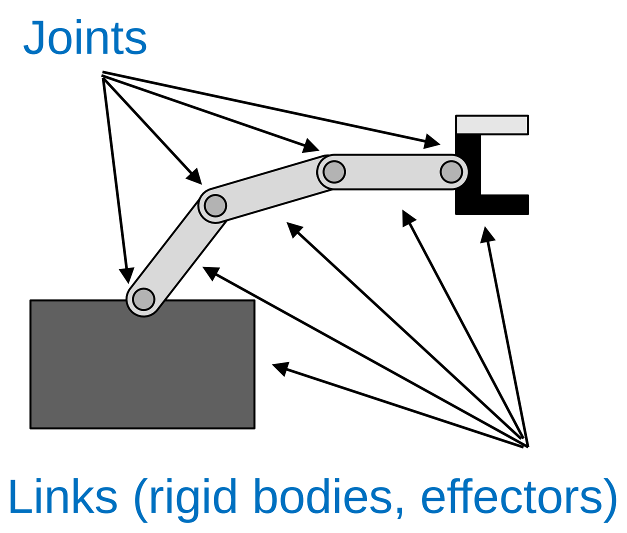


$$
\begin{aligned}
&dof \\
&= (dof_A + dof_B + dof_C + dof_D + \cdots) \\
&- (c_A + c_B + c_C + c_C + c_D + \cdots) \\
&= (3 + 3 + 3 + 3 + \cdots) \\
&- (0 + 1 + 2 + 3 + \cdots) = 6
\end{aligned}
$$

Degree of freedom of a **spatial rigid body** is 6

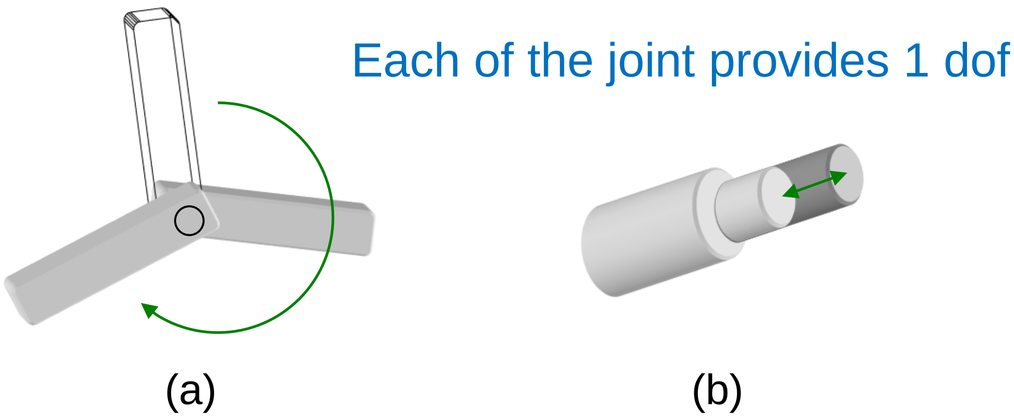
Degree of freedom (dof) of robots

- Robots are usually modelled as **articulated rigid bodies** (effectors), i.e. consist of rigid bodies connected by a chain of **joints**.
- The effectors (rigid bodies) are called **links**.
- *Dof* of a robot can be determined by the **Grubler's formula**.



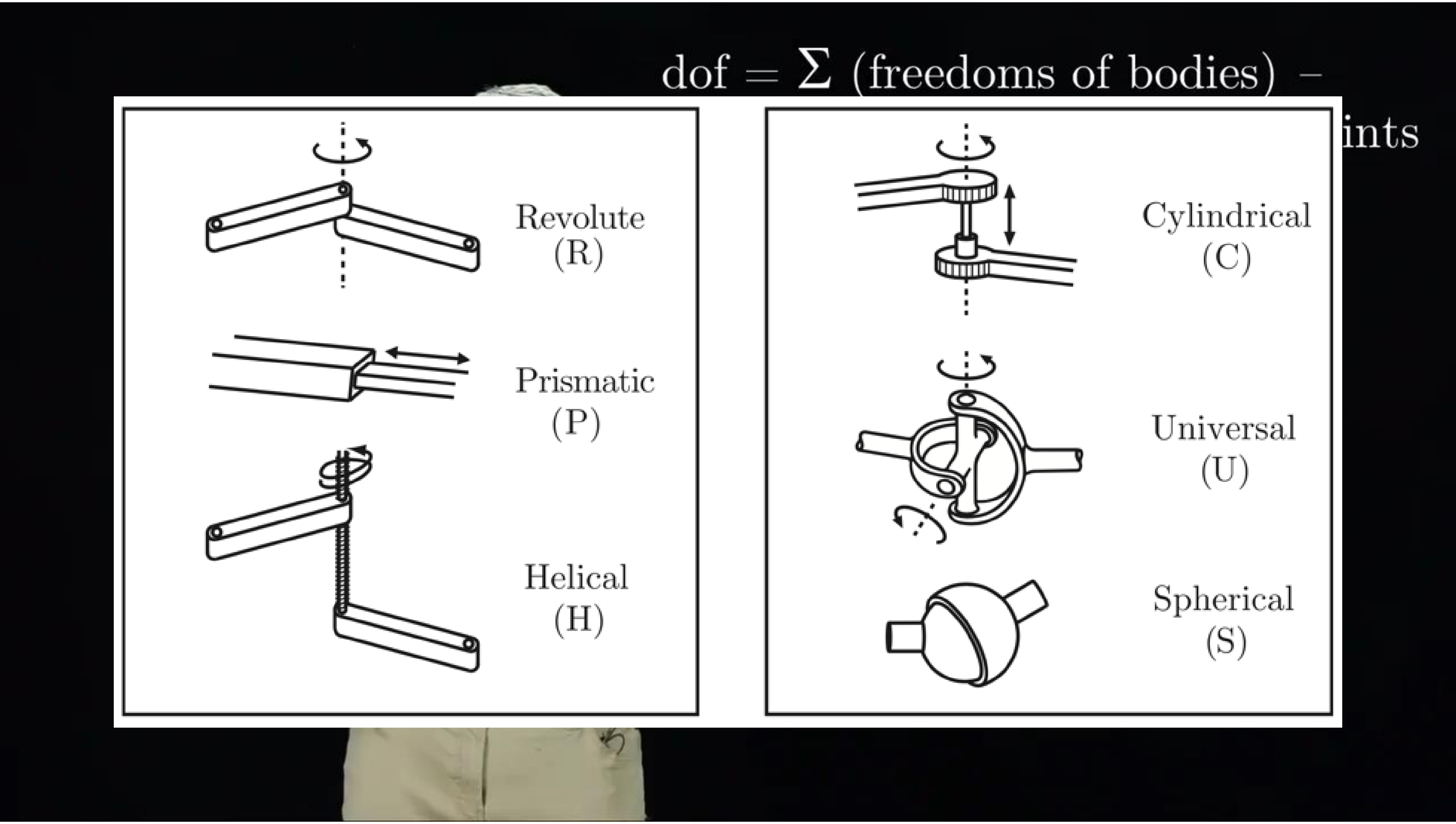
Robotic joints

- A **joint** contributes to the relative motion of the **links**. There are two **basic types of joint motion**:
  - (a) **Rotational** about a pivot. Referred as **revolute** (R) joint.
  - (b) **Translational** along a line. Referred as **linear** or **prismatic** (P) joint.



- Every joint connects exactly two links.
- A joint **constrain** the motion of two connected links with relative to each other.

Robotic joints



Source: Modern Robotics

Dof and constraints of different joint types

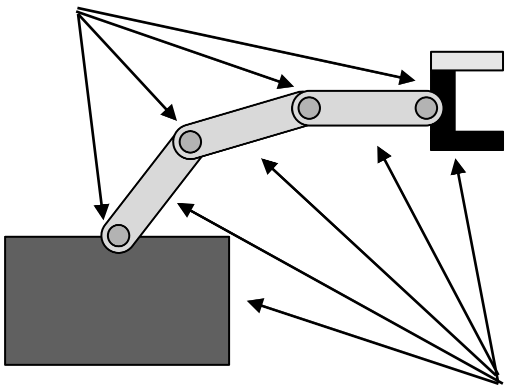
- For planar rigid body, it has $dof\ m = 3$. A joint imposes c constraints resulting in $dof\ f$.
- For spatial rigid body, $dof\ m = 6$.

$$f = m - c \qquad m = f + c \qquad c = m - f$$

Joint type	dof f	Constraints c between two planar rigid bodies	Constraints c between two spatial rigid bodies
Revolute (R)	1	2	5
Prismatic (P)	1	2	5
Helical (H)	1	N/A	5
Cylindrical (C)	2	N/A	4
Universal (U)	2	N/A	4
Spherical (S)	3	N/A	3

Source: Modern Robotics

Grubler's formula



$$
\begin{aligned}
&dof \\
&= (sum\ of\ dof\ of\ the\ bodies) \\
&- (number\ of\ independent\ constraints) \\
&= m(N-1) - \sum_{i=1}^{J} c_i \\
&= m(N-1) - \sum_{i=1}^{J} (m - f_i)
\end{aligned}
$$

$$dof = m(N - 1 - J) + \sum_{i=1}^{J} f_i$$

$m = 3\ if\ planar\ robot, 6\ if\ spatial\ robot$

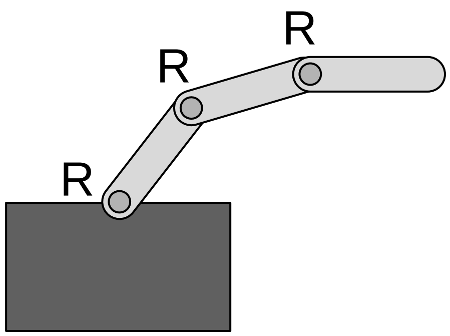
$N = No.of\ links$

$J = No.of\ joints$

$f_i = dof\ of\ each\ joint$

Dof example: Serial-chain RRR (3R) robot

$$dof = m(N - 1 - J) + \sum_{i=1}^{J} f_i$$



Open-chain mechanism
- also called **serial-chain**

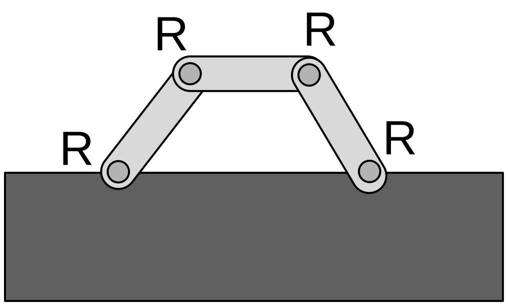
<u>Simple serial 3R robot (planar)</u>

m = 3 (planar)
N = 4 (include ground/base)
J = 3
fi = 1 each

$$dof = 3(4 - 1 - 3) + \sum_{i=1}^{3} 1 = 3$$

Dof example: Closed-chain RRRR (4R) robot

$$dof = m(N - 1 - J) + \sum_{i=1}^{J} f_i$$



Closed-chain mechanism

Simple closed-chain 4R robot (planar)

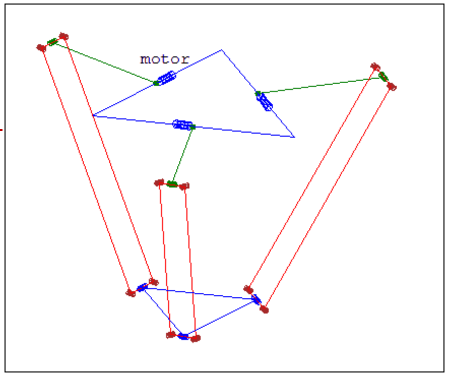
Also called 4-bar linkage

m = 3 (planar)
N = 4 (include ground/base)
J = 4
fi = 1 each

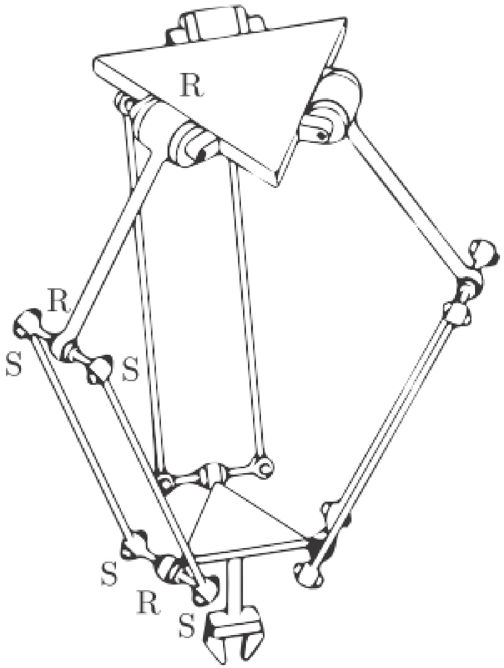
$$dof = 3(4 - 1 - 4) + \sum_{i=1}^{4} 1 = 1$$

Dof example: Delta robot

$$dof = m(N - 1 - J) + \sum_{i=1}^{J} f_i$$



Source: Wikipedia



Source: Modern Robotics

Delta robot (spatial)

3 x RRRSSSS (3 x 3R4S) parallel manipulators

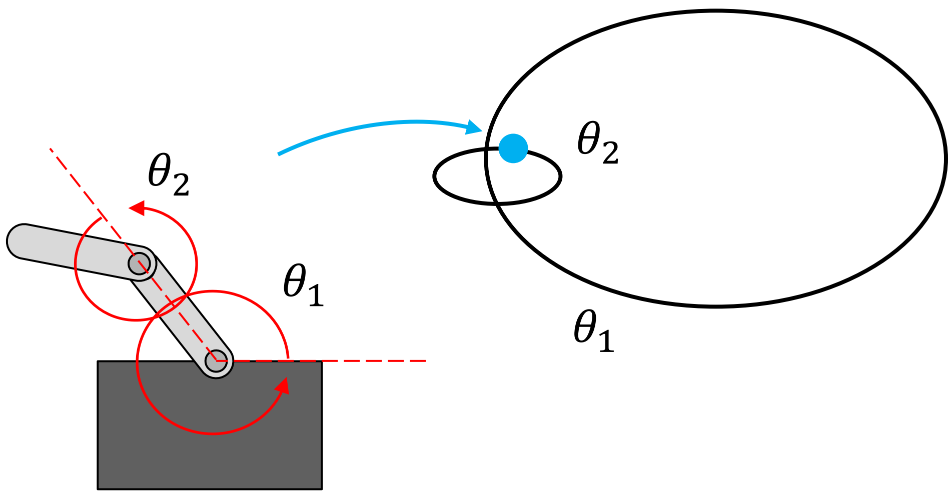
m = 6 (spatial)
N = 5 links each leg, top and bottom platforms
J = 7 each leg
fi = [(1 for R) x 3, (3 for S) x 4] each leg

$$\begin{aligned} dof &\\ &= 6\big((5 \times 3 + 2) - 1 - 7 \times 3\big) + 3(1 \times 3 + 3 \times 4) \\ &= -30 + 45 \\ &= 15 \end{aligned}$$

Note: Each joint connects two links

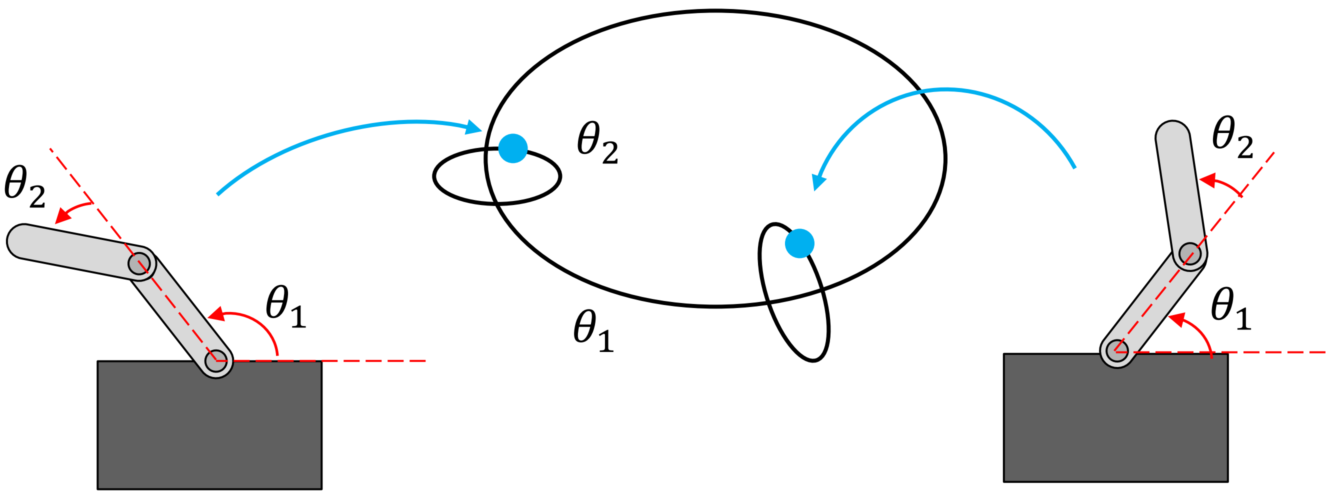
Topology of C-space

- C-space has two properties:
  - **Degree of freedom** (dof): dimension of the C-space
  - **Topology** (shape): shape of the distribution (and relative positions) of the configurations in C-space



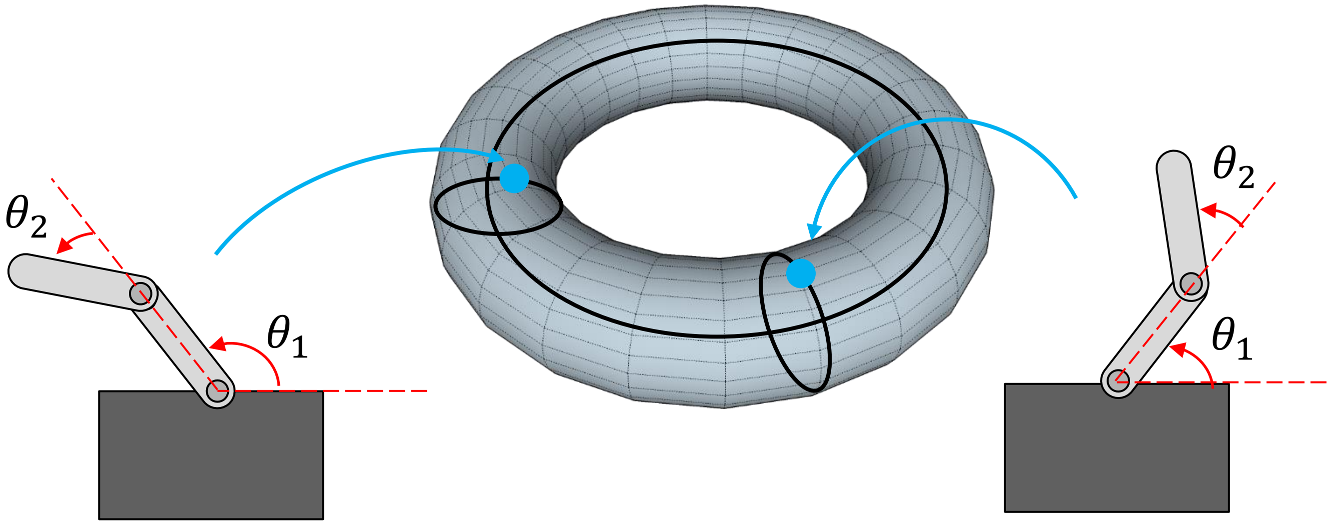
Topology of C-space

- C-space has two properties:
  - **Degree of freedom** (dof): dimension of the C-space
  - **Topology** (shape): shape of the distribution (and relative positions) of the configurations in C-space



Topology of C-space

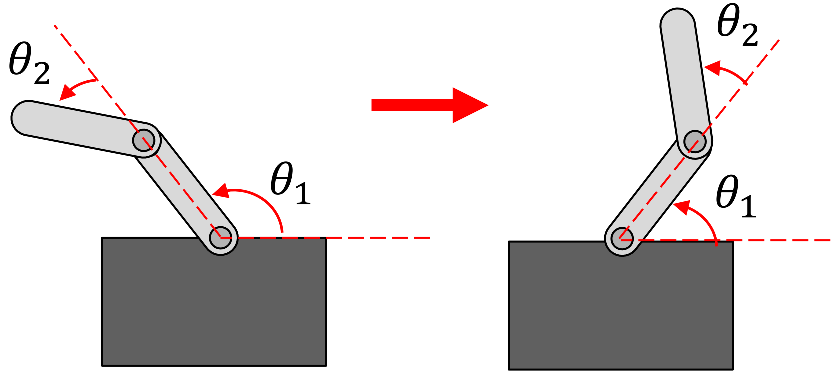
- C-space has two properties:
  - **Degree of freedom** (dof): dimension of the C-space
  - **Topology** (shape): shape of the distribution (and relative positions) of the configurations in C-space

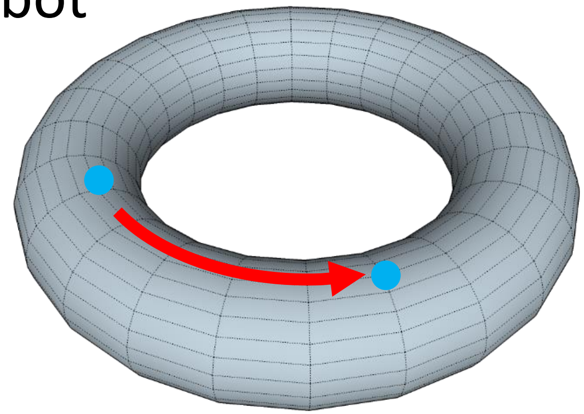


C-space of a 2R planar robot is on the **surface** of the torus (donut) shape.
The shape or **topology** of the C-space is a **torus** (donut).

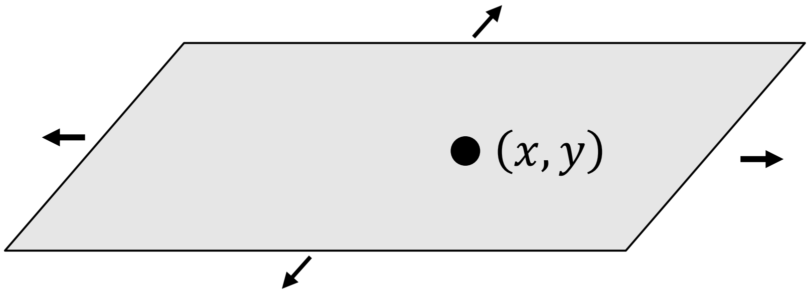
Topology of C-space

- Topology of C-space is useful for planning motion (sequence of configurations to move) of the robot

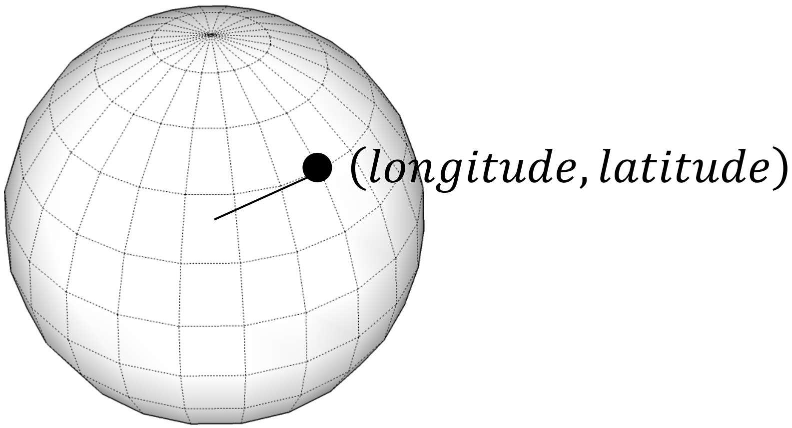




- Two C-spaces may have the same dof with different topology.



Point on a plane

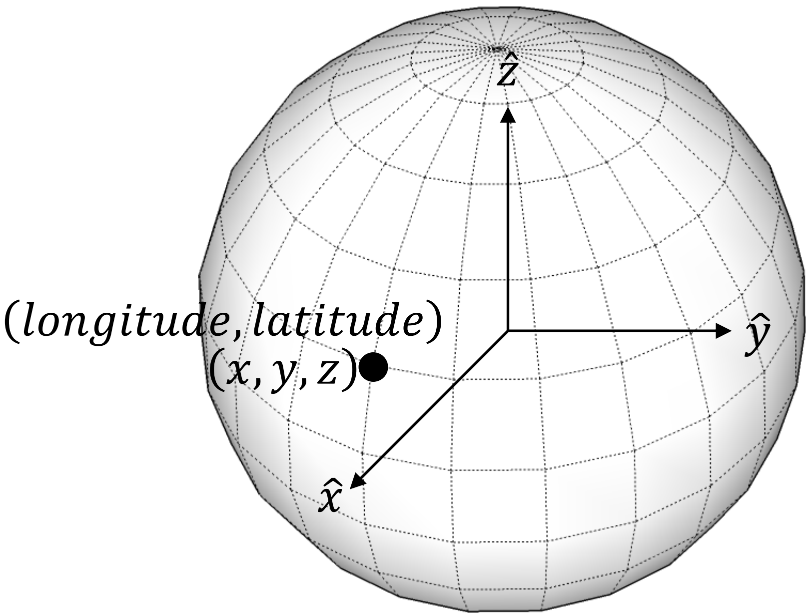


Point on a sphere
(E.g. Spherical pendulum)

Topology of C-space

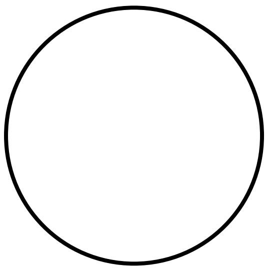
- Topology of two C-spaces are considered equivalent if they can smoothly transform from one to another without cutting and gluing.
- Topology is a fundamental property of a C-space, i.e. fix for a given C-space. However, a C-space can be represented in different ways.

Source: Wikipedia



C-space topologies and representations e.g.

Topologically distinct one-dimensional spaces

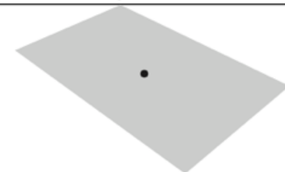
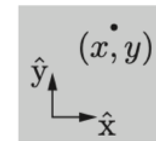
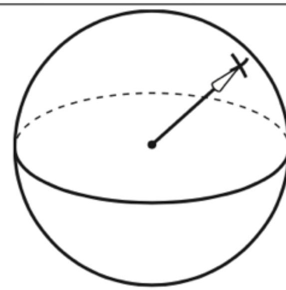
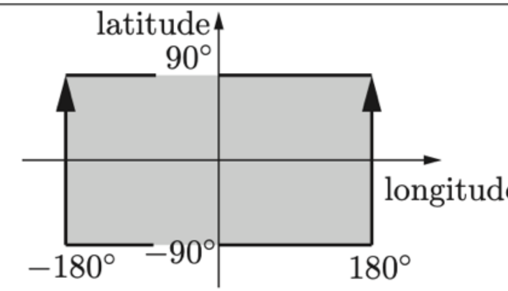
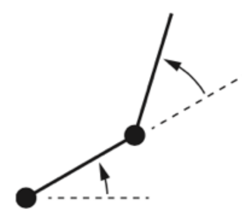
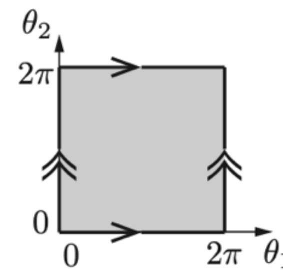
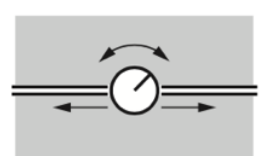
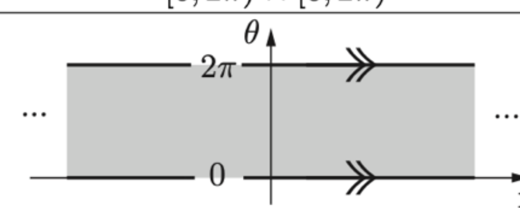


S or S^1

$\mathbb{E}$ or $\mathbb{E}^1$ or $\mathbb{R}$ or $\mathbb{R}^1$

I: $[a, b] \subset \mathbb{R}^1$

Topologically distinct two-dimensional spaces

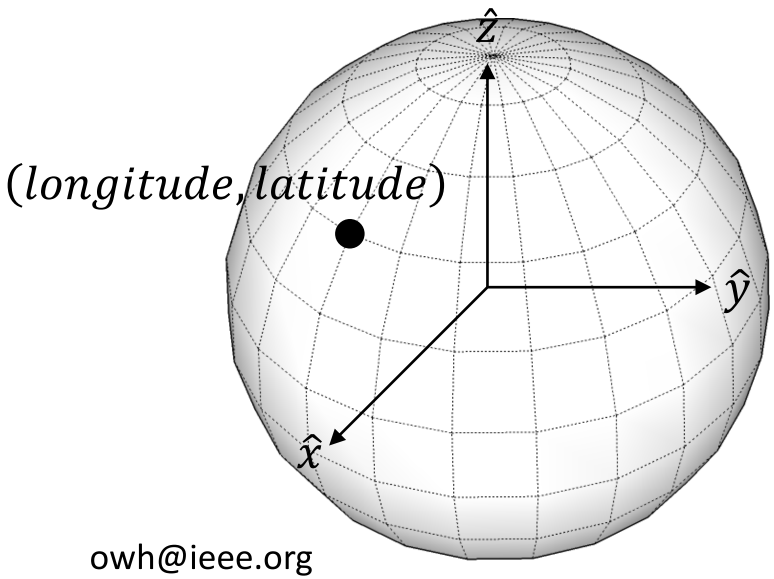
system	topology	sample representation
 point on a plane	 $\mathbb{E}^2$	 $\mathbb{R}^2$
 spherical pendulum	 S^2	 $[-180°, 180°) \times [-90°, 90°]$
 2R robot arm	 $T^2 = S^1 \times S^1$	 $[0, 2\pi) \times [0, 2\pi)$
 rotating sliding knob	 $\mathbb{E}^1 \times S^1$	 $\mathbb{R}^1 \times [0, 2\pi)$

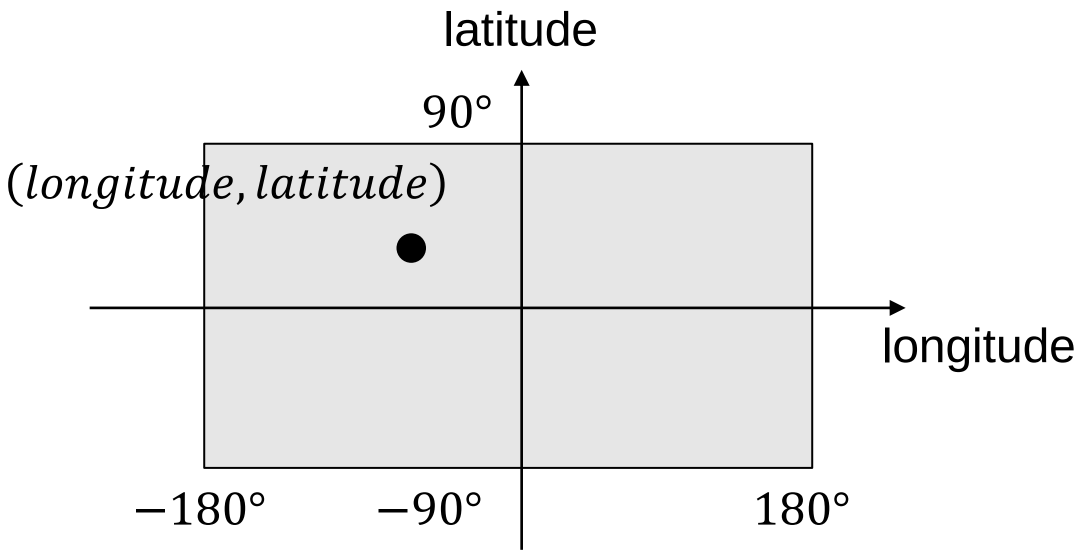
owh@ieee.org

Source: Modern Robotics

Explicit representation

- **Explicit** representation uses minimum parameters (e.g. 2 for 2d), however suffers from **singularities** in cases where the representation has different shape from the topology.
  - A **singularity** in representation is where the values are undefined (e.g. at a point of discontinuity)
  - (Related) A **mechanical singularity** is a position or configuration of a mechanism or a machine where the subsequent behavior cannot be predicted (undefined) – movement blocked in a certain dimension.

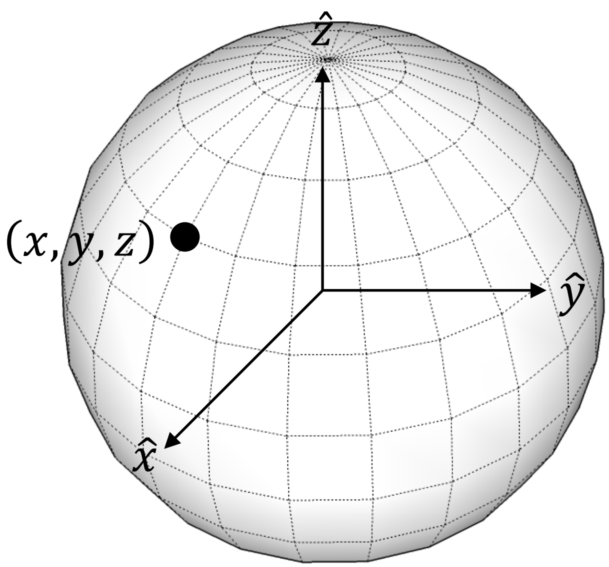


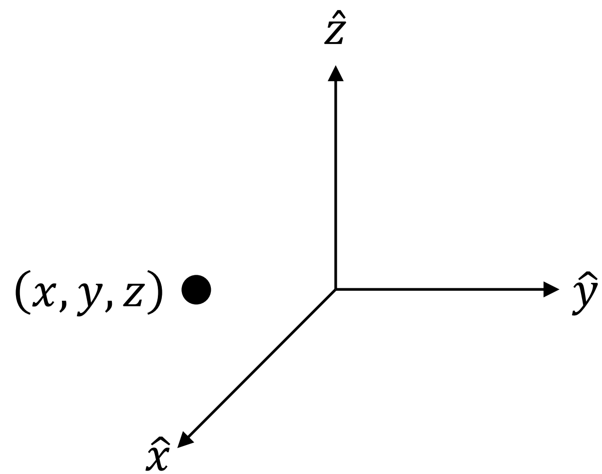


[Explicit representation]()

Implicit representation

- **Implicit** representation embed the C-space in **higher dimension** Euclidean space and impose constraints to restrict the dimension of the representation.
  - **Implicit** representation avoids **singularities**, however, is more complicated (not the simplest representation).



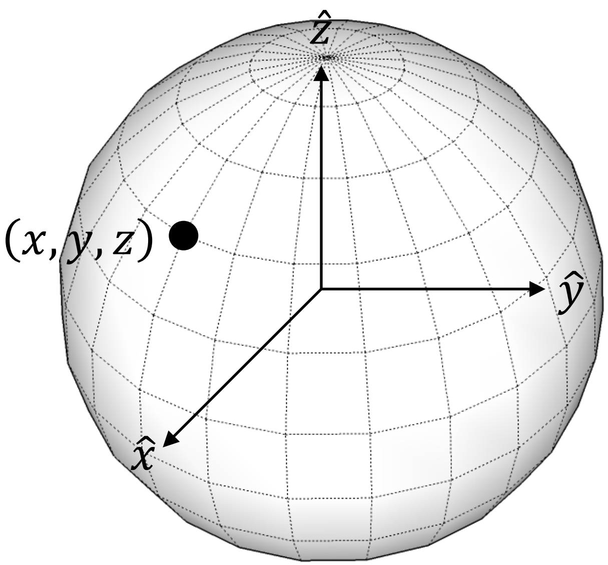


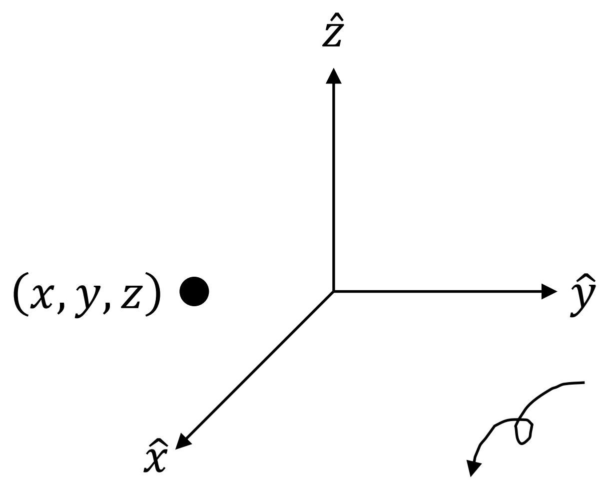
Constraint: $x^2 + y^2 + z^2 = r^2$
(Sphere equation)

[Implicit representation]

Configuration (Holonomic) constraints

- **Configuration constraints** reduce the dimension of C-space. In this case, it reduces the dimension of the representation C-space to that of the actual C-space.
- They are also called **Holonomic constraints** as they reduces the dof.





Configuration constraint

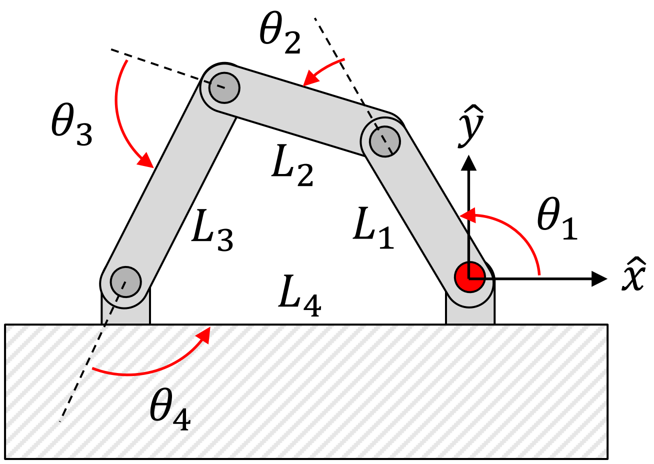
Constraint: $x^2 + y^2 + z^2 = r^2$

$$Actual\ C\ space\ dimension\ (DOF) = n - k\ where\ k \leq n$$

n=no. of parameters of the C-space, *k*=no. of constraints

E.g. For *n*=4 parameters, *k*=3 constraints, DOF = 4 – 3 = 1

Configuration (Holonomic) constraints



<u>Closed-chain four-bar linkage</u>

- Easier to represent implicit
- Use four angles, i.e. 4 parameters
- System is 1-dof
- Introduce 3 constraints base on:
  - L4 (ground link) always horizontal
  - End of L4 at origin (red dot)

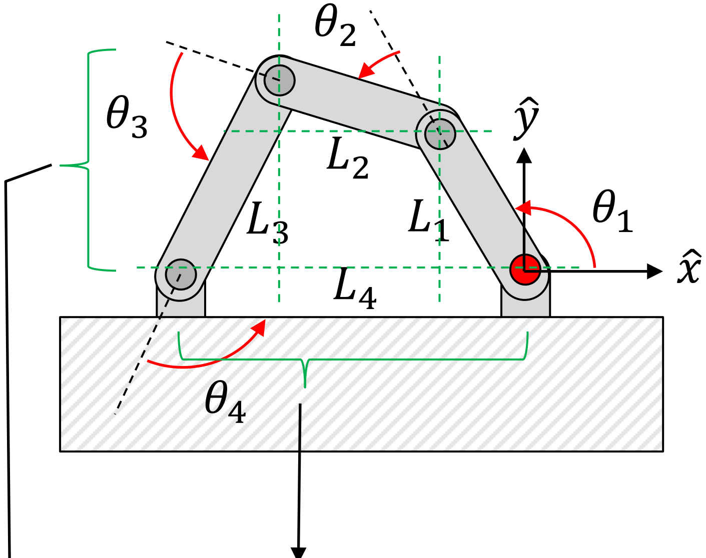
$$L_1 cos\theta_1 + L_2 cos(\theta_1 + \theta_2) + \cdots + L_4 cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) = 0$$

$$L_1 sin\theta_1 + L_2 sin(\theta_1 + \theta_2) + \cdots + L_4 sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) = 0$$

$$\theta_1 + \theta_2 + \theta_3 + \theta_4 - 2\pi = 0$$

(loop closure equations)

Configuration (Holonomic) constraints



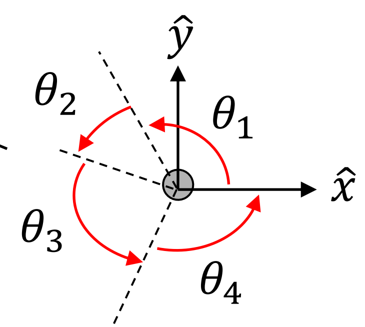
__Closed-chain four-bar linkage__

- Easier to represent implicit
- Use four angles, i.e. 4 parameters
- System is 1-dof
- Introduce 3 constraints base on:
  - L4 (ground link) always horizontal
  - End of L4 at origin (red dot)

$$L_1 cos\theta_1 + L_2 cos(\theta_1 + \theta_2) + \cdots + L_4 cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) = 0$$

$$L_1 sin\theta_1 + L_2 sin(\theta_1 + \theta_2) + \cdots + L_4 sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) = 0$$

$$\theta_1 + \theta_2 + \theta_3 + \theta_4 - 2\pi = 0$$



Actual DOF = 4 parameters – 3 constraints = 1

Configuration (Holonomic) constraints

Let constraint equations be defined as

$$g_1(\theta_1, \theta_2, \theta_3, \theta_4) = L_1 cos\theta_1 + L_2 cos(\theta_1 + \theta_2) + \cdots + L_4 cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) = 0$$

$$g_2(\theta_1, \theta_2, \theta_3, \theta_4) = L_1 sin\theta_1 + L_2 sin(\theta_1 + \theta_2) + \cdots + L_4 sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) = 0$$

$$g_3(\theta_1, \theta_2, \theta_3, \theta_4) = \theta_1 + \theta_2 + \theta_3 + \theta_4 - 2\pi = 0$$

Write in matrix form:

$$g(\theta) = \begin{bmatrix} g_1(\theta_1, \theta_2, \theta_3, \theta_4) \\ g_2(\theta_1, \theta_2, \theta_3, \theta_4) \\ g_3(\theta_1, \theta_2, \theta_3, \theta_4) \end{bmatrix} = 0$$

Generalize:

$$g(\theta) = \begin{bmatrix} g_1(\theta_1, \ldots, \theta_n) \\ \vdots \\ g_k(\theta_1, \ldots, \theta_n) \end{bmatrix} = 0$$

Velocity constraints

If the robot arm moves, we express the parameters (angles) as function of time *t*.

$$\theta(t) = [\theta_1(t) \quad \dots \quad \theta_n(t)]^T$$

The loop closure equations become

$$g\big(\theta(t)\big) = \begin{bmatrix} g_1\big(\theta_1(t), \dots, \theta_n(t)\big) \\ \vdots \\ g_k\big(\theta_1(t), \dots, \theta_n(t)\big) \end{bmatrix} = 0$$

Differentiating both side with respective to *t* gives the velocity of the parameters (joint angles' velocity), we can derive **Pfaffian constraints**

$$\frac{d}{dt} g\big(\theta(t)\big) = 0$$

$$\begin{bmatrix} \frac{\partial g_1}{\partial \theta_1}(\theta)\dot{\theta}_1 + \cdots + \frac{\partial g_1}{\partial \theta_n}(\theta)\dot{\theta}_n \\ \vdots \\ \frac{\partial g_k}{\partial \theta_1}(\theta)\dot{\theta}_1 + \cdots + \frac{\partial g_k}{\partial \theta_n}(\theta)\dot{\theta}_n \end{bmatrix} = 0$$

$$\begin{bmatrix} \frac{\partial g_1}{\partial \theta_1}(\theta) & \dots & \frac{\partial g_1}{\partial \theta_n}(\theta) \\ & \vdots & \\ \frac{\partial g_k}{\partial \theta_1}(\theta) & \dots & \frac{\partial g_k}{\partial \theta_n}(\theta) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} = 0$$

$$\frac{\partial g}{\partial \theta}(\theta)\dot{\theta} = 0$$

Pfaffian constraints:

$$A(\theta)\dot{\theta} = 0$$

Velocity constraints

- **Pfaffian constraints** are **velocity constraints**. It usually uses q instead of θ to generalize to non angular parameters.

$$A(q)\dot{q} = 0 \text{ where } A(q) = \frac{\partial g}{\partial q}(q) \text{ and } \dot{q} = [\dot{q}_1 \quad \cdots \quad \dot{q}_n]^T$$

Note $q = [q_1 \quad \cdots \quad q_n]^T \in \mathbb{R}^n$ is the parameters vector (e.g. joint angles), $A(q) \in \mathbb{R}^{k\times n}$ is differentiation wrt parameters q, $\dot{q} \in \mathbb{R}^n$ is differentiation wrt to time t (e.g. $\dot{q}_i$ is velocity of joint i).

$$g_1(\theta_1, \theta_2, \theta_3, \theta_4) = L_1 cos\theta_1 + L_2 cos(\theta_1 + \theta_2) + \cdots + L_4 cos(\theta_1 + \theta_2 + \theta_3 + \theta_4)$$

$$g_2(\theta_1, \theta_2, \theta_3, \theta_4) = L_1 sin\theta_1 + L_2 sin(\theta_1 + \theta_2) + \cdots + L_4 sin(\theta_1 + \theta_2 + \theta_3 + \theta_4)$$

$$g_3(\theta_1, \theta_2, \theta_3, \theta_4) = \theta_1 + \theta_2 + \theta_3 + \theta_4 - 2\pi$$

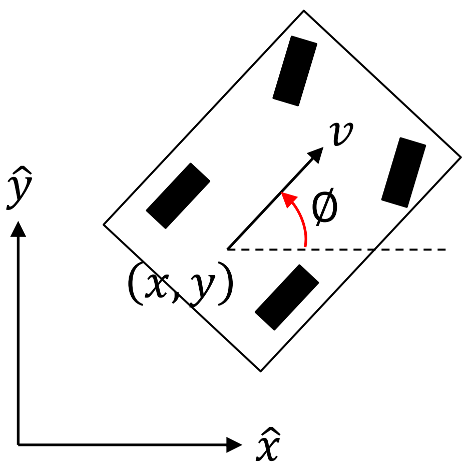
Holonomic and nonholonomic constraints

- If velocity constraints (Pfaffian constraints) can be integrated to equivalent configuration constraints, they are **holonomic**. If not, they are **nonholonomic.**
- **Nonholonomic constraints** reduce the dimension of the feasible **velocities**, but not the dimension of the C-space.

Nonholonomic constraint example

Pfaffian constraint: $A(q)\dot{q} = 0$ where $A(q) = \frac{\partial g}{\partial q}(q)$ and $\dot{q} = [\dot{q}_1 \quad \dots \quad \dot{q}_n]^T$

Consider a car driving on the road (plane). We determine the Pfaffian constraint.



Configuration of the car $q = [\emptyset \quad x \quad y]^T$

Note $[q_1 = \emptyset \quad q_2 = x \quad q_3 = y]$

Velocities in *x* and *y* directions

$$\dot{x} = v\,cos(\emptyset) \qquad \dot{y} = v\,sin(\emptyset)$$

Rearrange $\dot{y}$ and substitute into $\dot{x}$ yields below constraint

$$\dot{x} = \frac{\dot{y}}{sin(\emptyset)}\,cos(\emptyset) \;\rightarrow\; \dot{x}sin(\emptyset) - \dot{y}cos(\emptyset) = 0$$

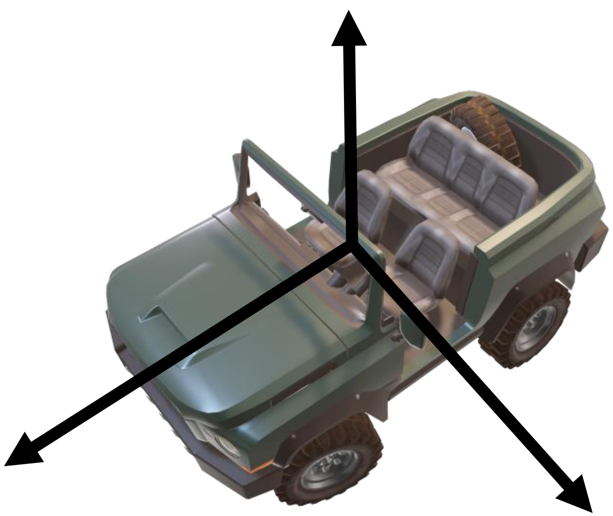
Rearrange above constraint equation in a form we can relate to $A(q)\dot{q}$

$$0.\dot{\emptyset} + sin(\emptyset)\dot{x} - cos(\emptyset)\dot{y} = 0 \quad \Rightarrow \quad 0.\dot{q}_1 + sin(q_1)\dot{q}_2 - cos(q_1)\dot{q}_3 = 0$$

$$[0 \quad sin(\emptyset) \quad -cos(\emptyset)]\begin{bmatrix}\dot{\emptyset}\\ \dot{x}\\ \dot{y}\end{bmatrix} = 0 \quad \Rightarrow \quad \underbrace{[0 \quad sin(q_1) \quad -cos(q_1)]}_{A(q)}\underbrace{\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix}}_{\dot{q}} = 0$$

Holonomic of robots

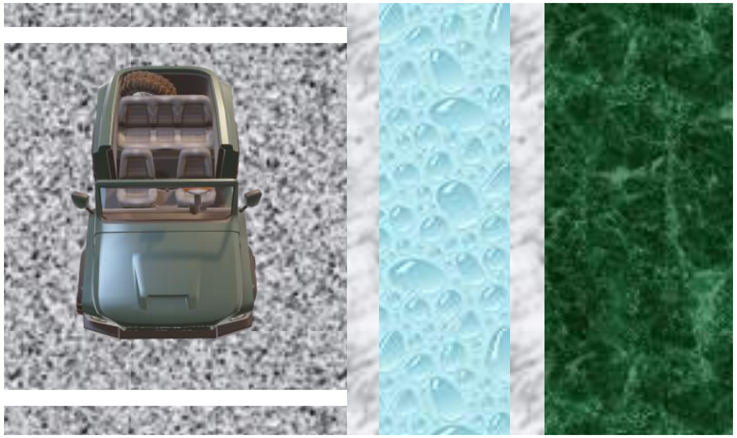
- **Holonomic** refers to the relationship between controllable and total degrees of freedom of a robot.
- One actuator gives one **controllable dof** (CDOF).
  - Not all DOF are controllable.
- Uncontrollable dof makes motion complex, i.e. a body has to take a series of controllable dof to achieve a desired motion.
  - That series of moving the body or effector is called the **trajectory**.



Total degree of freedom (TDOF) = 3 (on plane)
Controllable degree of freedom (CDOF) = 2 (forward/backward, turn)

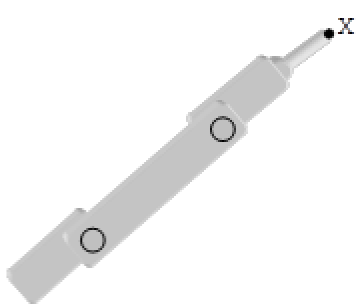
Holonomic of robots

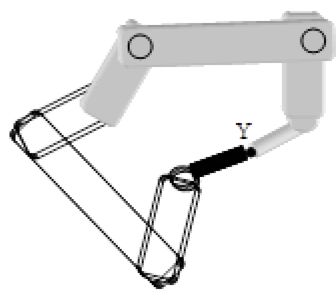
- A robot is **holonomic** if the number of controllable dof is equal to the number of dof of the robot. CDOF = TDOF
- A robot is **non-holonomic** if the number of controllable dof is less than the number of dof of the robot. CDOF < TDOF
- A robot is **redundant** if the number of controllable dof is more than the number of dof of the robot. CDOF > TDOF
- A system or robot may have holonomic **and/or** nonholonomic constraint(s).

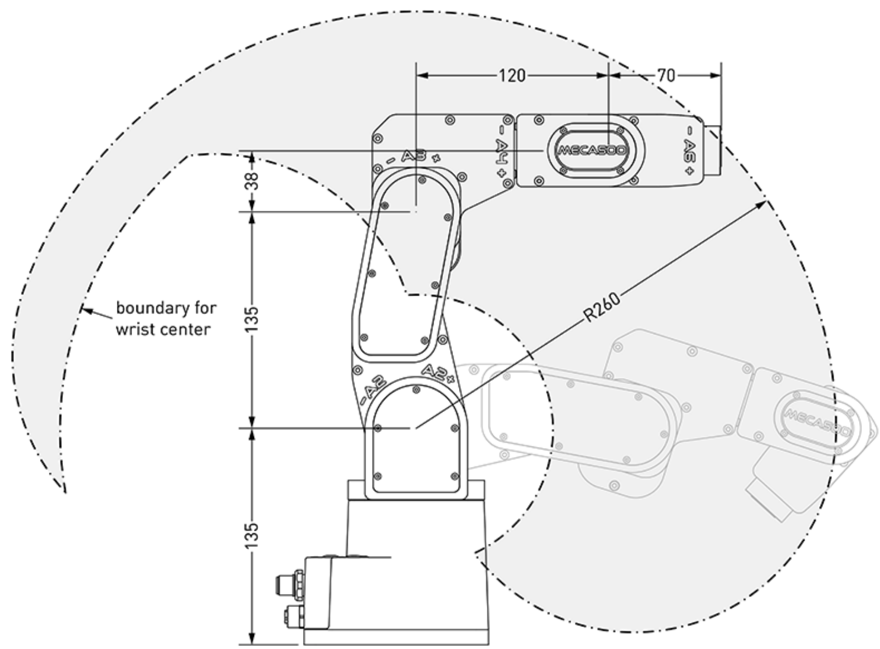


Workspace

- **Workspace**: the volume in space that a robot's **end-effector** can reach
  - **Dextrous workspace** – workspace where the end-effector can reach in any orientation
  - **Reachable workspace** – workspace where the end-effector can reach in at least one orientation
- Depends on robot structure.
- Usually position of end-effector, ignoring orientation.

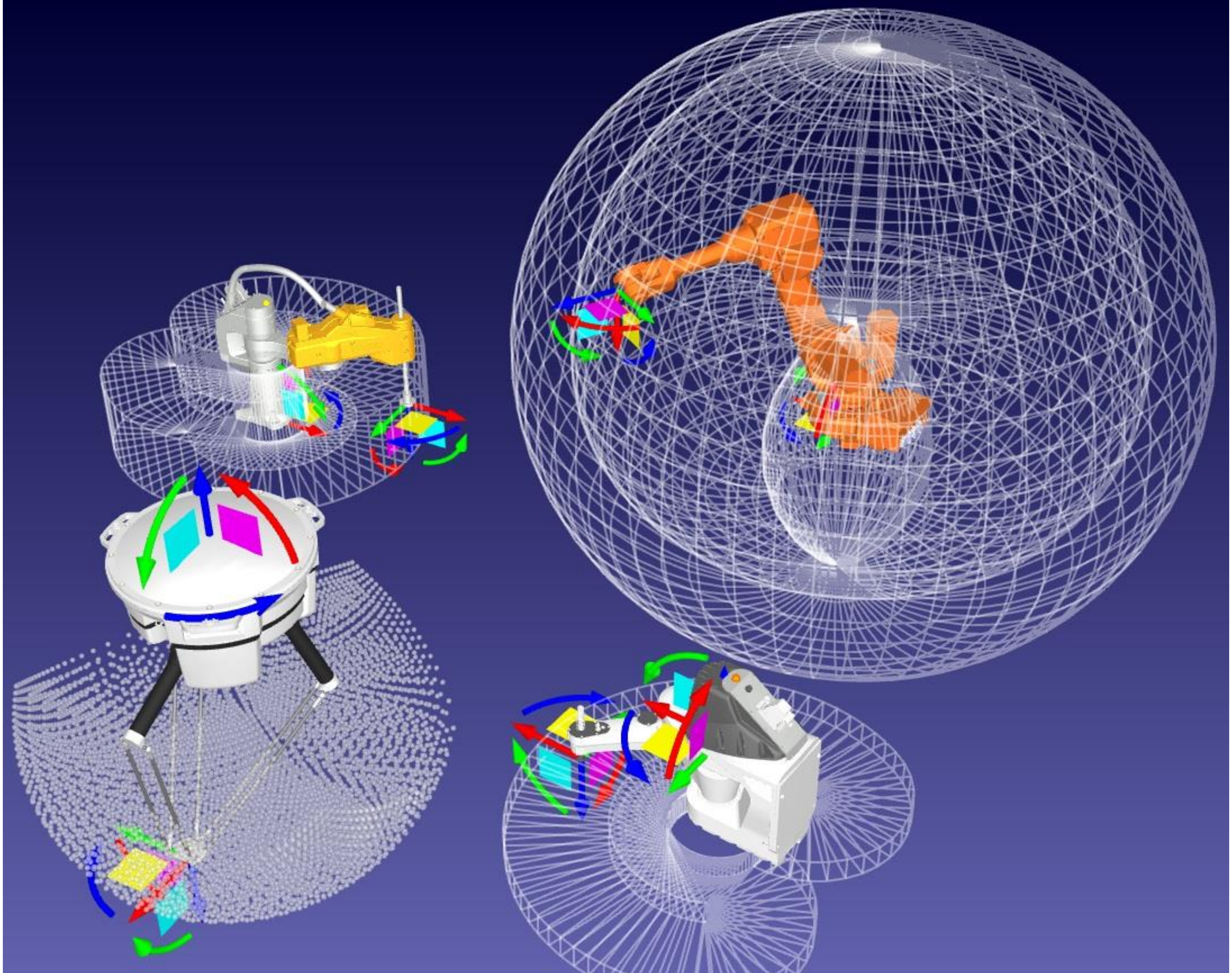






Source: Modern Robotics

A slice of a position-only workspace for a typical 6R robot, with joints limits



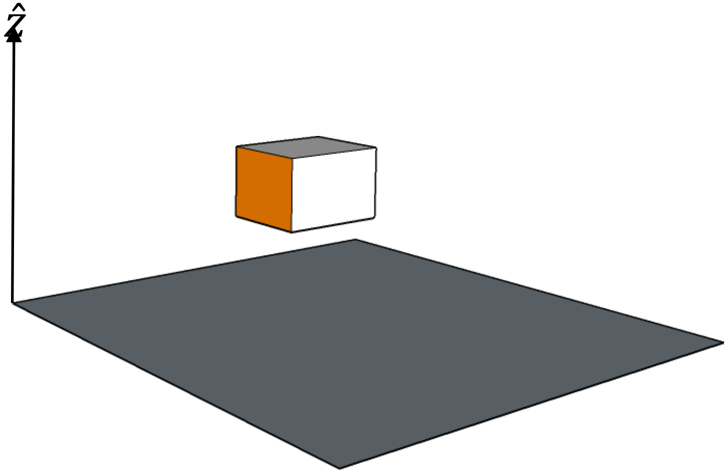
Source: https://robodk.com/doc/en/Tips-Tricks-ShowHide-Robot-Workspace.html

Task space

- **Task space**: a space in which the robot's task can be naturally expressed
  - depends on the task, not depend on the robot structure
  - it is possible that some part of the task space may not be reachable by a robot's C-space

Photo by Kvalifik on Unsplash

2 *dof*, $\mathbb{E}^2$

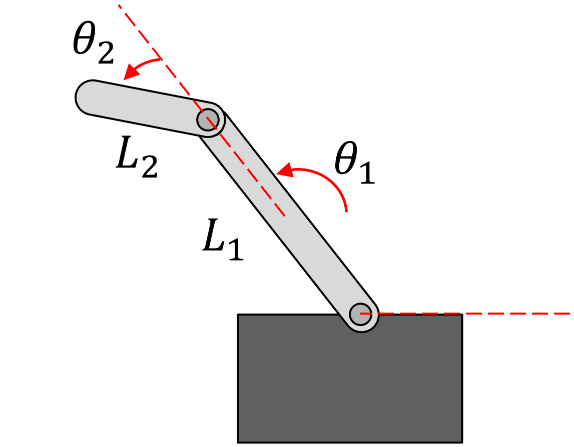


6 *dof*, $\mathbb{E}^3 \times \mathbb{R}^3$

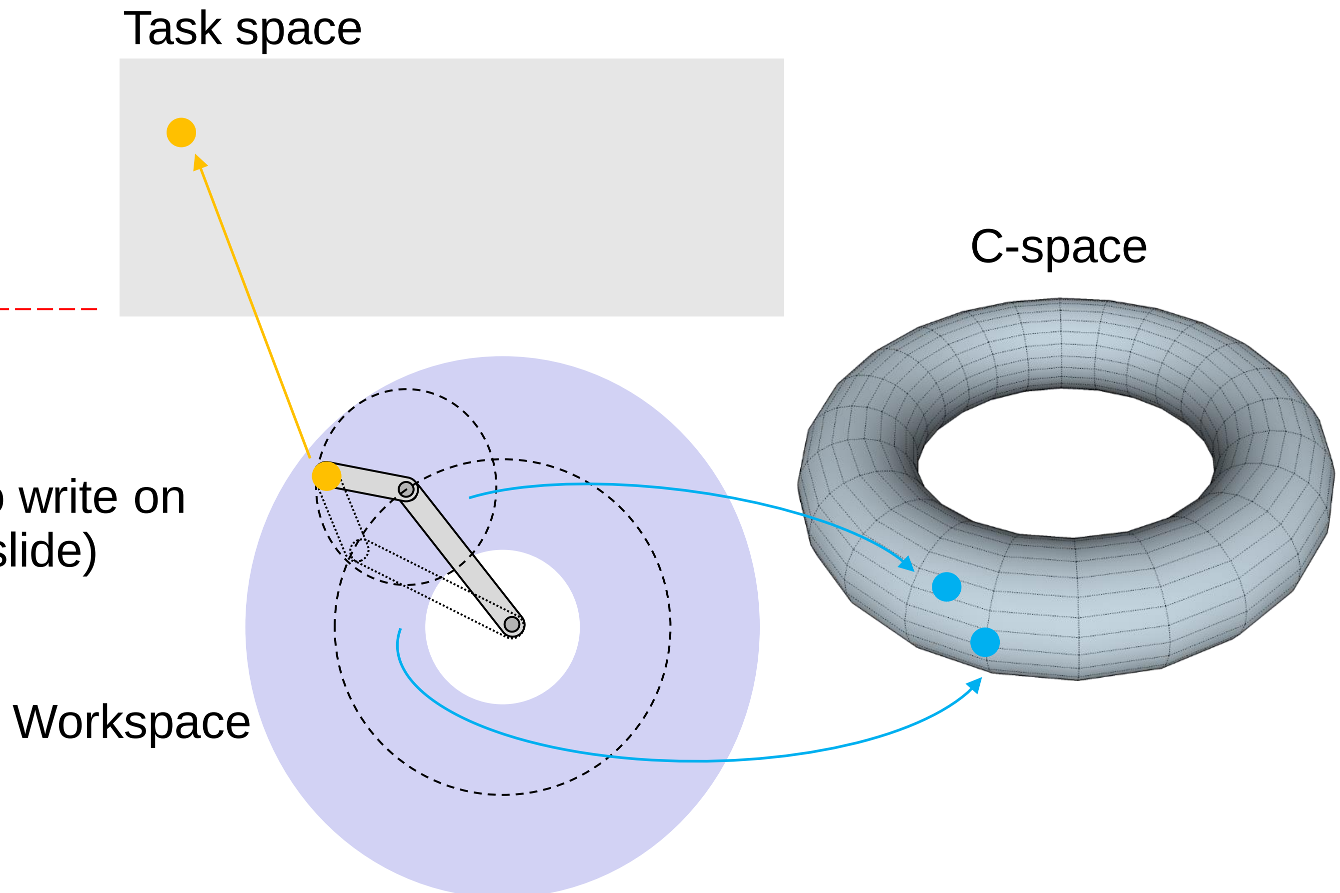
Source: https://hondanews.com

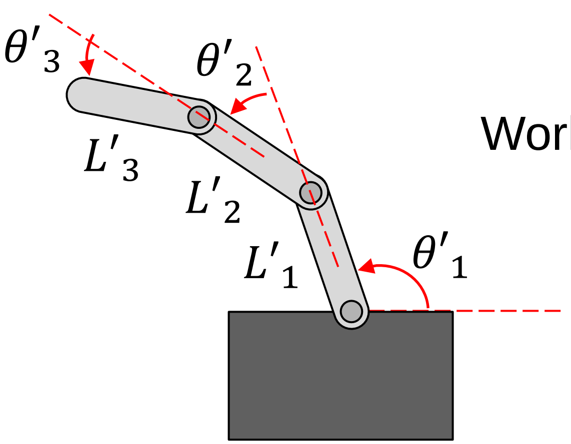
4 *dof*, $\mathbb{E}^3 \times \mathbb{R}^1$

C-space, workspace, task space are different



A 2R planar robot to write on a white board (this slide)





A 3R planar robot

If L’1+L’2=L1 and L’3=L2, the 3R and 2R robots have the same workspace. However, they have different C-space. If they are made to perform the same task, they have the same task space.

Summary (1/2)

- Robot's **configuration**: a specification of the positions of all points of the robot: position and orientation of its bodies.
- Describing **motion** of a robot requires information on its configuration, i.e. position and orientation of its bodies.
- The **configuration space** (C-space) is the space of all configurations of a robot.
- C-space has two fundamental properties: **degree of freedom** (dof) and **topology**.
- We can determine the dof using **Grubler's formula**.
- Robot configurations can be represented **explicitly** or **implicitly**.
- **Explicit** representation is simple however may suffer from **singularity**.

Summary (2/2)

- **Implicit** representation embds the C-space in higher dimension space and impose constraints. It avoids singularity.
- **Holonomic** constraint is constraint on dof of the C-space. **Nonholonomic** constraint is constraint on velocity of the configuration parameters.
- A robot's **workspace** is the volume in space that a robot's end-effector can reach.
- A **task space** is the space in which the robot's task is naturally express.
- C-space, workspace and task space are different.

Reading List

- Chapter 2 of Modern Robotics

To Do List

- Watch Chapter 2 videos of Modern Robotics on Coursera, or on YouTube https://www.youtube.com/playlist?list=PLggLP4f-rq02vX0OQQ5vrCxbJrzamYDfx