اصول علم ربات – جلسه سوم

Fundamentals of Robotics – Lecture 03

Pose, Configuration Space, Constraints, DOF, Representations 1

دکتر مهدی جوانمردی

زمستان ۱۴۰۰

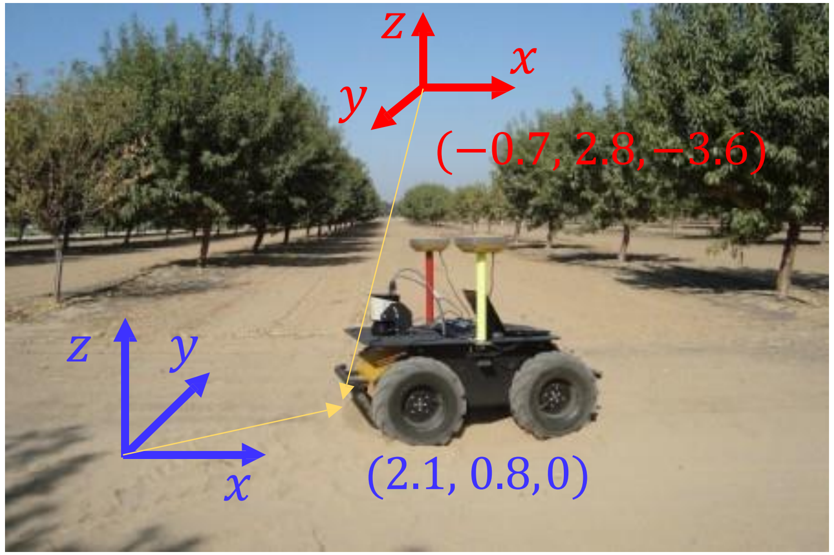
[slides adapted from Gianni Di Caro, @CMU with permission]

A robot is an embedded agent: we need its **Pose**

➢ A robot is embedded in a physical environment, its actions and their effects strictly depend on where the robot is precisely located in the environment and how it is oriented with respect to the surrounding elements

Pose: *position + orientation* of the robot, relative to some reference system

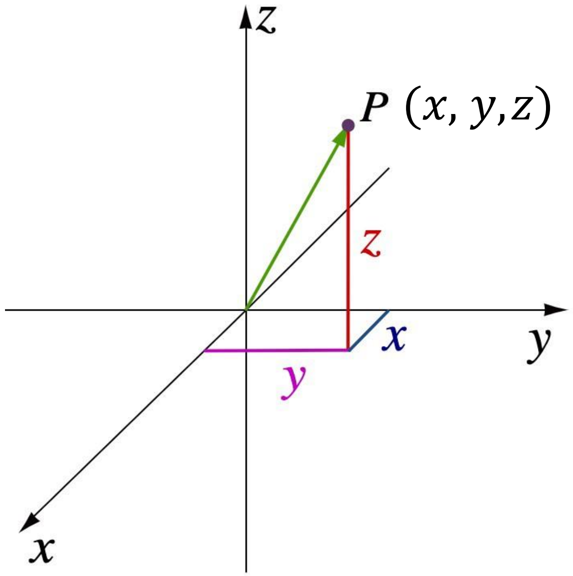
- Any description of a pose (of a robot, of an object) is made in relation to a selected coordinate frame, that in turn requires the selection of a **coordinate system** (a system of numbers, coordinates, to represent positions of points)



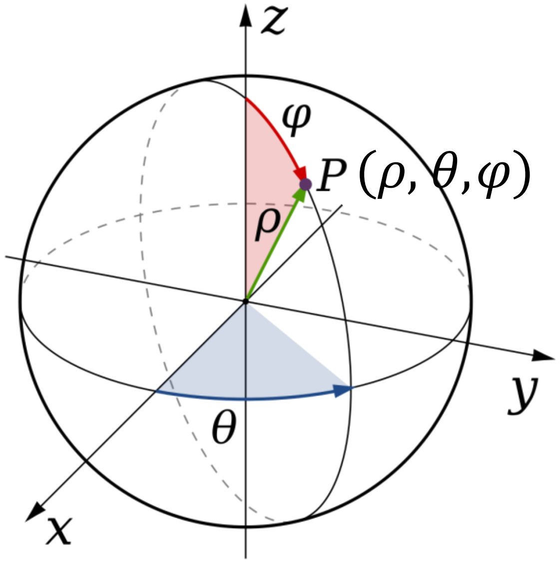
Examples of coordinate systems

- Any description of a pose (of a robot, of an object) is made in relation to a selected coordinate frame, that in turn requires the selection of a **coordinate system** (a system of numbers, coordinates, to represent positions of points)

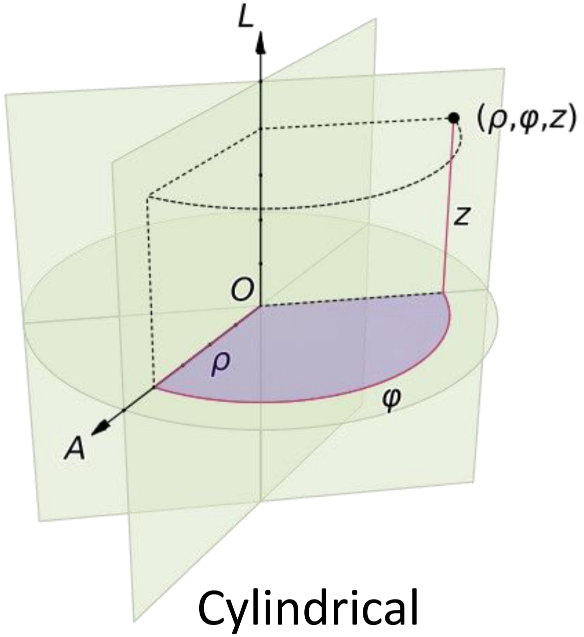
For a **point** (object/robot):



Cartesian



Spherical



Cylindrical

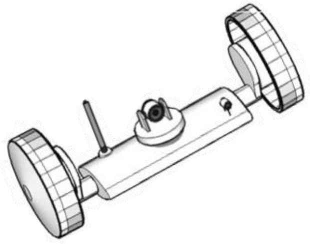
Rectilinear coordinate system

Curvilinear coordinate systems

Points vs. Robots

Point abstraction

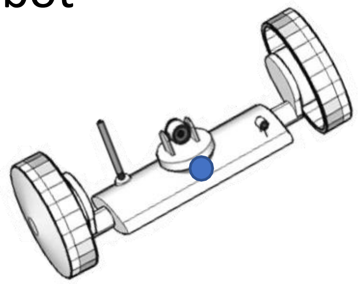
vs. Robot/Object



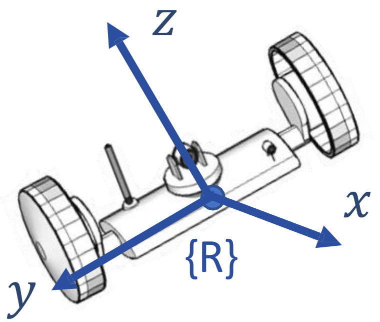
✓ Extended body

✓ Orientation

Select one **reference point** on the robot

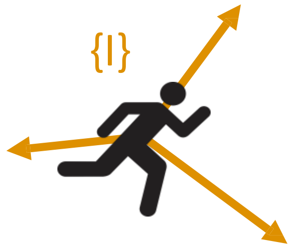


Attach a coordinate system to the reference point

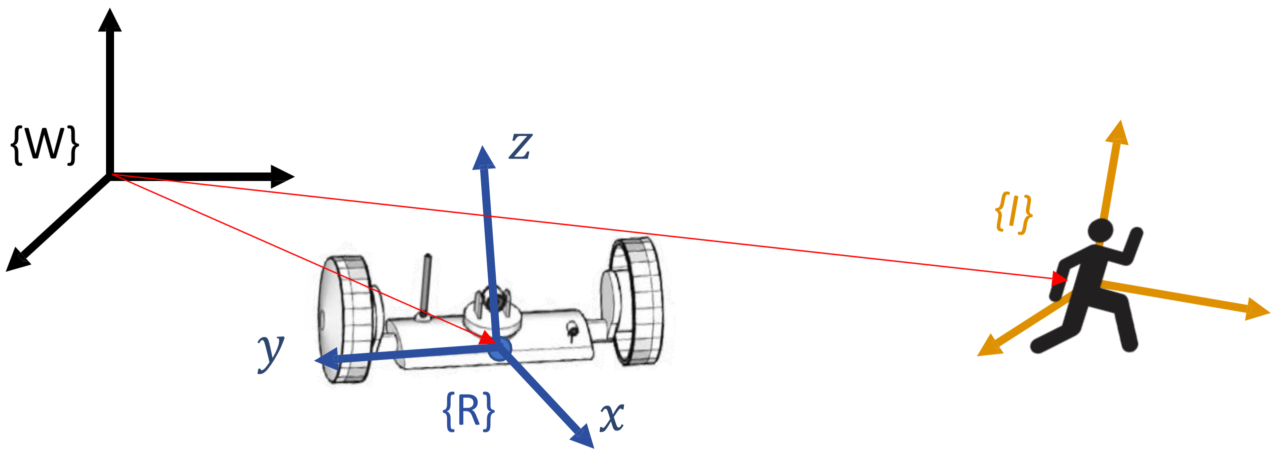


Oriented along some natural orientation of the robot

Can do the same with every object with parts/orientation



Robot / Object coordinate frames



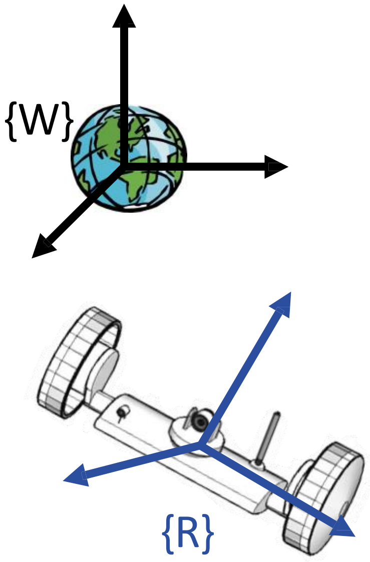
- Express the coordinates of the coordinate frame of each robot / object in some *world* reference frame {W}

- Three reference frames: {W}, {R}, {I}

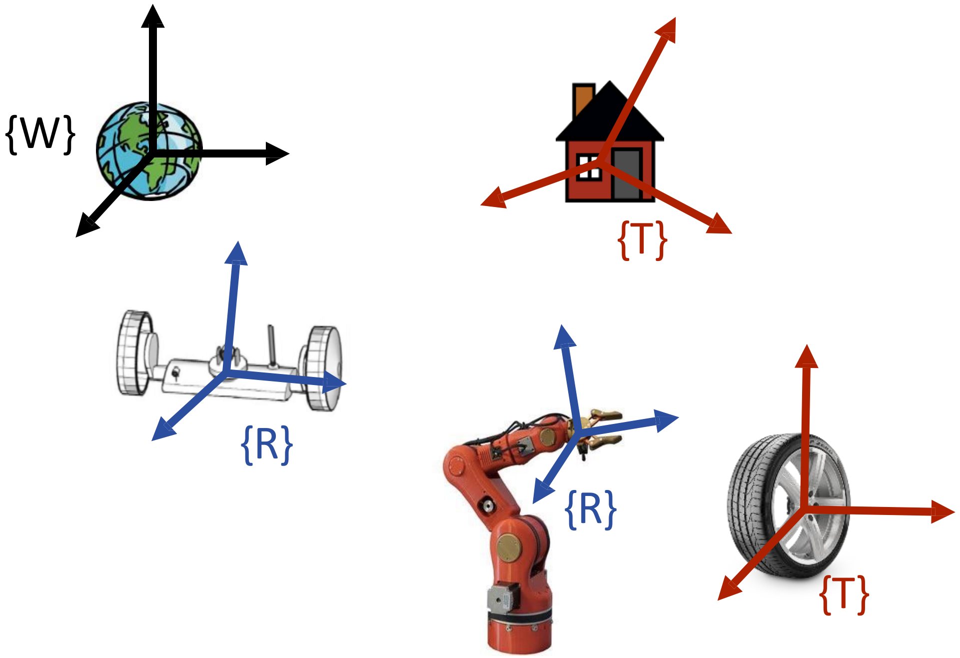
- Can use only the origin to represent the relative position (point abstraction)

Representing & computing (relative) pose: core problem in robotics

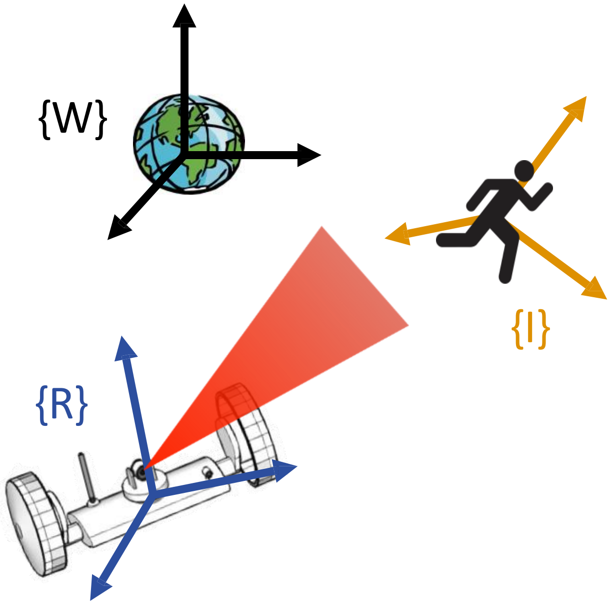
Where's the robot? What is robot's pose with respect to the world reference frame {W}?



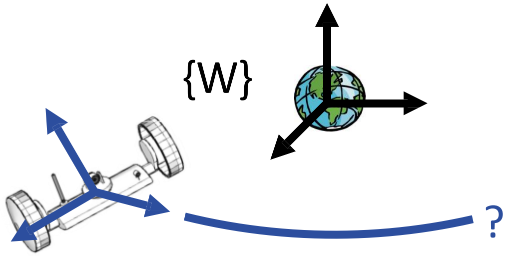
Robot's pose with respect to the external frame {T} (e.g., a target) ?



What is intruder's pose, observed using robot's lateral camera (local frame), in the world frame {W}?



Predict: What is robot's pose in {W} after moving at a velocity v for 1 minute?



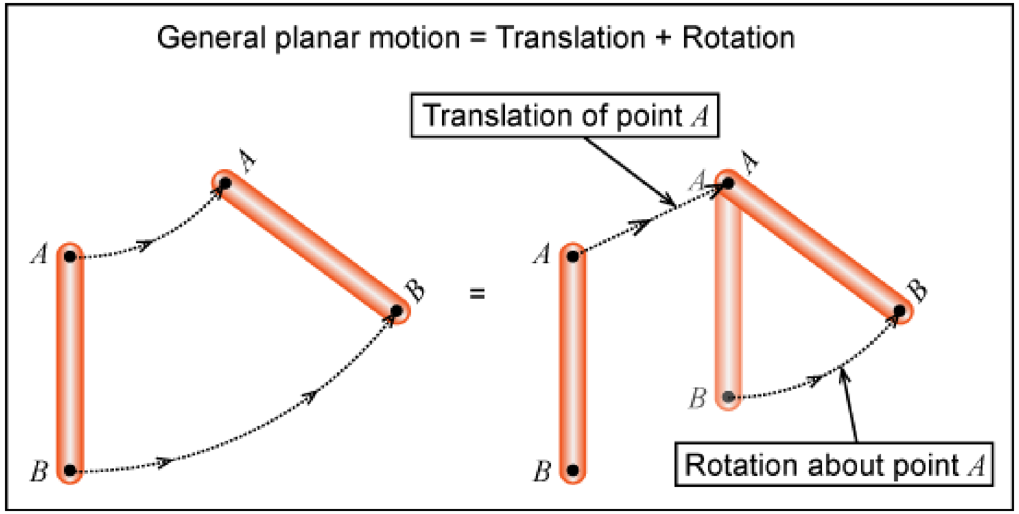
Plan: What is the velocity profile that allows to reach a pose ξ in {W}?

Rigid body assumption (in this course)

- Pose: Position and orientation of *all* robot points…

Robot ~ **Rigid (multi-)body**:

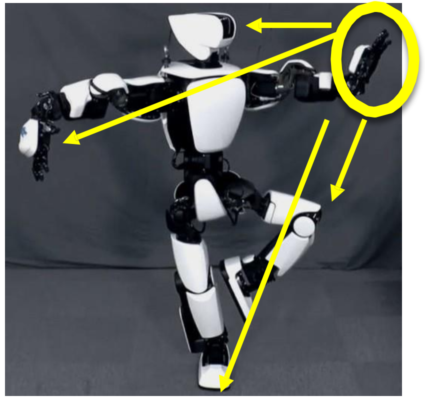
- The robot system is made of rigid parts (links), where constituent points maintain a constant relative position with respect to each other (and to part's coordinate frame)
- This holds under any translation and rotation applied to the rigid body
- **Point abstraction** can be applied to each individual body (link)



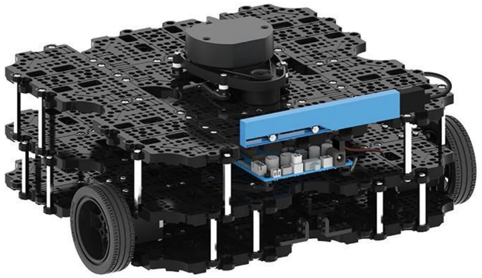
Robots with single vs. multiple rigid bodies

- A robot can have *multiple moving parts/bodies (links)* mechanically connected to each other using various types of *joints*

❖ Multi-body robot: Multiple moving rigid bodies (links) connected by joints



❖ Single (main) body robots

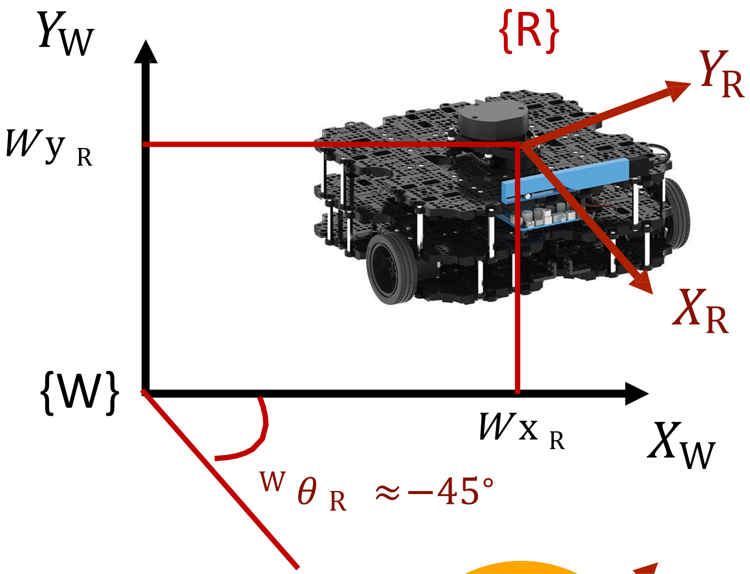


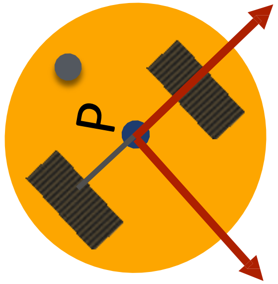
Point abstraction for a rigid body

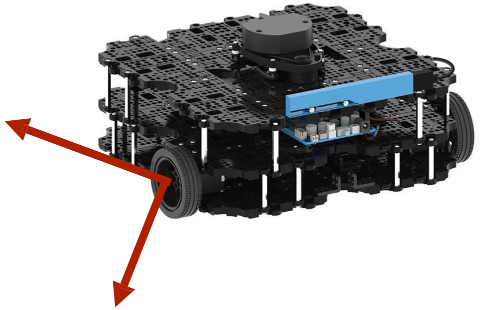
Robot (rigid body) **configuration** = Complete specification of location of every point

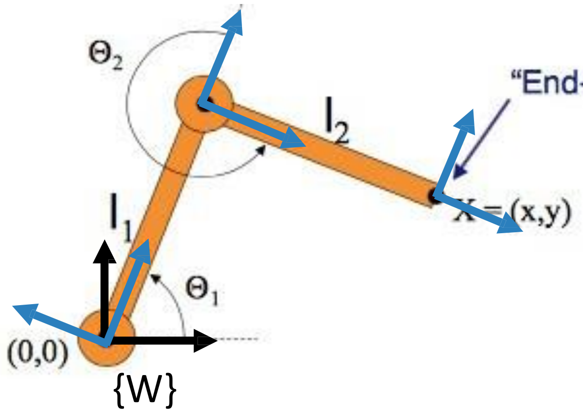
- ✓ It can be computed from a single reference point
- ✓ For pose representation, any *single-body robot shape* can be "reduced" to a point, used as reference point for the origin of a selected coordinate frame

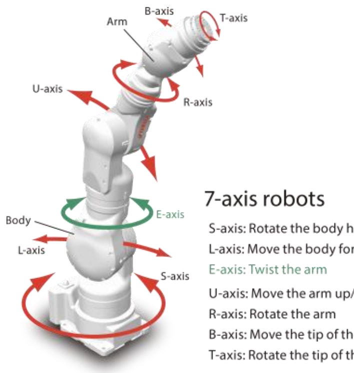
❖ *Any choice* for reference point and frame orientation...







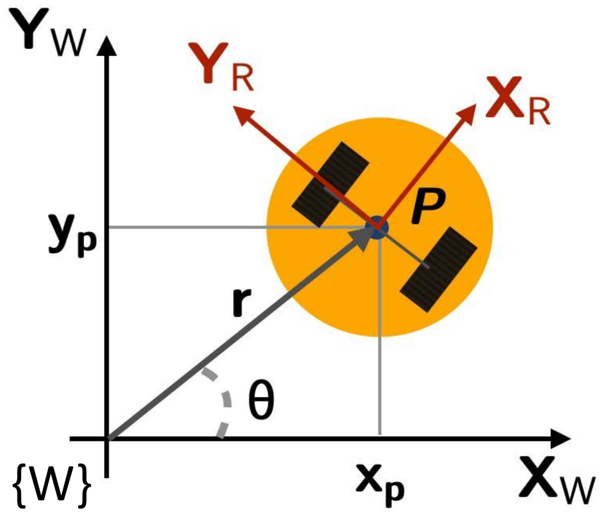




The configuration of a multi-body robot (of known structure) with n moving parts can be represented through n reference points

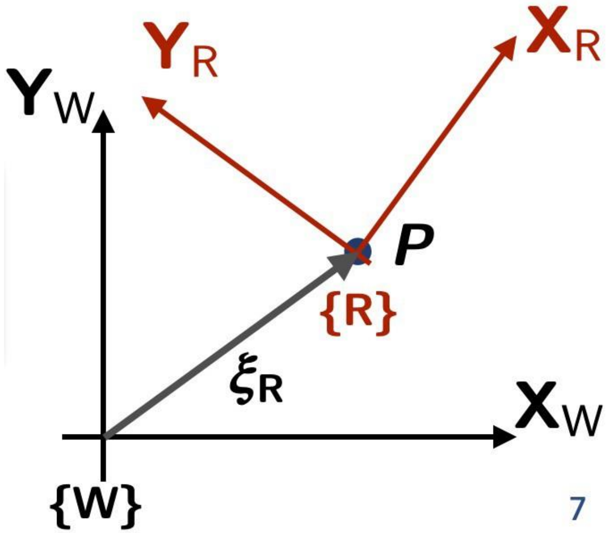
Robot Pose (for single body robots) step-by-step

- Choose a coordinate system, select a fixed world reference coordinate frame {W}
- Center a (local) coordinate frame {R} in a selected robot's reference point *P*, (possibly) oriented according to robot's natural orientation
- A point in space (= chosen reference point *P*) is described by a coordinate vector r representing the displacement of the point with respect to the *reference coordinate frame* {W}
- Different coordinate systems can be used based on convenience: *Cartesian Polar, Spherical, Cylindric,…*



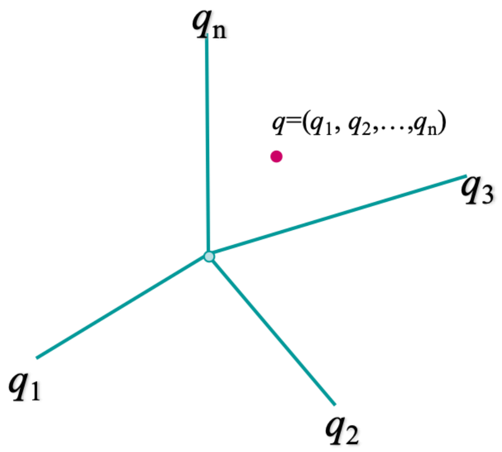
The **(relative) pose/configuration** of the object/robot in {W} is described by the position and orientation of the (local) coordinate frame {R} with respect to {W}

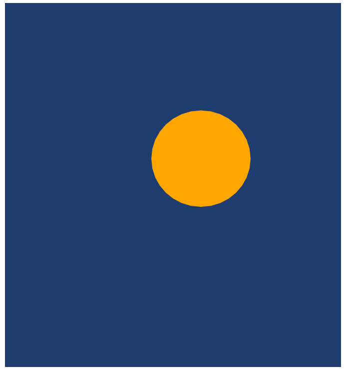
ξ_R is the relative pose of the frame/robot with respect to the reference coordinate frame



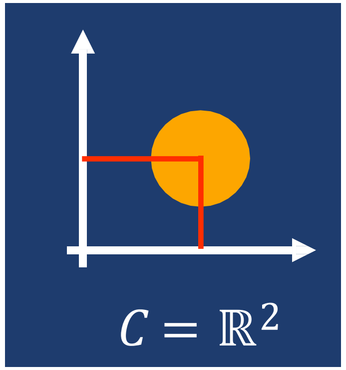
Generalized coordinates, C-Space

- **Generalized coordinates**: n parameters $\boldsymbol{q} = (q_1, q_2, \dots, q_n)$ that are sufficient to uniquely describe system configuration relative to some reference (frame, configuration)
- **Generalized velocities**: The time derivatives $\dot{\boldsymbol{q}}$ of the generalized coordinates.
- **State of the system**: (Generalized coordinates, Generalized velocities), represented in the phase space

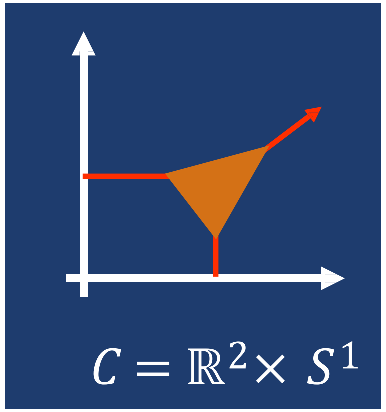




Omnidirectional robot in the plane



$q = (x, y)$



$q = (x, y, \theta)$

Configuration space (C-space): the n-dimensional space identified by the generalized coordinates defining the set of all possible robot configurations (based on robot's structure and environmental constraints). Usually, it is a *non-Euclidean manifold.*

S^1 group, unit circle (read it just for getting the general concept)

- In mathematics, the circle group, denoted by S^1 or $\mathbb{T}$ is the multiplicative group of all complex numbers with absolute value 1, that is, the unit circle in the complex plane or, more simply, the unit complex numbers:

$$S^1 = \{z \in \mathbb{C} : |z| = 1\}$$

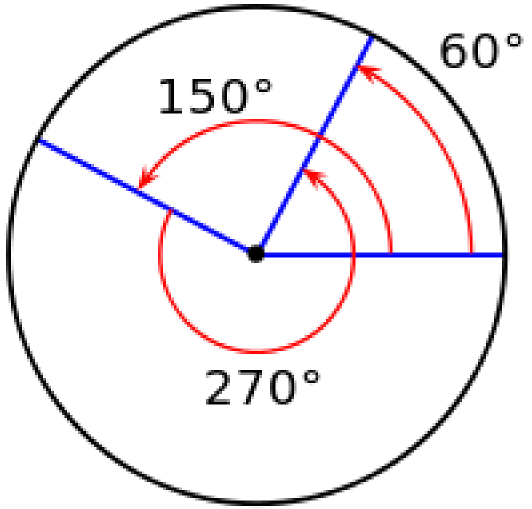
❖ This only for the sake of giving some mathematical details

- The circle group can be parametrized by the angle θ of rotation by

$$\theta \to z = e^{i\theta} = \cos\theta + i\sin\theta$$

This is the exponential map for the circle group.

➢ The circle group describes how to add *angles*, where only angles between 0° and 360° are permitted.
E.g., the diagram illustrates how to add 150° to 270°. The answer should be 150° + 270° = 420°, but when thinking in terms of the circle group, we need to "forget" the fact that we have wrapped once around the circle. Therefore, we adjust our answer by 360° which gives 420° = 60° (mod 360°).

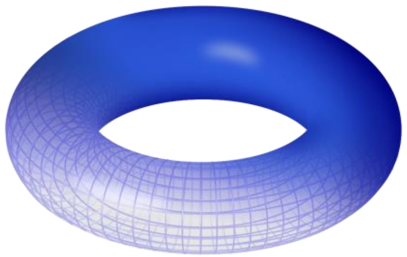
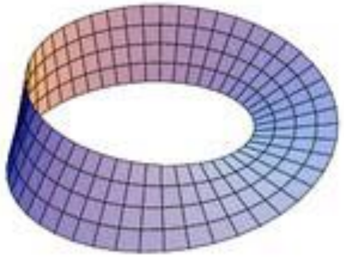
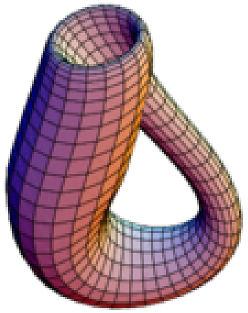
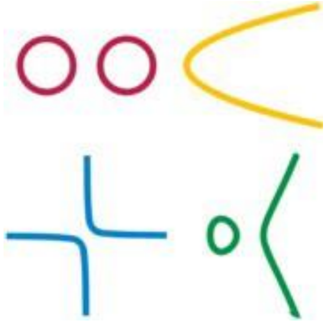


Manifolds (read it just for getting an intuition of the concept)

- A manifold is an abstract mathematical space (such as topological space) in which every point has a **local neighborhood which resembles a Euclidean space**, but in which the global structure may be more complicated (i.e., non-Euclidean, non-metric space)

- In a one-dimensional manifold (or one-manifold), every point has a neighborhood that looks like a **segment of a line**. Examples of one-manifolds include a line, a circle, and two separate circles.

- In a two-manifold, every point has a neighborhood that looks like a **disk**. Examples include a plane, the surface of a sphere, and the surface of a torus.

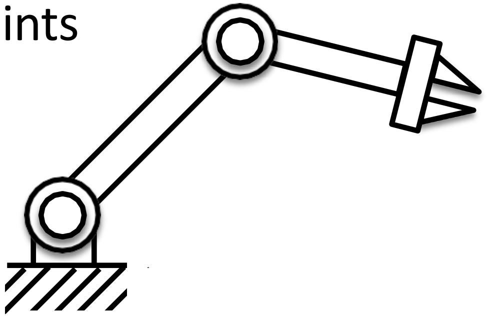





- Manifolds are important objects in mathematics and physics because they allow complicated structures to be expressed and understood in terms of the relatively well-understood properties of local, simpler spaces.

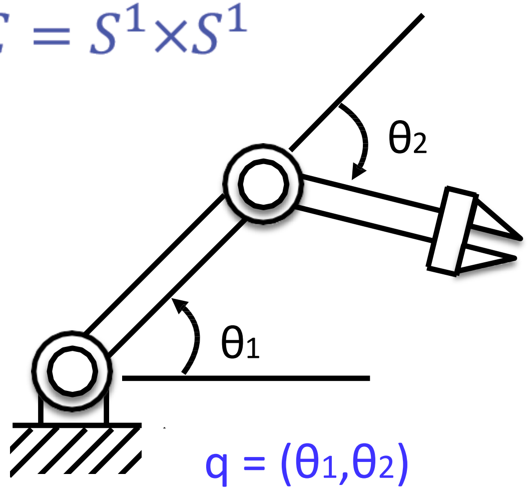
- Additional structures are often defined on manifolds. E.g., in differentiable manifolds one can do calculus, in Riemannian manifolds distances and angles can be defined, symplectic manifolds serve as the phase space in classical mechanics, the 4D pseudo-Riemannian manifolds model space-time in general relativity.

Generalized coordinates, dim of C-Space

Two-link robot arm,
two revolute joints



$C = S^1 \times S^1$

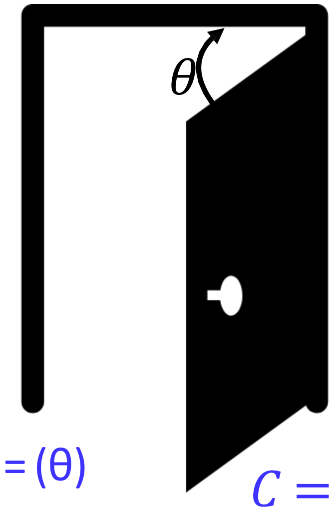


$q = (\theta_1, \theta_2)$

Accounting for joint limitations

$C = S_1 \times S_2, \qquad S_1, S_2 \subseteq S^1$

Door system



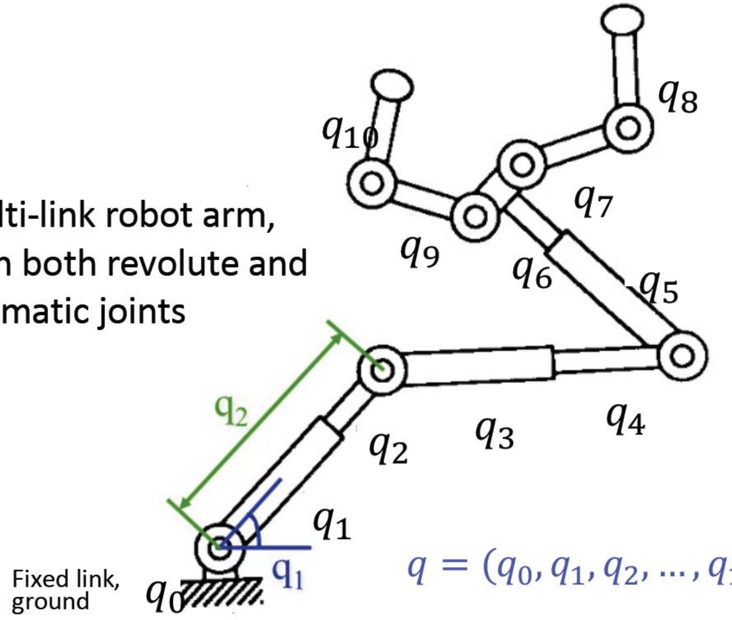
$q = (\theta)$

$C = S^1$

A coin in the plane

$q = (x, y, \theta)$

Multi-link robot arm, with both revolute and prismatic joints

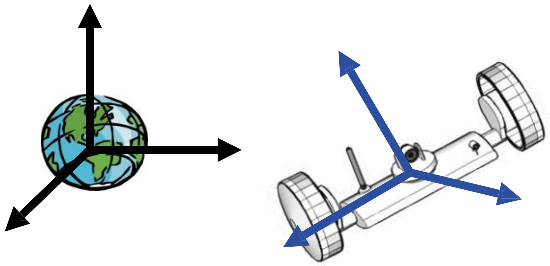


$q = (q_0, q_1, q_2, \ldots, q_{10})$

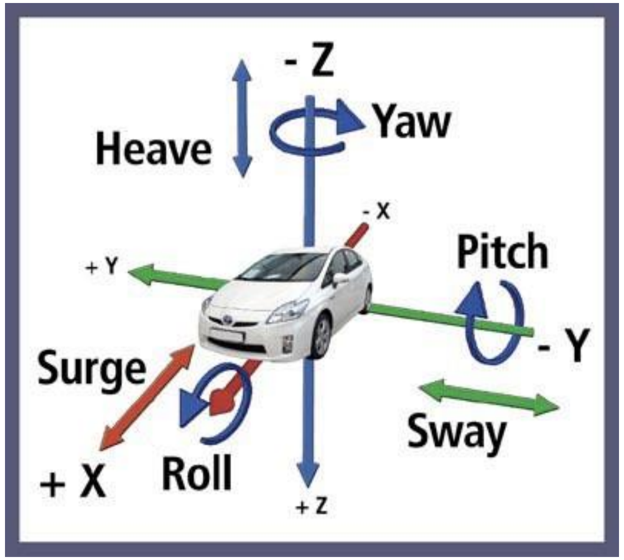
$C = S^1 \times S^1 \times \cdots S^1 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$

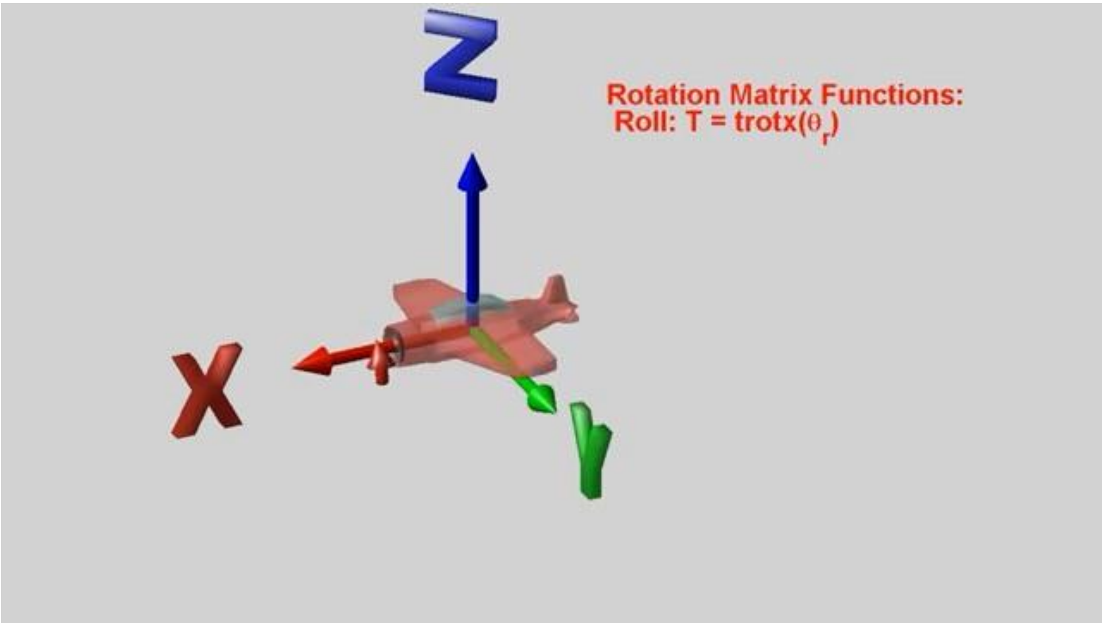
Generalized coordinates, dim of C-Space

A rigid body in 3D



$$\boldsymbol{q} = (x, y, z, \phi, \theta, \psi)$$



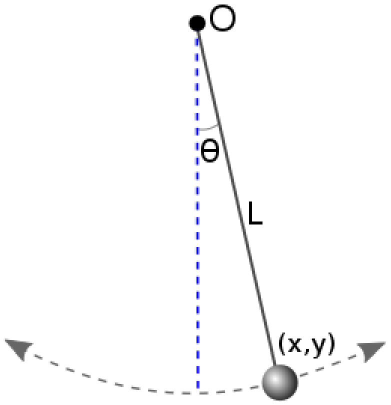


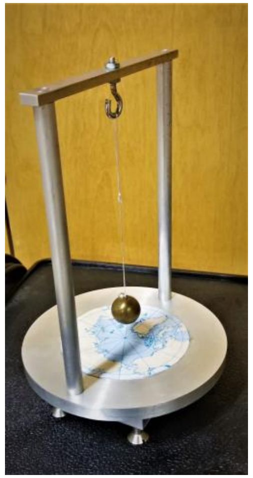
$$C = \mathbb{R}^3 \times S^3$$

Holonomic constraints

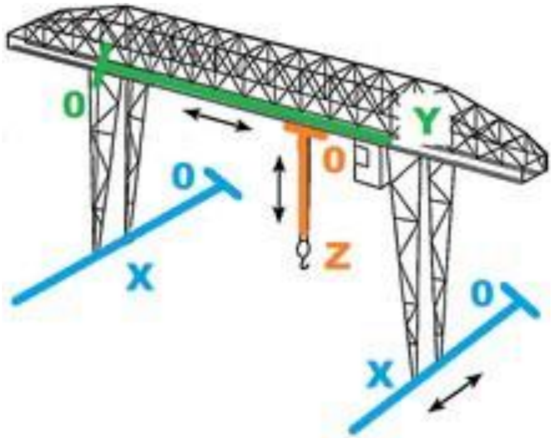
A **geometric constraint** imposes restrictions on the achievable configurations of the robot.
It is based on a functional relation among (some subset of) the configuration variables

Train





Pendulum, robotic arms



Gantry crane

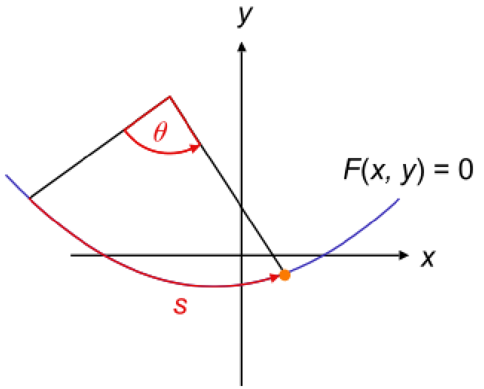
Holonomic constraints

- A *geometric / holonomic constraint* is expressed through "positional" variables, e.g., (α, β, ϕ_1, ϕ_2, x, y, θ, …), it only involves generalized coordinates, not their derivatives
- A holonomic constraint limits *the motion of the system to a manifold of the configuration space*, depending on the initial conditions

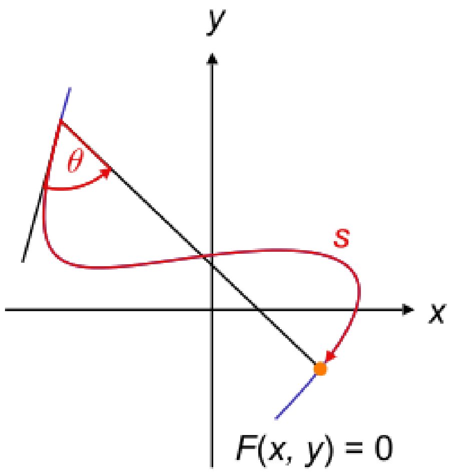
$$f(\mathbf{q}, t) = 0$$

- A constraint *not* depending on time is said scleronomic, rheonomic otherwise. We will focus on holonomic constraints that are scleronomic:

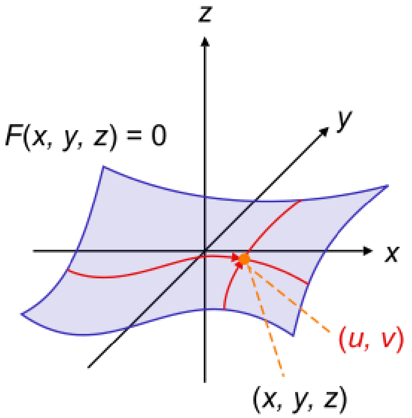
$$f(\mathbf{q}) = 0$$

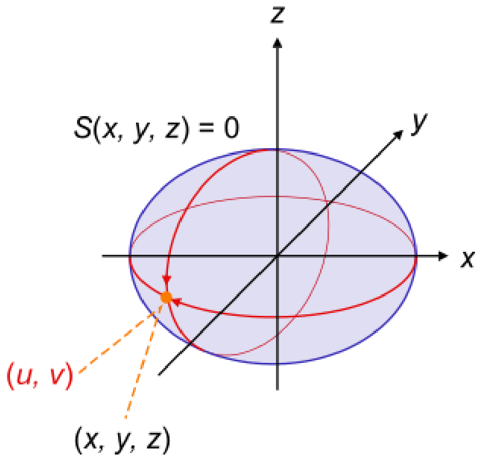


$q = (s)$ or $q = (\theta)$



$q = (s, \theta)$





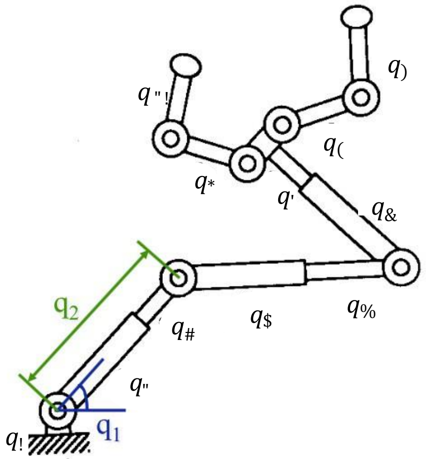
$q = (u, v)$

Need two variables to uniquely identify position in (x, y)

Degrees of freedom (DOF)

Dimension of the configuration space (C-space): (*minimal*) number of independent generalized coordinates that are sufficient to completely describe robot's (rigid body) configuration with respect to some reference coordinate system

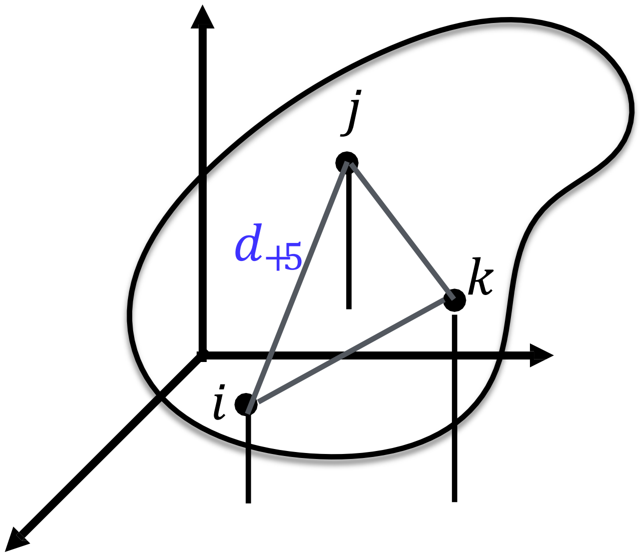
- ➢ Dimension of the C-space defines the number of parameters the robot (we) can independently act upon to change robot's configuration
- ✓ More in general, it says that the state of the system can be changed by operating on n independent variables (we have n available controls on different "dimensions")
- ✓ # freedoms the robot has to change and control its configuration



Degrees of freedom: A system whose configuration is described by n *independent* generalized coordinates has n degrees of freedom.

If there are *m independent functional relations* (holonomic constraints) among a chosen set of n generalized coordinates, the number of DOF is $n - m$: (number of variables - number of independent equations)

Degrees of freedom of a rigid body in 3D



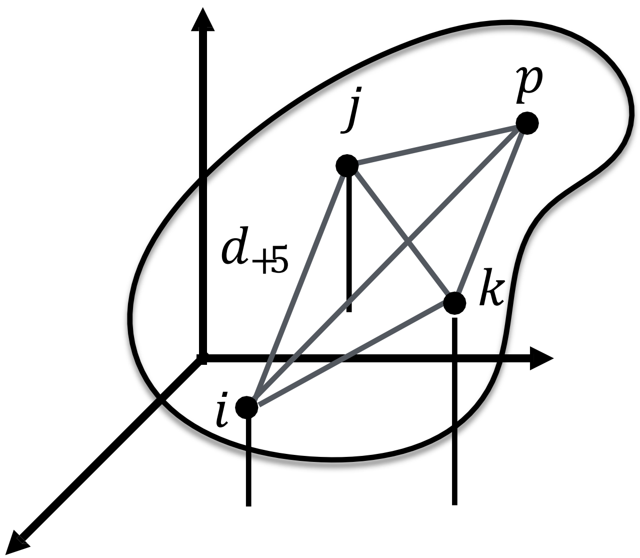
- A rigid body is modeled as a system of *at least* three non-collinear particles whose positions relative to one another remain fixed. i.e., distance d_{ij} between any two particles i and j remains constant throughout the motion (due to internal forces).
- In general, a rigid body is made of $N \gg 3$ particles
- To specify the position (x, y, z) of each particle , we would need $n = 3N$ generalized coordinates

❖ Distance constraint between all pairs of particles (holonomic, scleronomic):

$$d_{ij} = constant_{ij}, \forall i \neq j = 1,2, \ldots N \Rightarrow C_N = \frac{N(N-1)}{2} \text{ constraints}$$

- Are the # of DOF equal to $3N - C_{\mathrm{N}}$? No, not all C_{N} constraints are independent!
- We know that the rigid body has 6 DOF …

Degrees of freedom of a rigid body in 3D

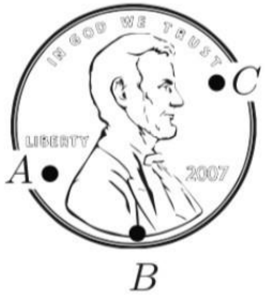


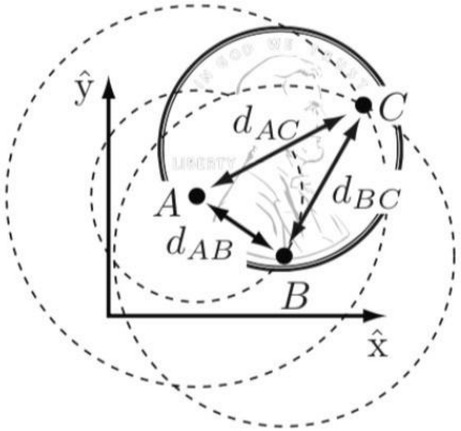
- A system of 3 particles in 3D needs 9 generalized coordinates. There are 3 independent distance constraints → 9 - 3 = 6 DOF

- What about a new particle p? → 3 more coordinates + 3 more constraints → 0 freedoms

- Any additional point would contribute with 3 more coordinates but will determine 3 more independent constraint equations (wrt the original three points, all other distances are fixed depending on these) → 0 freedoms

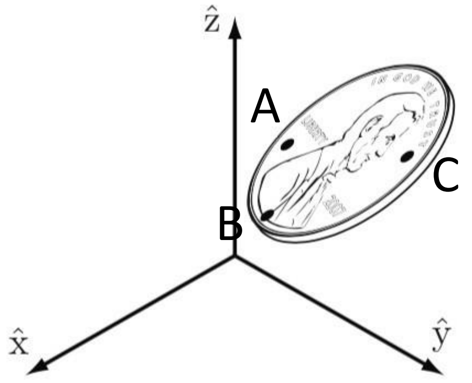
Degrees of freedom of a rigid body: Coin in a plane

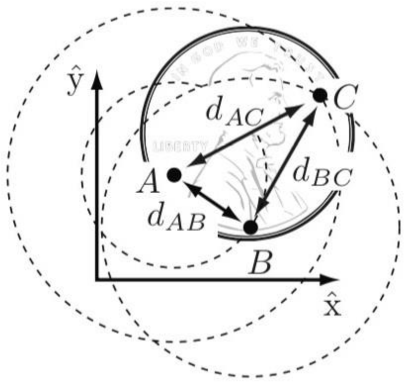




- DOF of a coin in a plane: freedoms choosing three reference points with given distances between them
- Once the location (x, y) of A is chosen (2 freedoms), B must lie on a circle of radius d_{AB} centered at A (1 freedom, angle θ)
- Once the location of B is chosen, C must lie at the intersection of circles centered at A and B → 0 freedom
- The coin in the plane has 3 DOF: (x, y, θ)

Degrees of freedom of a rigid body: Coin in 3D

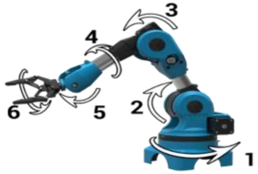




- Point A can be placed freely in the space → **3 freedoms** (x, y, z)
- Location of B is subject to the constraint d_{78}: it must lie on the *sphere* of radius d_{AB} centered at A→ 3-1 = **2 freedoms** (φ, ψ) (e.g., latitude and longitude on the sphere)
- Location of point C must lie at the intersection of spheres centered at A and B of radius $d_{\mathrm{AC}}, d_{\mathrm{BC}}$, respectively
- The intersection of two spheres is a *circle*, that can be parametrized by an angle → 1 **freedom** (θ)
- DOF = 3 + 2 + 1 = 6

DOF and robot control (we’ll see it later)








	dim $\mathcal{C}$	Degrees of freedom	Number of actuators	Actuation	Rolling constraints	Holonomic
Train	1	1	1	full		✓
2-joint robot arm	2	2	2	full		✓
6-joint robot arm	6	6	6	full		✓
10-joint robot arm	10	10	10	over		✓
Hovercraft	3	3	2	under		
Car	3	2	2	under	✓	
Helicopter	6	6	4	under		
Fixed wing aircraft	6	6	4	under		

DOF and robot control

	dim $\mathcal{C}$	Degrees of freedom	Number of actuators	Actuation	Rolling constraints	Holonomic
Train	1	1	1	full		✓
2-joint robot arm	2	2	2	full		✓
6-joint robot arm	6	6	6	full		✓
10-joint robot arm	10	10	10	over		✓
Hovercraft	3	3	2	under		
Car	3	2	2	under	✓	
Helicopter	6	6	4	under		
Fixed wing aircraft	6	6	4	under		

- DOF / dimension of the C-space defines the number of parameters the robot can independently act upon to change its configuration: If there is an actuator for each DOF then each DOF is **controllable**
- If not all DOF are directly controllable the control problems are (much) harder → Underactuation
- The number of controllable DOF determines how easy/hard the robot control problem will be

- Holonomic robots: # of controllable DOF is the same as the # DOF
- Non holonomic robots: # of controllable DOF is lesser than the # of DOF (we don’t have full controls!)
- Redundant robot: # of controllable DOF is larger then # of total DOF (over actuated robot)
- E.g., Human Arm 6 DOF – Position and orientation of the Fingertip in 3D space: 7 actuators - 3 shoulder, 1 elbow, 3 wrist (it would only require 6DOFs)

To be continued ....