

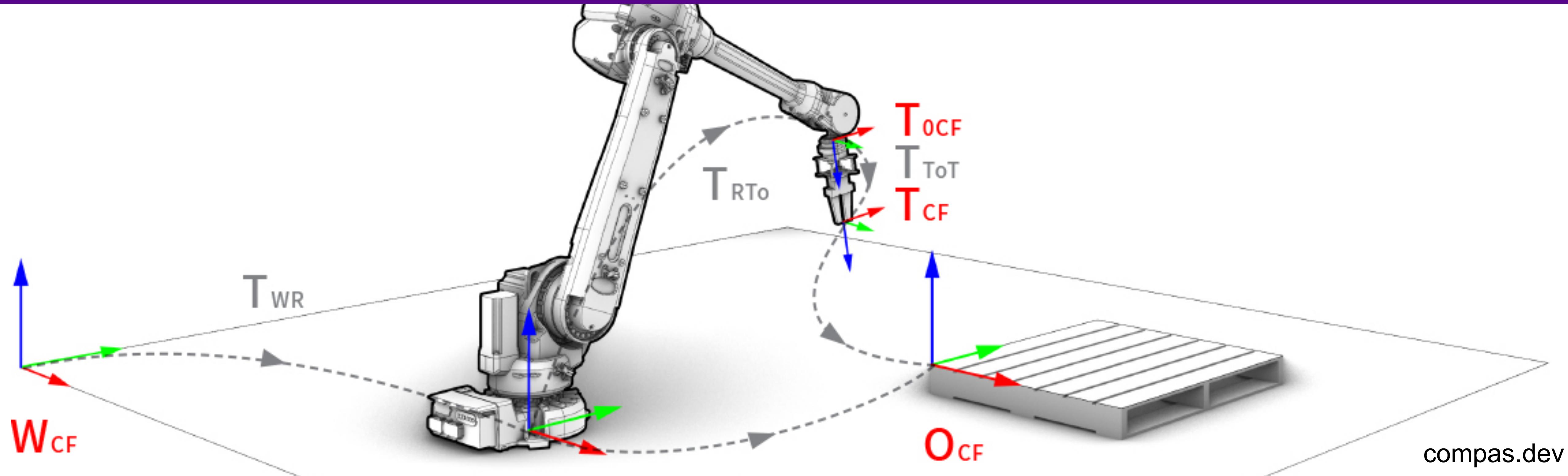


NYU

# ROB-UY 2004

Robotic Manipulation & Locomotion

## Lecture 02B - Mathematical Representations





NYU

# ROB-UY 2004

## Robotic Manipulation & Locomotion

### Agenda

1. Rigid Links and Joints
2. Configuration Space
3. Coordinate Frames



NYU

# ROB-UY 2004

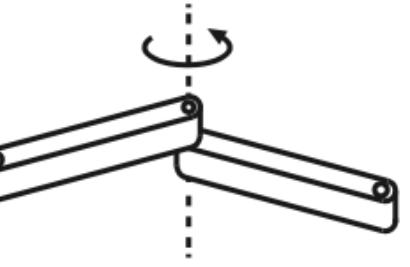
## Robotic Manipulation & Locomotion

### Agenda

1. Rigid Links and Joints
2. Configuration Space
3. Coordinate Frames

### Most common joints

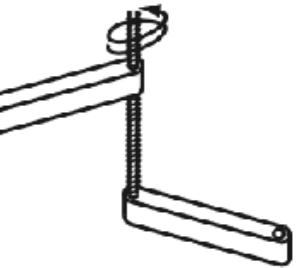
Revolute joint



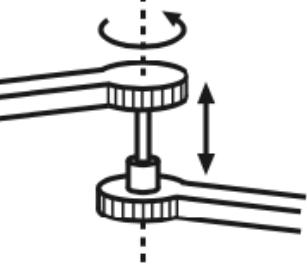
Prismatic joint



Helical joint



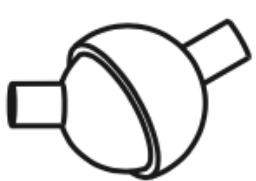
Cylindrical joint



Universal joint



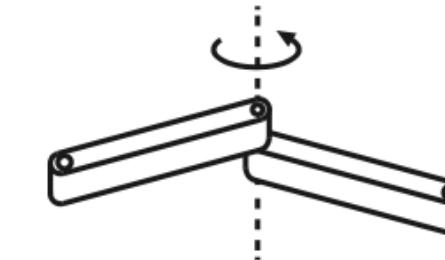
Spherical joint



## Degrees of Freedom

- The **Degrees of Freedom (DOF)** is the number of independent ways a robot can move, often as determined by its movable joints.

Revolute joint

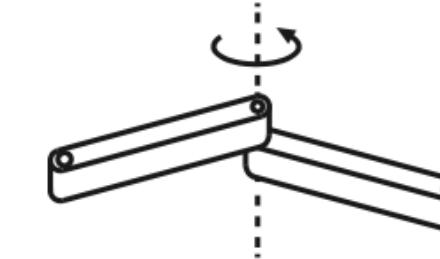


DOF  
1  
-

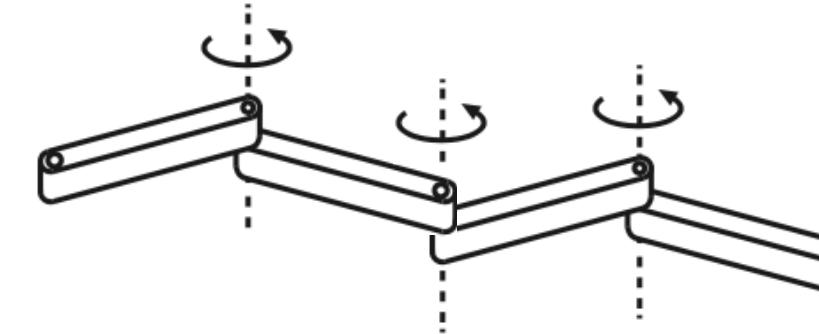
## Degrees of Freedom

- The **Degrees of Freedom (DOF)** is the number of independent ways a robot can move, often as determined by its movable joints.
  - E.g. adding one rotational joint adds one DOF.

Revolute joint



DOF  
1

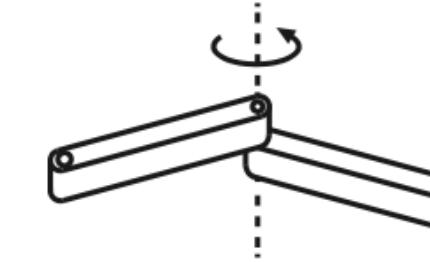


DOF  
3

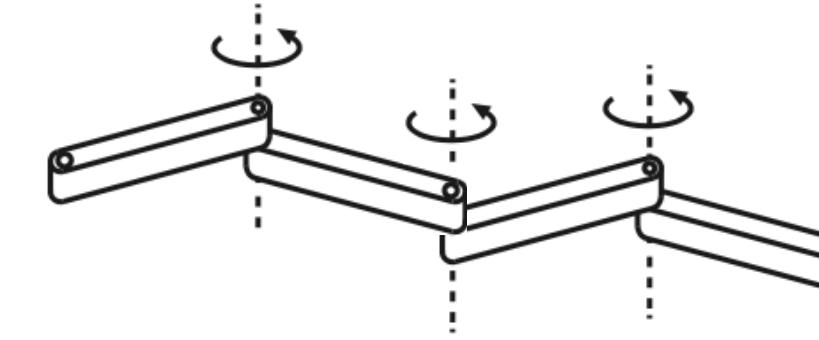
## Degrees of Freedom

- The **Degrees of Freedom (DOF)** is the number of independent ways a robot can move, often as determined by its movable joints.
  - E.g. adding one rotational joint adds one DOF.
- A robots DOF defines its **reachability**, and hence ability to accomplish different tasks.

Revolute joint



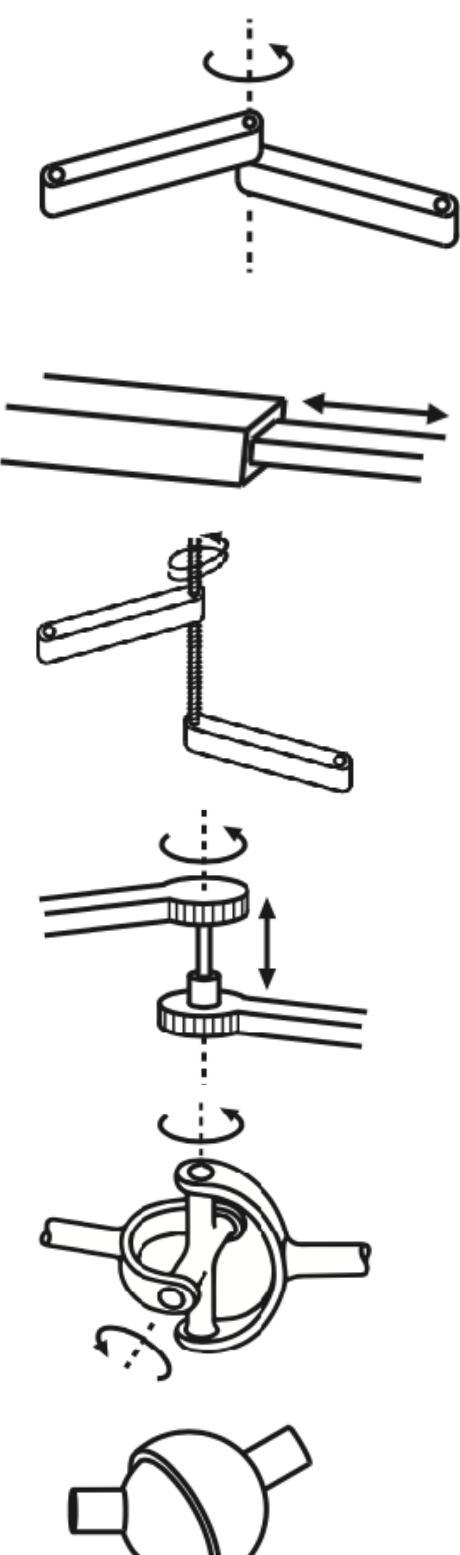
DOF  
1



DOF  
3

### Most common joints

	DOF
Revolute joint	1
Prismatic joint	1
Helical joint	1
Cylindrical joint	2
Universal joint	2
Spherical joint	3





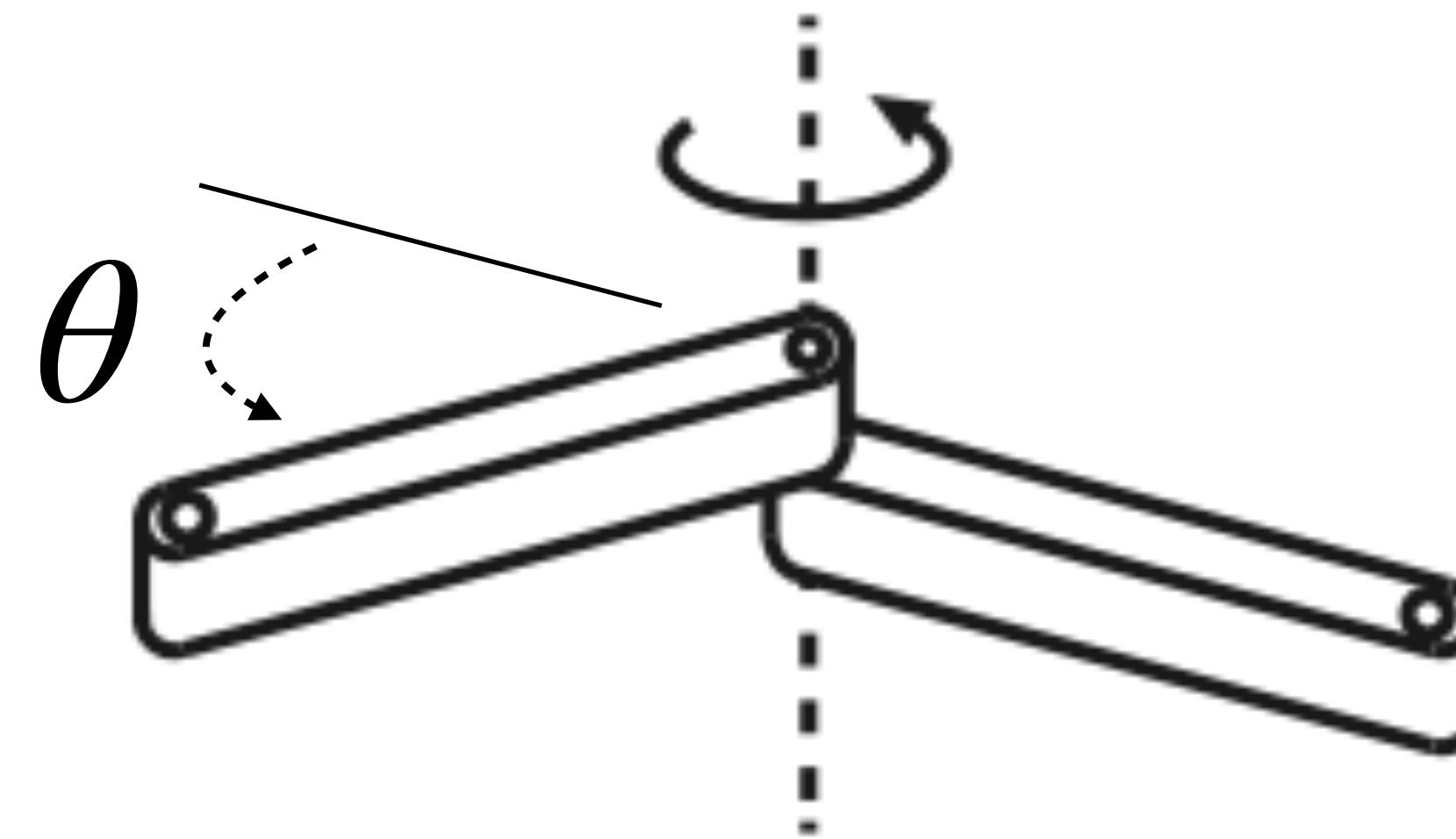
NYU

# ROB-UY 2004

## Robotic Manipulation & Locomotion

### Degrees of Freedom

- For each DOF, we typically use a variable to describe its configuration.
- We will be precise with the variables, leveraging coordinate frames ...



### Degrees of Freedom Examples



Iver2: 6  
DOF?



UniTree G1:  
43 + 6 DOF?



RichTech: 6  
DOF?



Waymo: 7  
DOF?



NYU

# ROB-UY 2004

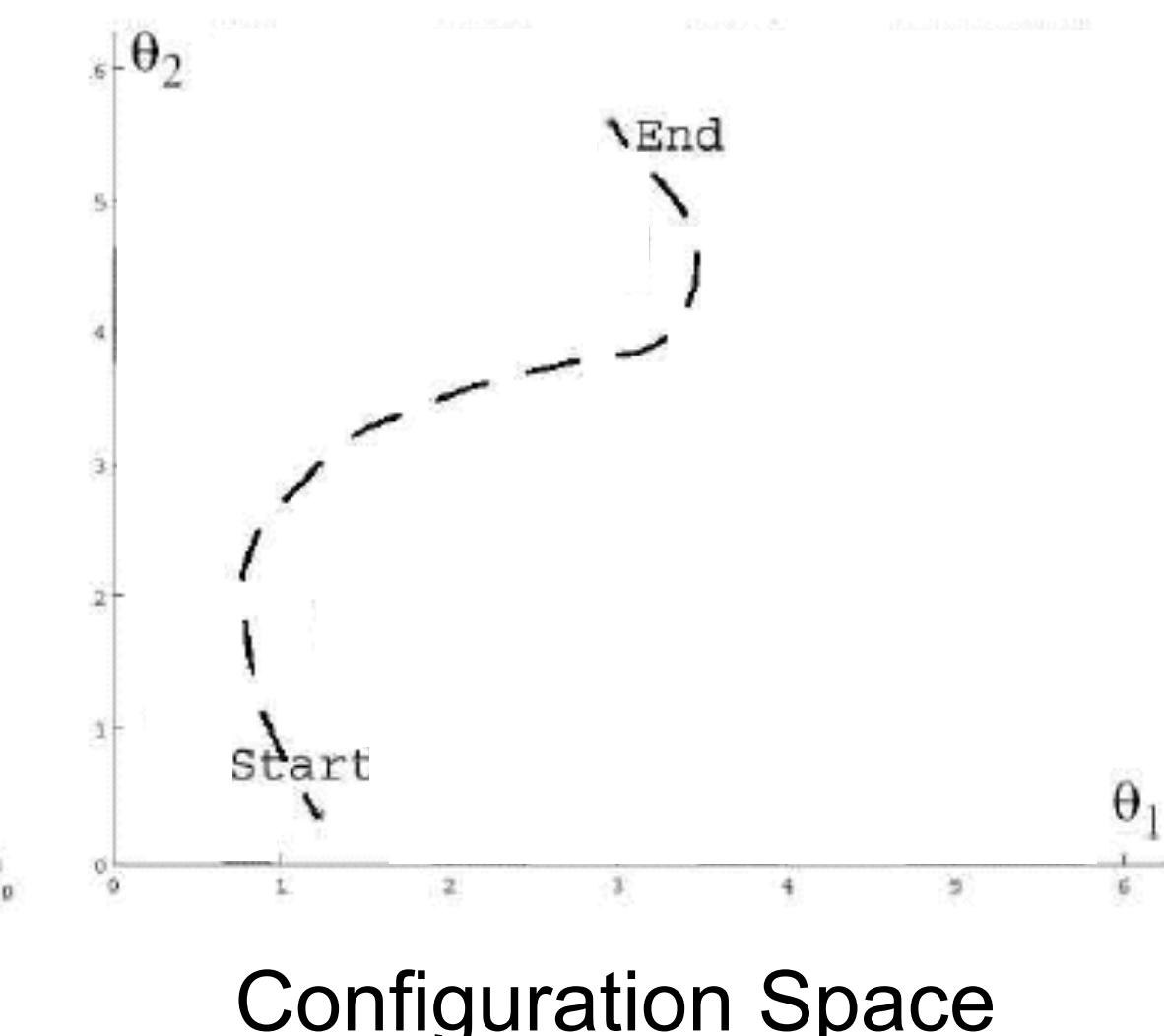
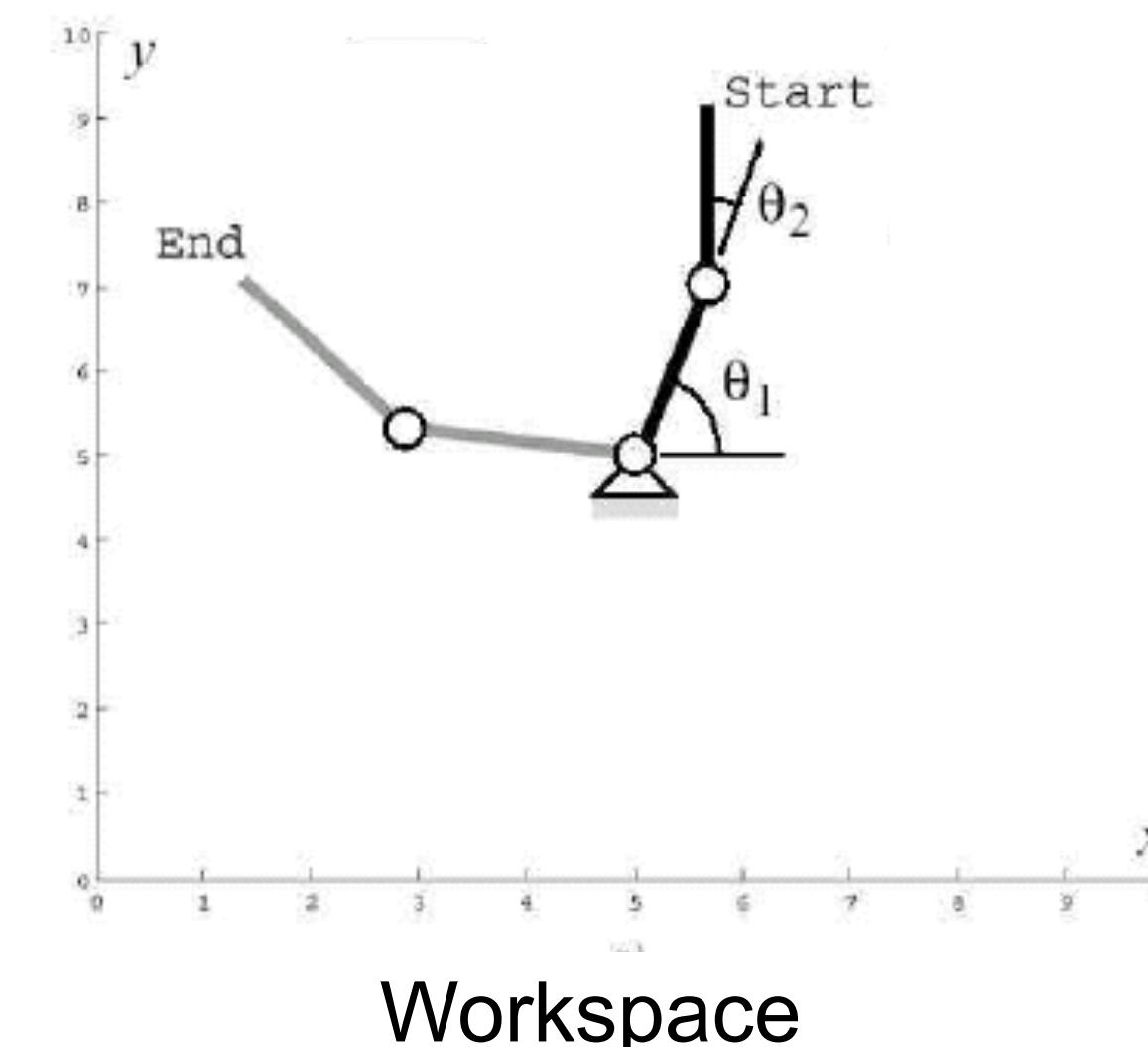
## Robotic Manipulation & Locomotion

### Agenda

1. Rigid Links and Joints
2. Configuration Space
3. Coordinate Frames

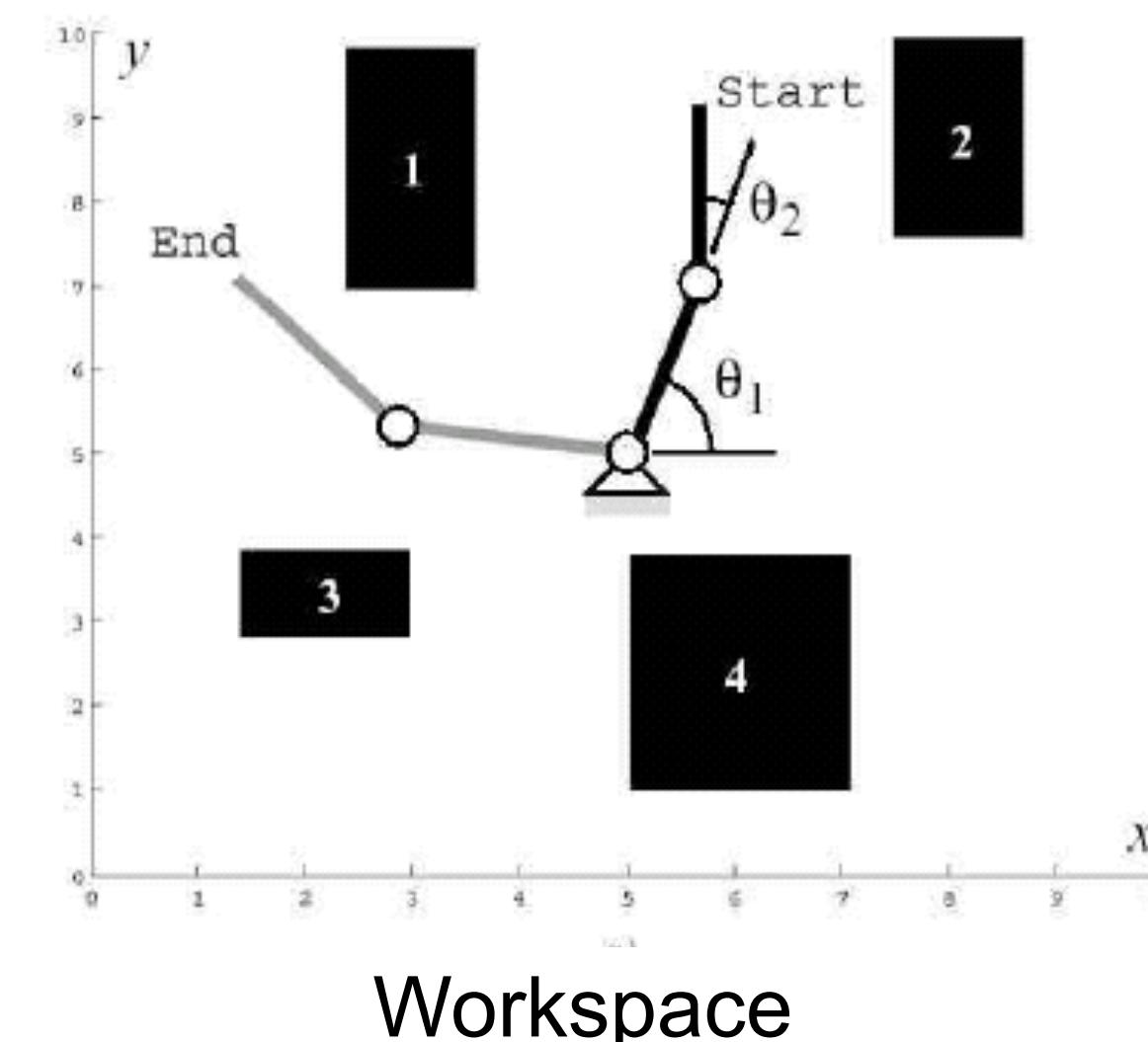
### Configuration Space

- A **configuration**  $q$  will completely define the position of every point on a robot
- The **configuration space**  $C$ , is the space of all possible configurations of the robot.
- The **free space**  $F \in C$ , is the portion of the free space which is collision-free.

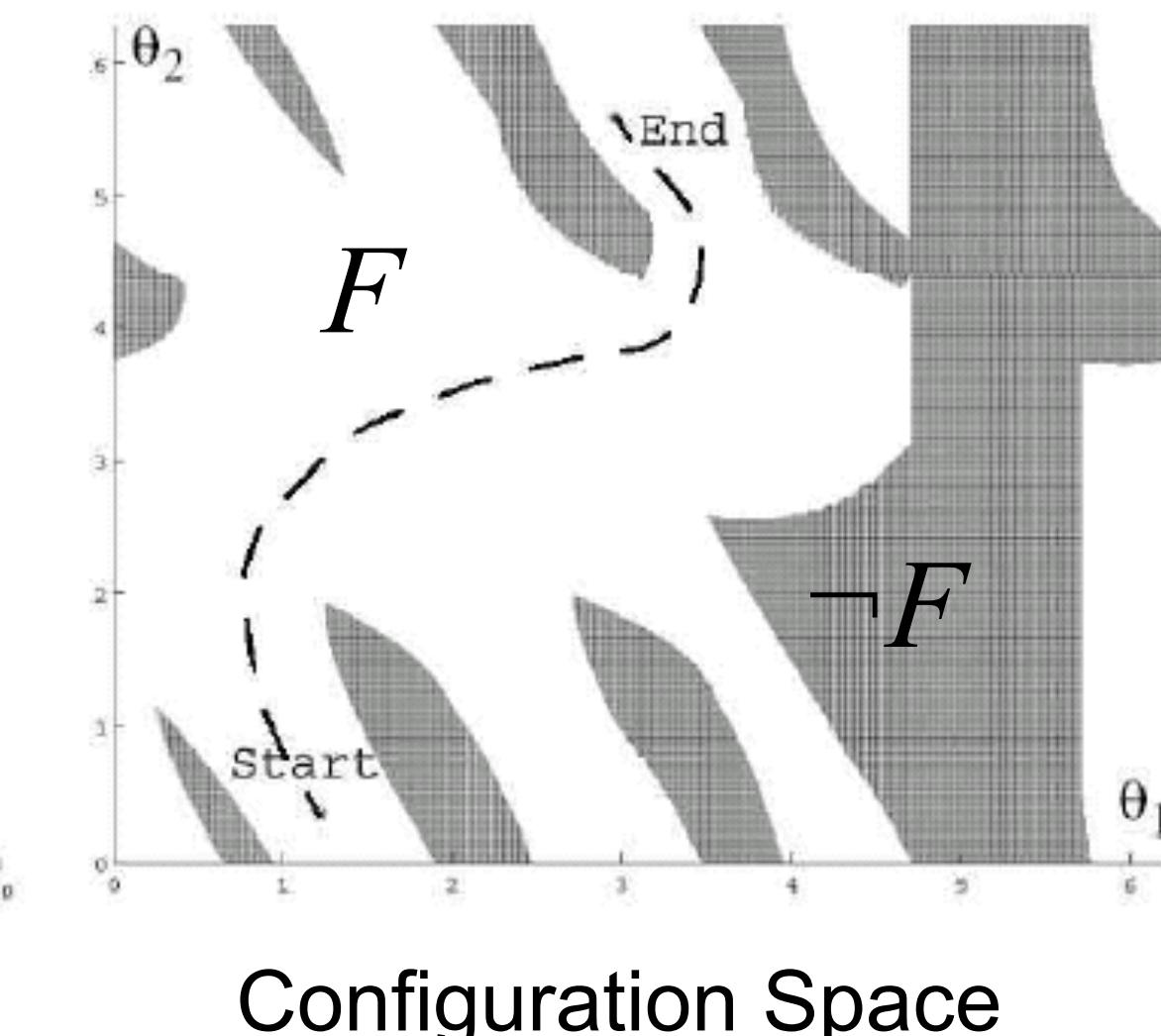


### Configuration Space

- A **configuration**  $q$  will completely define the state of a robot (e.g. mobile robot  $x, y, \theta$ )
- The **configuration space**  $C$ , is the space of all possible configurations of the robot.
- The **free space**  $F \in C$ , is the portion of the free space which is collision-free.



Workspace



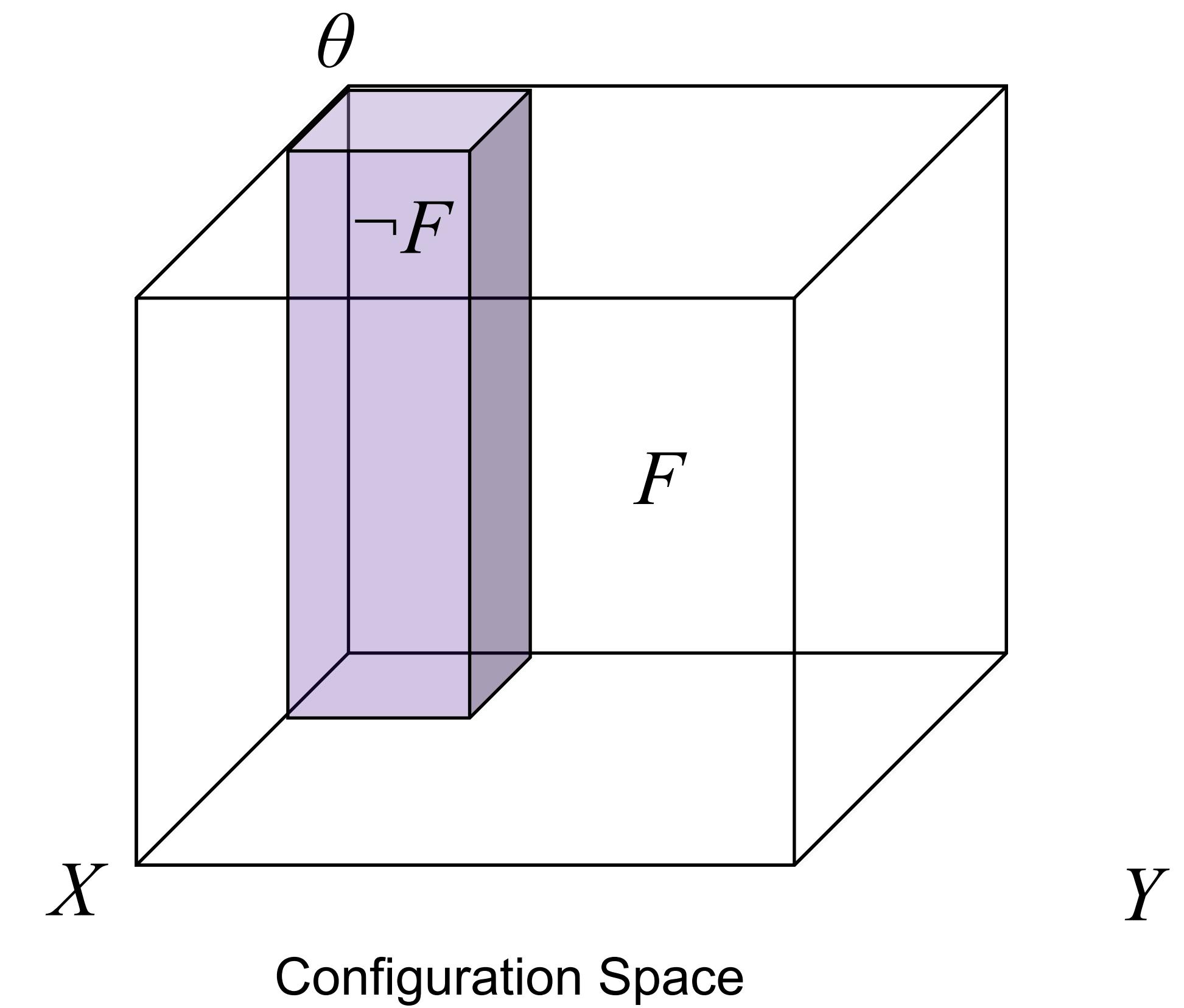
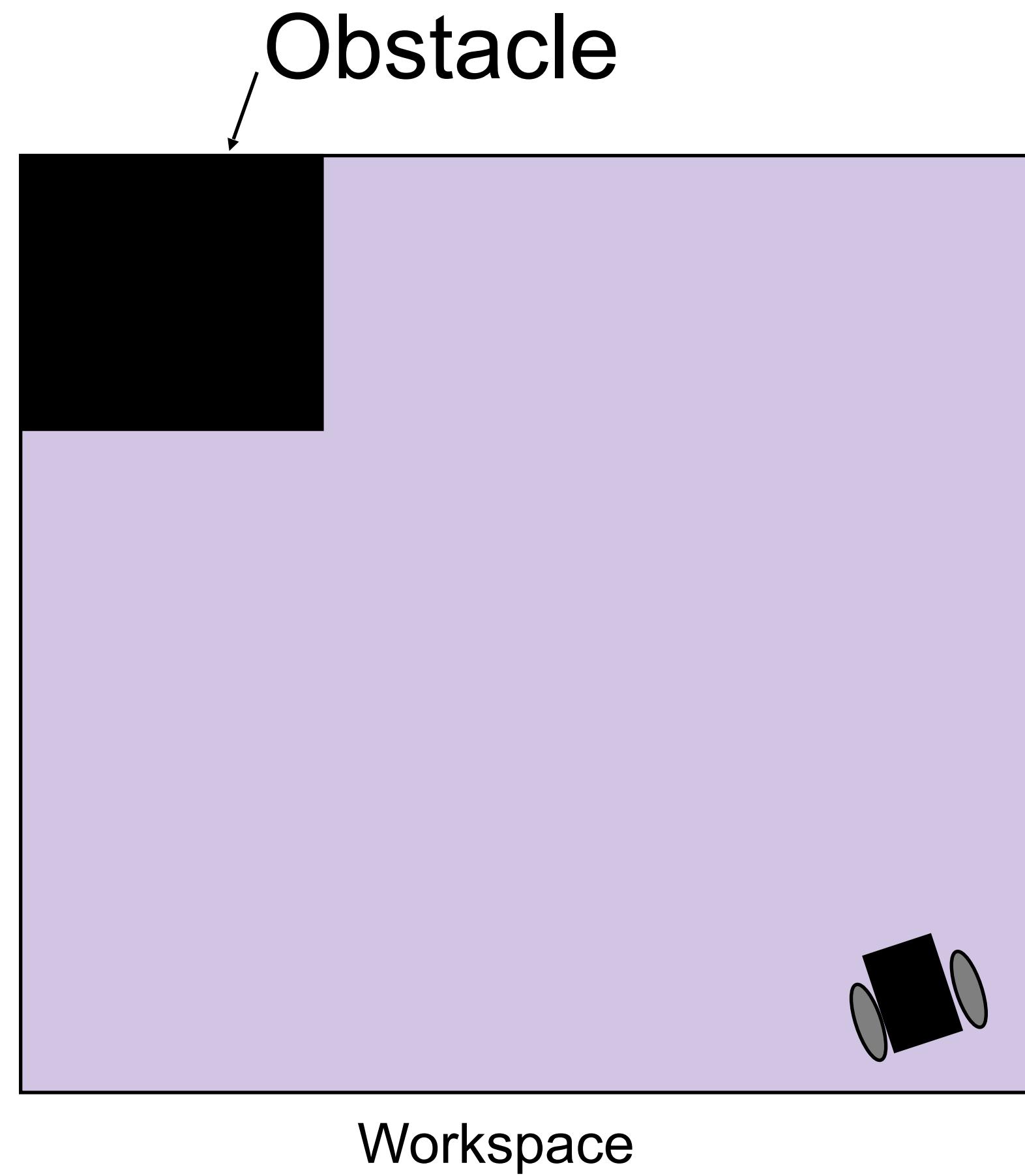
Configuration Space



NYU

# ROB-UY 2004

## Robotic Manipulation & Locomotion

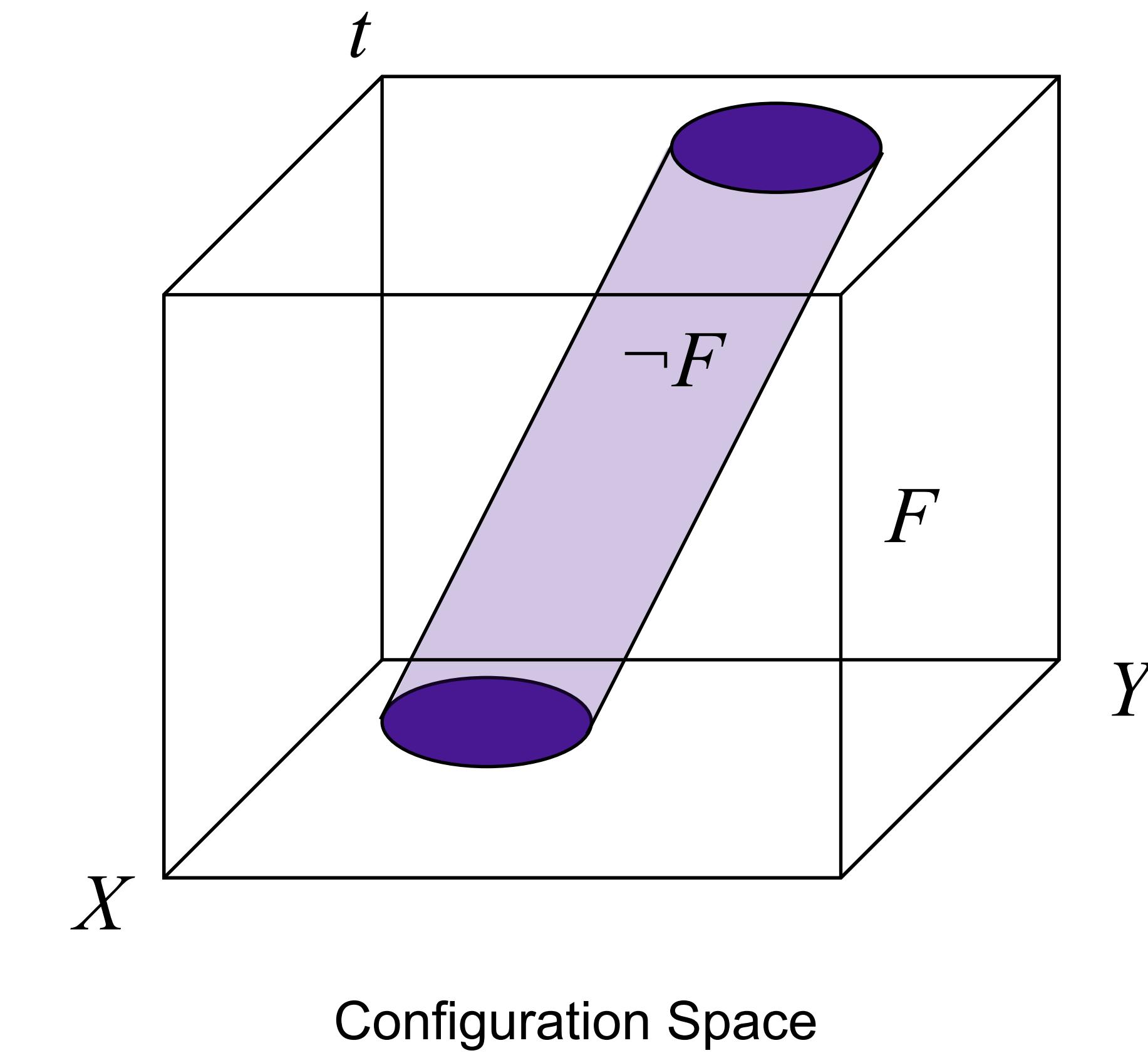
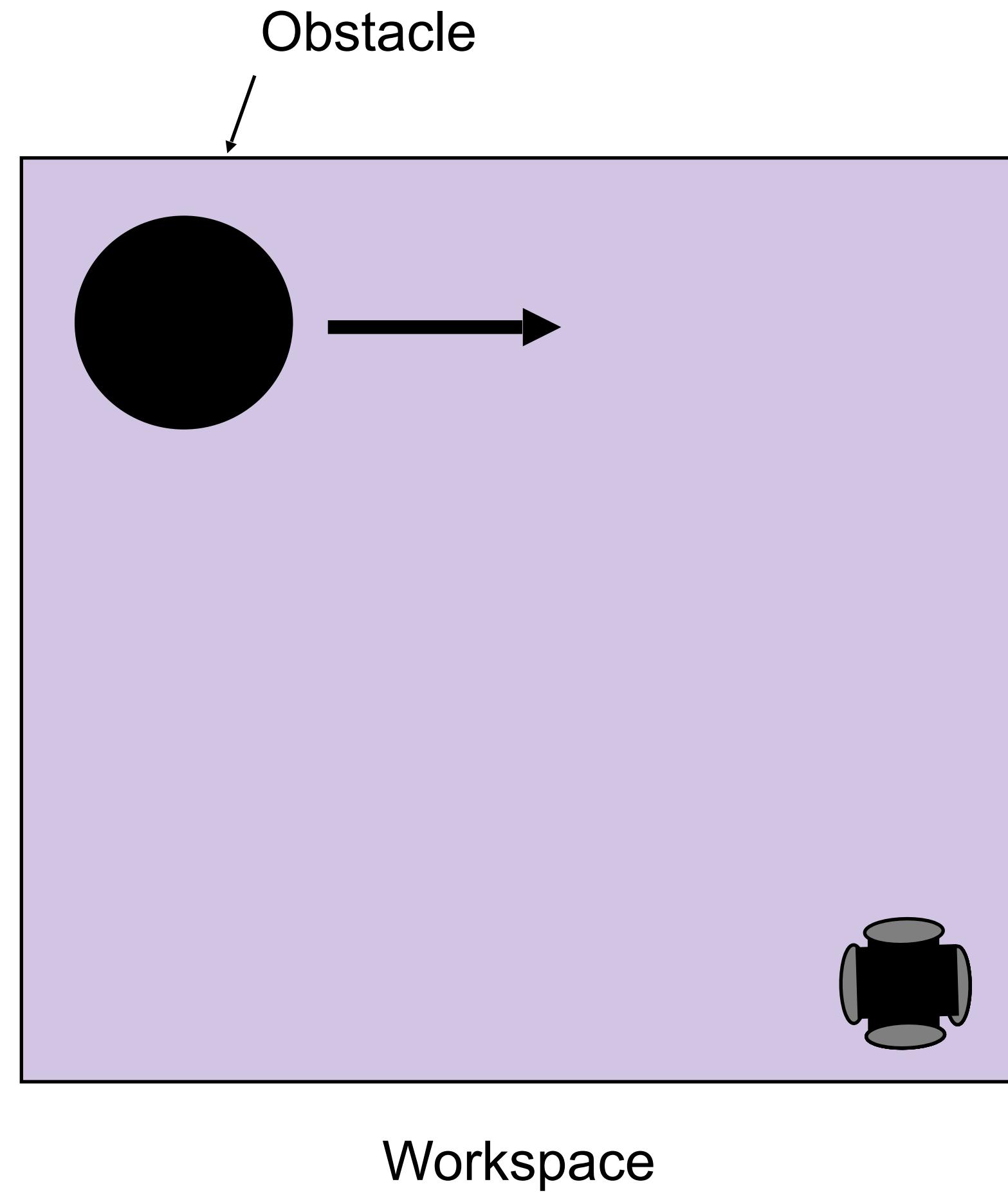




NYU

# ROB-UY 2004

## Robotic Manipulation & Locomotion





NYU

# ROB-UY 2004

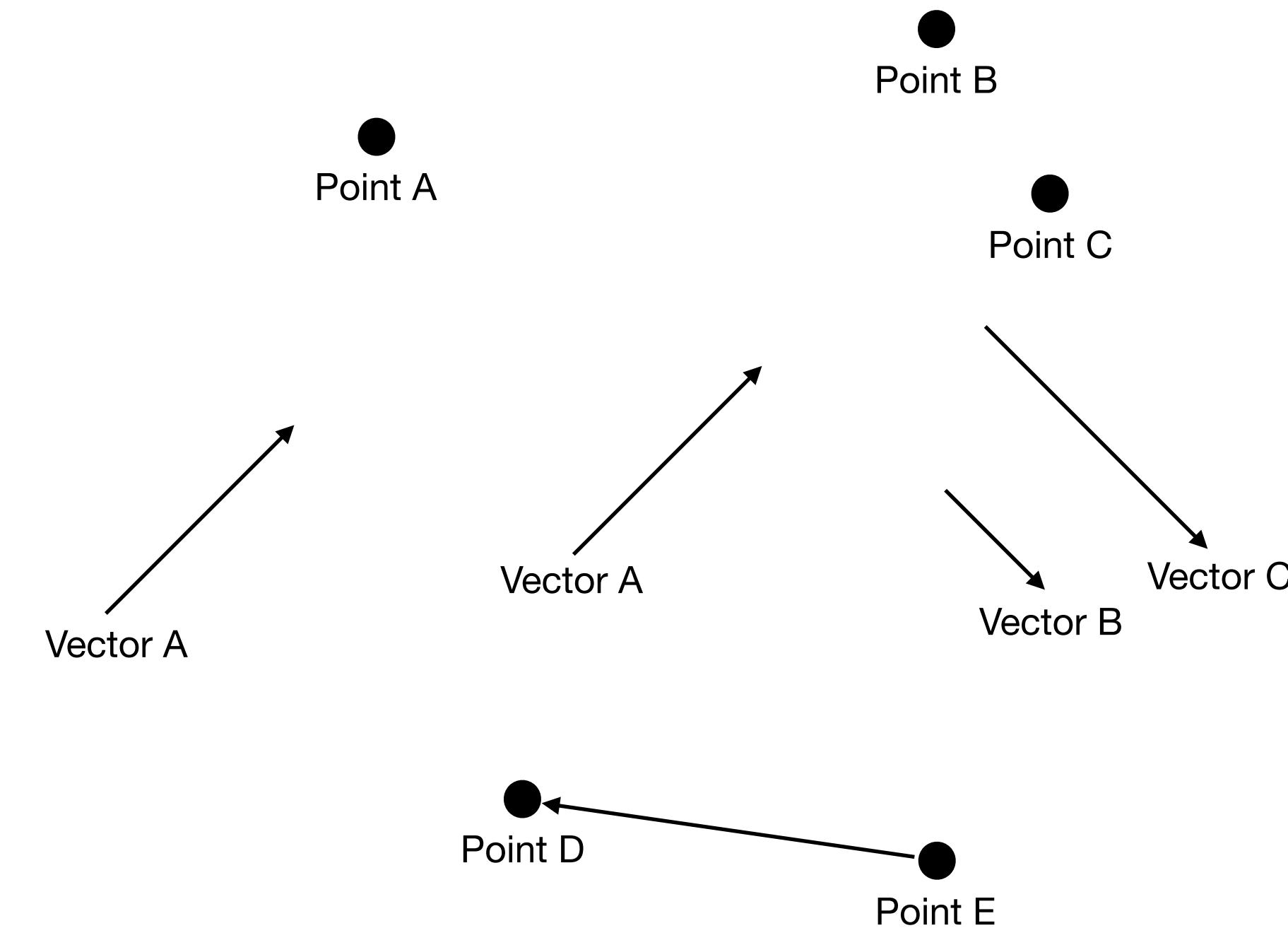
## Robotic Manipulation & Locomotion

### Agenda

1. Rigid Links and Joints
2. Configuration Space
3. **Coordinate Frames**

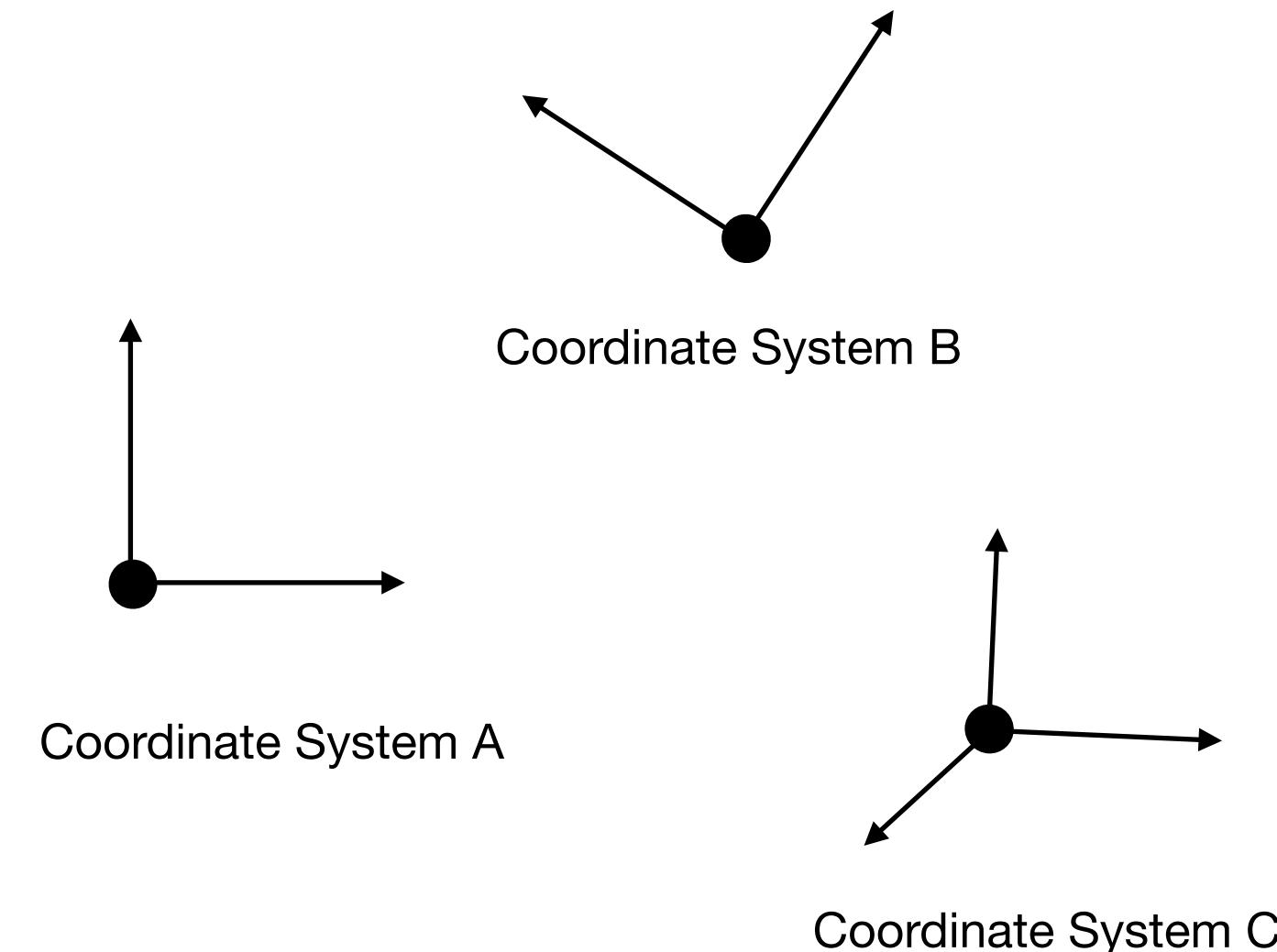
## Points & Vectors

- A **point** defines a specific **position in space**
- A **vector** is combination of a **direction and magnitude**
  - A vector can be used to define the position of one point in space relative to another.



## Coordinate Systems

- AKA a **Reference Frame**
- Defined by 1 point and 2 vectors for 2D spaces
- Defined by 1 point and 3 vectors for 3D spaces





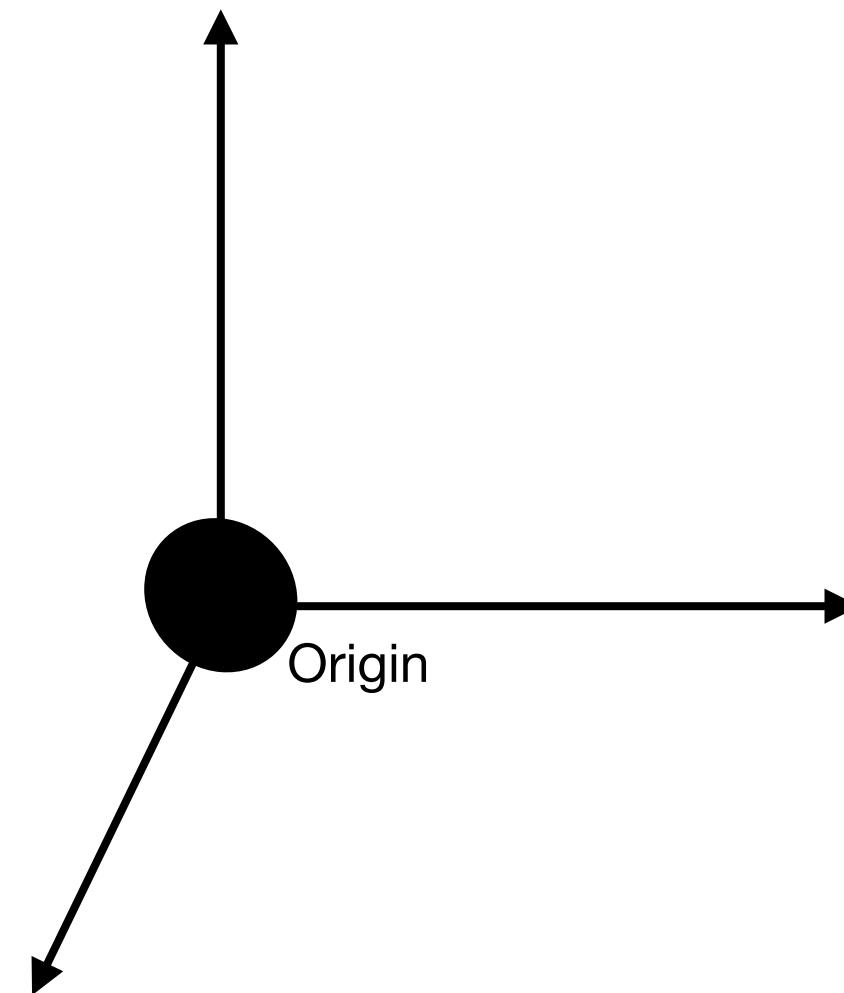
NYU

# ROB-UY 2004

## Robotic Manipulation & Locomotion

### Coordinate Systems

- The point is called the origin





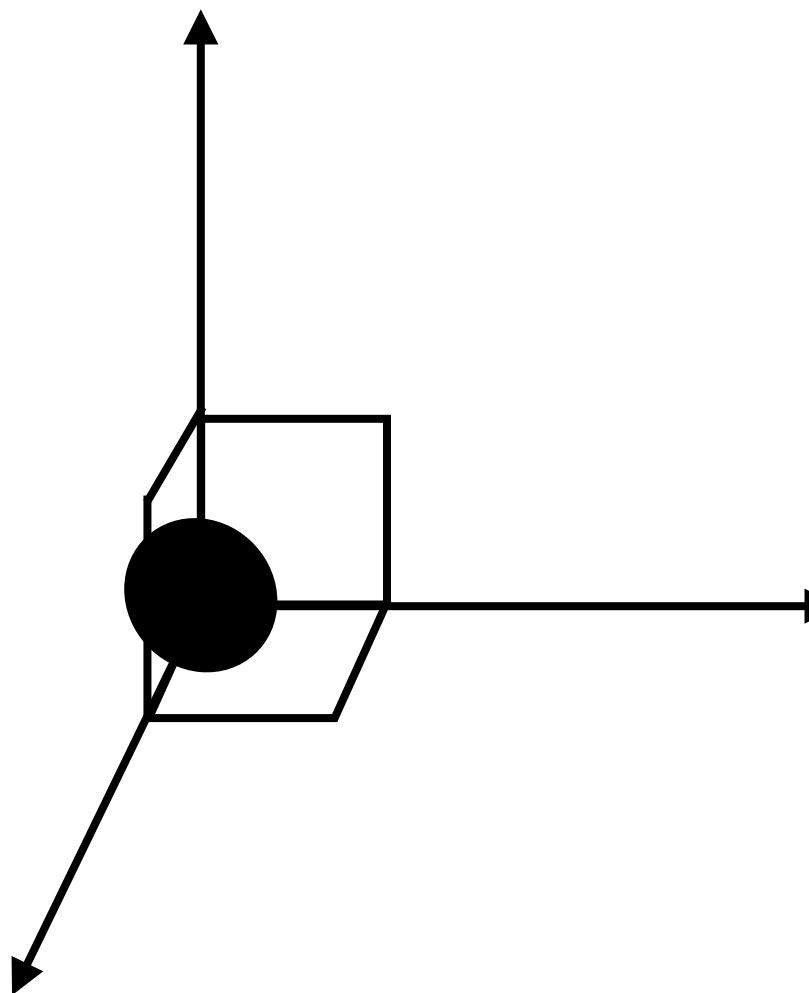
NYU

# ROB-UY 2004

## Robotic Manipulation & Locomotion

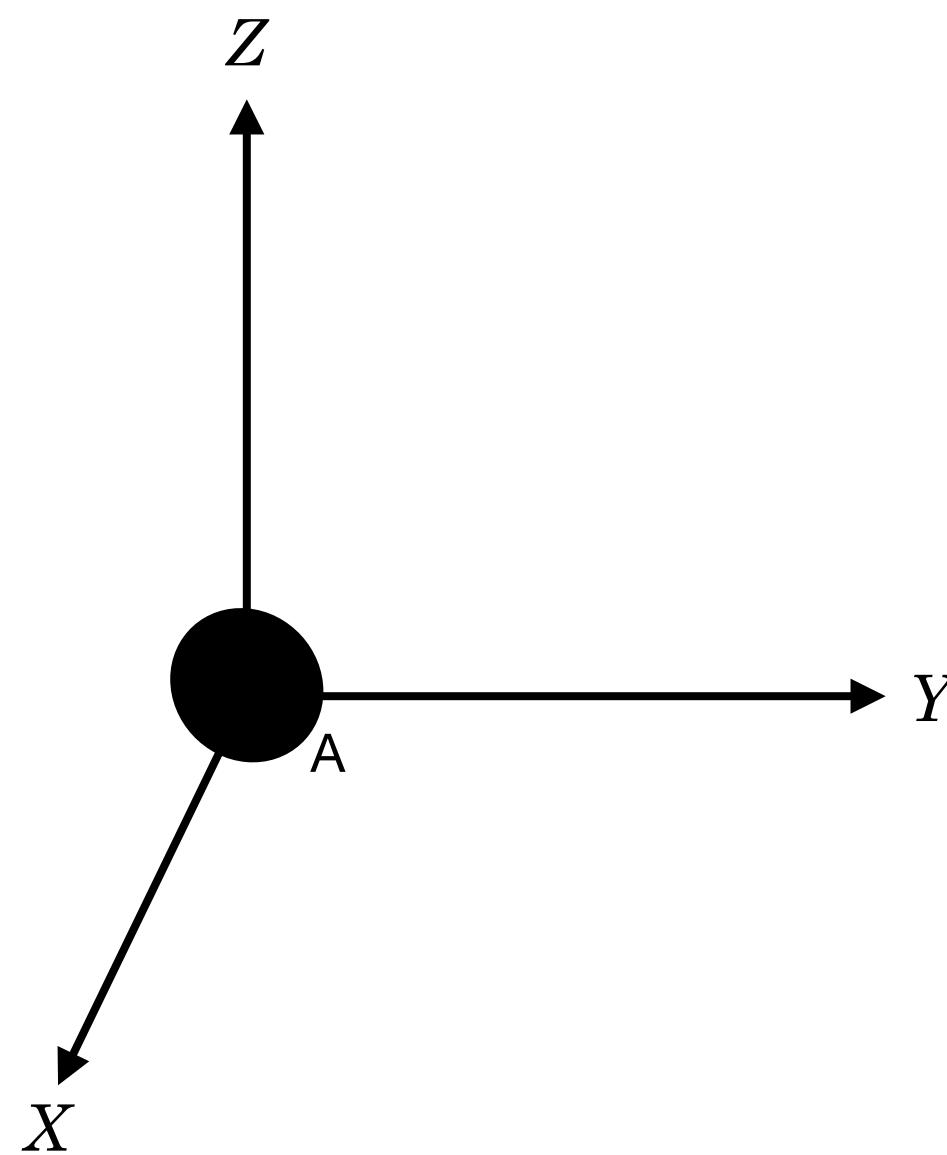
### Coordinate Systems

- The point is called the origin
- The vectors have perpendicular directions



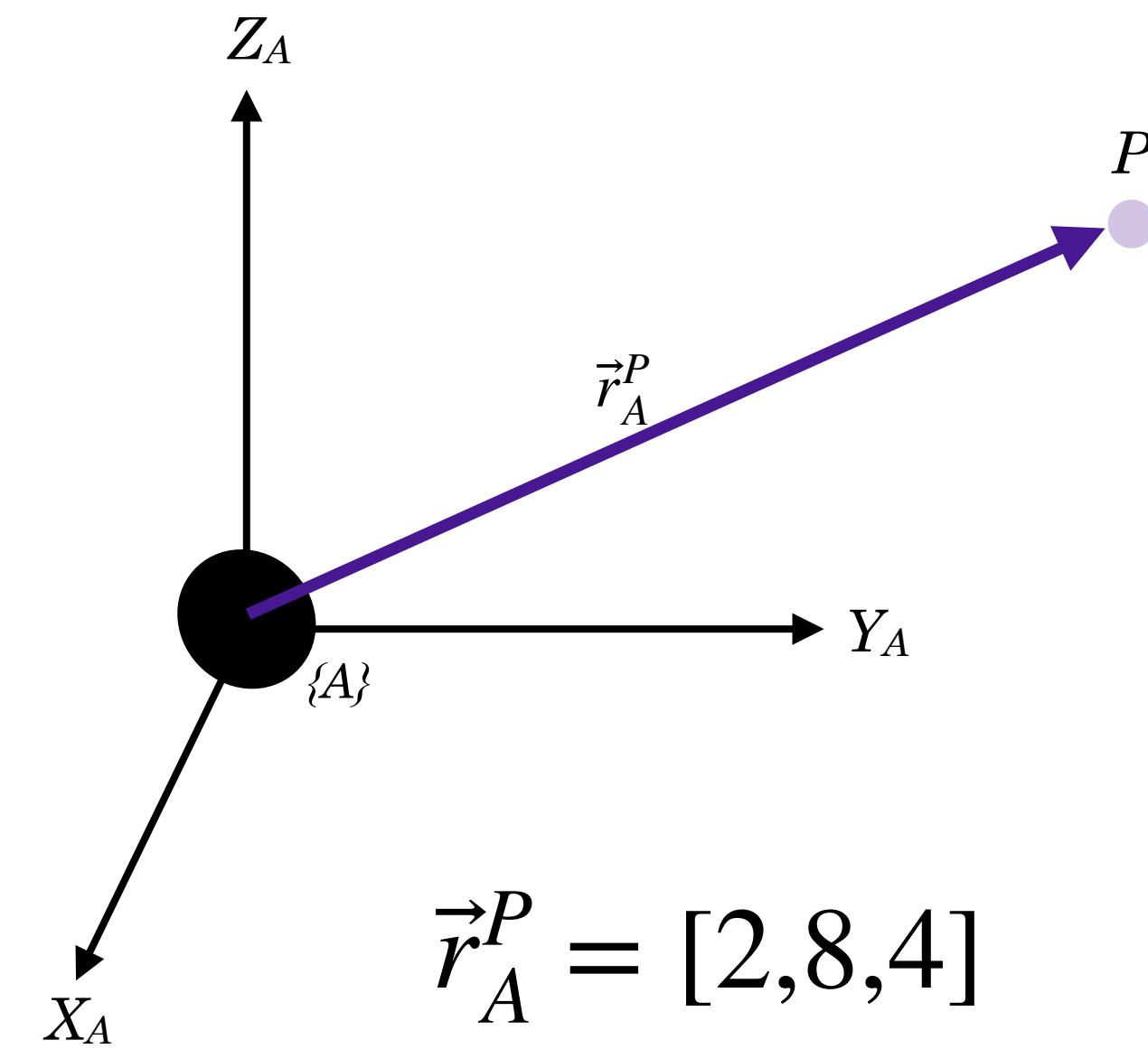
## Coordinate Systems

- The point is called the origin
- The vectors have perpendicular directions
- Each direction vector is assigned a label



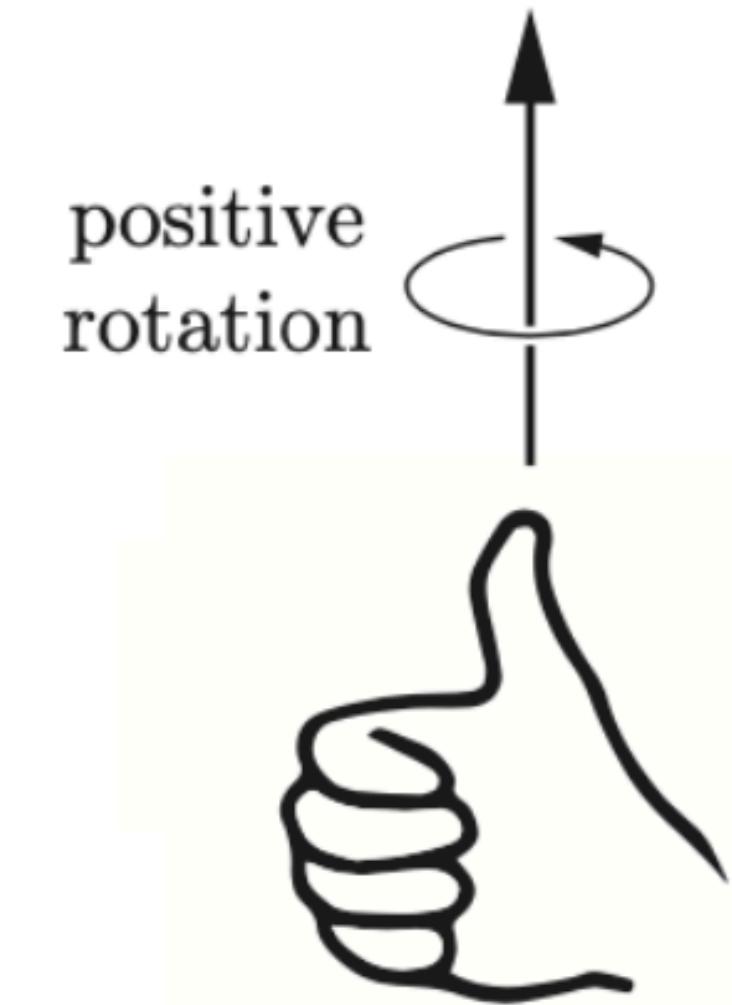
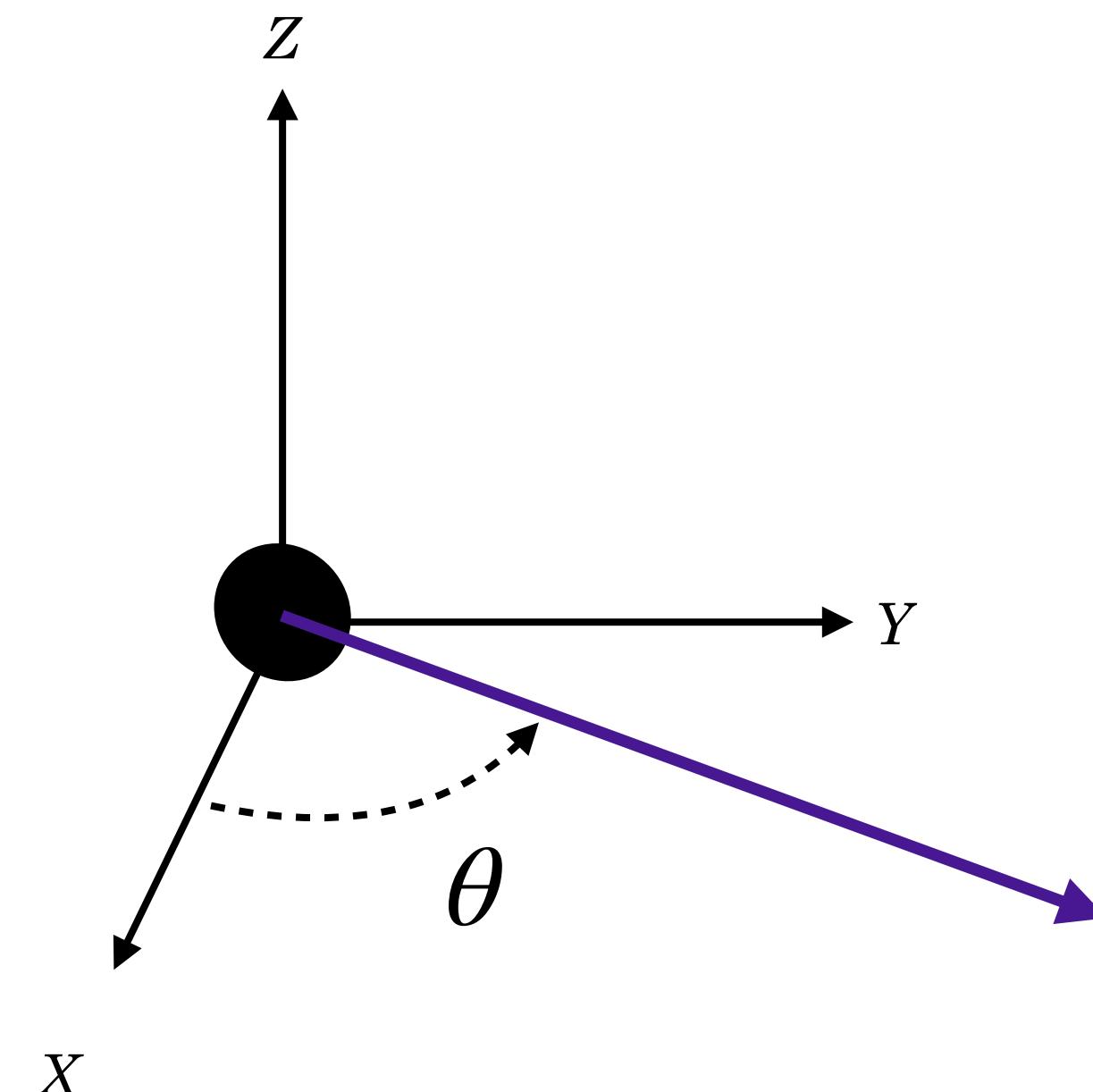
## Coordinate Systems

- The point is called the origin
- The vectors have perpendicular directions
- Each direction vector is assigned a label



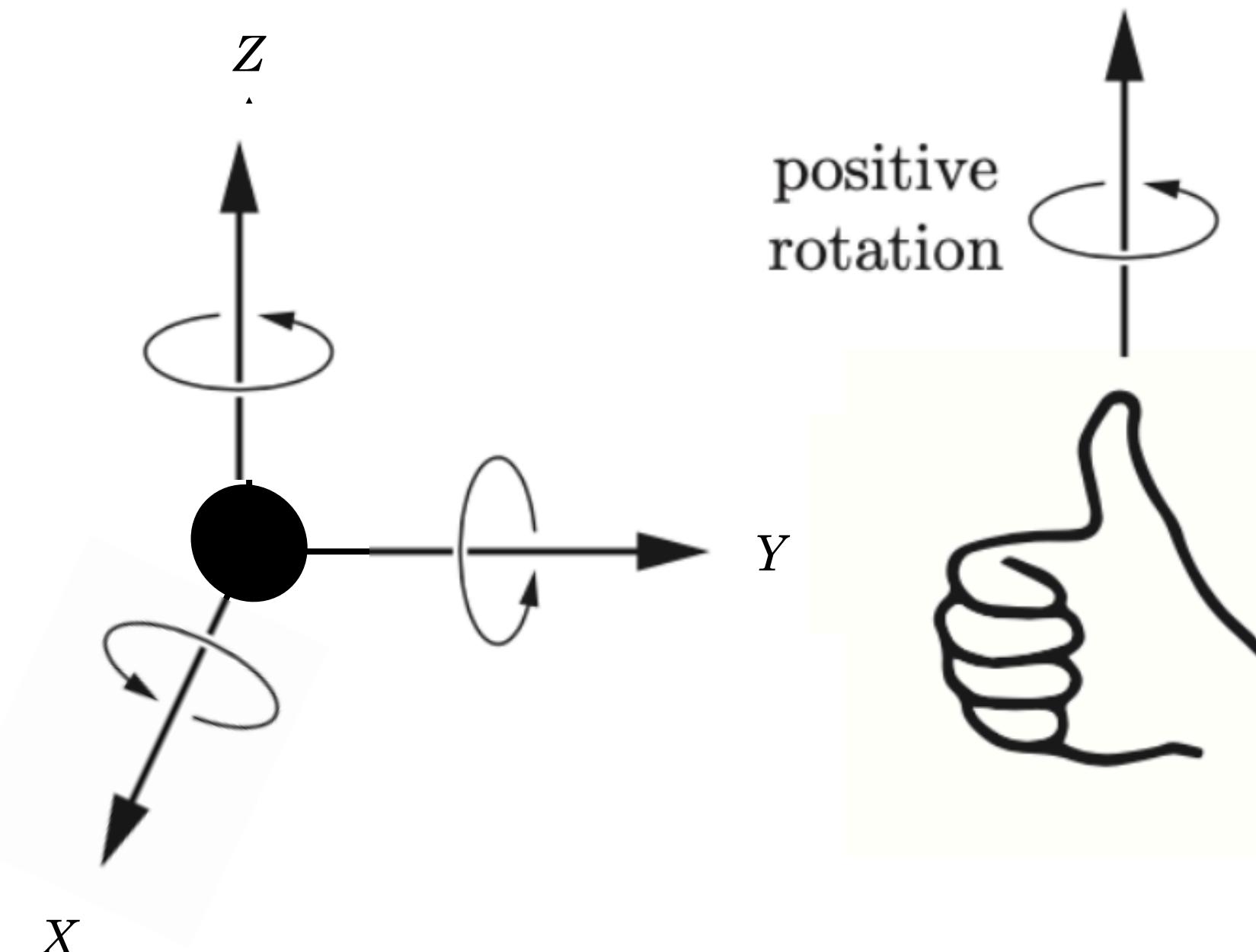
## Coordinate Systems

- The point is called the origin
- The vectors have perpendicular directions
- Each direction vector is assigned a label
- **Positive rotations** are CCW when looking in the negative direction of the axis of rotation.



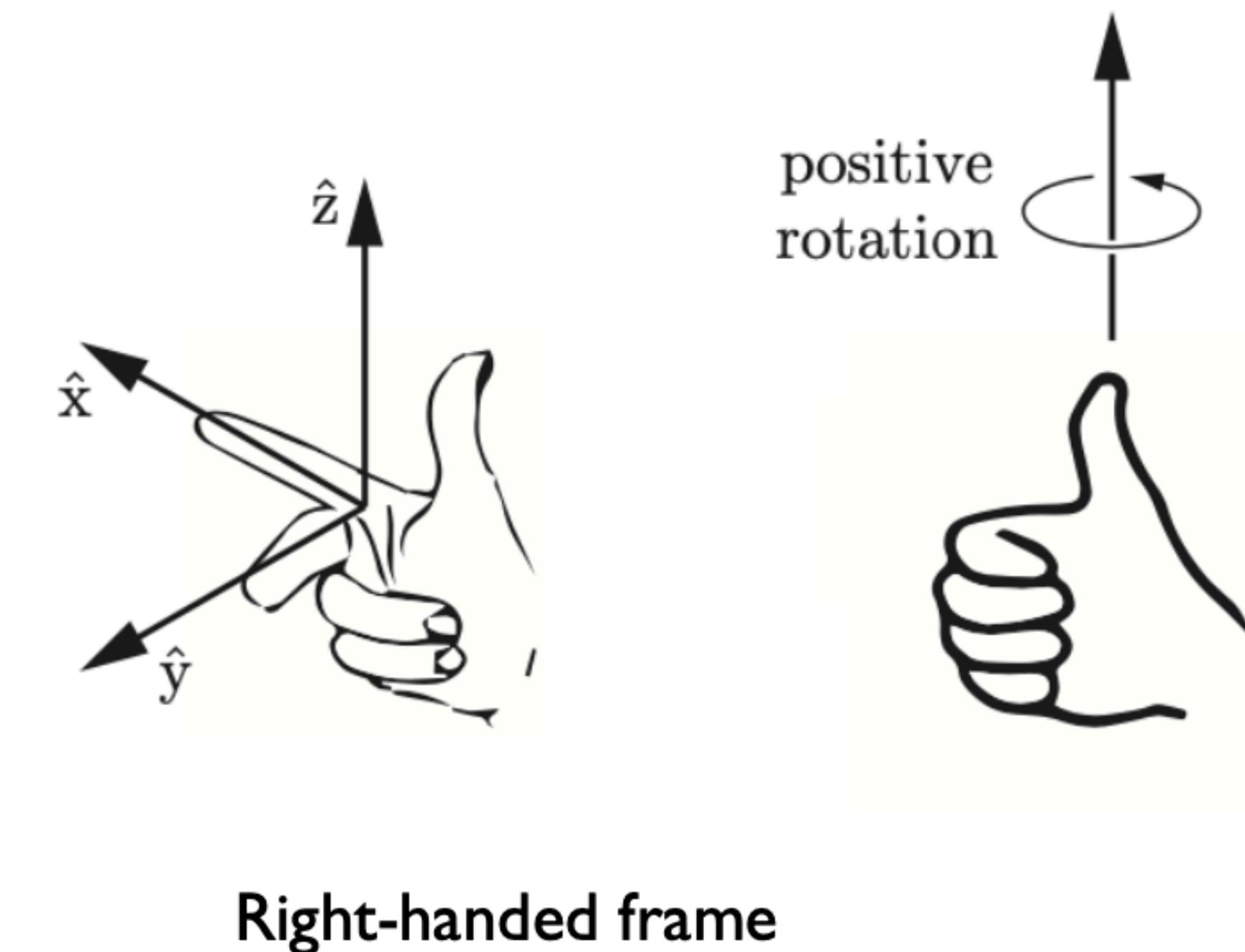
## Coordinate Systems

- The point is called the origin
- The vectors have perpendicular directions
- Each direction vector is assigned a label
- **Positive rotations** are CCW when looking in the negative direction of the axis of rotation.



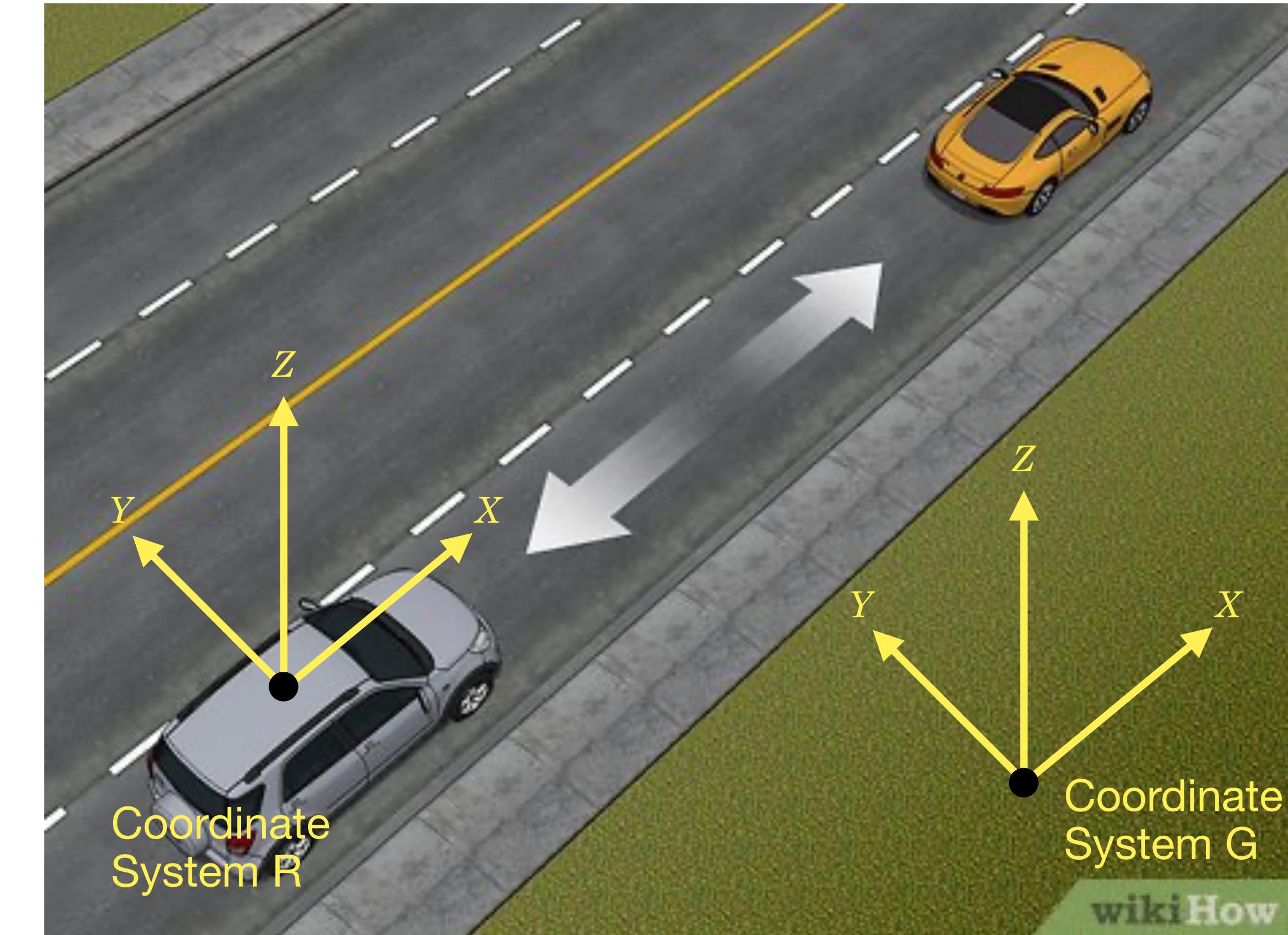
## Coordinate System Conventions in Robotics

- **Right hand coordinate frames** have
  - Rotating  $x$ -axis about the  $z$ -axis yields the  $y$ -axis
  - Rotating  $y$ -axis about the  $x$ -axis yields the  $z$ -axis
  - Rotating  $z$ -axis about the  $y$ -axis yields the  $x$ -axis



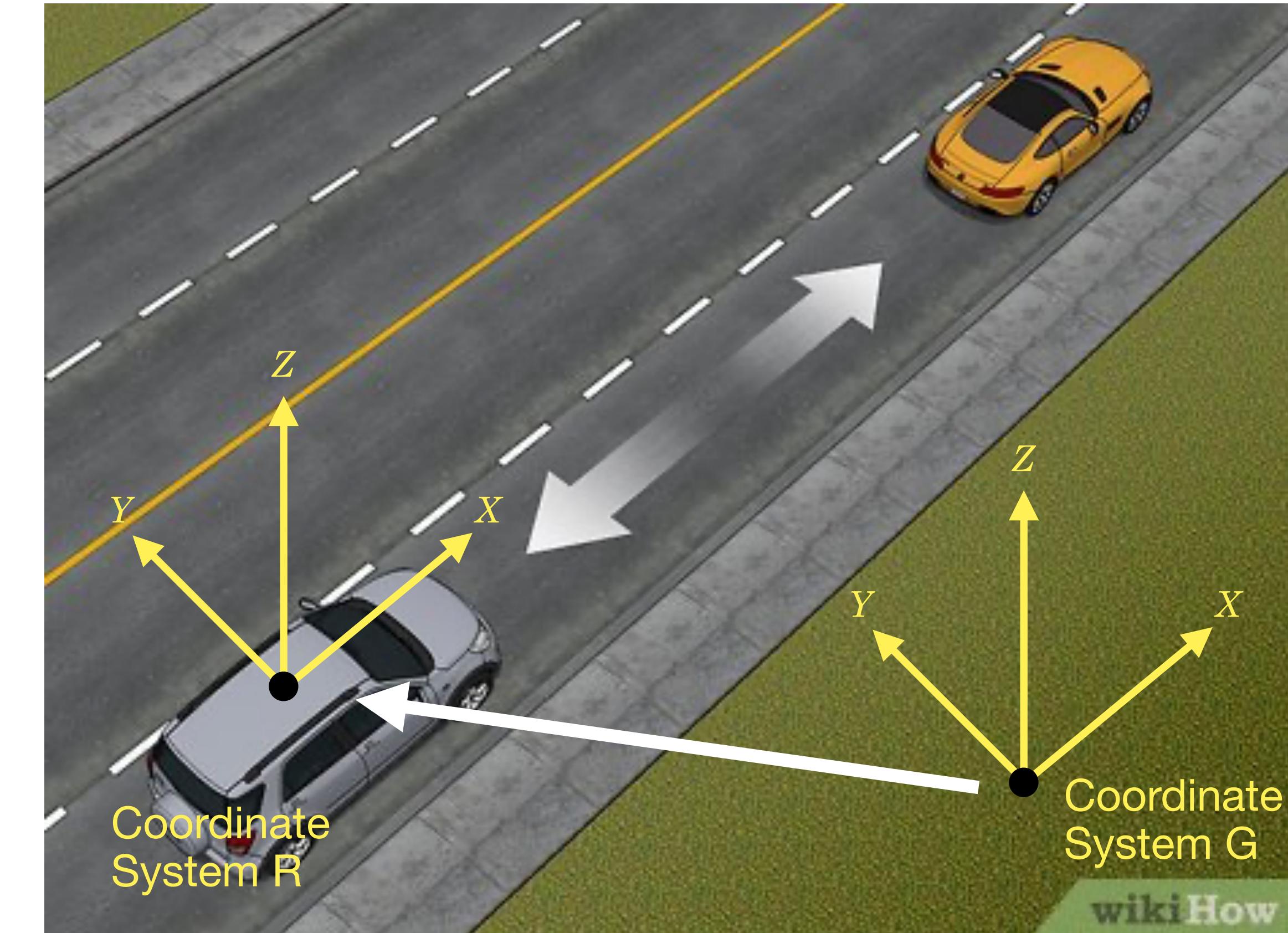
## Coordinate Systems

- In robotics, the frame is attached to something physical (e.g. a robot or the ground).



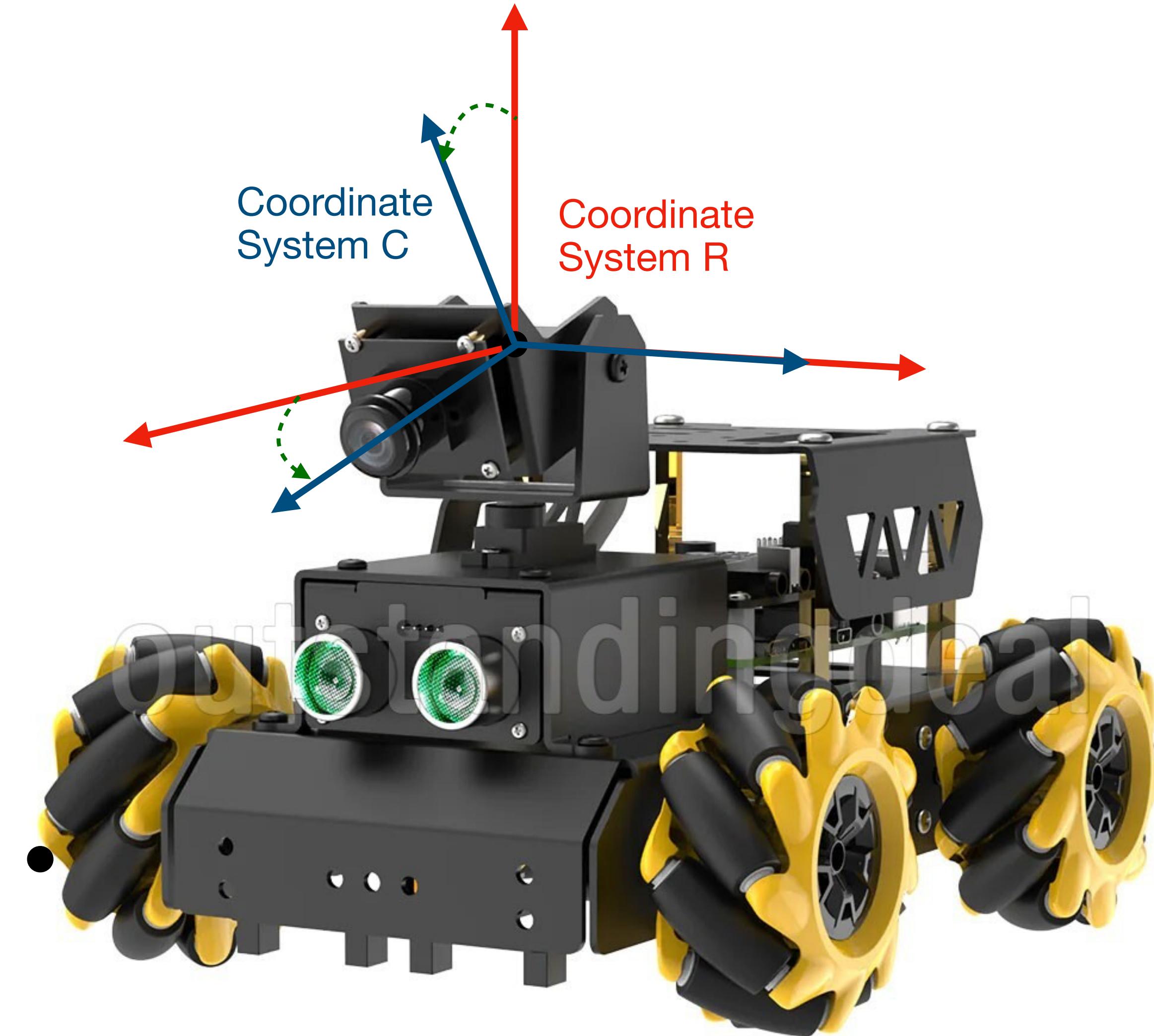
## Coordinate Systems

- In robotics, the frame is attached to something physical (e.g. a robot or the ground).
- It is helpful to be able to:
  - Describe the **relative position** of one frame with respect to another frame



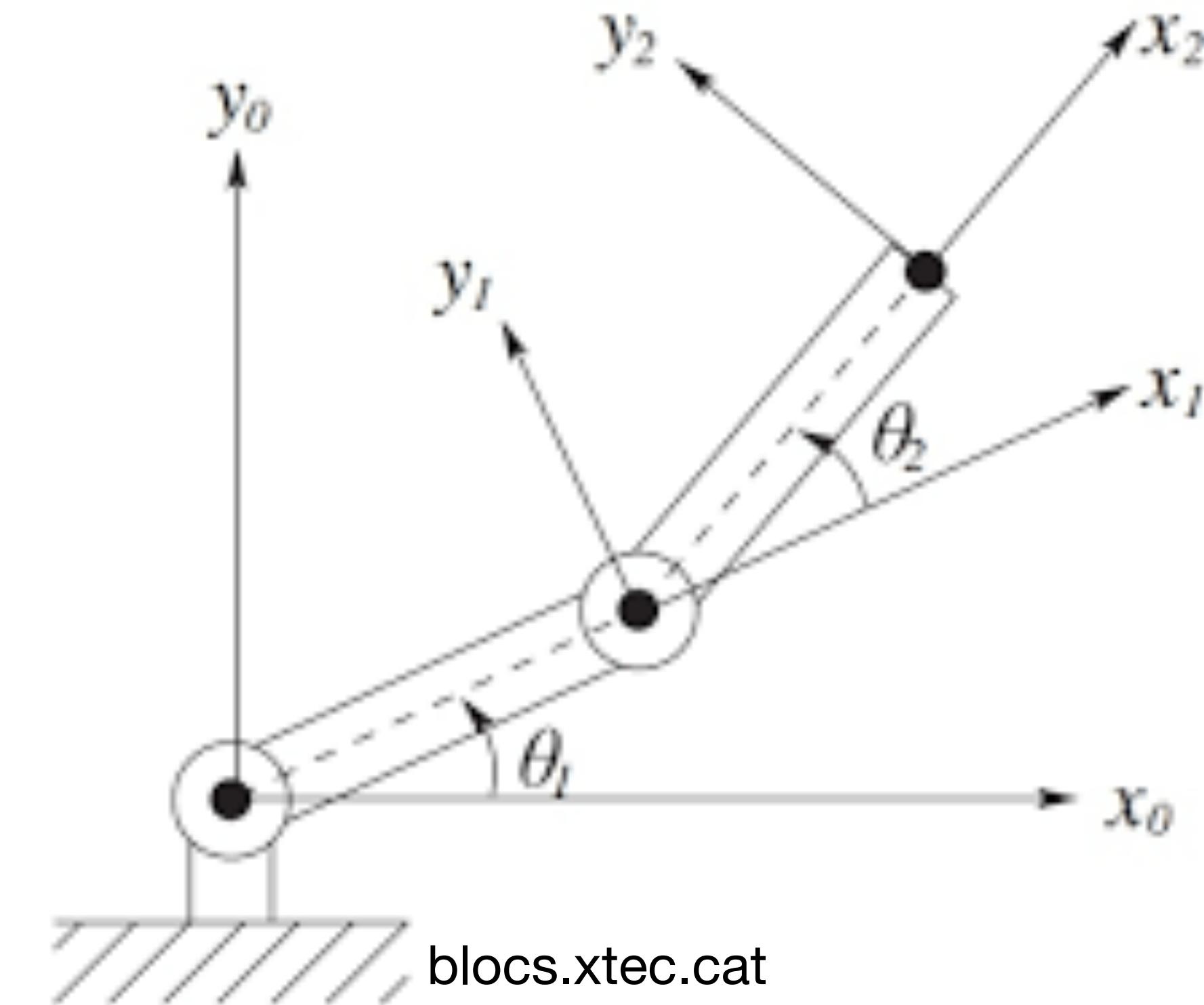
## Coordinate Systems

- In robotics, the frame is attached to something physical (e.g. a robot or the ground).
- It is helpful to be able to:
  - Describe the relative position of one frame with respect to another frame
  - Describe the **relative rotation** of one frame with respect to another frame



## Coordinate Systems

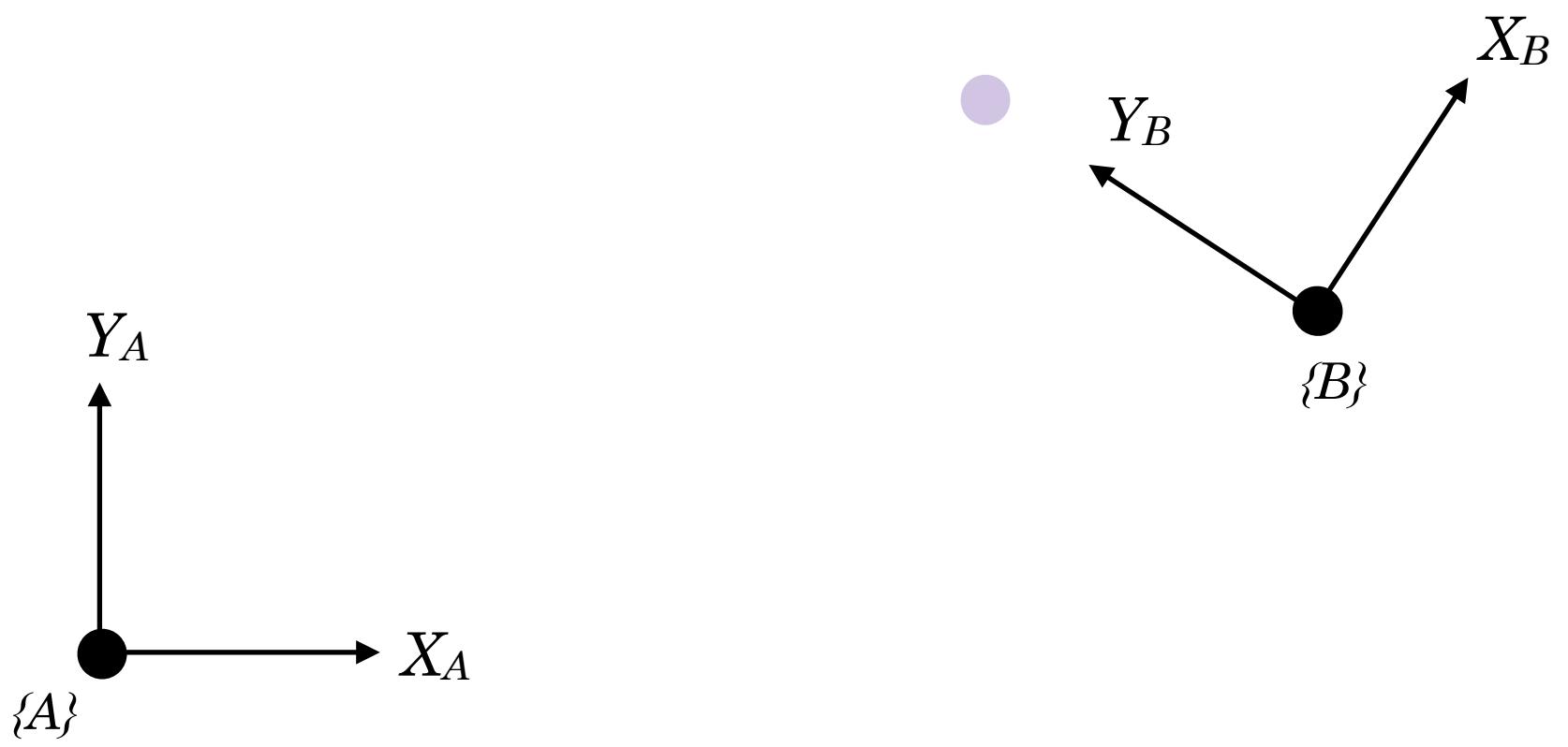
- In robotics, the frame is attached to something physical (e.g. a robot or the ground).
- It is helpful to be able to:
  - Describe the **relative position** and **relative rotation** of one frame with respect to another frame



blocs.xtec.cat

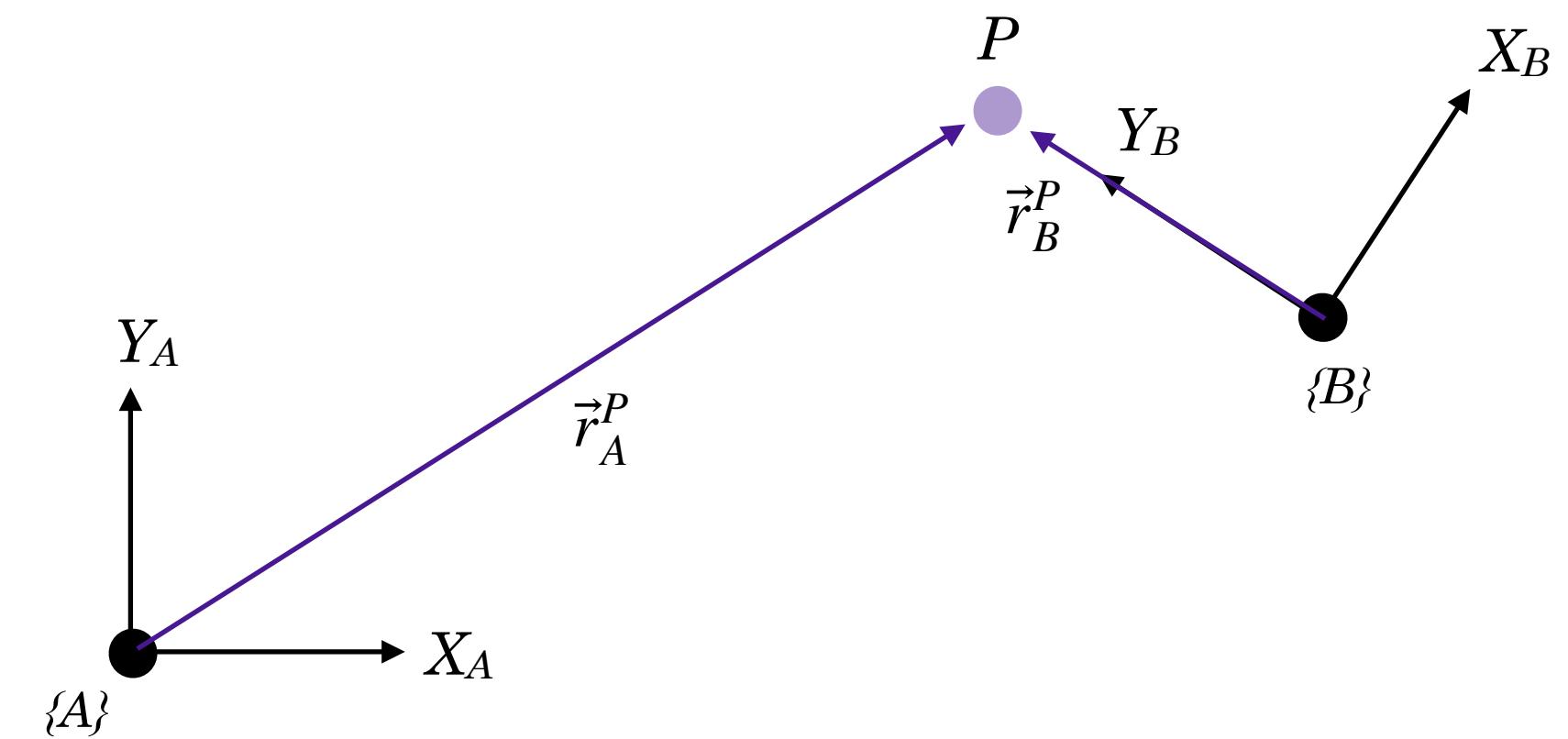
## Coordinate Systems

- It is important to note that a point in space looks very different from the perspective of one frame versus another.



## Coordinate Systems

- It is important to note that a point in space looks very different from the perspective of one frame versus another.



## Coordinate Systems

- It is important to note that a point in space looks very different from the perspective of one frame versus another.

