Introduction to Isaac Gym and Controls

June 3, 2025

What is a robot?

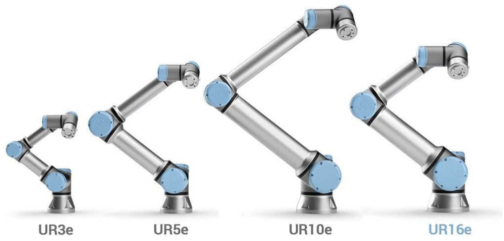
- A robot is an autonomous or semi-autonomous machine that can perceive, decide, and act on its environment to accomplish a task.

- Key Characteristics:

    1. **Sensing** the environment (e.g., cameras, force sensors)
    2. **Processing** data and making decisions
    3. **Acting** using motors, grippers, or wheels
    4. Often programmable and may adapt over time

Types of Robot

Type	Description	Example Robots
Mobile Robots	Move through environment on wheels or legs	TurtleBot, Spot, Roomba
Manipulators	Stationary arms for precise motion & tasks	Franka Emika, UR5
Humanoids	Human-like form, often used in research	NAO, Pepper, Atlas
Industrial Arms	High-speed, high-precision manipulators for factories	KUKA, ABB, Fanuc

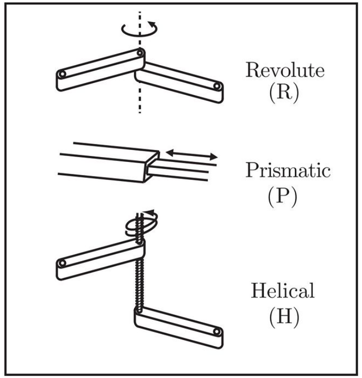


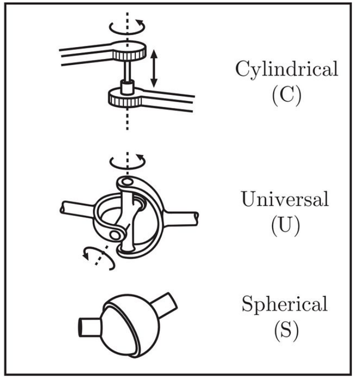
Robot Kinematics and Control

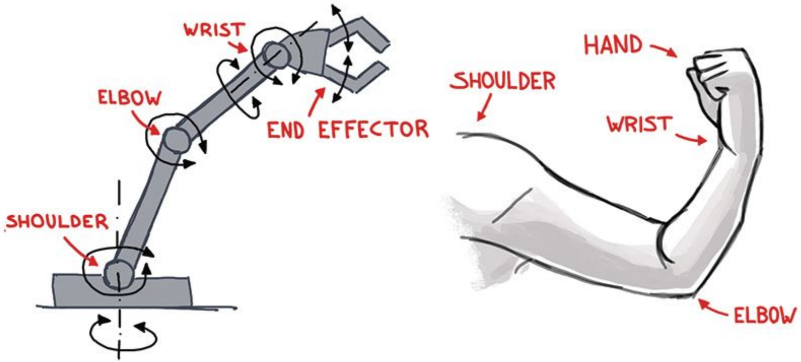
- **Degrees of Freedom (DOF)**: The number of independent ways a robot can move. Each joint adds 1 DOF (e.g., a 6-DOF arm = 6 joints).

- Types of joints: Revolute (rotational), Prismatic (linear) etc

💡 *More DOFs = More flexible, but harder to control*





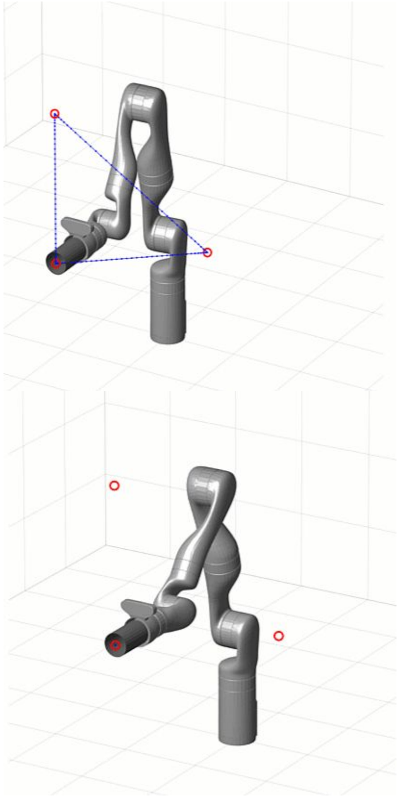


Robot Kinematics and Control

Joint Space	Cartesian Space
Robot's configuration (angles)	Position & orientation in 3D space
Easier to control directly	Task-relevant, human-readable
E.g., *[q1, q2, ..., qn]*	E.g., *[x, y, z, roll, pitch, yaw]*

💡 *Robot operates in **joint space**, but we often **command it in Cartesian space**.*

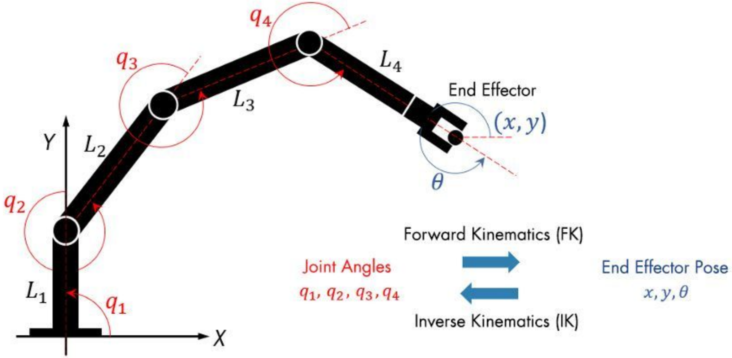
- **Position Control**: Set joint angles or end-effector position
- **Velocity Control**: Control joint or end-effector velocity
- **Torque Control**: Control the joint torques



Robot Kinematics and Control

Concept	Definition	Use Case Example
Forward Kinematics	Compute end-effector position from joint angles	“If I set these joint angles, where is the hand?”
Inverse Kinematics	Compute joint angles needed for desired end-effector pose	“I want the hand here — what should the joint angles be?”

💡 *IK is harder, since it may have multiple or no solutions, and is often nonlinear.*



Robot Kinematics and Control

Inverse Kinematics (IK)

- Here, we use IK to obtain the joint positions for the desired end-effector pose which is set to the object pose.

- Using the relationship between joint velocities and eef velocities,

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}}, \quad \dot{\mathbf{x}} \in \mathbb{R}^6, J \in \mathbb{R}^{6\times 7}, \dot{\mathbf{q}} \in \mathbb{R}^7$$

$$\Delta\mathbf{q} \approx J^+ \Delta\mathbf{x}$$

$$\boxed{\Delta\mathbf{q} \approx J^T(JJ^T + \lambda^2 I)^{-1}\Delta x}$$

* Robot *Dynamics* and Control

Operational Space Control (OSC)

- Operational Space Control (OSC) is a control method designed to command a robot's end-effector in task space rather than directly controlling the robot's joint torques or velocities.

$$\boxed{\tau_{\text{task}} = J^T M_{\text{eef}} \, f_{\text{task}}}$$

$$M_{\text{eef}} := (J M^{-1} J^T)^{-1}$$

$$f_{\text{task}} := K_p \Delta x + K_d \Delta \dot{x}$$

- Need to tune the gains Kp and Kd for improved performance.

Isaac Gym Simulation

Clone the below repo:

https://github.com/FaizalKarim280280/isaac-gym-controls