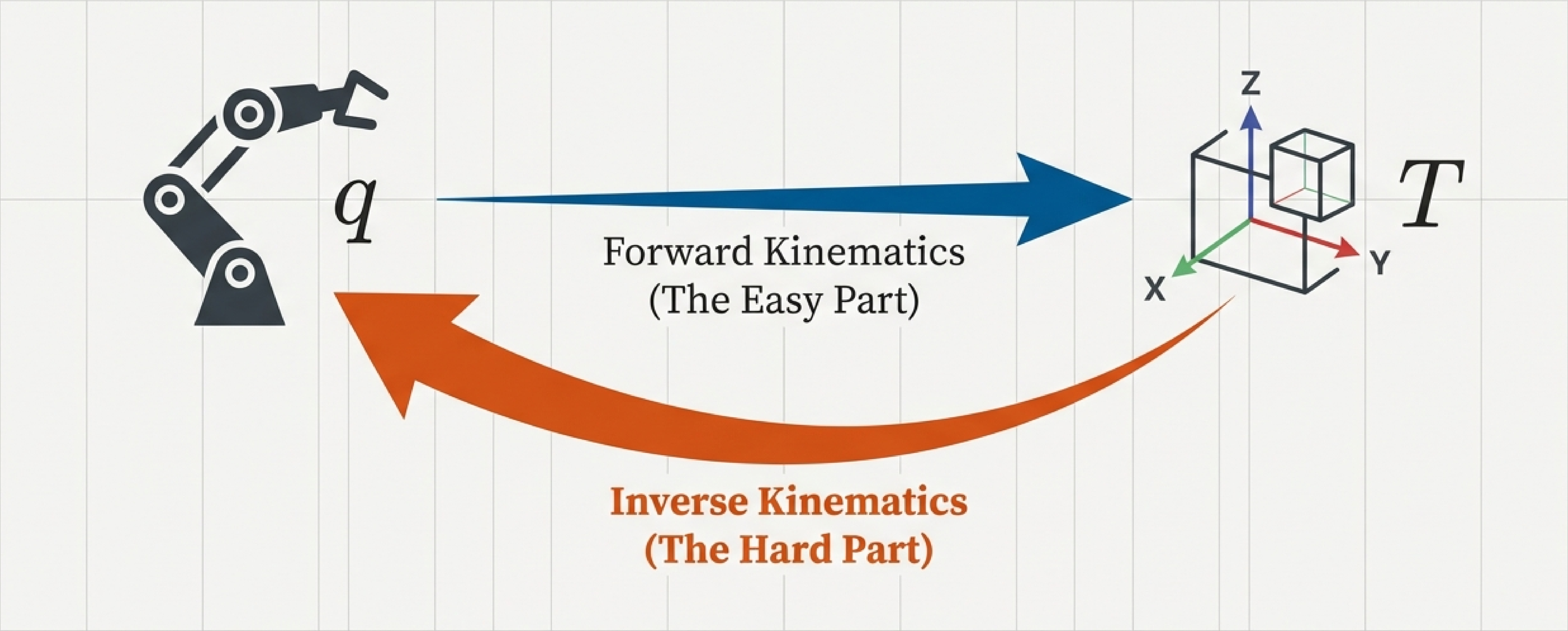
Inverse Kinematics: The Journey Backwards

Finding the Joint Commands for a Desired Pose



From Joint Space to Task Space, and Back Again

Recap: Forward Kinematics (FK)

Given a set of joint angles, what is the position and orientation of the end-effector?

Input: Joint Vector, $\mathbf{q} = [\theta_1, \theta_2, \ldots, \theta_n]$

Process: Apply Denavit-Hartenberg (DH) convention to compute the total transformation matrix.

$$ {}^{0}_{N}T = {}^{0}_{1}T(\theta_1) \cdot {}^{1}_{2}T(\theta_2) \cdot \ldots \cdot {}^{N-1}_{N}T(\theta_n) $$

Output: End-Effector Pose, $T = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix}$

A single, unique solution.
Computationally straightforward.

The Challenge: Inverse Kinematics (IK)

Given a desired position and orientation for the end-effector, what are the required joint angles?

Input: Desired End-Effector Pose,
$T_{\text{des}} = [\mathbf{R}_{\text{des}}\ \mathbf{p}_{\text{des}}; 0\ 1]$

Process: Solve a system of non-linear trigonometric equations.

Output: Joint Vector(s), $\mathbf{q} = [\theta_1, \theta_2, \ldots, \theta_n]$

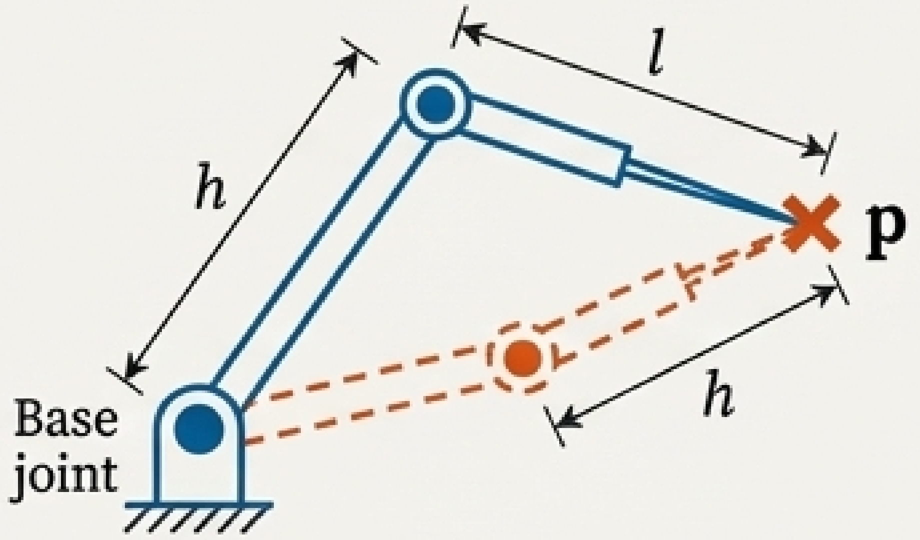
A complex, non-linear problem. The solution may not be unique, or may not exist at all.

The Complexities of the Inverse Problem

Multiplicity of Solutions

A single end-effector pose can often be achieved by multiple distinct robot configurations. A common example is the “elbow up” vs. “elbow down” configuration in a simple arm. For a 6-DOF arm, there can be up to 8 or more solutions.

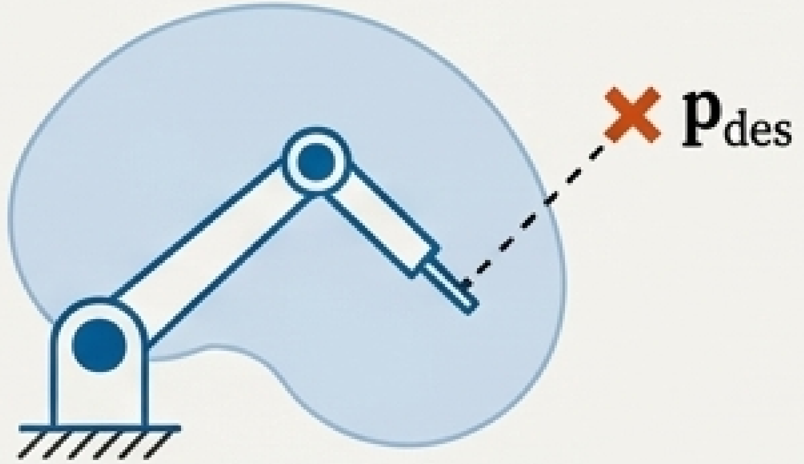
Implication: How do we choose the “best” solution? We often select the one closest to the current configuration to minimize joint travel.



Existence of Solutions (Workspace)

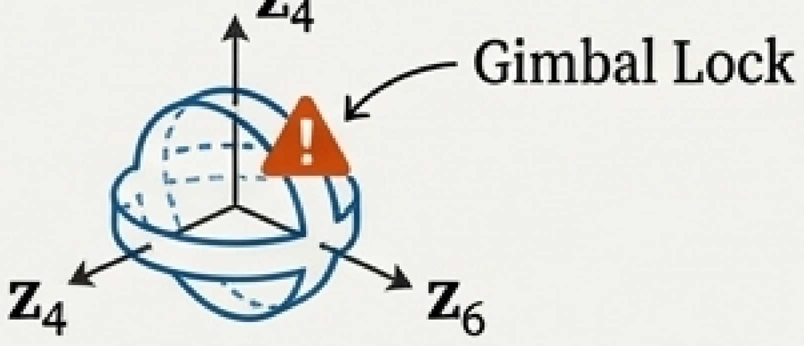
A desired pose may be unreachable if it lies outside the robot’s workspace.

Implication: The IK algorithm must be able to determine if a solution exists. If not, the target pose is physically impossible to attain.

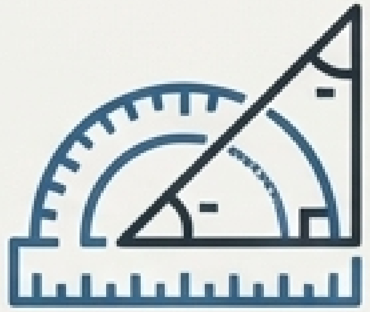


Singularities

At certain configurations (singularities), the robot loses one or more degrees of freedom. A common example is “gimbal lock” in a spherical wrist, where two joint axes align, causing infinite possible solutions for their individual angles.



Approaches to Finding Analytical Solutions



Method 1: Geometric Solutions

Core Idea: Decompose the spatial geometry of the manipulator into a set of plane trigonometry problems.

Process: Use geometric relationships like the Law of Cosines and trigonometric identities to solve for joint angles directly from the robot's geometry.

Strengths: Highly intuitive and easy to visualize.

Limitations: Generally only feasible for simple robots with few degrees of freedom (typically ≤ 3 DOF) and specific geometries (e.g., intersecting joint axes).

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \left\{ \begin{aligned} f(\mathbf{q}) &= x_{\text{des}} \\ g(\mathbf{q}) &= y_{\text{des}} \end{aligned} \right\}$$

Method 2: Algebraic Solutions

Core Idea: Use the forward kinematics equations (e.g., from DH parameters) to establish a system of non-linear equations.

Process: Equate the FK transformation matrix ${}^{0}_{N}T(\mathbf{q})$ with the desired pose matrix T_{des} and solve the resulting system of equations for the unknown joint variables $\mathbf{q}$.

Strengths: A more general and powerful method that can be applied to complex, high-DOF manipulators (like 6-DOF industrial arms).

Limitations: Less intuitive; the ‘magical algebra’ can be complex and requires careful manipulation of equations.

Case Study 1: The Planar 2R Robot

Problem Statement

For the planar two-link manipulator shown, find the joint angles q_1 and q_2 that place the end-effector at a desired position (x, y).

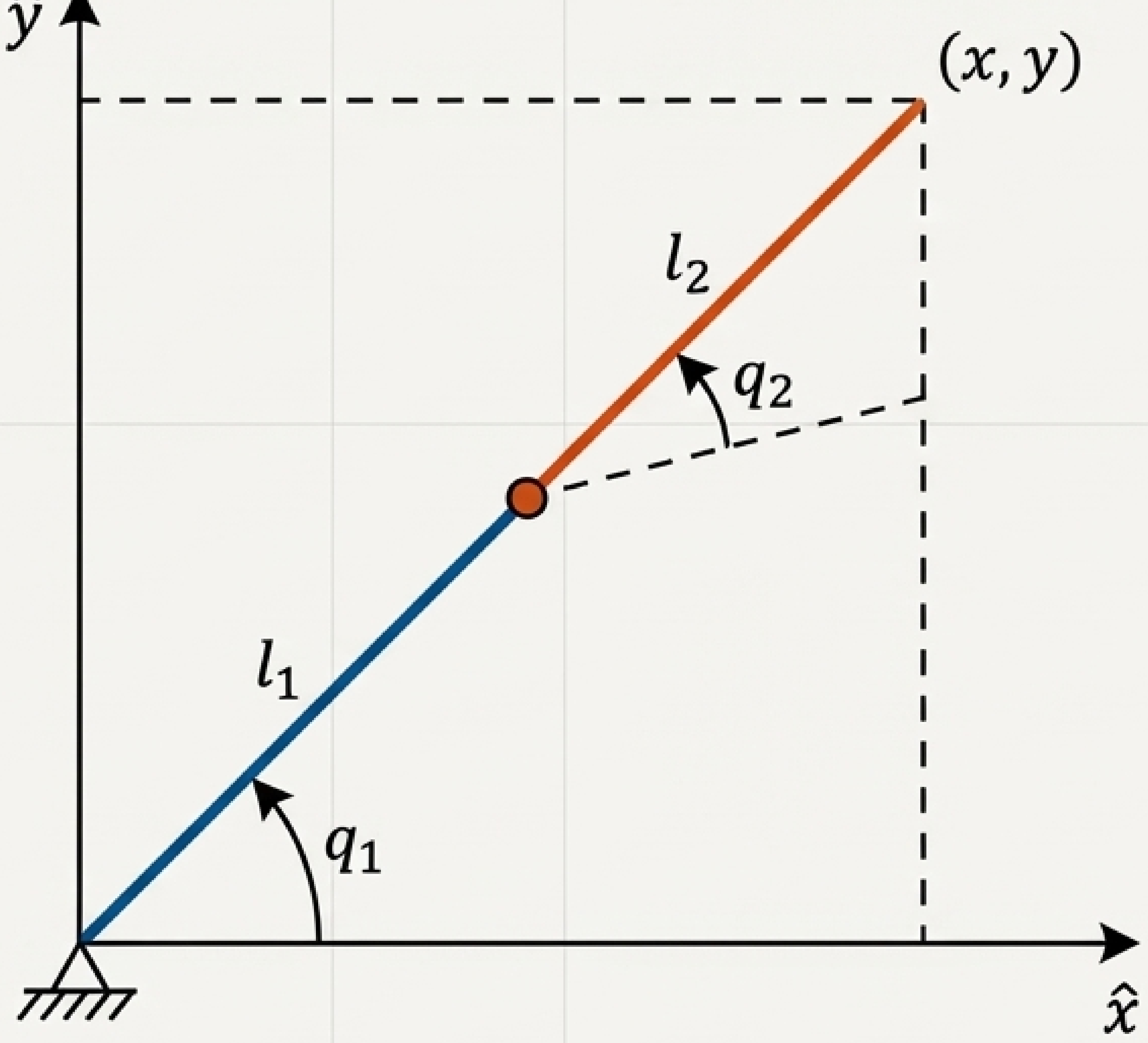
Knowns:

- Link lengths: l_1, l_2
- Desired end-effector position: (x, y)

Unknowns:

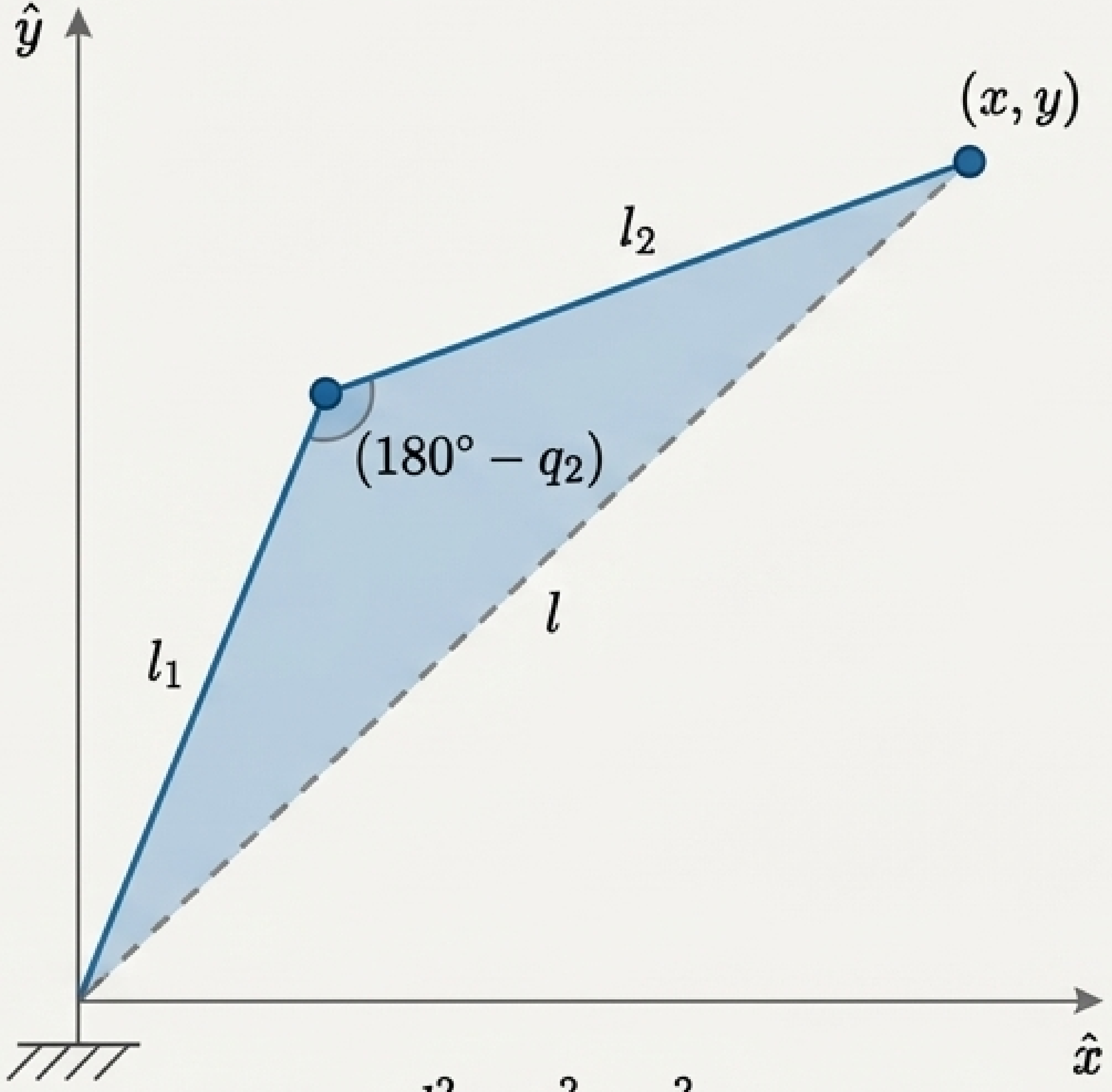
- Joint angles: q_1, q_2

Approach: We will use a geometric solution.



Solving the 2R Robot: Finding q_2 with the Law of Cosines

Geometric Insight



$$l^2 = x^2 + y^2$$

Derivation

Step 1: Apply the Law of Cosines

$$l^2 = l_1^2 + l_2^2 - 2l_1l_2\cos(180° - q_2)$$

Since $\cos(180° - \theta) = -\cos(\theta)$, this simplifies:

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2\cos(q_2)$$

Step 2: Solve for $\cos(q_2)$ (c_2)

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}$$

Step 3: Find $\sin(q_2)$ (s_2)

Using the identity $s_2^2 + c_2^2 = 1$, we get:

$$s_2 = \boxed{\pm}\sqrt{1 - c_2^2}$$

This $\pm$ is the source of the two solutions: "elbow up" ($s_2 > 0$) and "elbow down" ($s_2 < 0$).

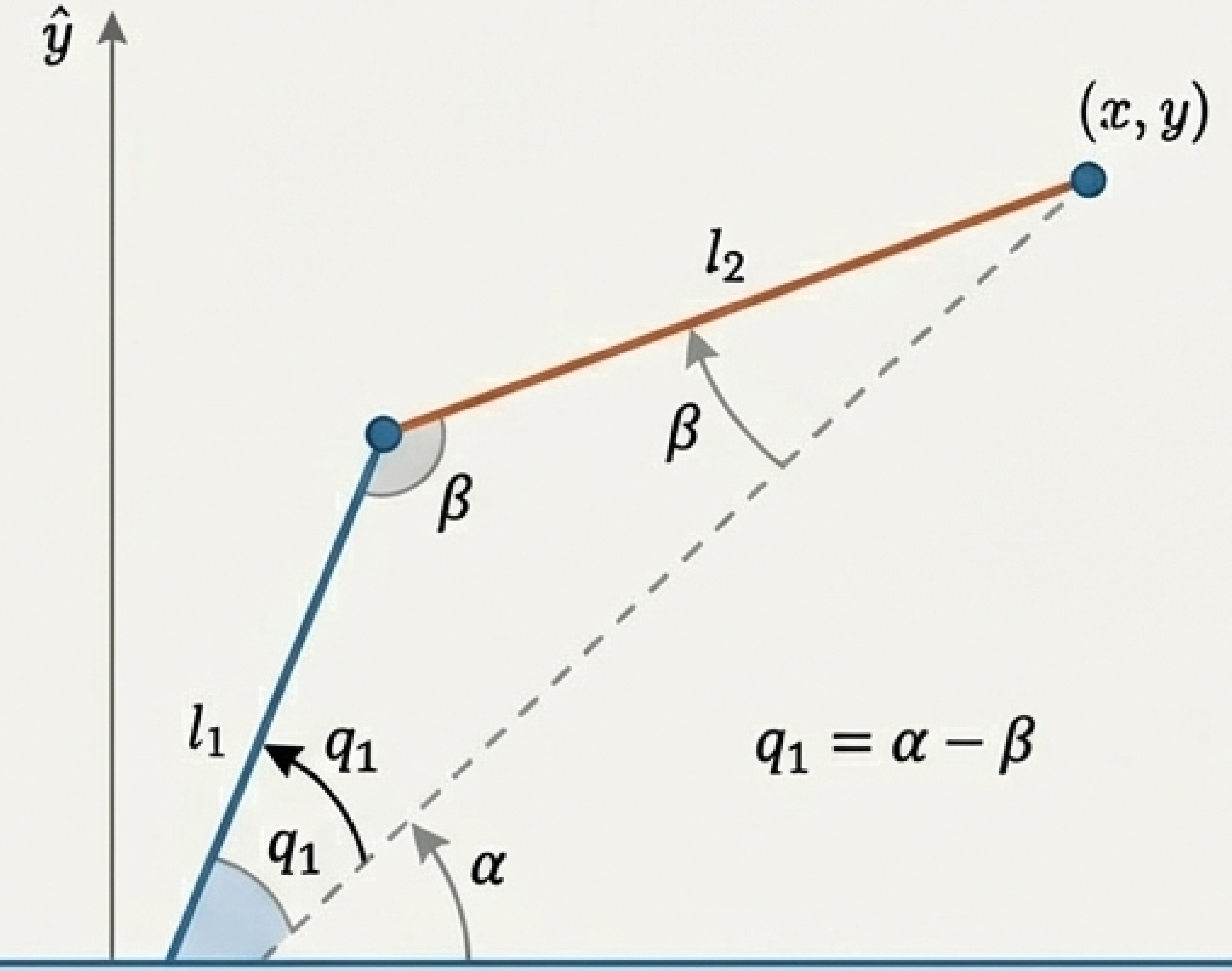
Step 4: Solve for q_2

Using the two-argument arctangent function for robustness:

$$q_2 = \text{atan2}(s_2, c_2)$$

Solving the 2R Robot: Finding q_1

Geometric Insight



Derivation

Step 1: Solve for angle α

From basic trigonometry, α is the angle of the vector (x, y).

$$\alpha = \operatorname{atan2}(y, x)$$

Step 2: Solve for angle β

Using the geometry of the triangle, we can express the opposite side ($l_2 s_2$) and adjacent side ($l_1 + l_2 c_2$) of angle β.

$$\beta = \operatorname{atan2}(l_2 s_2, l_1 + l_2 c_2)$$

Step 3: Combine to find q_1

$$q_1 = \alpha - \beta = \operatorname{atan2}(y, x) - \operatorname{atan2}(l_2 s_2, l_1 + l_2 c_2)$$

Complete IK Solution for Planar 2R Robot

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2} \qquad s_2 = \pm\sqrt{1 - c_2^2} \qquad q_2 = \operatorname{atan2}(s_2, c_2) \qquad q_1 = \operatorname{atan2}(y, x) - \operatorname{atan2}(l_2 s_2, l_1 + l_2 c_2)$$

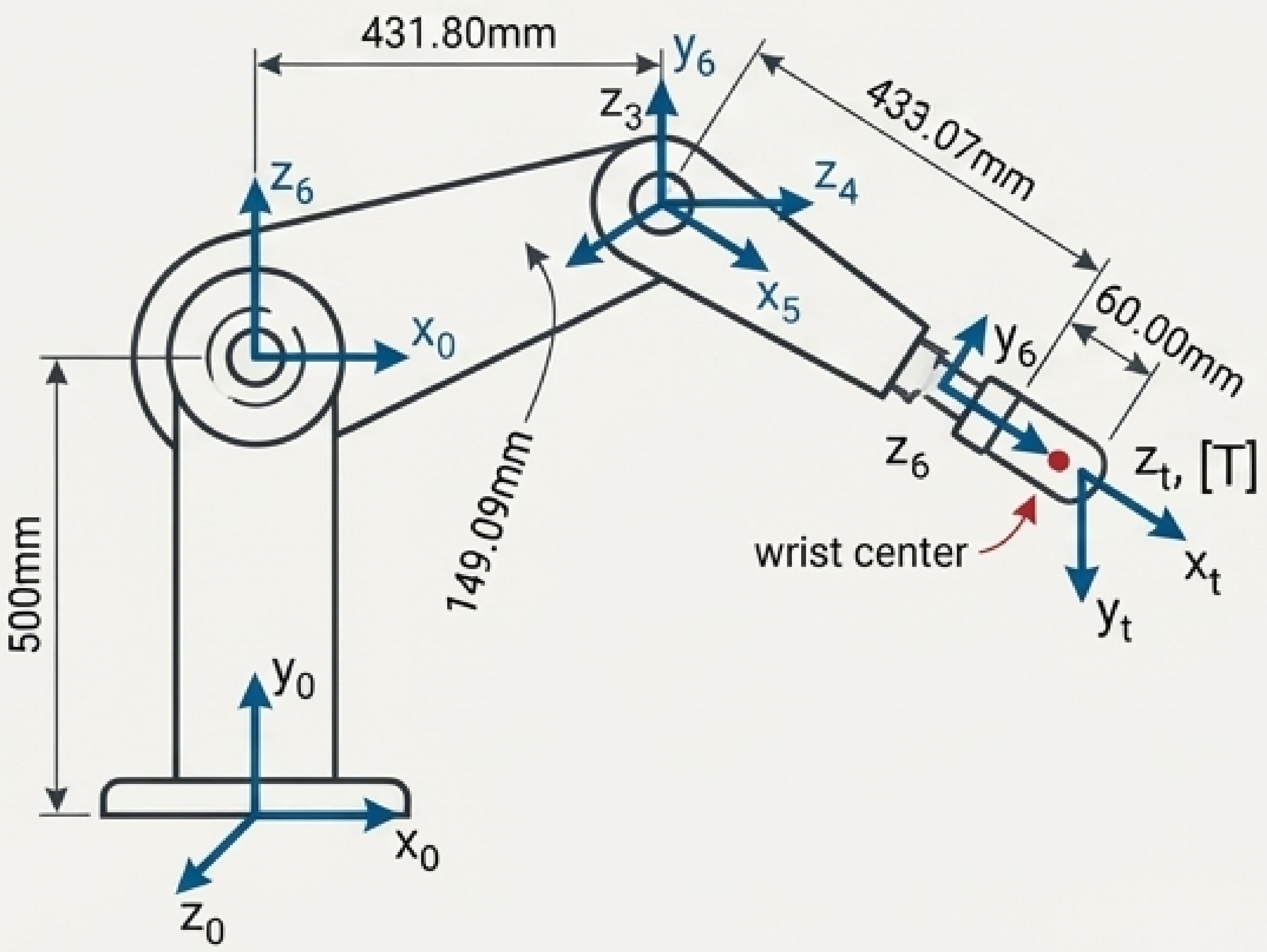
(**Note:** There are two solutions for (q_1, q_2) corresponding to the sign choice for s_2.)

Scaling Up: Inverse Kinematics for a 6-DOF Manipulator

The geometric approach provided clear intuition for the 2R arm. However, for manipulators with more degrees of freedom and complex, non-planar geometry, this method becomes intractable.

Case Study 2: The PUMA 560 Robot

A classic 6-DOF articulated robot arm. Its structure includes offsets and non-intersecting joint axes, making simple geometric decomposition impossible. We must turn to a more powerful algebraic approach based on its DH parameters.



Link	α_i (deg)	a_i (mm)	d_i (mm)	θ_i (deg)
1	0	0	0	θ_1
2	-90	0	0	θ_2
3	0	431.8	149.09	θ_3
4	-90	20.3	433.07	θ_4
5	90	0	0	θ_5
6	-90	0	0	θ_6

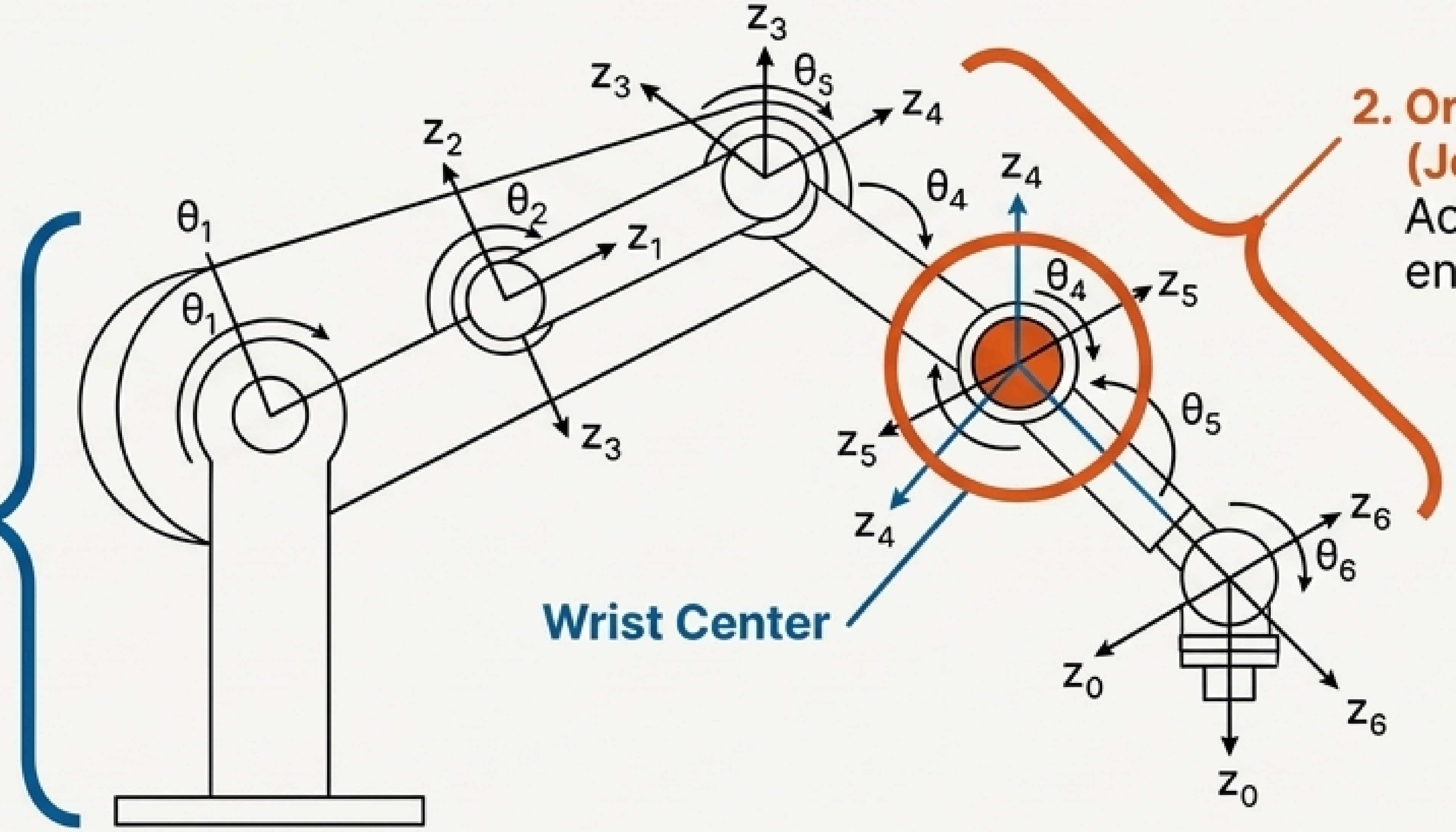
The Algebraic Strategy: Kinematic Decoupling

The Challenge We need to solve the matrix equation ${}^0_6T(\theta_1, \ldots, \theta_6) = T_{\text{des}}$ for six unknown angles. Solving this directly is extremely difficult.

The Insight Many 6-DOF robots, including the PUMA 560, are designed with a spherical wrist. This means the last three joint axes (z_4, z_5, z_6) intersect at a single point, the wrist center.

The Strategy: Decouple the Problem

1. Positioning Problem (Joints $\theta_1, \theta_2, \theta_3$):
Position the wrist center at the correct location.



2. Orientation Problem (Joints $\theta_4, \theta_5, \theta_6$):
Achieve the desired end-effector orientation.

Solving for Position: Joints θ_1, θ_2 and θ_3

Goal: Find $(\theta_1, \theta_2, \theta_3)$ that place the wrist center $\mathbf{p}_{wc}$ at the desired location derived from T_{des}.

Step 1: Solve for θ_1

Insight: By viewing the robot from above (in the x_0-y_0 plane), θ_1 determines the plane in which the rest of the arm operates.

$$-\sin(\theta_1) \cdot p_x + \cos(\theta_1) \cdot p_y = d_3$$

Annotation: This is of the form $A\cos\theta + B\sin\theta = C$, which yields two solutions for θ_1, corresponding to a front-side or back-side reach.

Step 2: Solve for θ_3

Insight: With θ_1 known, we analyze the arm's geometry in the vertical plane defined by θ_1. Using a 3D version of the Law of Cosines, we relate the distance from the second joint to the wrist center with θ_3.

$$\left(p_x^2 + p_y^2 + p_z^2\right) - \left(a_2^2 + a_3^2 + d_3^2 + d_4^2\right) = 2a_2(a_3\cos\theta_3 - d_4\sin\theta_3)$$

Annotation: This also has the form $A\cos\theta + B\sin\theta = C$, yielding two solutions for θ_3 (elbow up/down).

Step 3: Solve for θ_2

Insight: For each of the four (θ_1, θ_3) pairs, θ_2 is now constrained. We can form two linear equations in $\sin(\theta_2)$ and $\cos(\theta_2)$ and solve for them uniquely.

$$\theta_2 = \operatorname{atan2}(s_2, c_2)$$

Result: We have found 4 possible sets of $(\theta_1, \theta_2, \theta_3)$ to position the wrist center correctly.

NotebookLM

Solving for Orientation: The Spherical Wrist θ_4, θ_5, and θ_6

Goal: For each of the 4 arm configurations, find $(\theta_4, \theta_5, \theta_6)$ to match the desired end-effector orientation $\mathbf{R}_{des}$.

Step 1: Isolate the Wrist Rotation

Insight: We know the total desired rotation ${}^0_6R$ (from $\mathbf{T}_{des}$). For a given $(\theta_1, \theta_2, \theta_3)$, we can compute the rotation of the arm ${}^0_3R$. The required wrist rotation ${}^3_6R$ is then found by:

$$ {}^3_6R = ({}^0_3R)^{-1} \cdot {}^0_6R = ({}^0_3R)^T \cdot {}^0_6R $$

Step 2: Solve as a ZYZ Euler Angle Problem

Insight: The rotation matrix ${}^3_6R$ corresponds to successive rotations about z, y, and z axes. By equating the elements of the calculated ${}^3_6R$ matrix with the symbolic ZYZ Euler matrix, we can extract the angles.

$$ \theta_5 = \pm\arccos(R_{3,3}) $$

$$ \theta_4 = \operatorname{atan2}\left(\frac{R_{2,3}}{\sin\theta_5}, \frac{R_{1,3}}{\sin\theta_5}\right) $$

$$ \theta_6 = \operatorname{atan2}\left(\frac{R_{3,2}}{\sin\theta_5}, -\frac{-R_{3,1}}{\sin\theta_5}\right) $$

Note: A singularity occurs if $\sin\theta_5 = 0$, where $\theta_5 = 0$ or $180°$.

Result: For each of the 4 arm configurations, we find 2 possible wrist configurations, giving a total of $4 \times 2 = 8$ solutions.

The Journey Backwards: A Recap

Inverse Kinematics is the inverse of Forward Kinematics.

Given a desired pose T, we solve for the joint angles q required to achieve it.

It is a fundamentally non-linear and challenging problem.

Solutions may not exist (workspace limits).
Solutions are often not unique (multiple configurations).

Analytical solutions provide exact joint values.

Geometric Approach: Intuitive, visual, best for simple manipulators (e.g., Planar 2R).
Algebraic Approach: Powerful, general, necessary for complex robots (e.g., PUMA 560).
Uses kinematic decoupling for robots with spherical wrists.

The Big Picture: A robust IK solver is a cornerstone of robotics, enabling tasks like path following, trajectory generation, and interaction with the environment.

Next Lecture: Motion Planning. Now that we can find the joint angles for a *single* pose, how do we generate smootlh trajectories between poses?