Autonomous Robotics

Inverse Kinematics

Slides inspired from Mike Hagenow's Intro to Robotic Systems IK lecture

Josiah Hanna
University of Wisconsin — Madison

Announcements

Midterm in one week!

Grading: HW 2 is underway, everything else has been graded and returned to you.

Midterm survey released. Please complete ASAP!

https://forms.gle/CBxwteEa74ge5xMe6

Learning Outcomes

After today's lecture, you will:

- Be able to define the inverse kinematics problem.
- Understand analytical and geometric solutions to the IK problem.
- Understand optimization approaches to IK.

Forward Kinematics

Task space:

Position of a robot's end-effector. Assume $\mathbb{R}^n$.

Joint space:

Space of possible robot configurations (e.g., angle of all joints). Assume $\mathbb{R}^m$.

Forward kinematics is the mapping from joint space to task space:

$r = f(q)$, where $r \in \mathbb{R}^m$ and $q \in \mathbb{R}^n$.

Given a robot's joint configuration, determine where its end-effector is relative to a base frame of reference.

Inverse Kinematics

Task space:

Position of a robot's end-effector. Assume $\mathbb{R}^n$.

Joint space:

Space of possible robot configurations (e.g., angle of all joints). Assume $\mathbb{R}^m$.

Inverse kinematics is the mapping from a point in task space to joint space:

$q = f^{-1}(r)$, where $r \in \mathbb{R}^m$ and $q \in \mathbb{R}^n$.

Given a desired position of the end-effector, determine the robot's joint configuration to put the end-effector there. Why useful?

Note: not all desired positions are achievable. Set of achievable positions = the robot's workspace.

IK: Number of Solutions

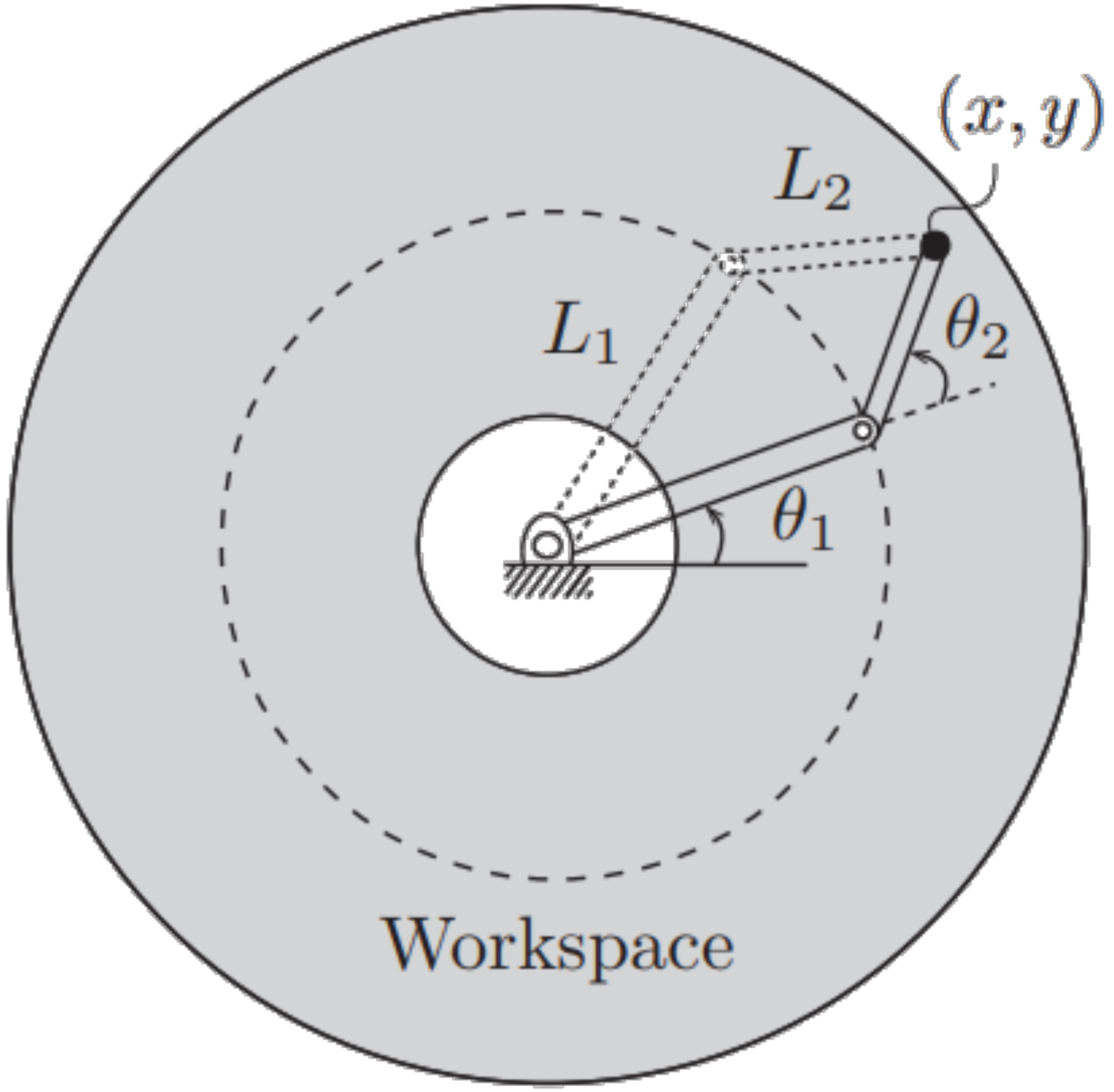
Depends on the joints (e.g., lengths, number/redundancy)

Number of solutions:

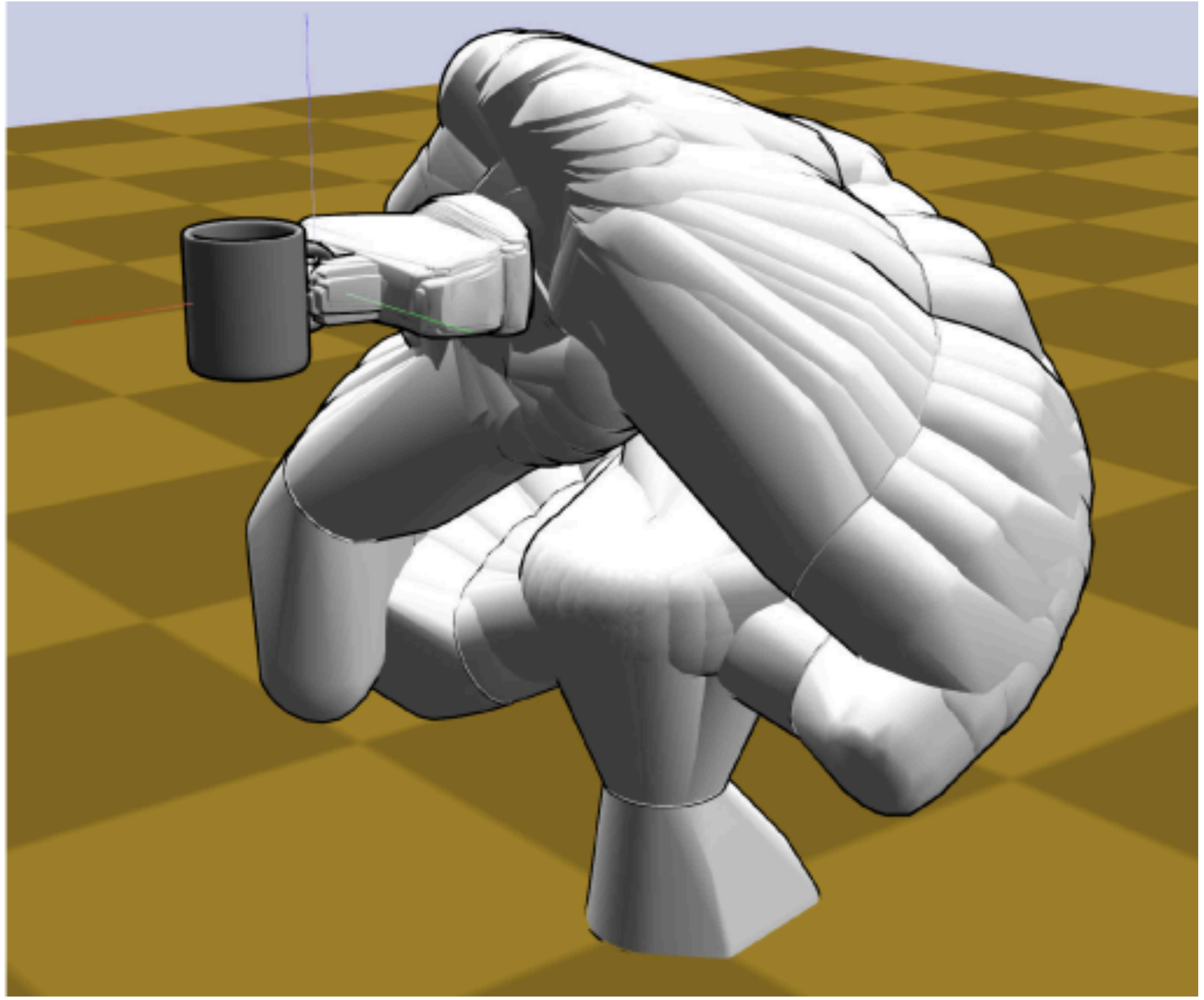
- Zero: outside of workspace

- A finite number of solutions

- Infinite solutions: when have redundant joints.

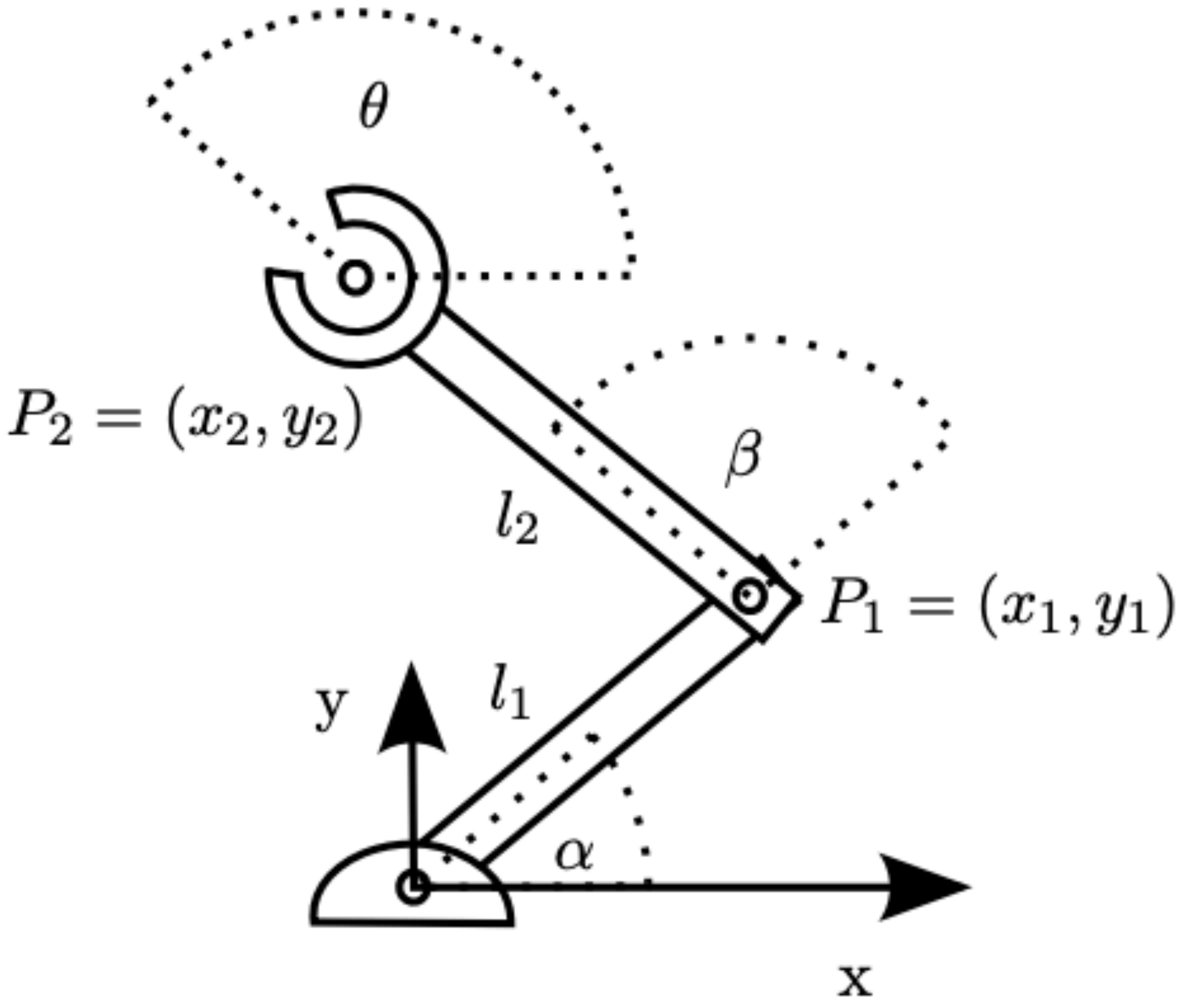


Redundancy Example



https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793576

IK Example



Given: (x, y, θ)

Determine: α, β

$$\left.\begin{aligned} \cos\theta &= \cos(\alpha+\beta) \\ x &= l_2\cos(\alpha+\beta) + l_1\cos\alpha \\ y &= l_2\sin(\alpha+\beta) + l_1\sin\alpha \end{aligned}\right\}$$

From forward kinematics

Then solve for α, β:

$$\theta = \alpha + \beta$$

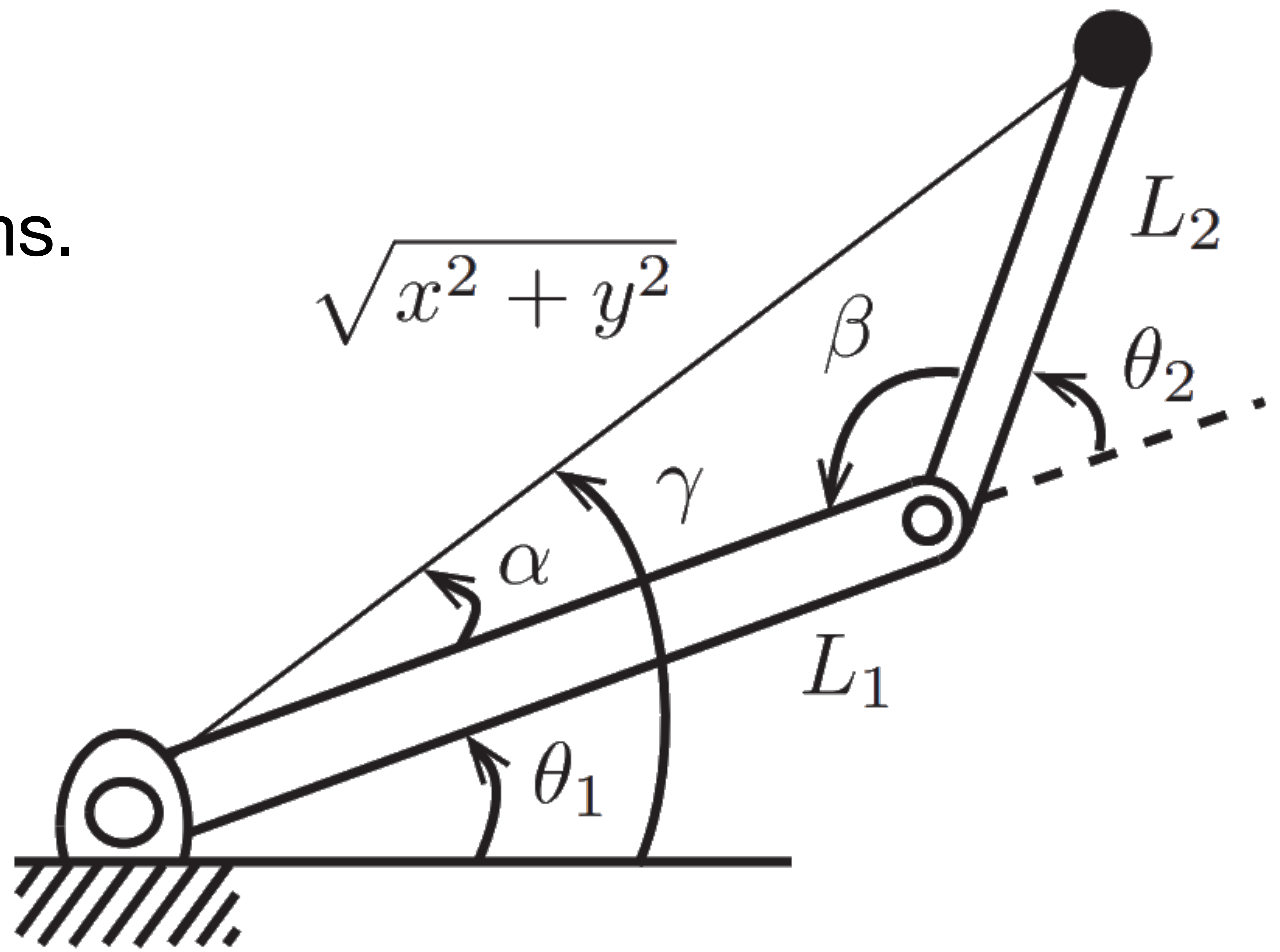
$$\cos\alpha = \frac{l_2\cos\theta - x}{l_1} \quad \sin\alpha = \frac{l_2\sin\theta - y}{l_1}$$

Inverse Kinematics

Let's try it without θ.

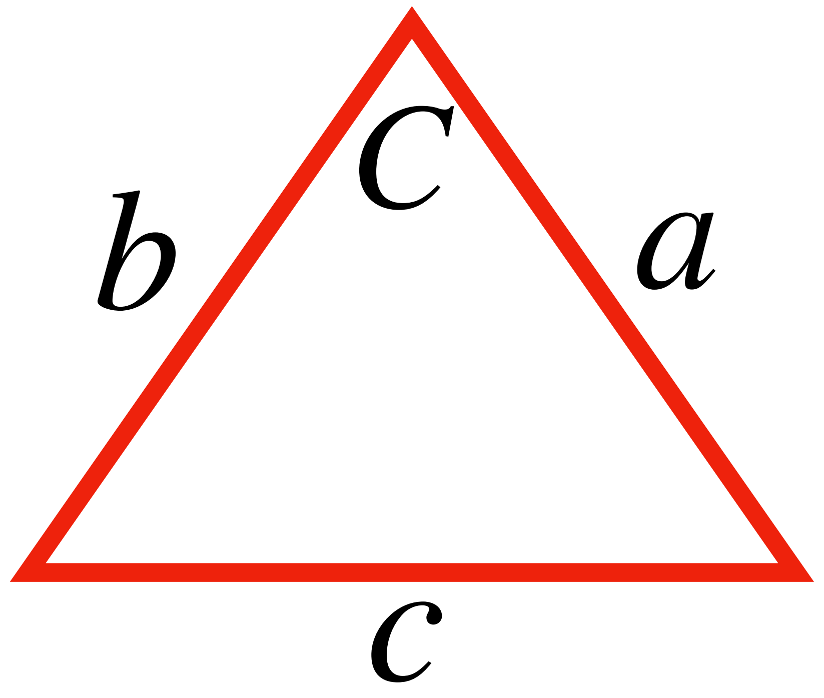
Can be non-trivial even with simple arms.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$



Law of Cosines:

$$c^2 = a^2 + b^2 - 2ab\cos C$$



Analytic IK

3D Printers

Kuka KR 210

General IK Approach

In general, it may be challenging to solve for $f^{-1}(r)$ directly.

Instead, try to find $\hat{q} \approx f^{-1}(r)$.

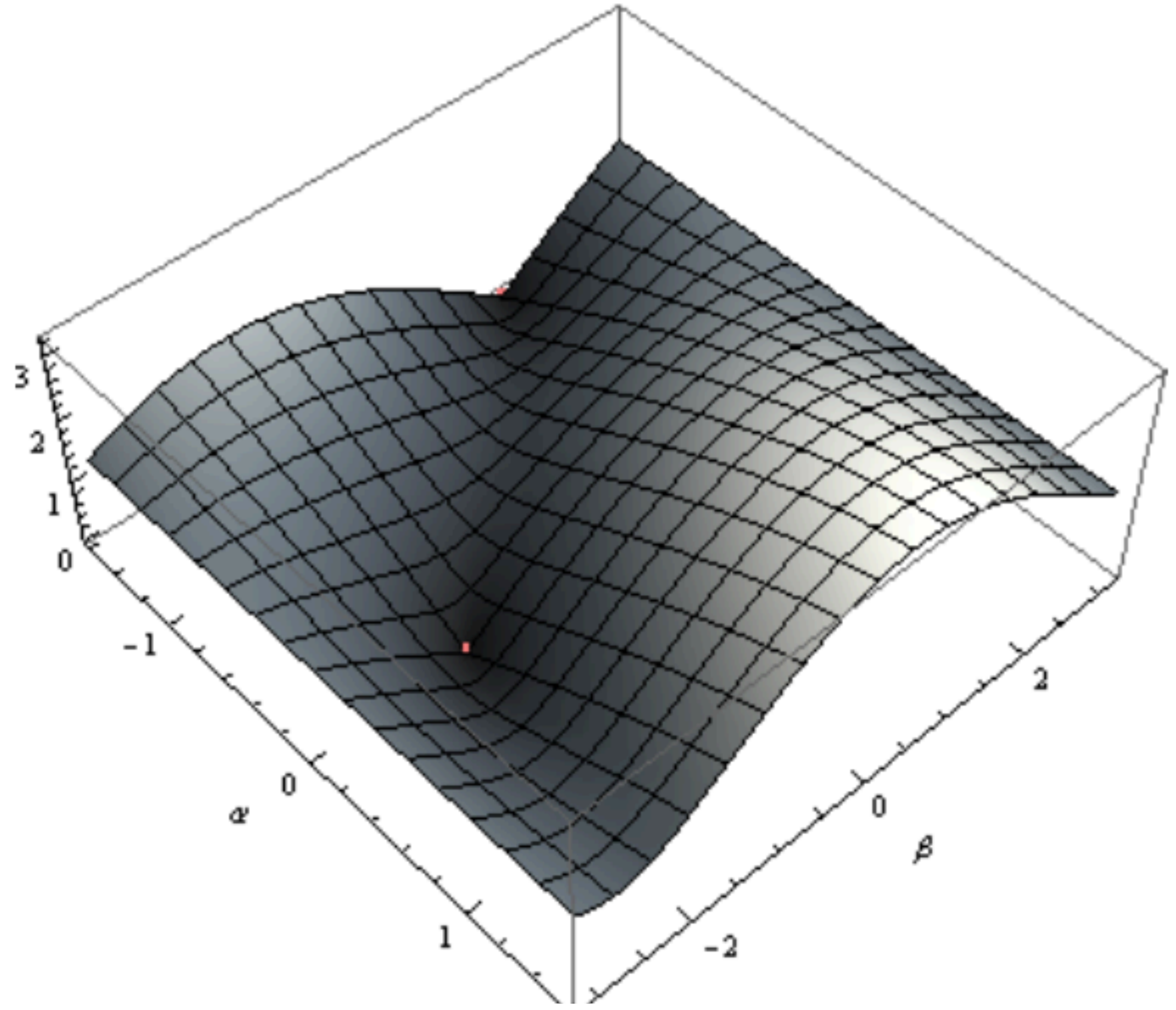
Specify a loss function:

$$l_r(q) = ||r - f(q)||_2$$

Use optimization to find the minimizer, q^*,

such that $f(q^*) \approx r$.

Ex: Newton Raphson, Levenberg-Marquardt



$$f_{x,y}(\alpha, \beta) = \sqrt{(s_{\alpha\beta} + s_\alpha - y)^2 + (c_{\alpha\beta} + c_\alpha - x)^2}$$

Assuming $l_1 = l_2 = 1$

$\sin \alpha\beta = \sin(\alpha + \beta)$

Differential IK

Recall differential kinematics: $\dot{r} = J(q) \cdot \dot{q}$ where $J(q) \in \mathbb{R}^{m \times n}$ is the Jacobian of $f(q)$ and $\dot{q}$ is the time derivative of q (i.e., rate of change in q).

Differential IK: given a desired velocity in task space, determine the desired velocity that achieves it.

First idea: use $J^{-1}(q) \cdot \dot{r} = \dot{q}$. Problem?

Second idea: use the pseudo-inverse, $J^{+}(q)$. Problem?

Third idea: damp joint velocities. $\Delta q = (J^{\top} J + \lambda^2 I)^{-1} J^{+} e$

IK for Differential Drive

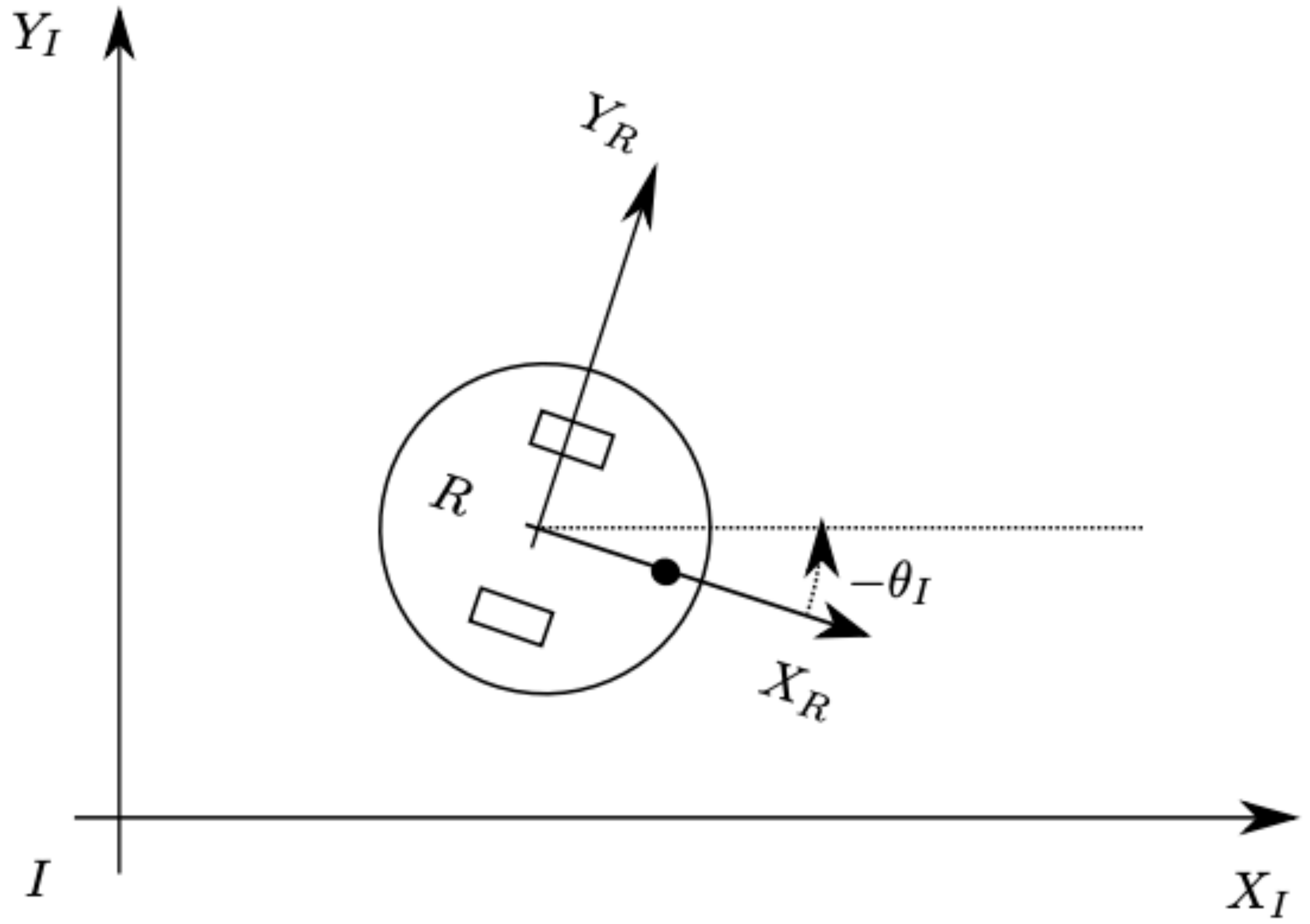
Given desired change in x, y, and θ, compute velocity for left and right wheel to achieve it.

Step 1: transform desired $\dot{x}, \dot{y}, \dot{\theta}$ to desired $\dot{x}_R, \dot{\theta}_R$.

Step 2: solve for $\dot{\phi}_l, \dot{\phi}_r$

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} \dot{\phi}_l \\ \dot{\phi}_r \end{bmatrix} = \begin{bmatrix} \frac{\partial x_R}{\partial \dot{\phi}_l} & \frac{\partial x_R}{\partial \dot{\phi}_r} \\ \frac{\partial y_R}{\partial \dot{\phi}_l} & \frac{\partial y_R}{\partial \dot{\phi}_r} \\ \frac{\partial \theta}{\partial \dot{\phi}_l} & \frac{\partial \theta}{\partial \dot{\phi}_l} \end{bmatrix} \begin{bmatrix} \dot{\phi}_l \\ \dot{\phi}_r \end{bmatrix}$$

Note: ignore $\dot{y}_R$ as we assume the robot cannot move sideways.



$$\dot{\phi}_l + \dot{\phi}_r = 2\dot{x}_R / r$$
$$\dot{\phi}_l - \dot{\phi}_r = d\dot{\theta}_R / r$$

IK for Differential Drive

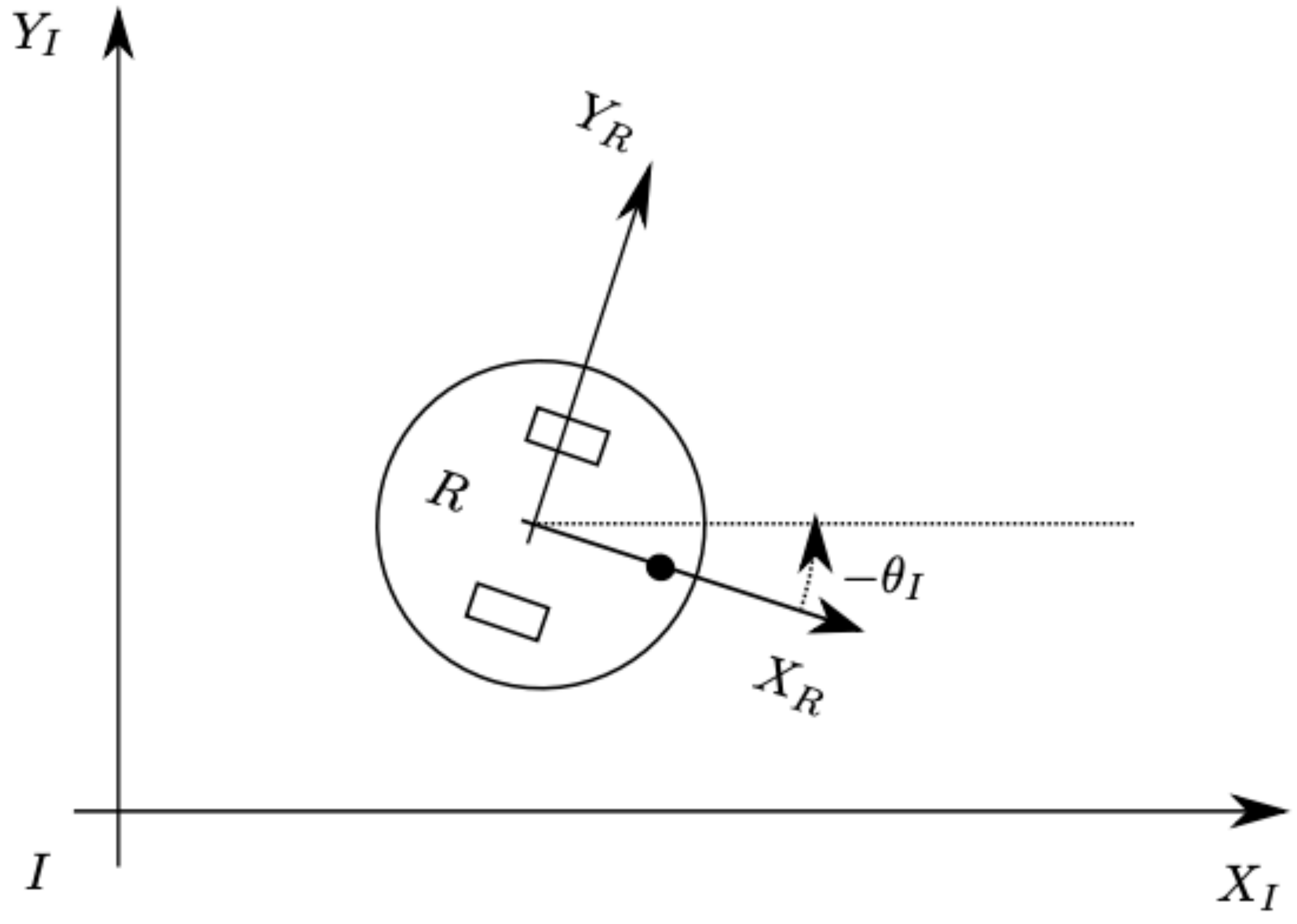
How to obtain a desired pose in inertial frame?

First determine error:

$$\rho = \sqrt{(x_r - x_g)^2 + (y_r - y_g)^2}$$

$$\alpha = \tan^{-1} \frac{y_g - y_r}{x_g - x_r} - \theta_r$$

$$\eta = \theta_g - \theta_r$$



Then use controller to minimize error:

$$\dot{x} = p_1 \rho \qquad \dot{\theta} = p_2 \alpha + p_3 \eta$$

IK in Practice

Targets not set in isolation.

Can use task knowledge to select joint configurations strategically.

Perform a *reconfiguration*

https://graphics.cs.wisc.edu/Papers/2024/WSG24/

https://graphics.cs.wisc.edu/Papers/2018/RMG18a/

Summary

- Introduced inverse kinematics problem.
- Saw how to compute IK solutions
- Introduced differential inverse kinematics.

Action Items

- Planning reading for next week; send a reading response by 12 pm on Monday.

- Midterm: on Tuesday!

$$(l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2), l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2))$$