NYU

Introduction to Robot Intelligence
[Spring 2023]

Inverse Kinematics

March 2, 2023

Lerrel Pinto

Recap of Forward Kinematics

- Forward kinematics: The use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters.

Today's class – Inverse of the problem

Image credits: Matlab Simulink, Najam Syed.

Today's class – Inverse of the problem

- Inverse kinematics: The use of the kinematic equations of a robot to compute the joint parameters from specified values of position of the end-effector.

Image credits: Matlab Simulink, Najam Syed.

Today's class – Inverse of the problem

- Inverse kinematics: The use of the kinematic equations of a robot to compute the joint parameters from specified values of position of the end-effector.

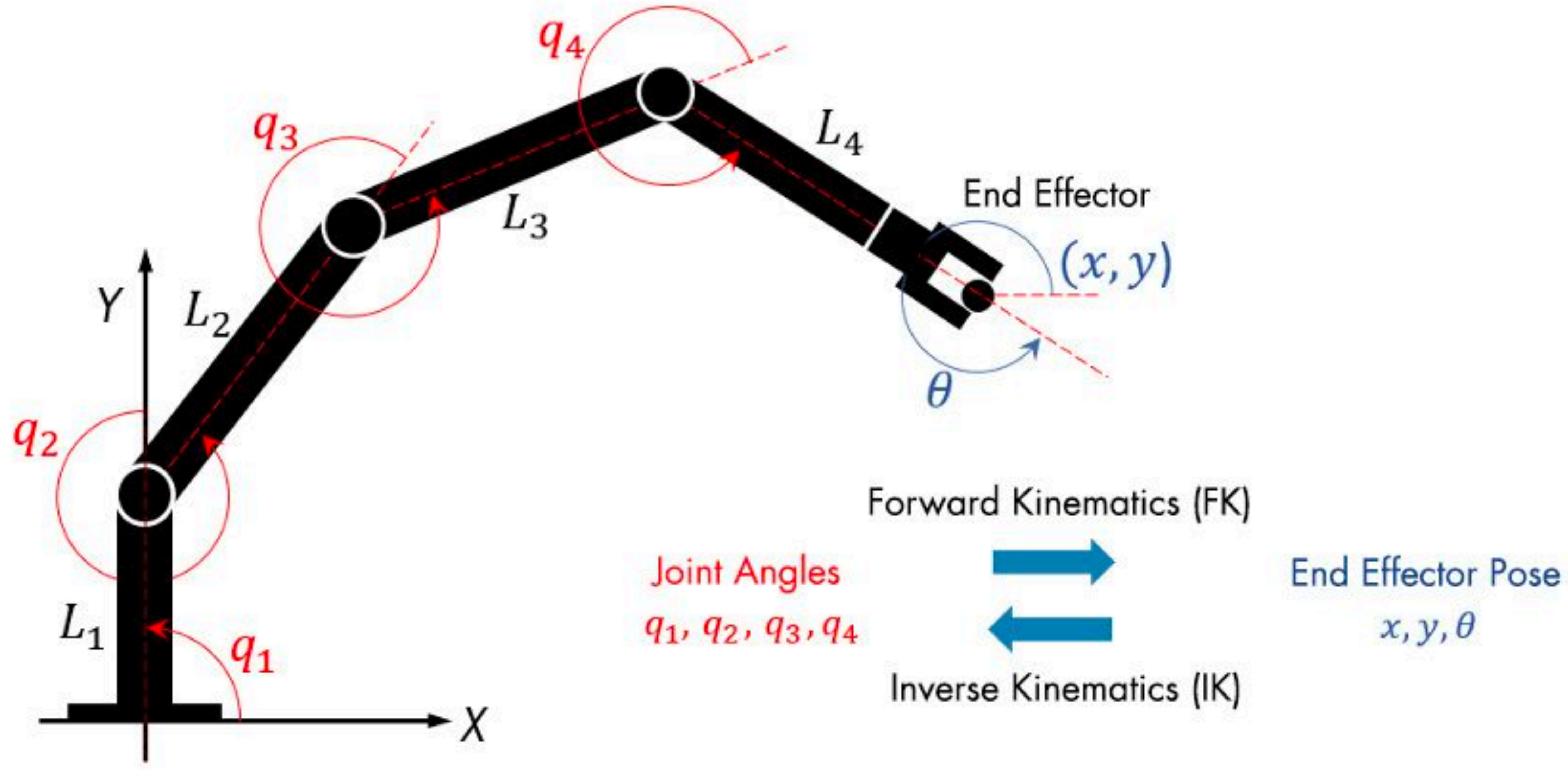
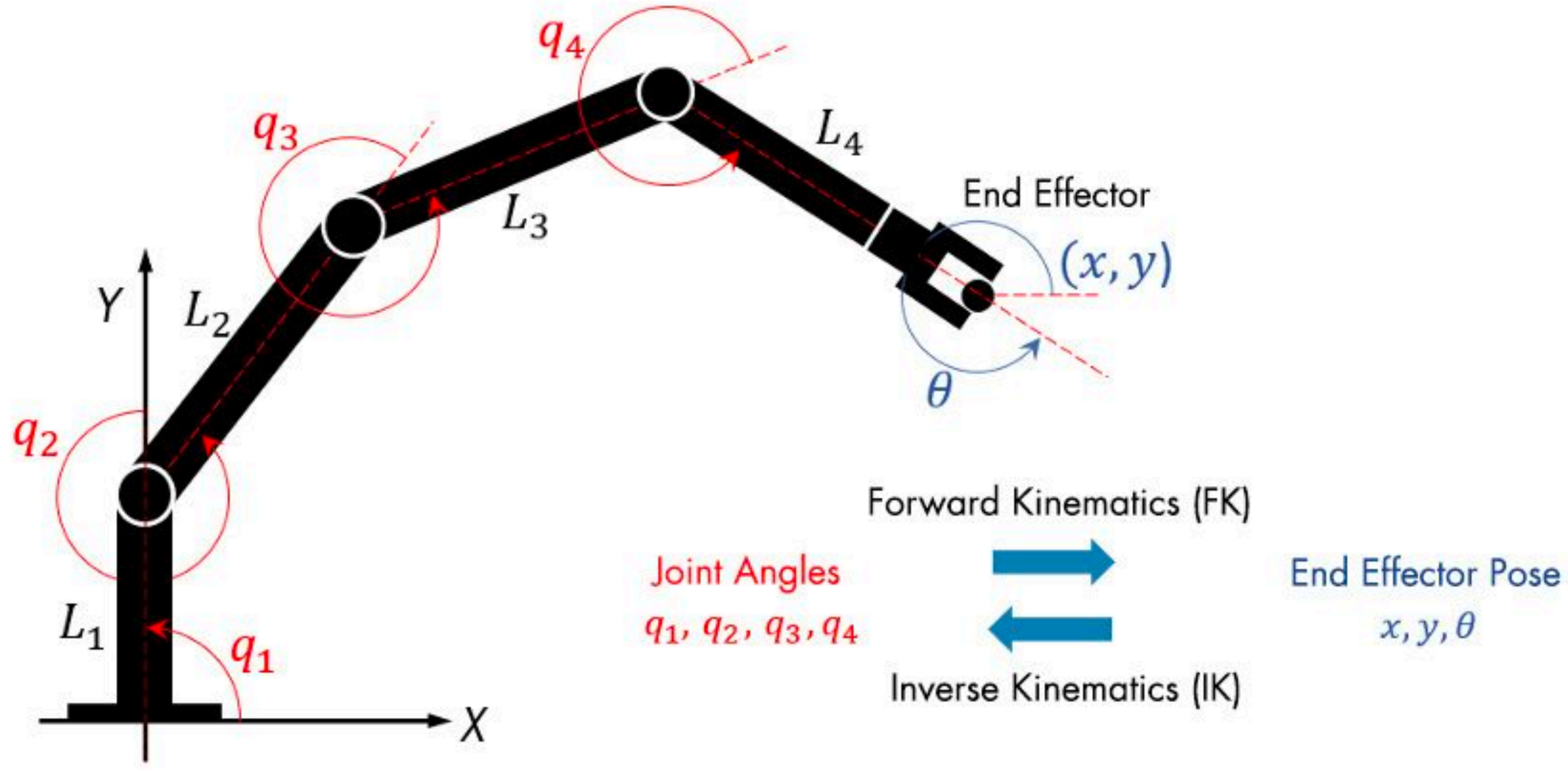


Image credits: Matlab Simulink, Najam Syed.

Today's class – Inverse of the problem

- Inverse kinematics: The use of the kinematic equations of a robot to compute the joint parameters from specified values of position of the end-effector.



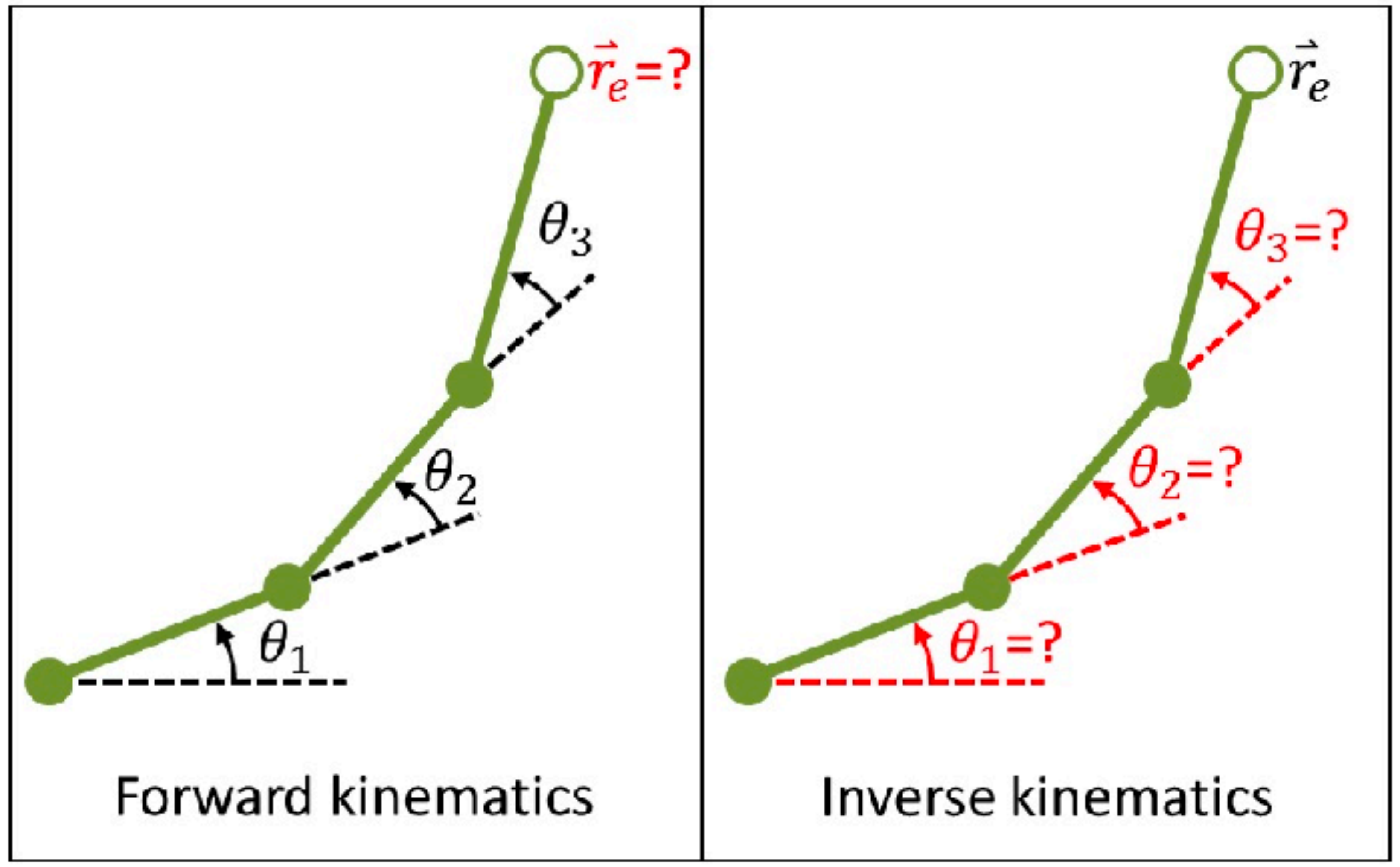
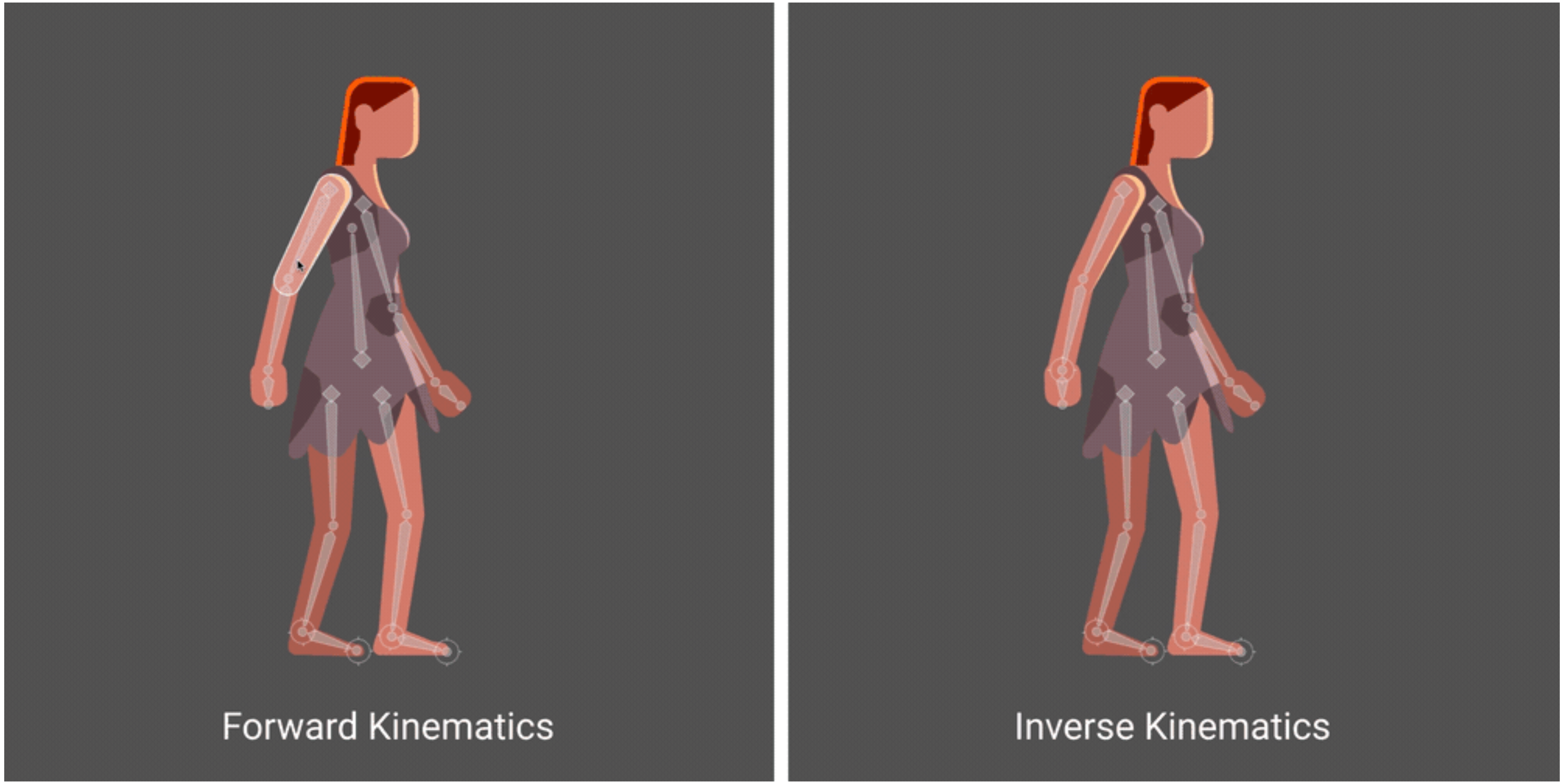


Image credits: Matlab Simulink, Najam Syed.

Today's class – Inverse of the problem

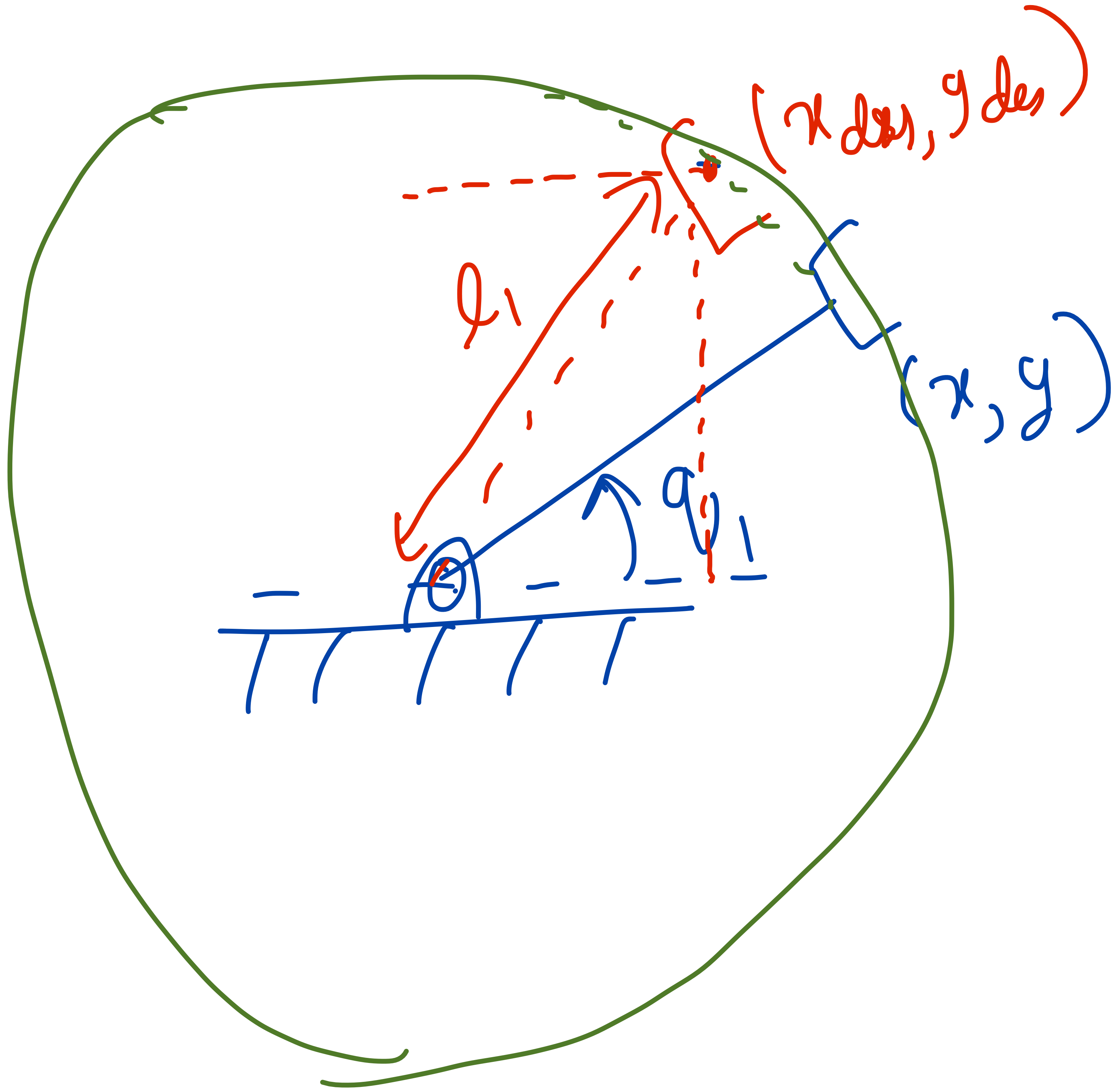


https://help.rive.app/editor/constraints/ik-constraint

Challenges with IK

- While FK is easy to compute, IK is not.
- There may be several solutions for IK.
- There may be no solution for IK.
- If there is a solution, it may require expensive computations to find.

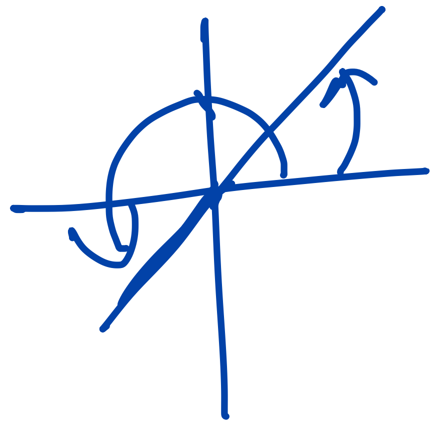
Solution 1: Analytic Methods



Single joint robot

(x_{des}, y_{des})

what is the best q_1 value



$$l_1 \cos q_1 = x_{des} \Rightarrow \cos q_1 = \frac{x_{des}}{l_1}$$

$$l_1 \sin q_1 = y_{des} \Rightarrow \sin q_1 = \frac{y_{des}}{l_1}$$

$-$ $+$

$-$ $+$

(i) $\sqrt{x_{des}^2 + y_{des}^2} = l_1$

(ii) $q_1 = \operatorname{atan2}\left(\frac{x_{des}}{l_1}, \frac{y_{des}}{l_1}\right)$

$$q_1 = \tan^{-1}\left(\frac{x_{des}}{l_1}, \frac{y_{des}}{l_1}\right)$$

$$q_1 = \operatorname{atan2}\left(\frac{x_{des}}{l_1}, \frac{y_{des}}{l_1}\right)$$

Two linked planar robot

$(a+b)^2 = a^2 + b^2 + 2ab$

$\cos q_1 \to C_1$

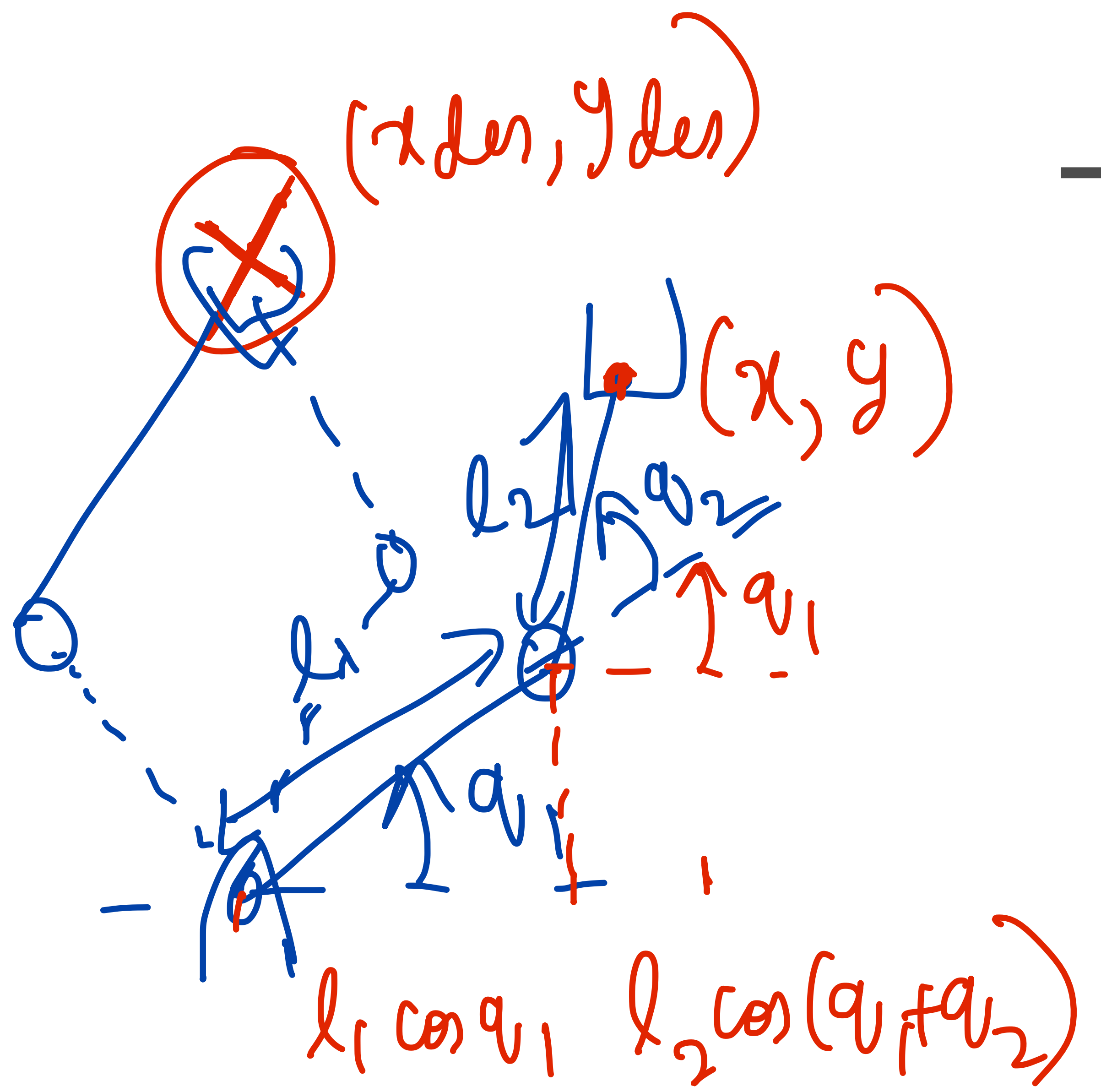
$\sin q_1 \to S_1$

$\cos(q_1+q_2) \to C_{12}$

$\sin(q_1+q_2) \to S_{12}$

$\cos(A+B) = \cos A \cos B - \sin A \sin B$

$\sin(A+B) = \sin A \cos B + \sin B \cos A$



$$x_{des} = l_1 \cos q_1 + l_2 \cos(q_1+q_2) \quad (1)$$

$$y_{des} = l_1 \sin q_1 + l_2 \sin(q_1+q_2) \quad (2)$$

$(1)^2 + (2)^2$

$$x_{des}^2 + y_{des}^2 = l_1^2 \underbrace{(S_1^2 + C_1^2)}_{=1} + l_2^2 \underbrace{(C_{12}^2 + S_{12}^2)}_{=1} + 2 l_1 l_2 C_1 C_{12} + 2 l_1 l_2 S_1 S_{12}$$

$$= l_1^2 + l_2^2 + 2 l_1 l_2 (C_1 C_{12} + S_1 S_{12})$$

$(\cos q_1 \cos(q_1+q_2) + \sin q_1 \sin(q_1+q_2))$

2 solutions for q_2

$$x_{des}^2 + y_{des}^2 = l_1^2 + l_2^2 + 2 l_1 l_2 C_2$$

$$\Rightarrow \cos q_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

$$\sin q_2 = \pm\sqrt{1 - \cos^2 q_2}$$

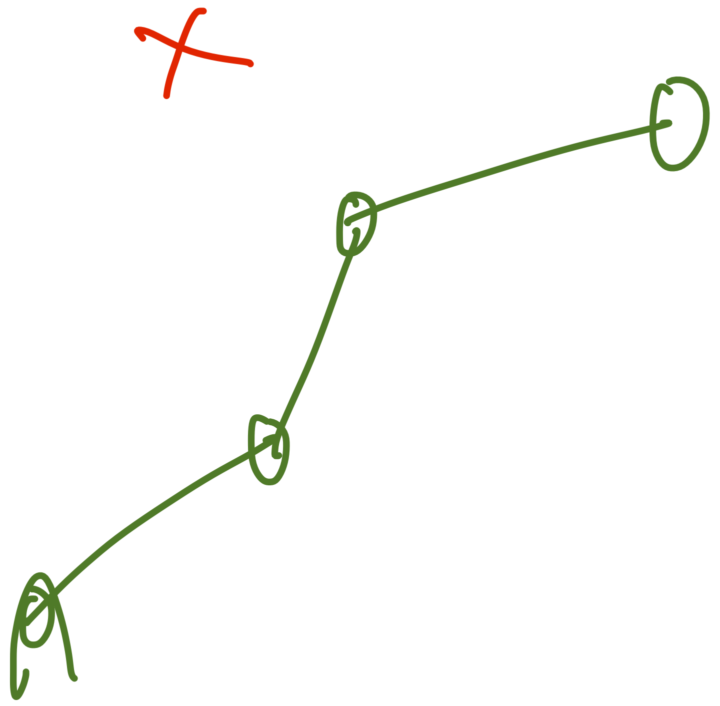
$\cos q_2$

$q_2 = \operatorname{atan2}(\cos q_2, \sin q_2)$

Two linked planar robot

$\cos(A+B) = \cos A \cos B - \sin A \sin B$

$\sin(A+B) = \sin A \cos B + \sin B \cos A$



q_2 $\quad q_1$?

$$x_{des} = l_1 c_1 + l_2 c_{12}$$

$$y_{des} = l_1 s_1 + l_2 s_{12}$$

$$x_{des} = l_1 c_1 + l_2 c_1 c_2 - l_2 s_1 s_2$$

$$x_{des} = (l_1 + l_2 c_2) c_1 - l_2 s_2 s_1 \quad \text{--- } (3)$$

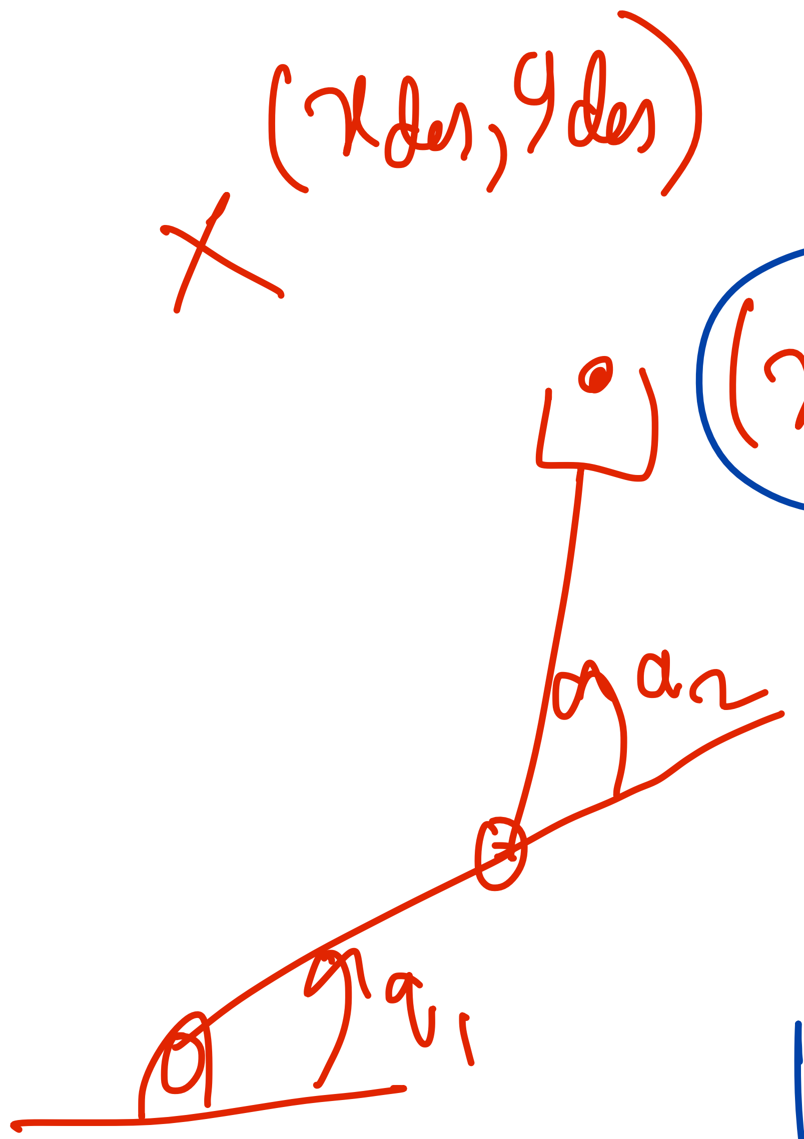
$$y_{des} = l_1 s_1 + l_2 s_1 c_2 + l_2 s_2 c_1$$

$$y_{des} = (l_1 + l_2 c_2) s_1 + l_2 s_2 c_1 \quad \text{--- } (4)$$

$$A \begin{bmatrix} l_1 + l_2 c_2 & -l_2 s_2 \\ l_2 s_2 & l_1 + l_2 c_2 \end{bmatrix} \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = \begin{bmatrix} x_{des} \\ y_{des} \end{bmatrix} \rightarrow \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = A^{-1} \begin{bmatrix} x_{des} \\ y_{des} \end{bmatrix}$$

$$q_1 = \text{atan2}(c_1, s_1)$$

Connection to Optimization



(x, y)

Objective

$$\min_{q_1, q_2} \left\| \underline{FK(q_1, q_2)} - \begin{bmatrix} x_{des} \\ y_{des} \end{bmatrix} \right\|^2$$

$$\min_{\vec{q}} \left\| \underline{F.K.(\vec{q})} - \vec{e} \right\|^2 \longrightarrow \vec{q}_{des} \rightarrow \text{I.K solution}$$

Solution 2a: Black box optimization

1. Python — https://github.com/facebookresearch/nevergrad
2. Matlab — https://www.mathworks.com/help/optim/ug/fmincon.html

Solution 2b: Numerical Methods (Gradient Descent)

Gradient Descent for IK

Vanilla Gradient Descent

How do I do this practically?

- Step 1: Figure out the Forward Kinematics

How do I do this practically?

- Step 1: Figure out the Forward Kinematics
- Step 2: Differentiate it!
    - Option 1: Do it by hand.
    - Option 2: Do it numerically.

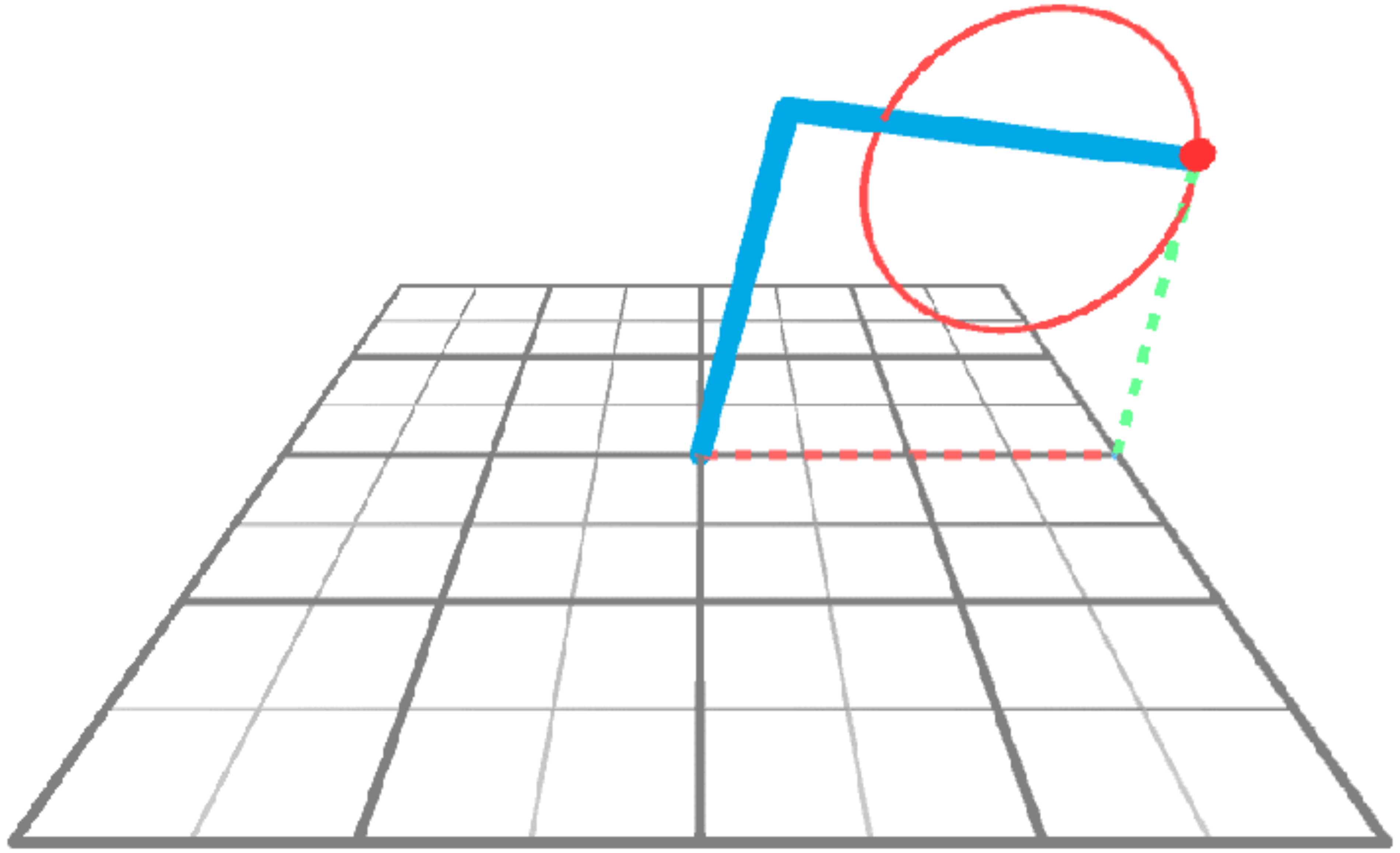
How do I do this practically?

- Step 1: Figure out the Forward Kinematics
- Step 2: Differentiate it!
  - Option 1: Do it by hand.
  - Option 2: Do it numerically.
- Step 3: Choose learning rate.

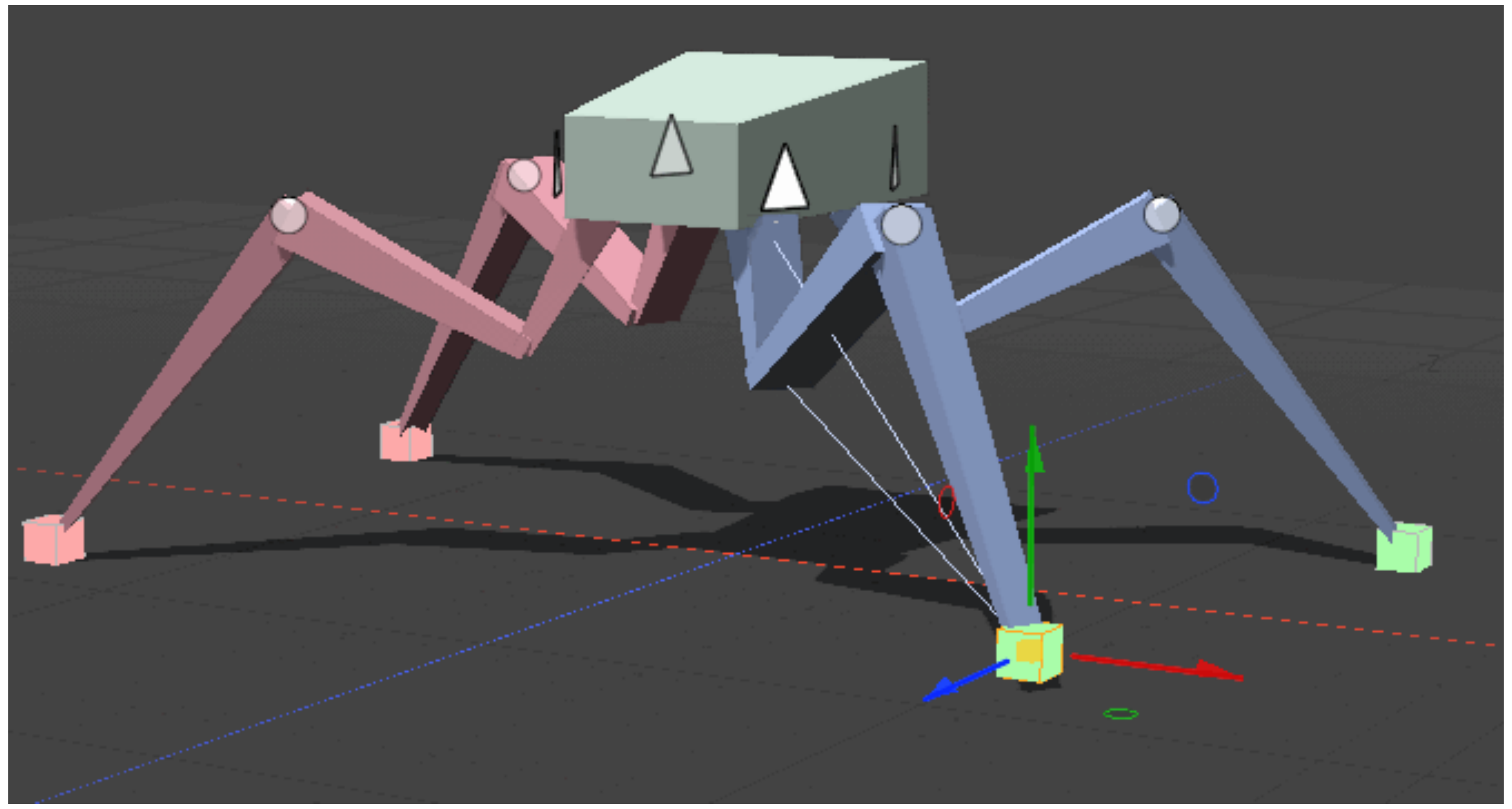
How do I do this practically?

- Step 1: Figure out the Forward Kinematics
- Step 2: Differentiate it!
    - Option 1: Do it by hand.
    - Option 2: Do it numerically.
- Step 3: Choose learning rate.
- Step 4: Run optimizer.

Examples

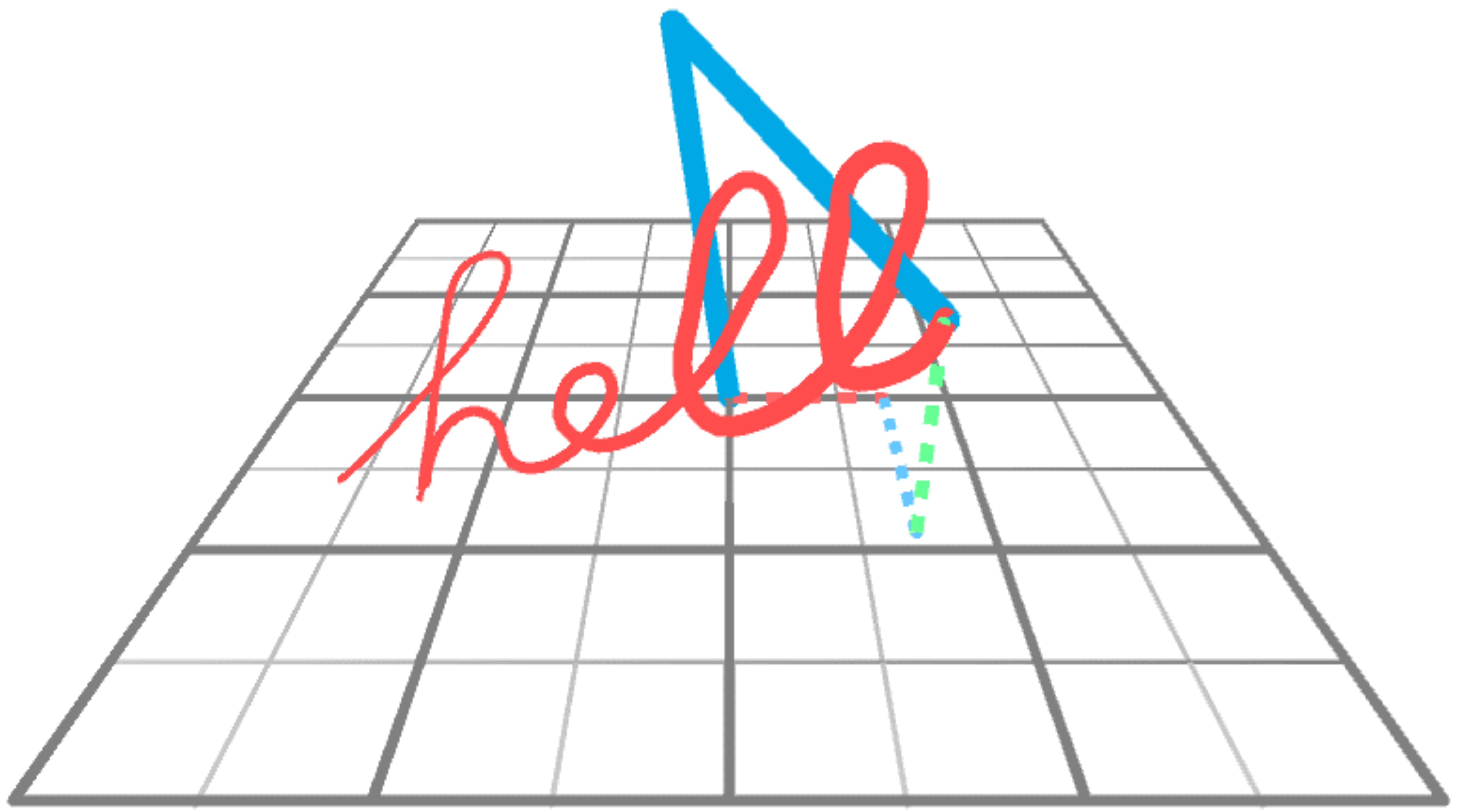


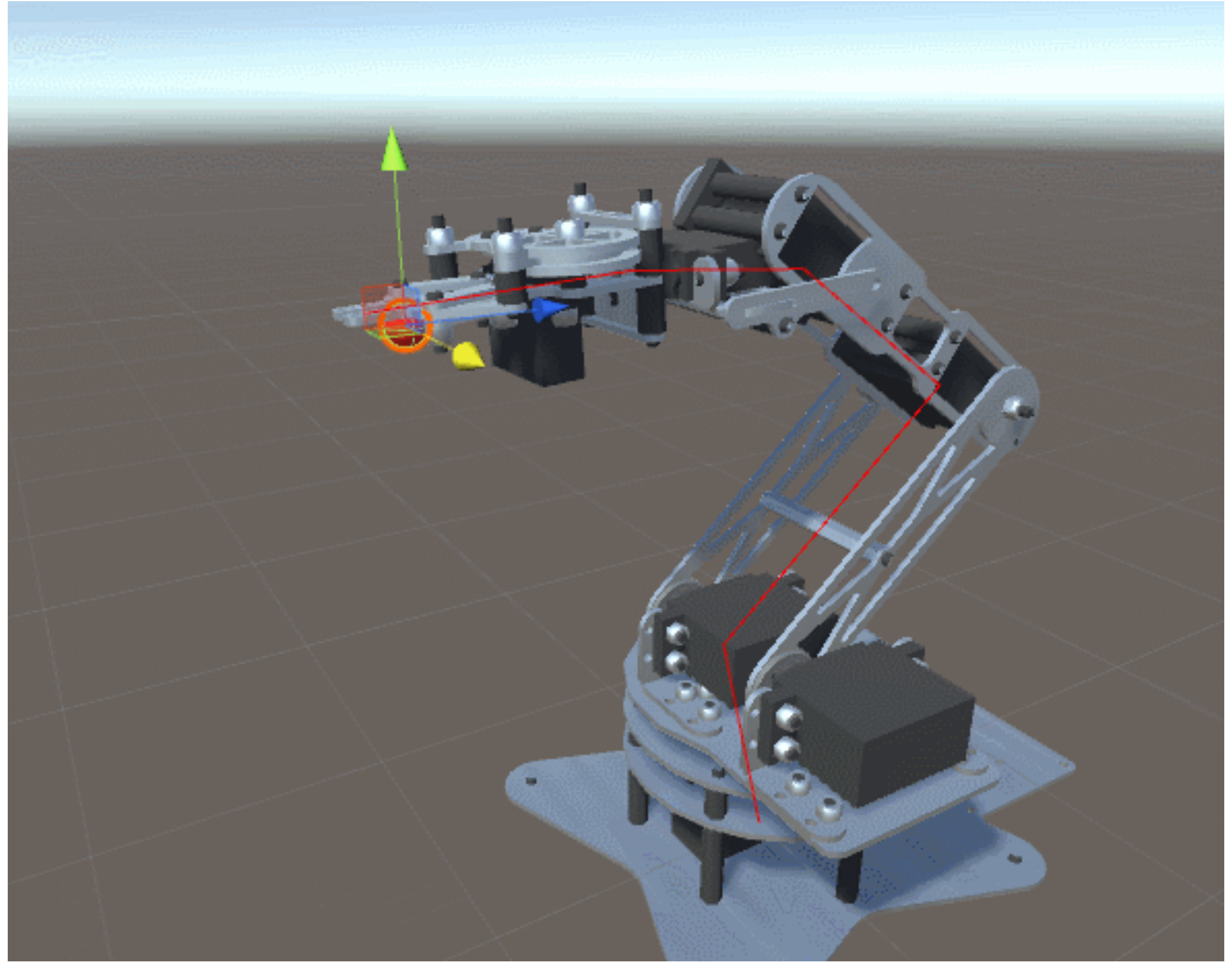
https://www.alanzucconi.com/



https://learn.foundry.com/

Examples





https://www.alanzucconi.com/

Additional Reading

- http://motion.cs.illinois.edu/RoboticSystems/Kinematics.html
- Textbook: http://hades.mech.northwestern.edu/images/2/2a/Park-lynch.pdf