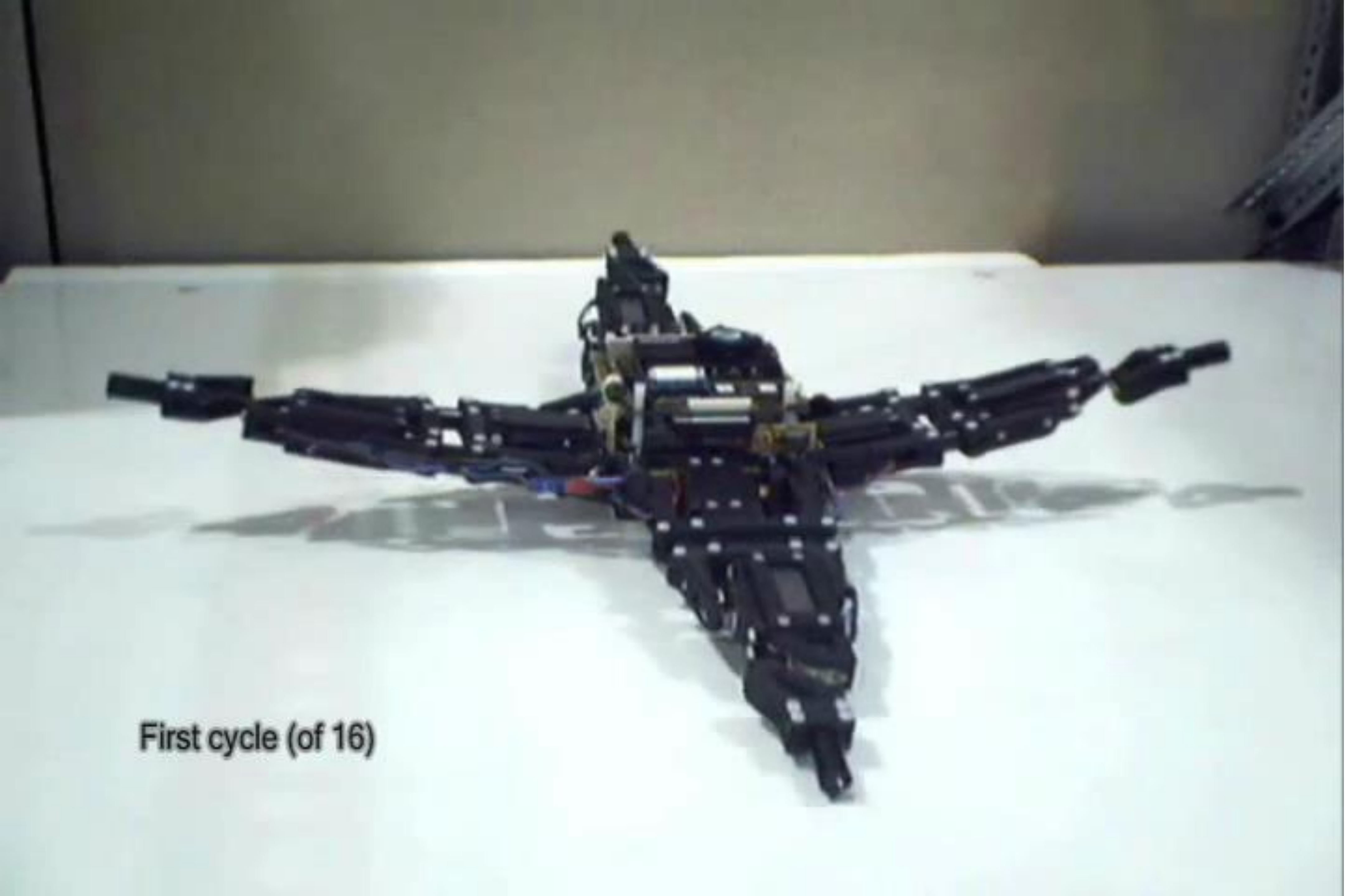
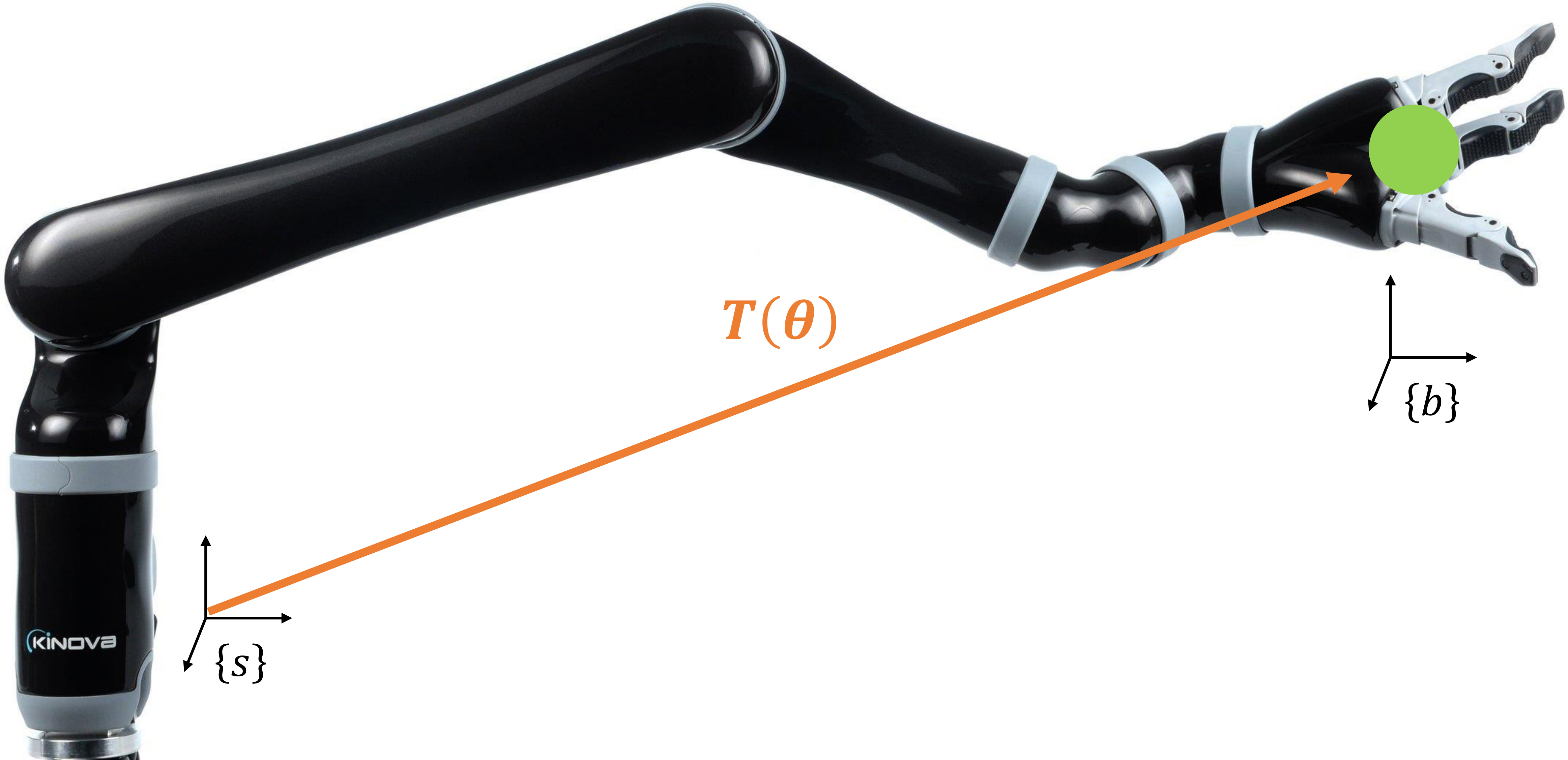
Numerical Inverse Kinematics

Reading: Modern Robotics 6.2

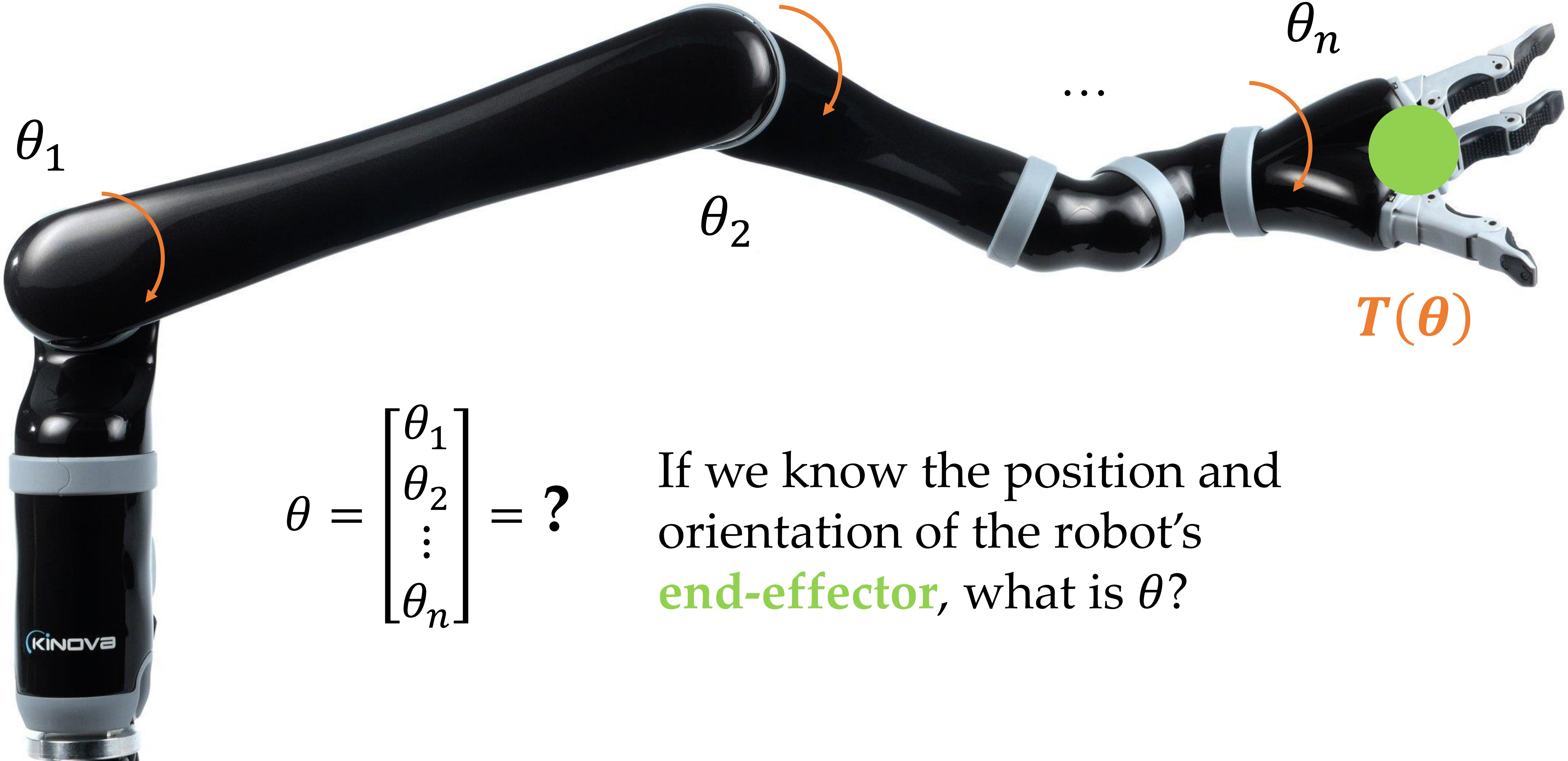

First cycle (of 16)

This Lecture

- How can we use the Jacobian pseudoinverse to get inverse kinematics?
- Can we develop a numerical algorithm for inverse kinematics?
- When does our algorithm fail?

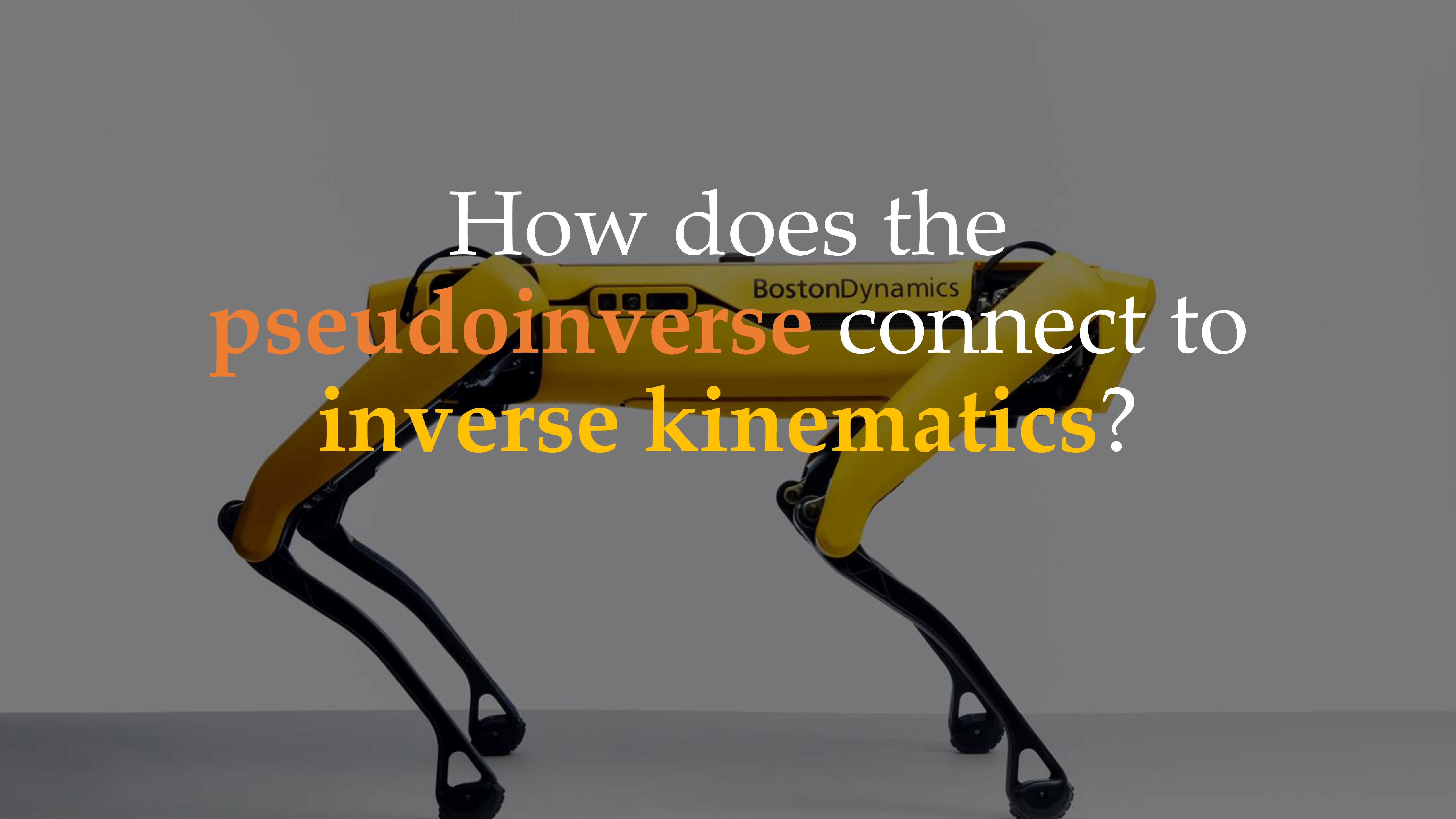


$T(\theta)$
$\{s\}$
$\{b\}$
KINOVA

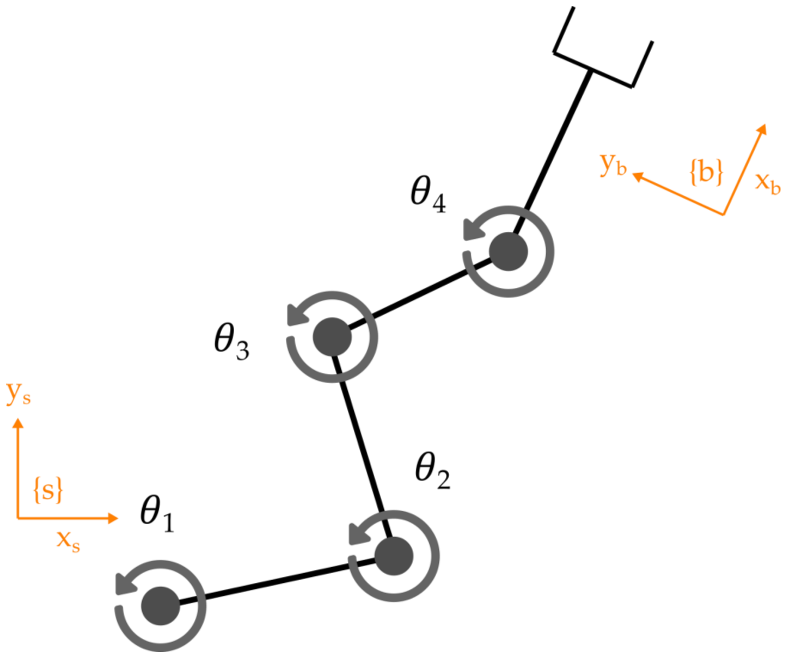


$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} = \textbf{?}$$

If we know the position and orientation of the robot's **end-effector**, what is θ?


How does the pseudoinverse connect to inverse kinematics?
BostonDynamics

Using the Pseudoinverse



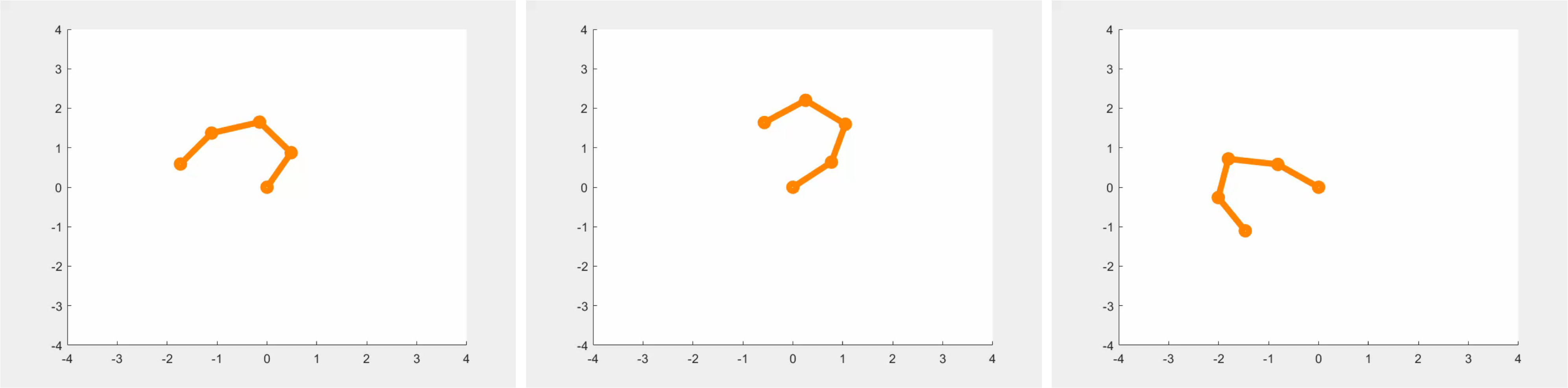
$$\dot{\theta} = \underline{J^+V} + \underline{(I - J^+J)b}$$

With this pseudoinverse, we can convert a twist V into a joint velocity $\dot{\theta}$

Using the Pseudoinverse

$$\dot{\theta} = J^{+}V$$

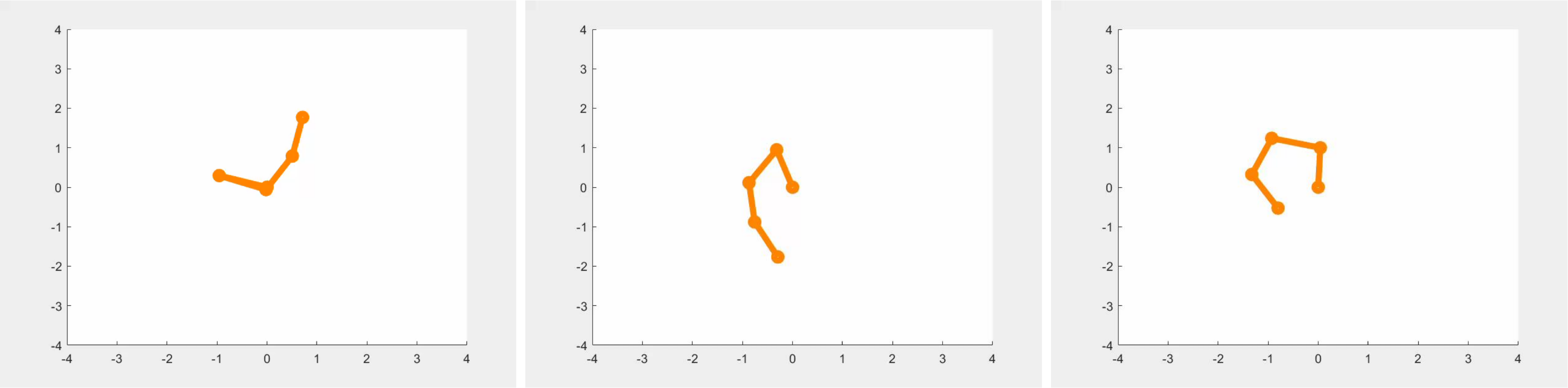
$$V = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$



Using the Pseudoinverse

$$\dot{\theta} = J^{+}V$$

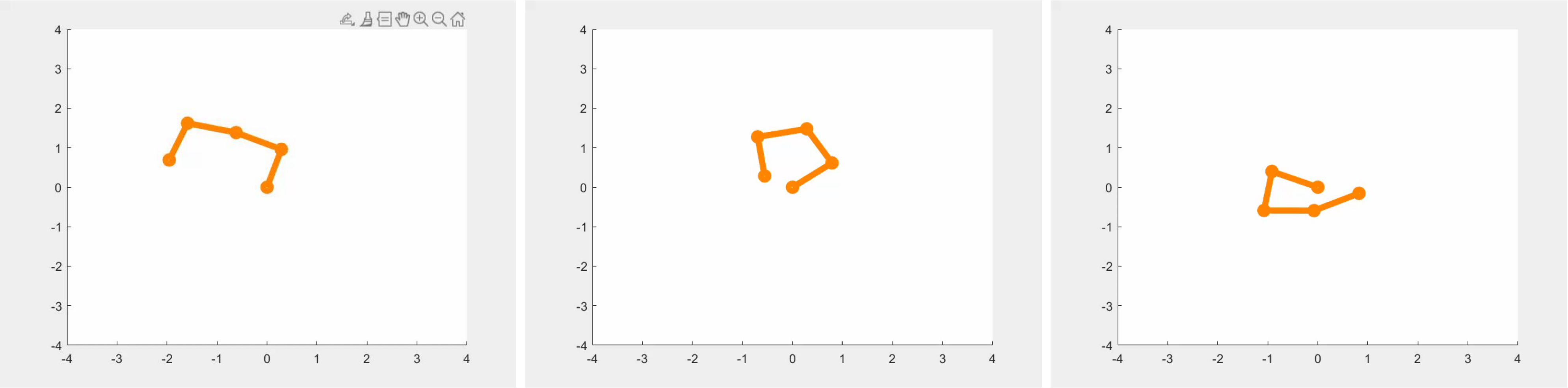
$$V = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$



Using the Pseudoinverse

$$\dot{\theta} = J^{+}V$$

$$V = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Using the Pseudoinverse

$$\dot{\theta} = J^{+}V$$

Applying to inverse kinematics:

Given a desired T_d and our current T,
we first need a twist V that moves from T towards T_d

1 Convert T to X

2 Convert T_d to X_d

3 Subtract to get V

For any transformation matrix T, there exists a screw axis S and scalar θ so that:

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = e^{[S]\theta}$$

1 Convert T to X

2 Convert T_d to X_d

3 Subtract to get V

For any transformation matrix T, there exists a screw axis S and scalar θ so that:

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = e^{[S]\theta}$$

In other words, there is a **matrix logarithm** that takes us from T to $S\theta$:

$$\log T \rightarrow S\theta$$

1 Convert T to X

2 Convert T_d to X_d

3 Subtract to get V

For any transformation matrix T, there exists a screw axis S and scalar θ so that:

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = e^{[S]\theta}$$

We are going to cover a version of this that is compatible with geometric twists, i.e., velocity of $\{b\}$ *expressed in* $\{s\}$*. This is so we get a* V *that works with our* ***geometric Jacobian***.

1 Convert T to X

2 Convert T_d to X_d

3 Subtract to get V

For any transformation matrix T, there exists a screw axis S and scalar θ so that:

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = e^{[S]\theta}$$

In the special case where $R = I$:

$$\log T = \begin{bmatrix} 0 \\ p \end{bmatrix}$$

1 Convert T to X

2 Convert T_d to X_d

3 Subtract to get V

For any transformation matrix T, there exists a screw axis S and scalar θ so that:

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = e^{[S]\theta}$$

Otherwise find the axis-angle for R:

$$\log T = \begin{bmatrix} \hat{\omega}\theta \\ p \end{bmatrix}, \qquad R = e^{[\hat{\omega}]\theta}$$

1. Convert T to X

$$\log T = X = \begin{bmatrix} \omega_{x1} \\ \omega_{y1} \\ \omega_{z1} \\ p_{x1} \\ p_{y1} \\ p_{z1} \end{bmatrix}$$

2. Convert T_d to X_d

$$\log T_d = X_d = \begin{bmatrix} \omega_{x2} \\ \omega_{y2} \\ \omega_{z2} \\ p_{x2} \\ p_{y2} \\ p_{z2} \end{bmatrix}$$

3. Subtract to get V

$$V = X_d - X$$

Now that we know the **direction we want to go** and the **pseudoinverse**, we can find **numerical inverse kinematics**

Numerical Solution

$X_d \leftarrow \log T_d$

Map the desired transformation to 6×1 *vector (previous slides)*

Numerical Solution

$X_d \leftarrow \log T_d$

$\theta \leftarrow \theta_0$

$X \leftarrow \log T(\theta)$

Choose a random initial value for θ. This is your first guess at the inverse kinematics solution. Then use forward kinematics to get $T(\theta)$, and map to X

Numerical Solution

$X_d \leftarrow \log T_d$

$\theta \leftarrow \theta_0$

$X \leftarrow \log T(\theta)$

while $\|X_d - X\| > \varepsilon$

While the 6×1 *representation of* $T(\theta)$ *does not match the* 6×1 *representation of* T_d

Numerical Solution

$X_d \leftarrow \log T_d$

$\theta \leftarrow \theta_0$

$X \leftarrow \log T(\theta)$

while $\|X_d - X\| > \varepsilon$

$V \leftarrow X_d - X$

Get a twist moving from T towards T_d

Numerical Solution

$X_d \leftarrow \log T_d$

$\theta \leftarrow \theta_0$

$X \leftarrow \log T(\theta)$

while $\|X_d - X\| > \varepsilon$

$V \leftarrow X_d - X$

$\Delta\theta = J^+V + (I - J^+J)b$

Use pseudoinverse to map twist V to a change in θ.
Should be the geometric Jacobian because of how we defined V.
If you have a redundant robot, you can use the null space as well.

Numerical Solution

$X_d \leftarrow \log T_d$

$\theta \leftarrow \theta_0$

$X \leftarrow \log T(\theta)$

while $\|X_d - X\| > \varepsilon$

$V \leftarrow X_d - X$

$\Delta\theta = J^{+}V + (I - J^{+}J)b$

$\theta \leftarrow \theta + \alpha\Delta\theta$

Increment your guess of the inverse kinematics solution.
Here α is a small positive number of your choice.

Numerical Solution

$X_d \leftarrow \log T_d$

$\theta \leftarrow \theta_0$

$X \leftarrow \log T(\theta)$

while $\|X_d - X\| > \varepsilon$

$V \leftarrow X_d - X$

$\Delta\theta = J^+V + (I - J^+J)b$

$\theta \leftarrow \theta + \alpha\Delta\theta$

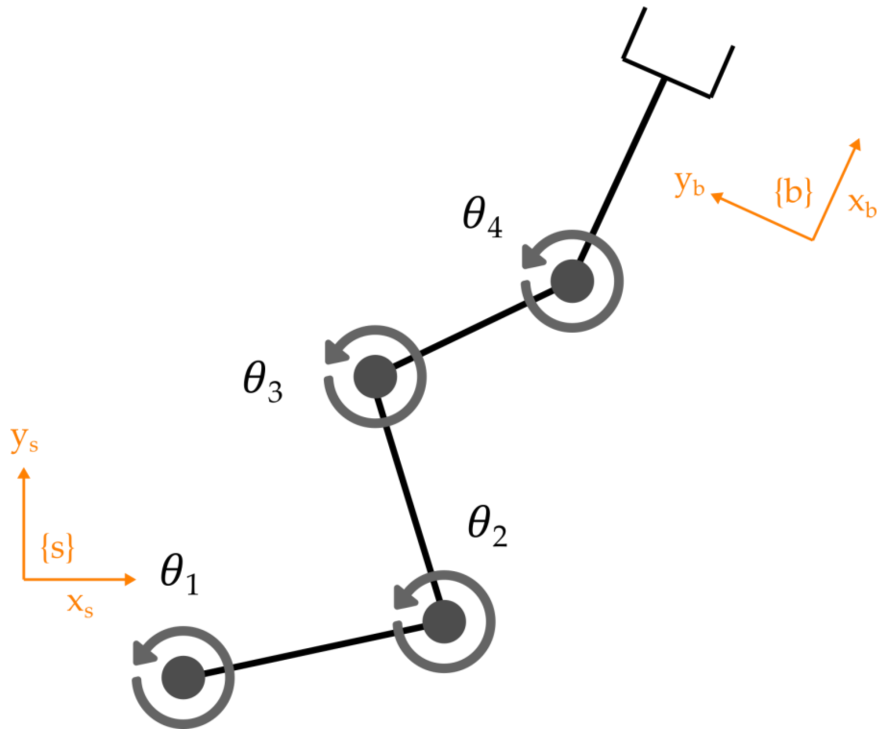
$X \leftarrow T(\theta)$

Use forward kinematics to find the new end-effector position.

Example

4-DoF **redundant** robot moving in a plane.

Given a desired end-effector pose, can we get joint positions?

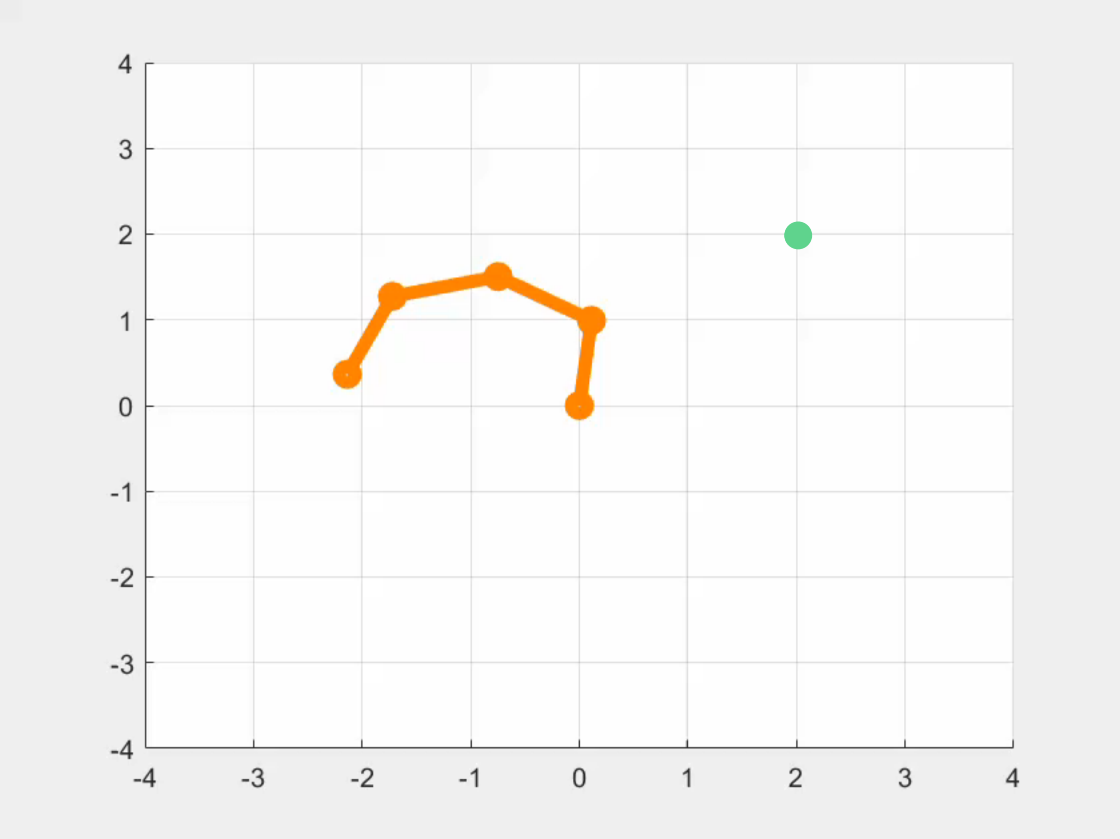


Example

What joint position gives us:

$$T_{sb} = \begin{bmatrix} \text{rotz}\left(\frac{pi}{4}\right) & \begin{matrix} 2 \\ 2 \\ 0 \end{matrix} \\ 0 & 1 \end{bmatrix}$$

Initial: $\theta_0 = \begin{bmatrix} 1.4590 \\ 1.1472 \\ 0.7675 \\ 0.9087 \end{bmatrix}$ Final: $\theta = \begin{bmatrix} -0.480 \\ 1.310 \\ 0.998 \\ -0.985 \end{bmatrix}$

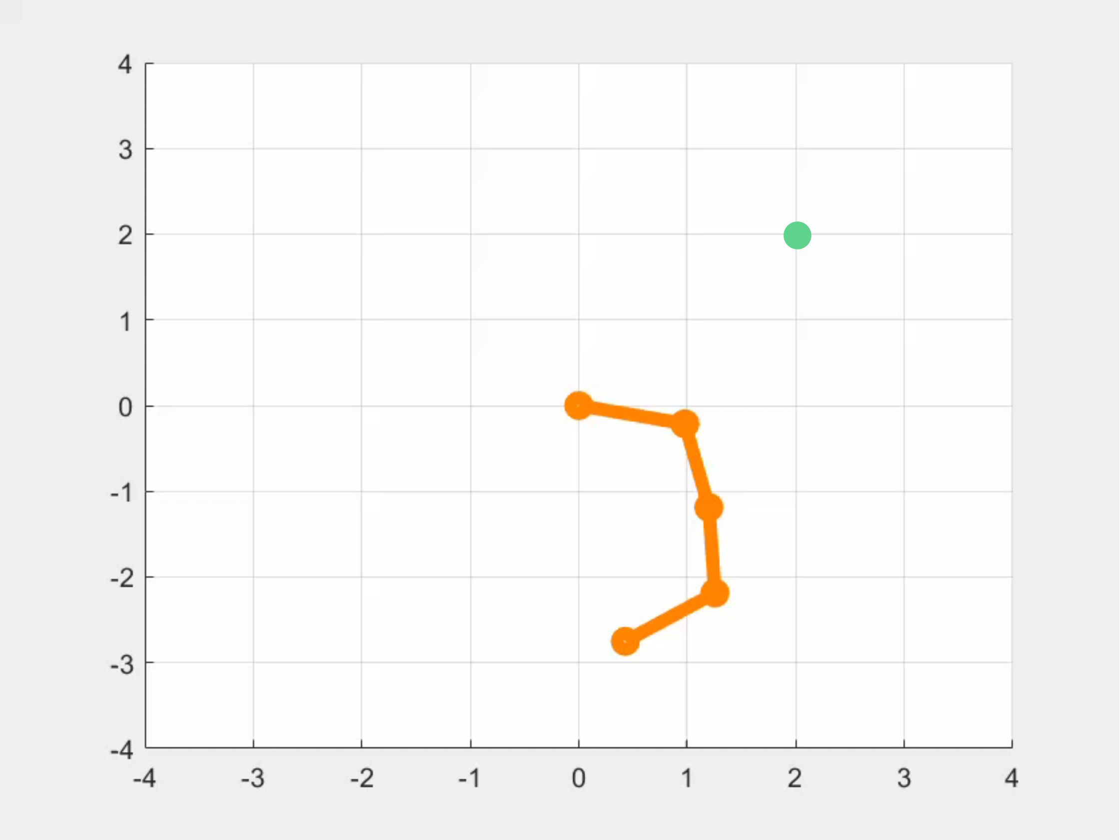


Example

What joint position gives us:

$$T_{sb} = \begin{bmatrix} \text{rotz}\left(\frac{pi}{4}\right) & \begin{matrix} 2 \\ 2 \\ 0 \end{matrix} \\ 0 & 1 \end{bmatrix}$$

Initial: $\theta_0 = \begin{bmatrix} -0.214 \\ -1.133 \\ -0.167 \\ -1.027 \end{bmatrix}$ Final: $\theta = \begin{bmatrix} -1.460 \\ -0.357 \\ -1.819 \\ -4.746 \end{bmatrix}$

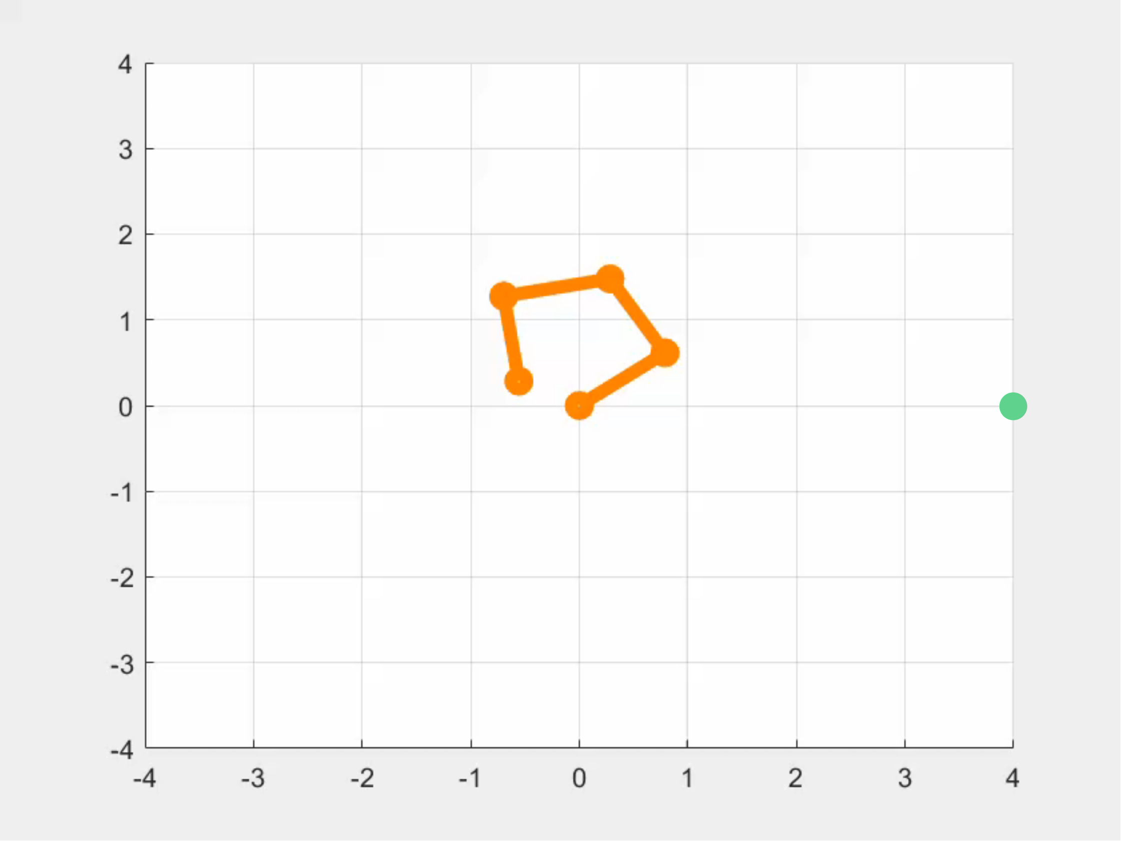


Example

What joint position gives us:

$$T_{sb} = \begin{bmatrix} & & 4 \\ & I & 0 \\ & & 0 \\ 0 & & 1 \end{bmatrix}$$

We know the answer is $\theta = 0$. But why is our algorithm jumping around?



Using the Pseudoinverse

$$\dot{\theta} = J^{+}V + (I - J^{+}J)b$$

Remember that $J^{+} = J^{T}(JJ^{T})^{-1}$
At **singularities** the rank of J decreases, so the rank of JJ^{T} also decreases
Hence, $(JJ^{T})^{-1} \to \infty$ as the robot approaches singularities

This Lecture

- How can we use the Jacobian pseudoinverse to get inverse kinematics?
- Can we develop a numerical algorithm for inverse kinematics?
- When does our algorithm fail?

Next Lecture

- So far we've talked about position and velocity…

  …what about forces and torques?