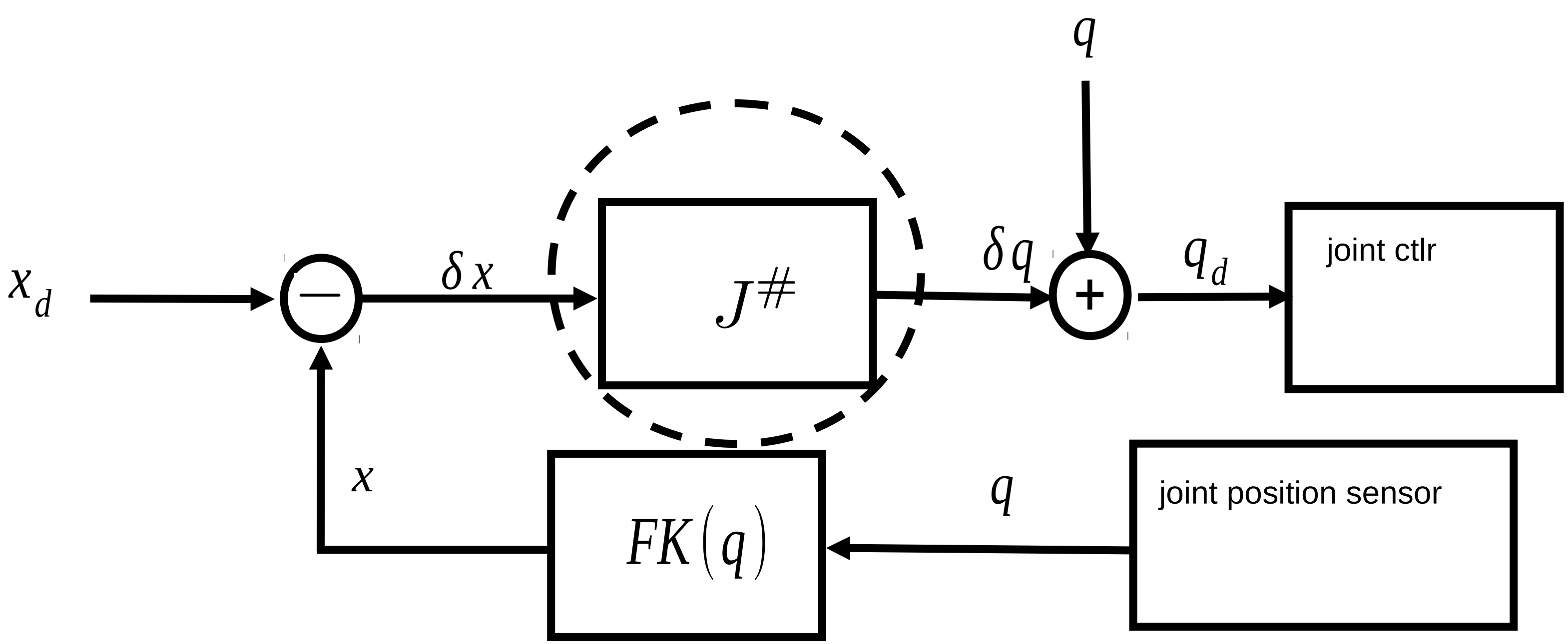
# Cartesian Control (Orientation)

Robert Platt
Northeastern University

# Controlling Cartesian Position



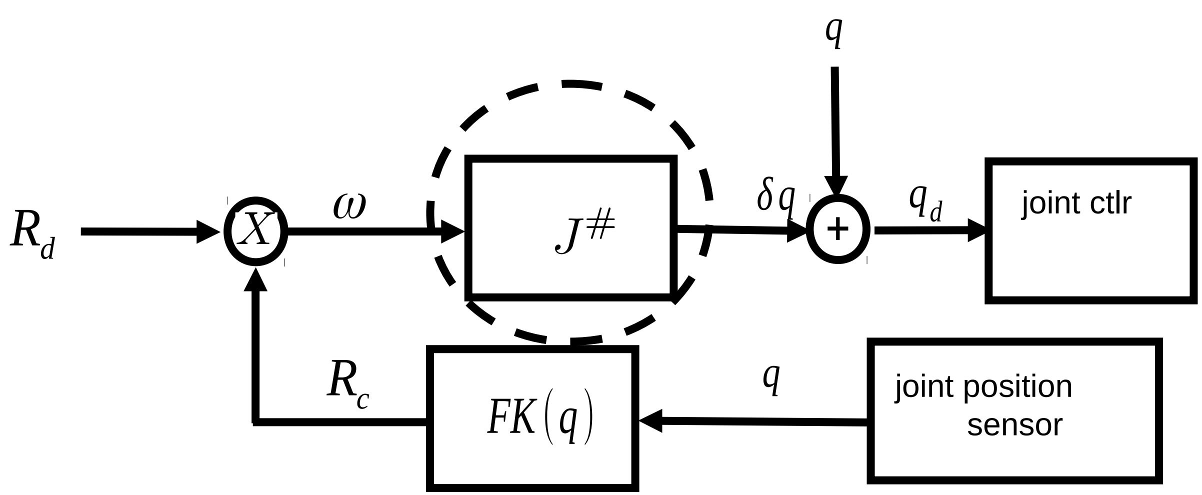

Procedure for controlling position:

1. Calculate position error: $x_{err}$
2. Multiply by a scaling factor: $\delta x_{err} = \alpha x_{err}$
3. Multiply by the velocity Jacobian pseudoinverse: $\dot{q} = J_v{}^{\#} \alpha x_{err}$

# Controlling Cartesian Orientation

How does this strategy work for orientation control?

- Suppose you want to reach an orientation of $R_d$
- Your current orientation is $R_c$
- You've calculated a difference: $R_{cd} = R_c^T R_d$
- How do you turn this difference into a desired angular velocity to use in $\dot{q} = J^{\#}\omega$ ?

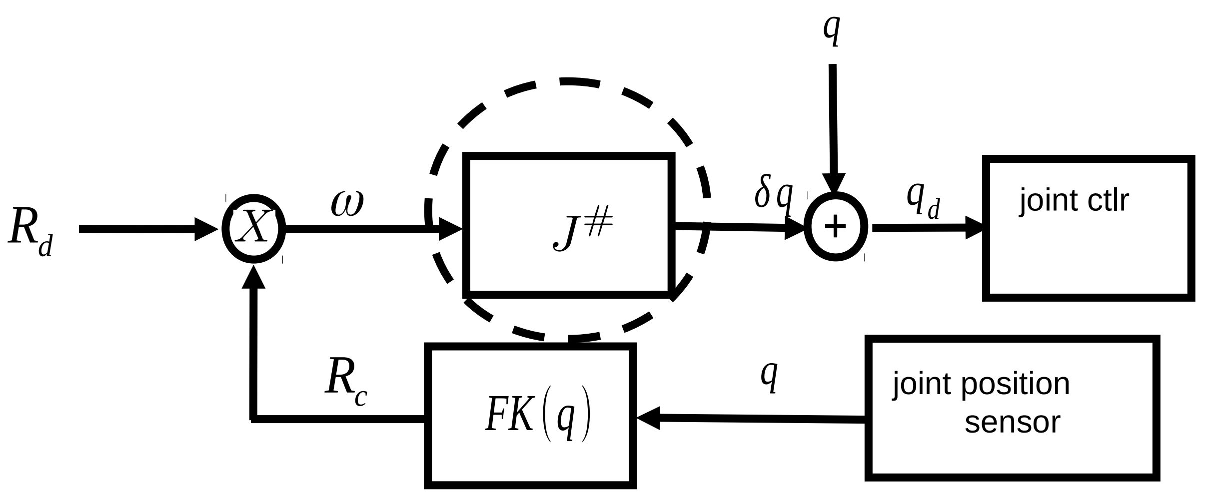


# Controlling Cartesian Orientation

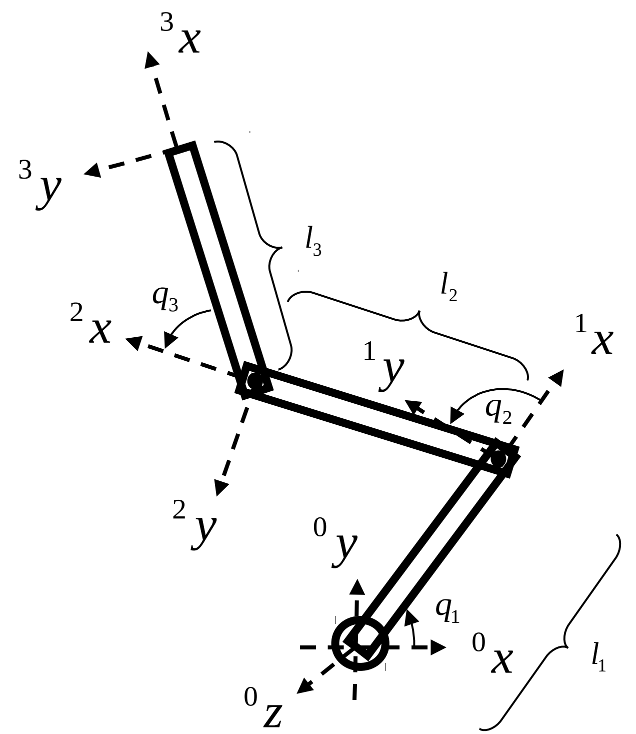
You **can't** do this:

- Convert the difference to ZYZ Euler angles: $r_{\phi\theta\psi}$
- Multiply the Euler angles by a scaling factor and pretend that they are an angular velocity: $\delta q = \alpha J^{\#} r_{\phi\theta\psi}$

Remember that in general: $J_{\omega} \neq \dfrac{\partial r_{\phi\theta\psi}}{\partial q}$



# Controlling Cartesian Orientation




The easiest way to handle this Cartesian orientation problem is to represent the error in axis-angle format

$$\delta r_k = J_\omega \dot{q}$$

Axis angle delta rotation

Procedure for controlling rotation:

1. Represent the rotation error in axis angle format: $r_{err}$
2. Multiply by a scaling factor: $\delta r_{err} = \alpha r_{err}$
3. Multiply by the angular velocity Jacobian pseudoinverse: $\dot{q} = J_\omega^{\#} \alpha r_{err}$

# Controlling Cartesian Orientation

Why does axis angle work?

- Remember Rodrigues' formula from before:

$$R_{k\theta} = e^{S(k)\theta} = I + S(k)\sin(\theta) + S(k)^2(1 - \cos(\theta))$$

axis angle

Compare this to the definition of angular velocity: $^b\dot{p} = S(^b\omega)^b p$

The solution to this FO diff eqn is: $^bR_{\omega t} = e^{S(^b\omega)t}$

Therefore, the angular velocity gets integrated into an axis angle representation