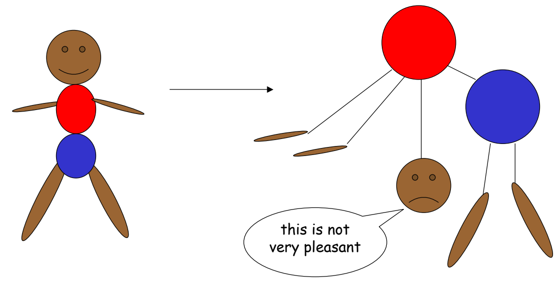
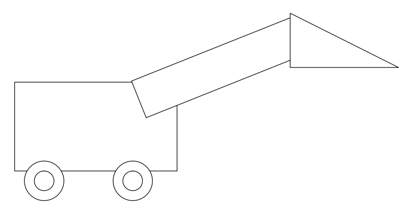
Articulated Figures & Kinematics

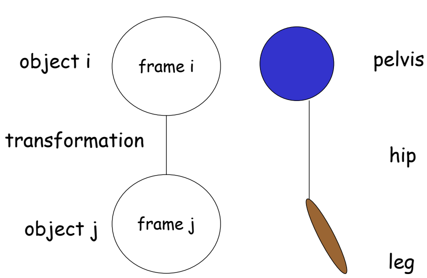
Articulated Figure

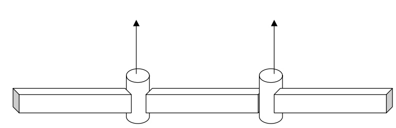


- nodes correspond to body parts
- links correspond to joints

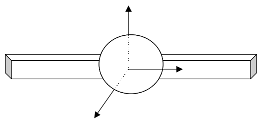
exercise: draw the hierarchical representation



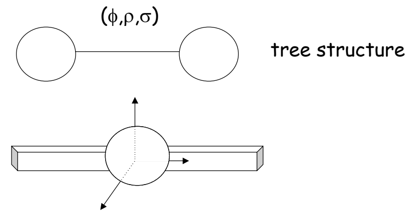




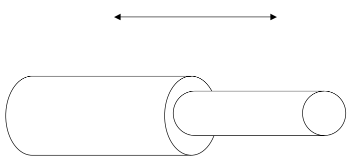
revolute joints – transformation is a rotation



revolute joints – transformation is a rotation with up to three degrees of freedom

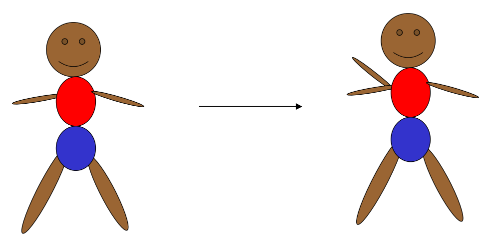


revolute joints – transformation is a rotation with up to three degrees of freedom



prismatic joints – not used (often) in animation

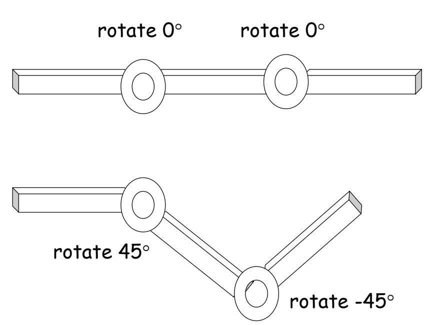
Animating an Articulated Figure



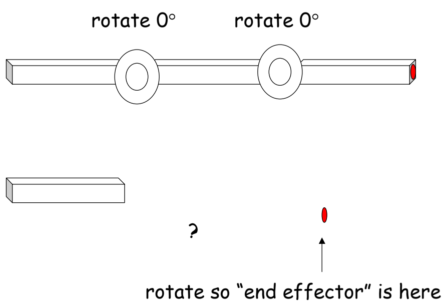
specifying change

- Forward kinematics: Specify node parameters as a function of time
  - Keyframe (motion capture or manually generated)
  - Physically-based simulation
- Inverse kinematics: Specify "end-effect" and let system compute remaining parameters subject to constraints

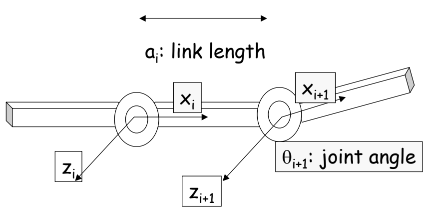
forward kinematics

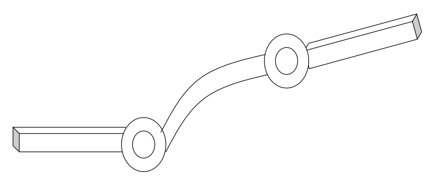


inverse kinematics



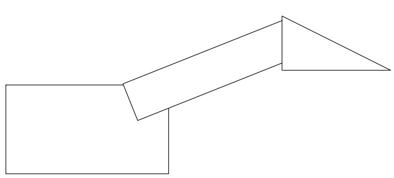
Denavit-Hartenberg (DH) notation



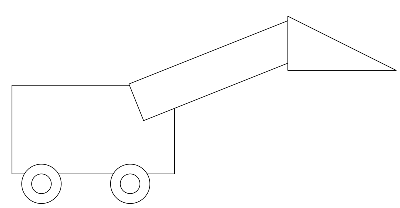


can be extended to non-planar body parts

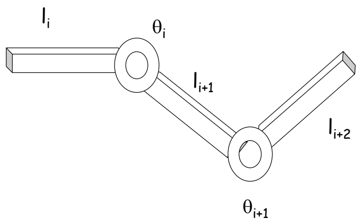
exercise: draw the hierarchical structure and augment with DH notation



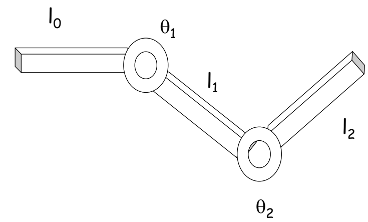
what is wrong with this picture?



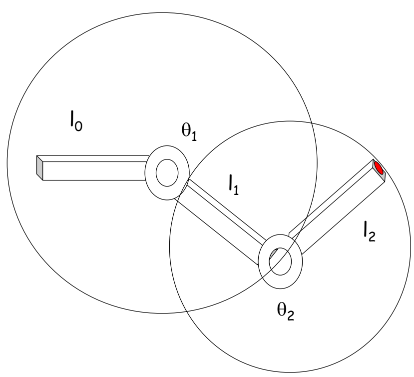
axis position (AP):
joint position, orientation, pointers to connected body parts

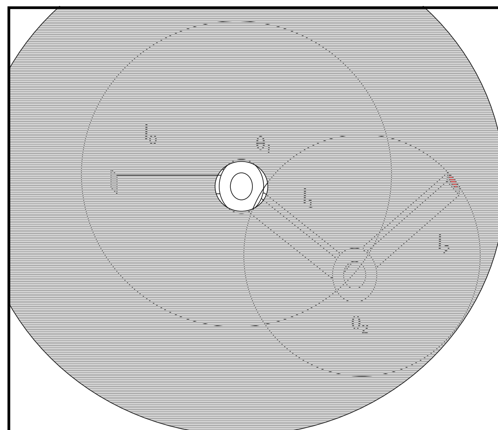


our simple model: single linkage, planar body parts, revolute joints with one degree of freedom

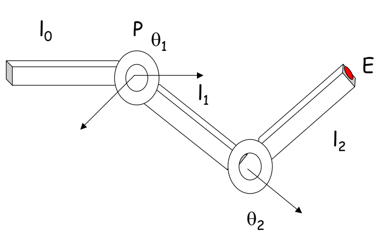


what are possible orientations assuming part 0 is fixed?





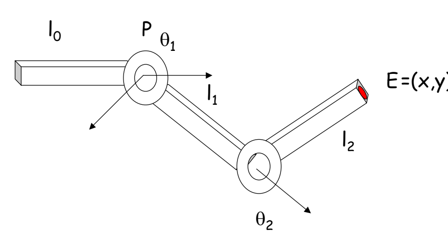
FORWARD KINEMATICS



write E in terms of P=(0,0), θ_1, θ_2, l_1, l_2

$$E_x = l_1\cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$
$$E_y = l_1\sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

INVERSE KINEMATICS



write θ_1, θ_2 in terms of l_1, l_2, x, y

$$\theta_1 = \cos^{-1}(x^2y^2+l_1^2-l_2^2)/(2l_1(x^2+y^2)^{1/2})$$
$$\theta_2=\cos^{-1} (l_1^2 + l_2^2 -(x^2+y^2)0/(2l_1l_2)$$

In general

change in angles → easy computation → change in joint positions/ orientations

$dX = J(\theta)d\theta$

↑ Jacobian

Problem: invert $J(\theta)$

$E_x = l_1\cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$
$E_y = l_1\sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$

write dE in terms of $d\theta$

$dE_x = -l_1\sin\theta_1\, d\theta_1 - l_2\sin(\theta_1 + \theta_2)(d\theta_1 + d\theta_2)$

$dE_y = l_1\cos\theta_1\, d\theta_1 + l_2\cos(\theta_1 + \theta_2)(d\theta_1 + d\theta_2)$

$$dE = \begin{bmatrix} -l_1\sin\theta_1 - l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix}$$

note this gives position but not orientation at E

In General ...

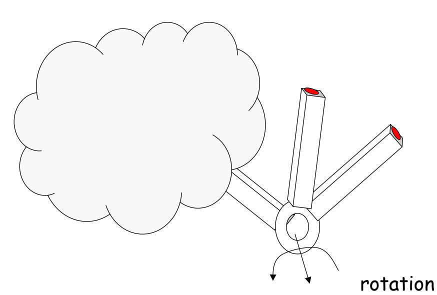
$$\begin{bmatrix} dE_x & dE_y & dE_z & dE_\phi & dE_\rho & dE_\sigma \end{bmatrix} = J(\theta) \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_m \end{bmatrix}$$

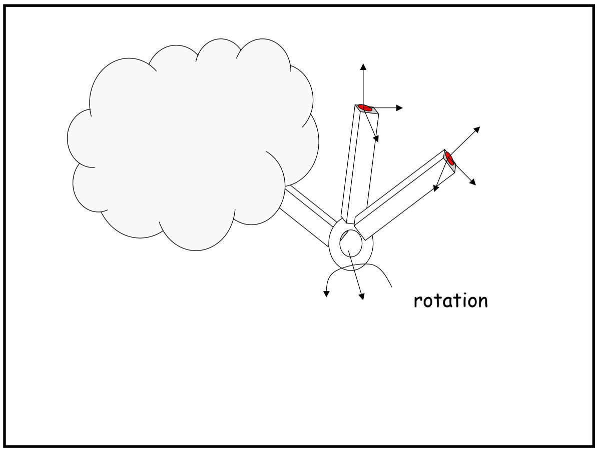
↑ 6 X m

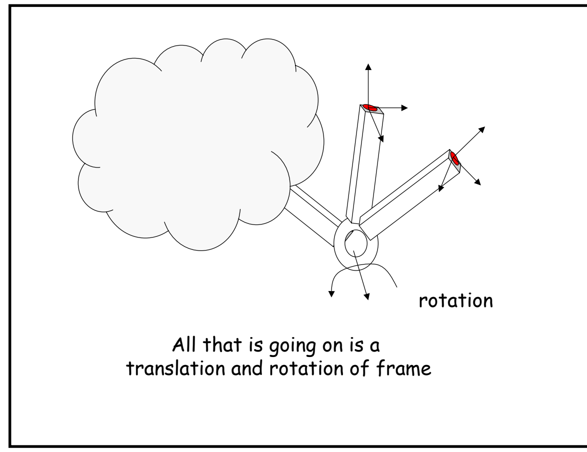
We need an algorithm for computing $J(\theta)$!

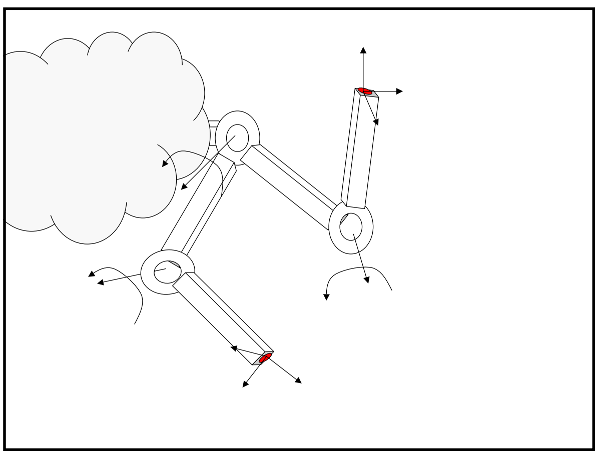
Intuition: The Jacobian represents velocity of end-effector in terms of state velocities

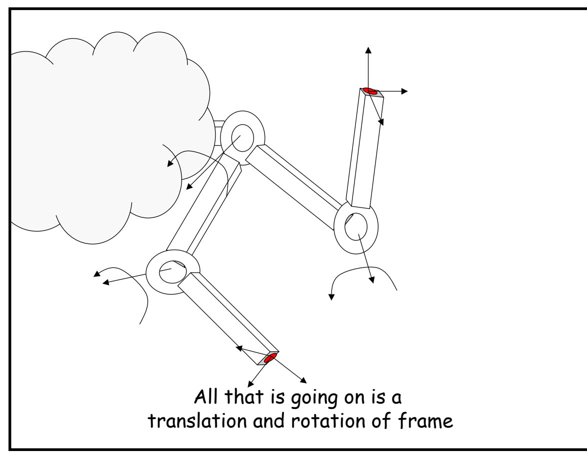
---

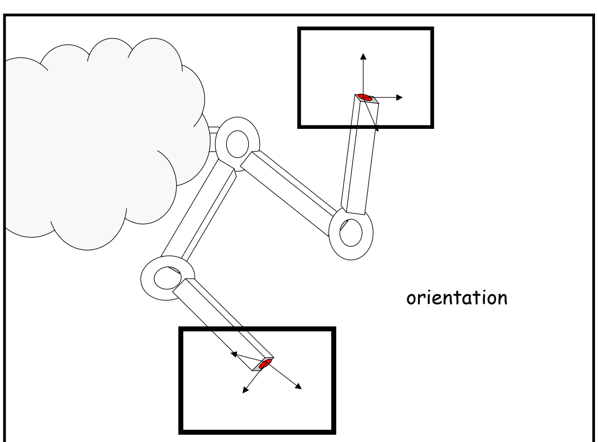


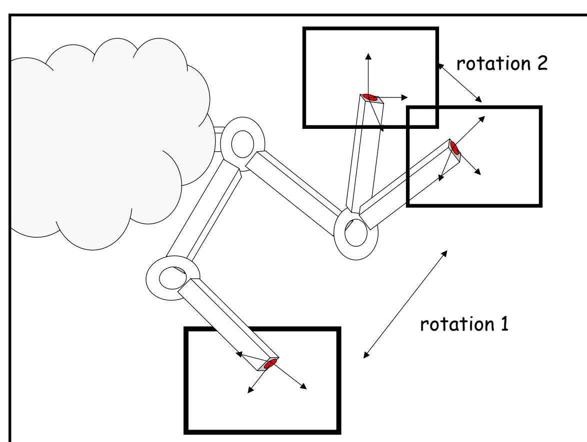

rotation

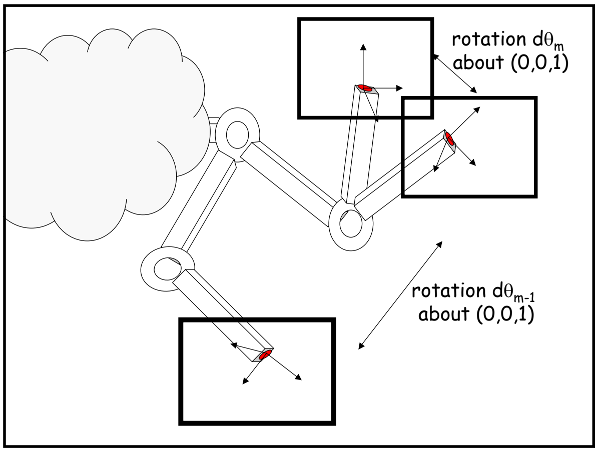

rotation
All that is going on is a
translation and rotation of frame



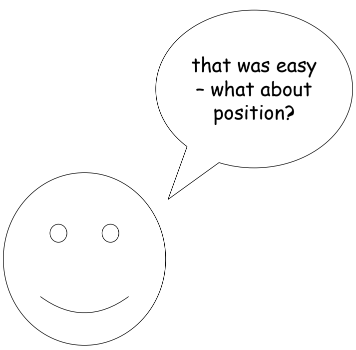

All that is going on is a
translation and rotation of frame

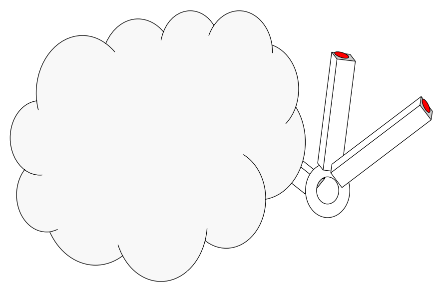

orientation

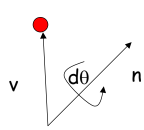

rotation 2
rotation 1



change in orientation of end-effector is simply the sum of the rotations at all joints!

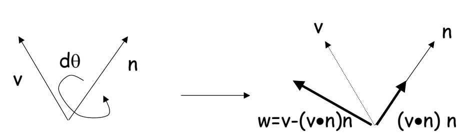




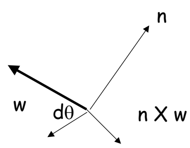


suppose v is rotated about n by $d\theta$ to yield u

write u in terms of v, n, and $d\theta$

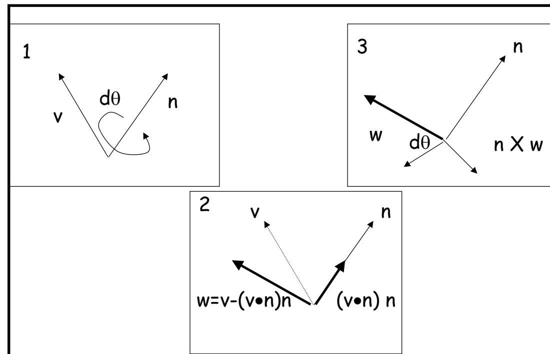


we only need to rotate $w=v-(v\bullet n)n$



let w'=w/||w|| then rotated w is

$w \cos d\theta + \sin d\theta\ (n \times w)$



$u = (v \bullet n)\, n + w \cos d\theta + \sin d\theta\ (n \times w) =$

$v \cos d\theta + (1-\cos d\theta)(v \bullet n)n + \sin d\theta\ (n \times v)$

suppose v is rotated about n by dθ to yield u

write u in terms of v, n, and θ

$u = v \cos d\theta + (1-\cos d\theta)(v \bullet n)n + \sin d\theta\ (n \times v)$

when dθ is small $\cos d\theta \approx 1$ and $\sin d\theta \approx d\theta$ so

$u \approx v + \underbrace{d\theta\ (n \times v)}_{\Delta v}$

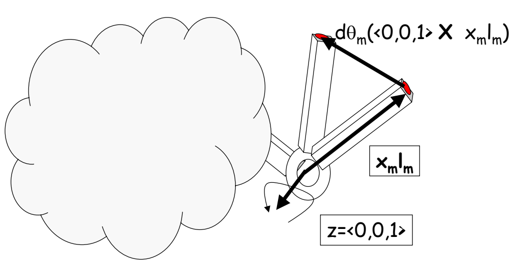
the change in v when rotated some small θ about n

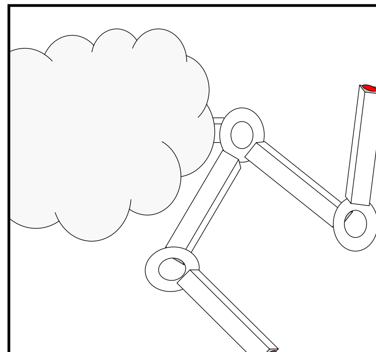
LINEAR!!

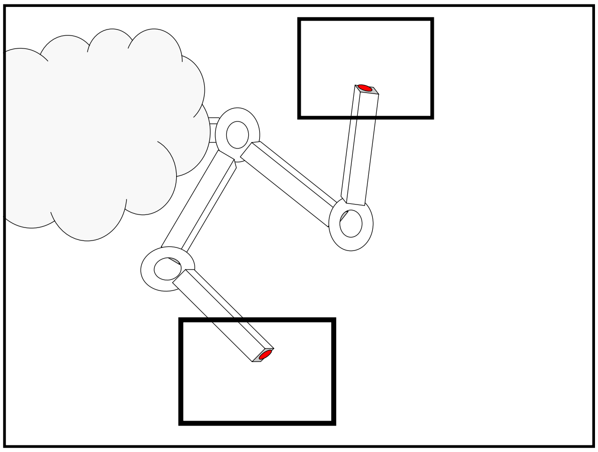
when dθ is small

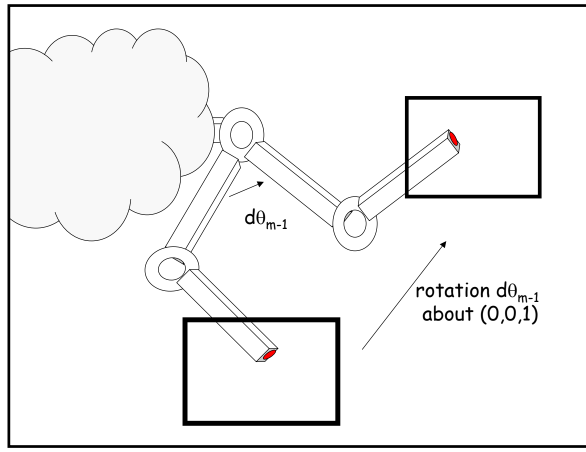
$\Delta v \approx d\theta\ (n \times v)$

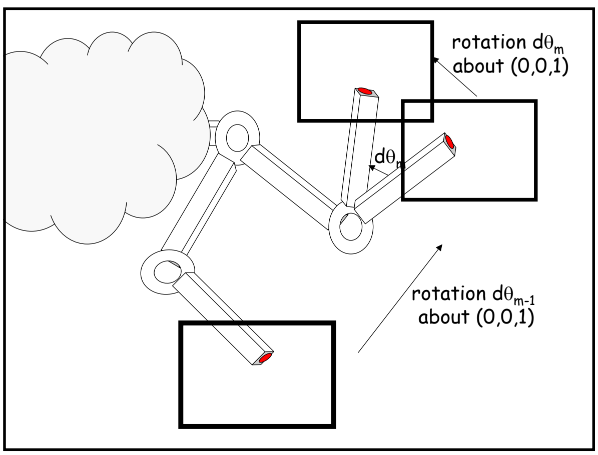
the change in v when rotated some small dθ about n

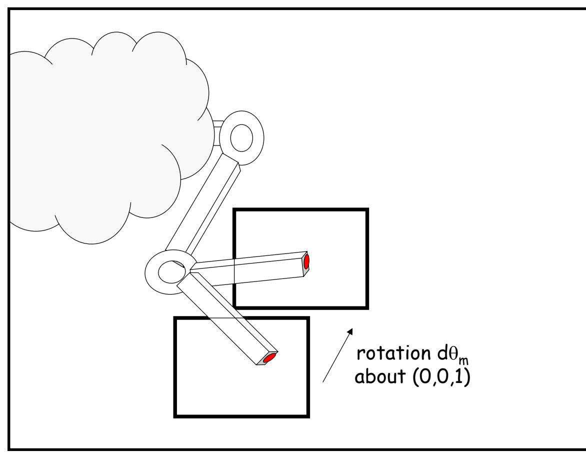


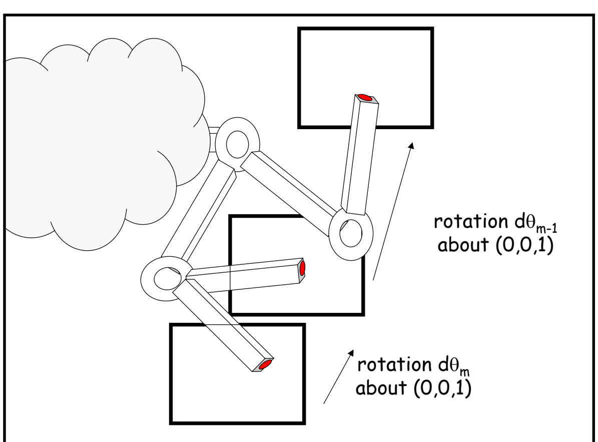






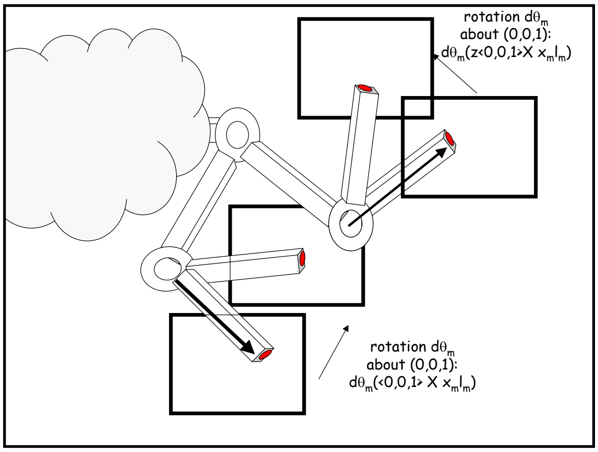


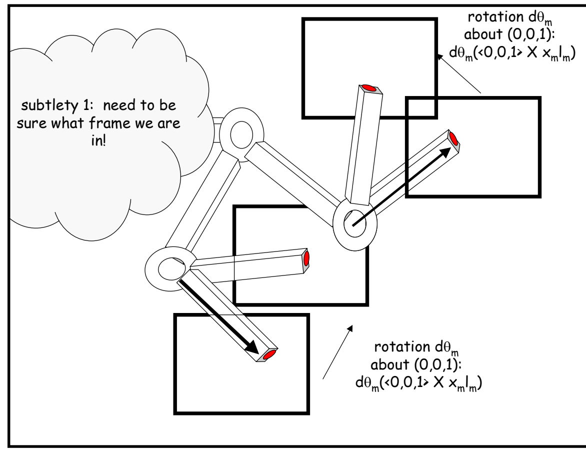


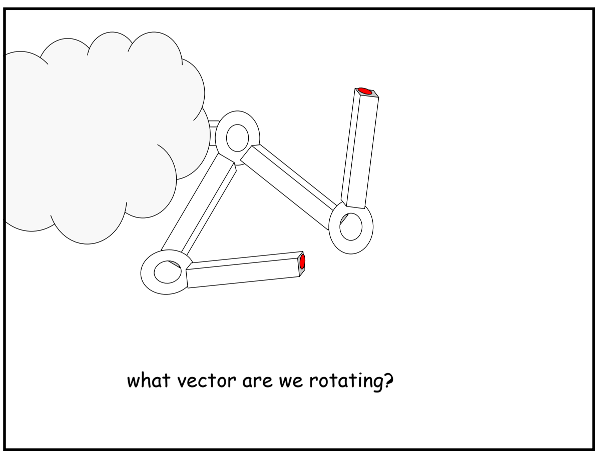


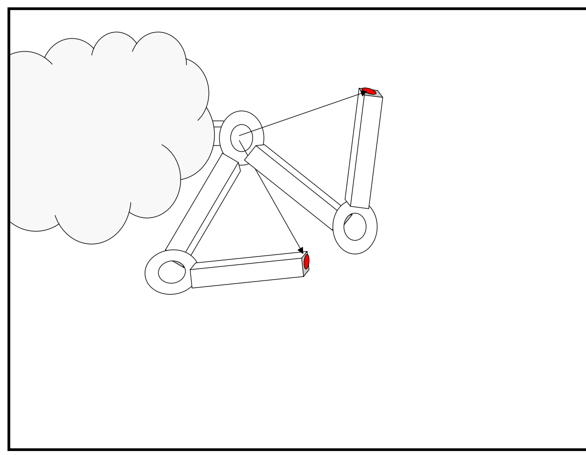
Question: What is the difference?

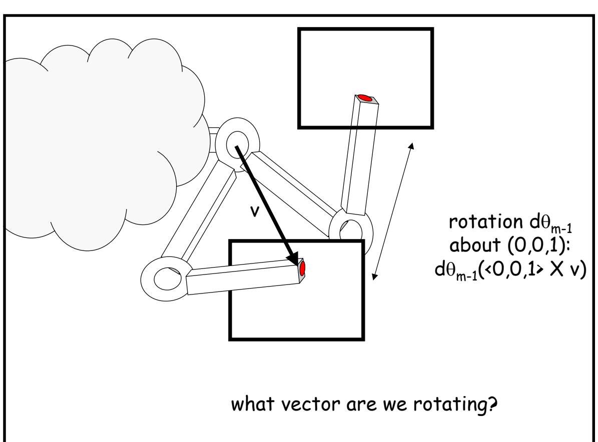
Hint: What vector are we rotating?

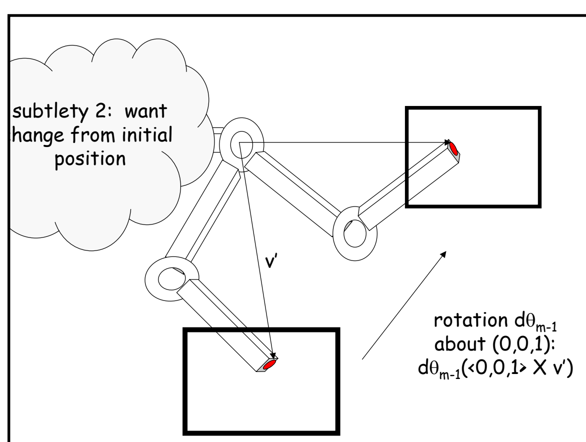

rotation $d\theta_m$
about (0,0,1):
$d\theta_m(z\langle 0,0,1\rangle X\ x_m l_m)$
rotation $d\theta_m$
about (0,0,1):
$d\theta_m(\langle 0,0,1\rangle\ X\ x_m l_m)$

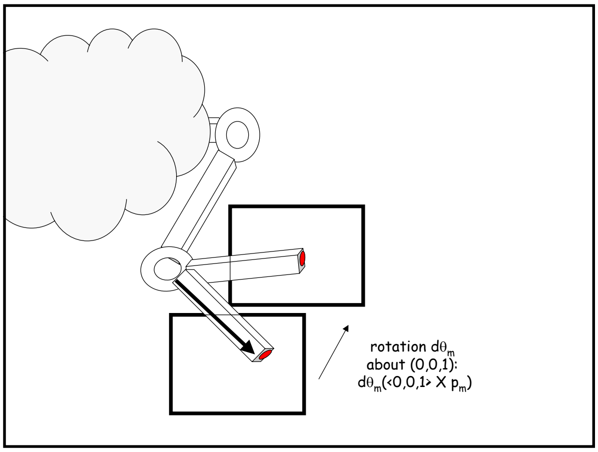

rotation $d\theta_m$
about (0,0,1):
$d\theta_m(\langle 0,0,1\rangle\ X\ x_m l_m)$
subtlety 1: need to be sure what frame we are in!
rotation $d\theta_m$
about (0,0,1):
$d\theta_m(\langle 0,0,1\rangle\ X\ x_m l_m)$

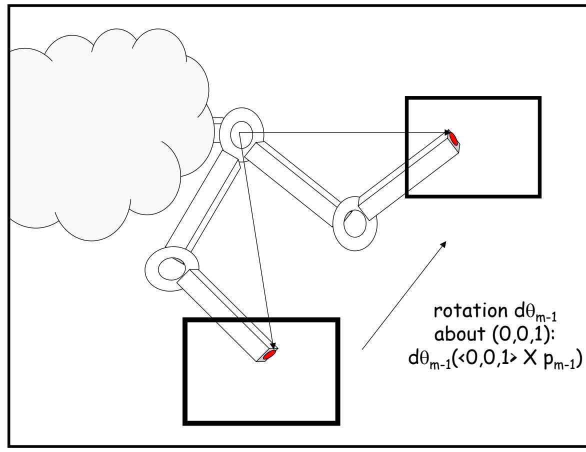

what vector are we rotating?

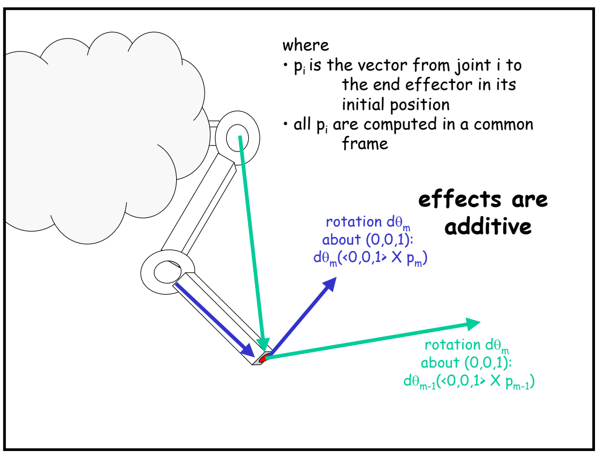



v
rotation $d\theta_{m-1}$
about (0,0,1):
$d\theta_{m-1}(\langle 0,0,1\rangle\ X\ v)$
what vector are we rotating?


subtlety 2: want hange from initial position
v'
rotation $d\theta_{m-1}$
about (0,0,1):
$d\theta_{m-1}(\langle 0,0,1\rangle\ X\ v')$







We need an algorithm for computing $J(\theta)$!

Intuition: The Jacobian represents velocity of end-effector in terms of state velocities

In General ...

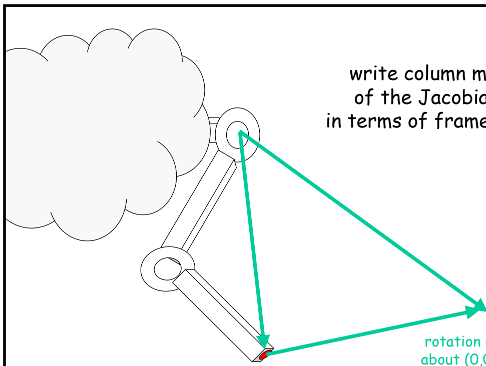
$$\begin{bmatrix} dE_x \\ dE_y \\ dE_z \\ dE_\phi \\ dE_\rho \\ dE_\sigma \end{bmatrix} = J(\theta) \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_m \end{bmatrix}$$

$J(\theta)$: 6 X m

In General ...

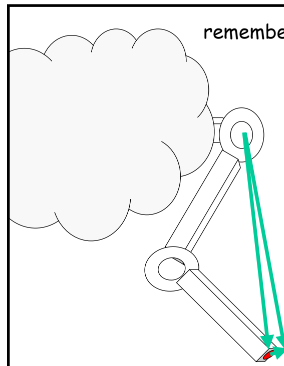
$$\begin{bmatrix} dE_x \\ dE_y \\ dE_z \\ dE_\phi \\ dE_\rho \\ dE_\sigma \end{bmatrix} = \begin{bmatrix} dE_x/d\theta_1 & dE_x/d\theta_2 & \dots & dE_x/d\theta_m \\ dE_y/d\theta_1 & dE_y/d\theta_2 & & dE_y/d\theta_m \\ \vdots & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_m \end{bmatrix}$$

write column m-1
of the Jacobian
in terms of frame m-1



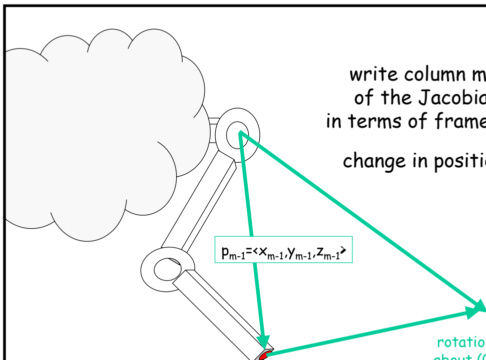
remember change in angles are small

write column m-1
of the Jacobian
in terms of frame m-1



write column m-1
of the Jacobian
in terms of frame m-1

change in position



change in position

rotation $d\theta_m$
about (0,0,1):
$d\theta_{m-1}(\langle 0,0,1\rangle \times \langle x_{m-1},y_{m-1},z_{m-1}\rangle)$

$(\langle 0,0,1\rangle \times \langle x_{m-1},y_{m-1},z_{m-1}\rangle)_x$
$(\langle 0,0,1\rangle \times \langle x_{m-1},y_{m-1},z_{m-1}\rangle)_y$
$(\langle 0,0,1\rangle \times \langle x_{m-1},y_{m-1},z_{m-1}\rangle)_z$