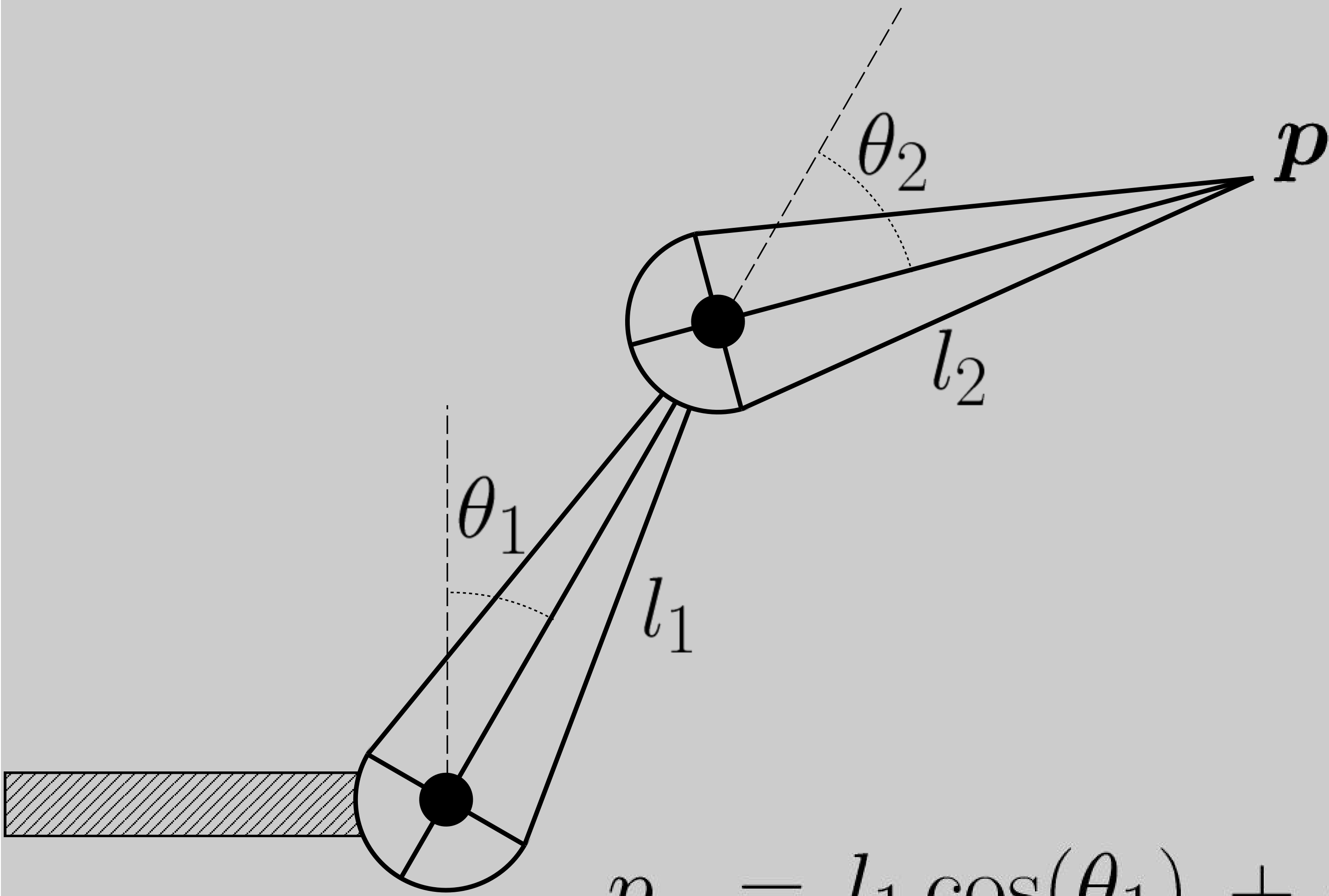
Inverse Kinematics

Computer Graphics/Animation

Prof. James O'Brien

Simple System: A Two Segment Arm

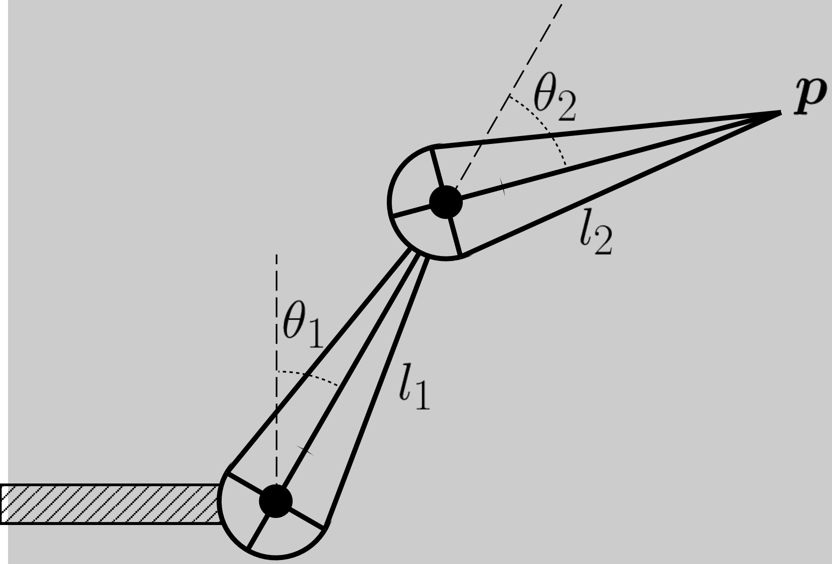


$$p_z = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$p_x = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Warning: Z−up Coordinate System

Direct IK: Solve for θ_1 and θ_2

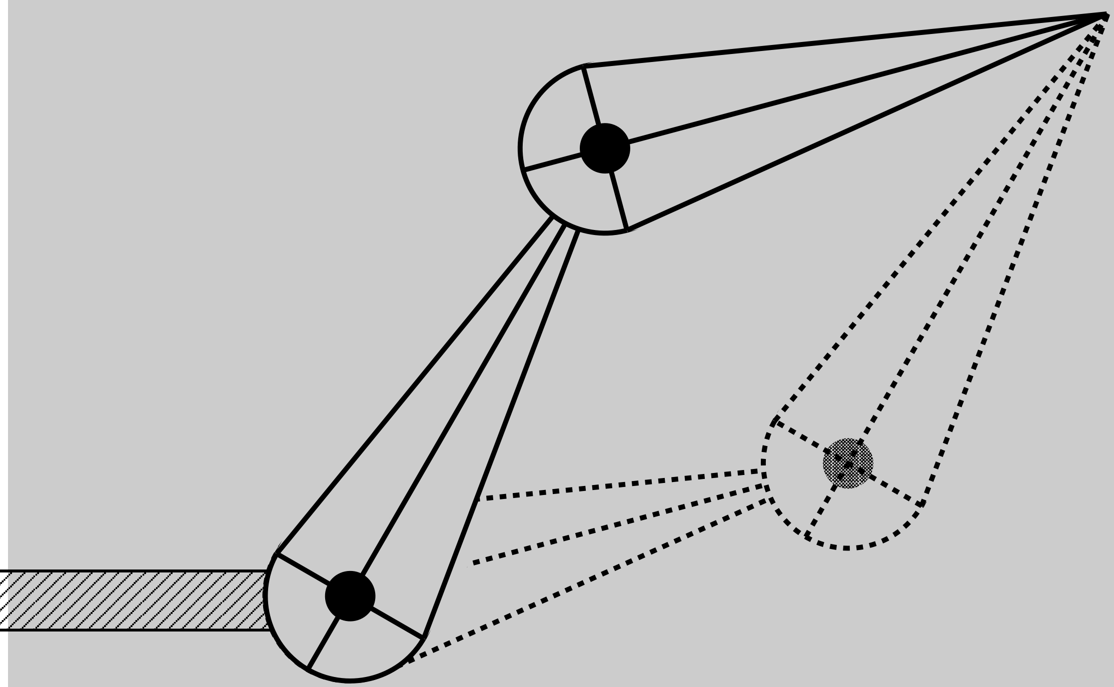


$$\theta_2 = \cos^{-1}\left(\frac{p_z^2 + p_x^2 - l_1^2 - l_2^2}{2l_1 l_2}\right)$$

$$\theta_1 = \frac{-p_z l_2 \sin(\theta_2) + p_x(l_1 + l_2\cos(\theta_2))}{p_x l_2 \sin(\theta_2) + p_z(l_1 + l_2\cos(\theta_2))}$$

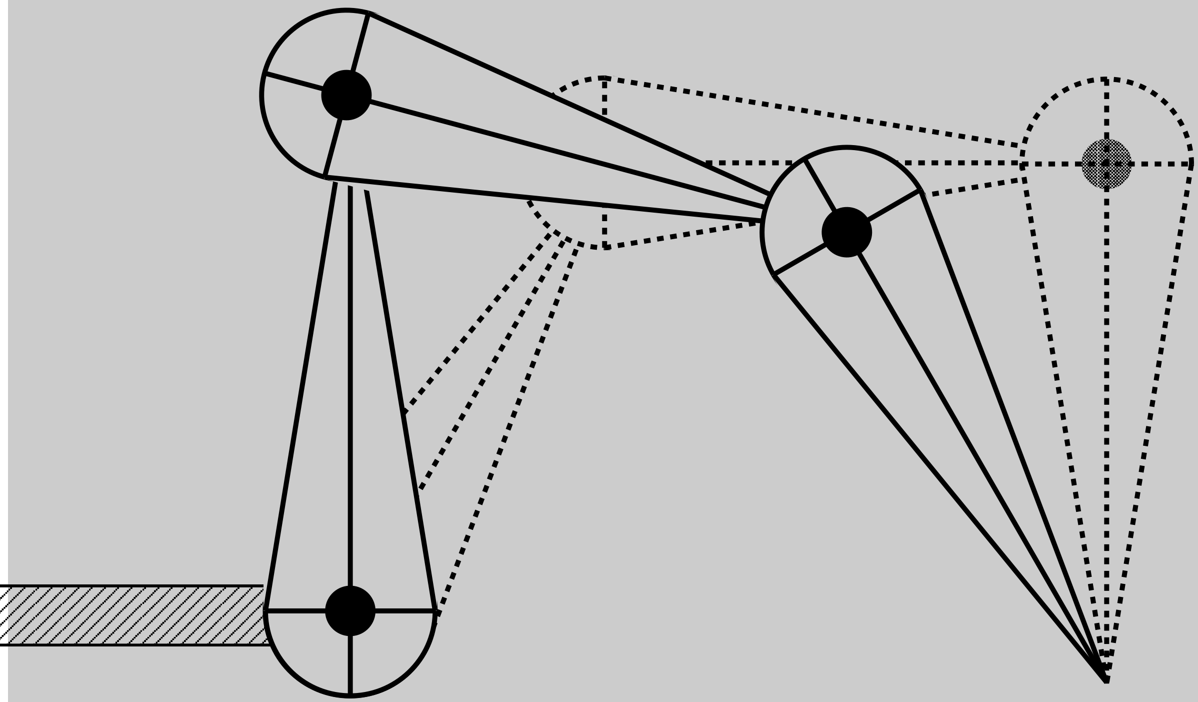
Why is this a hard problem?

Multiple solutions separated in configuration space



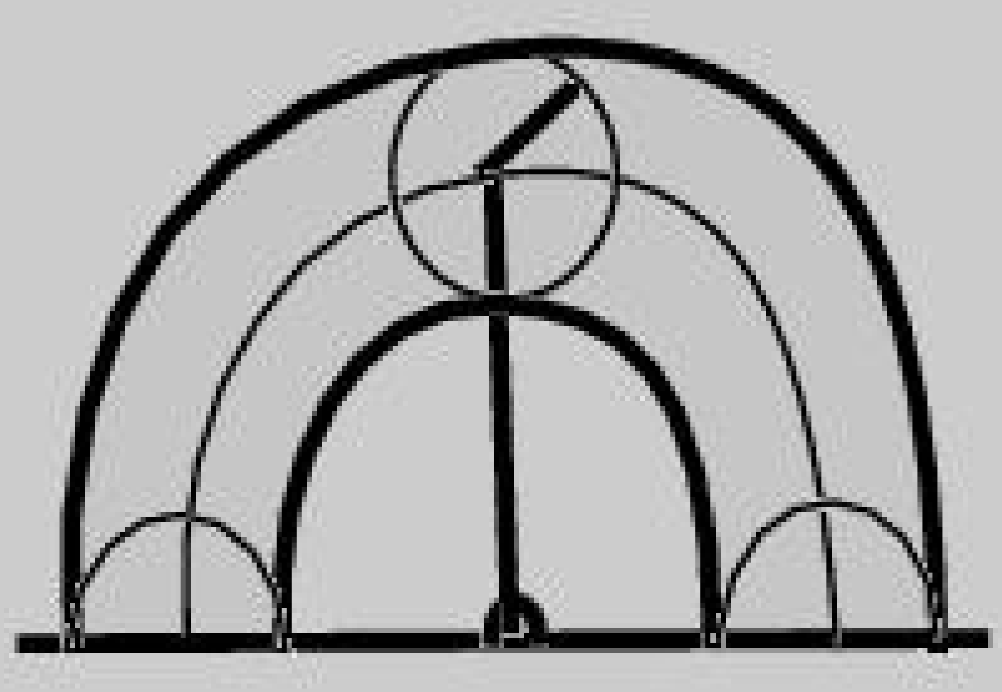
Why is this a hard problem?

Multiple solutions connected in configuration space



Why is this a hard problem?

Solution may not exist



From Parent, page 185

Numerical Solution

Start in some initial configuration

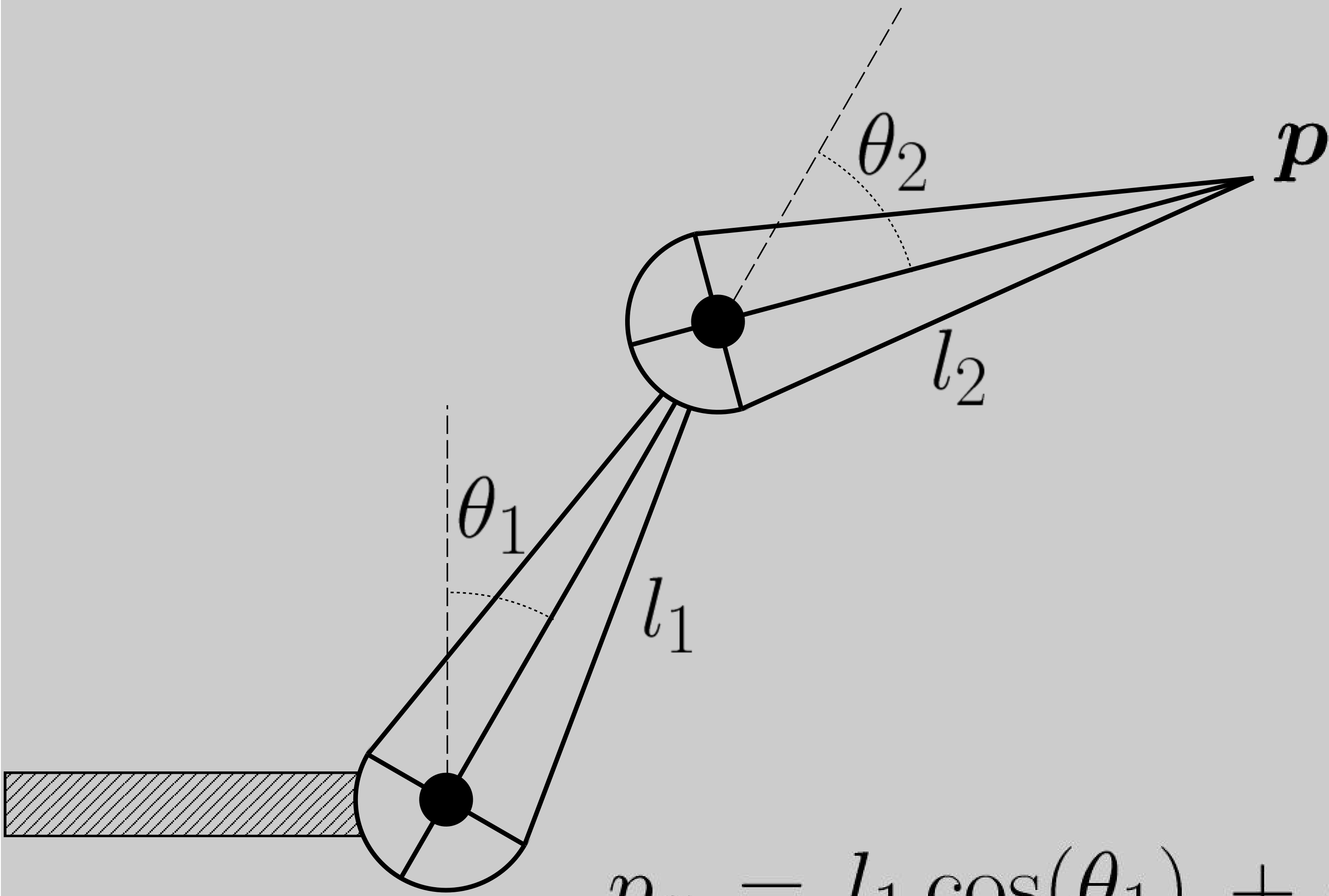
Define an error (e.g. goal pos − current pos)

Compute Jacobian of error w.r.t inputs

Use some numerical method to eliminate error as if Jacobian were constant

Iterate...

Simple System: A Two Segment Arm

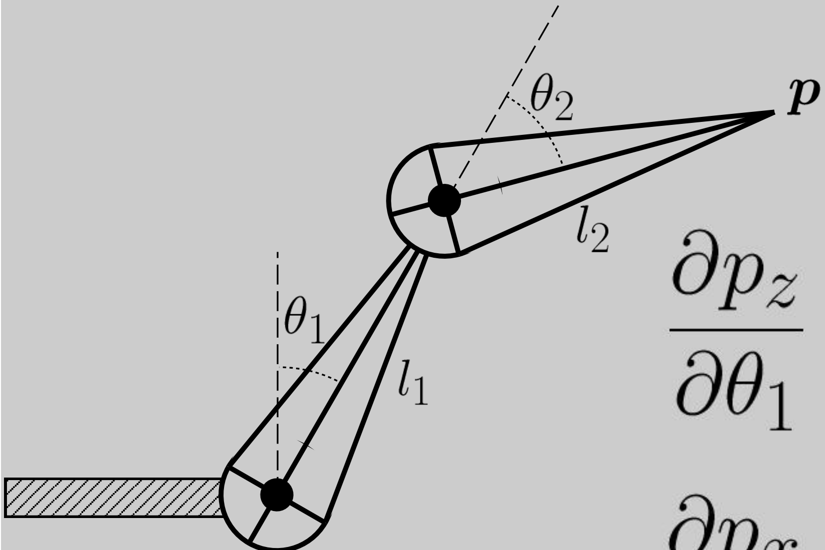


$$p_z = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$p_x = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Warning: Z−up Coordinate System

Simple System: A Two Segment Arm



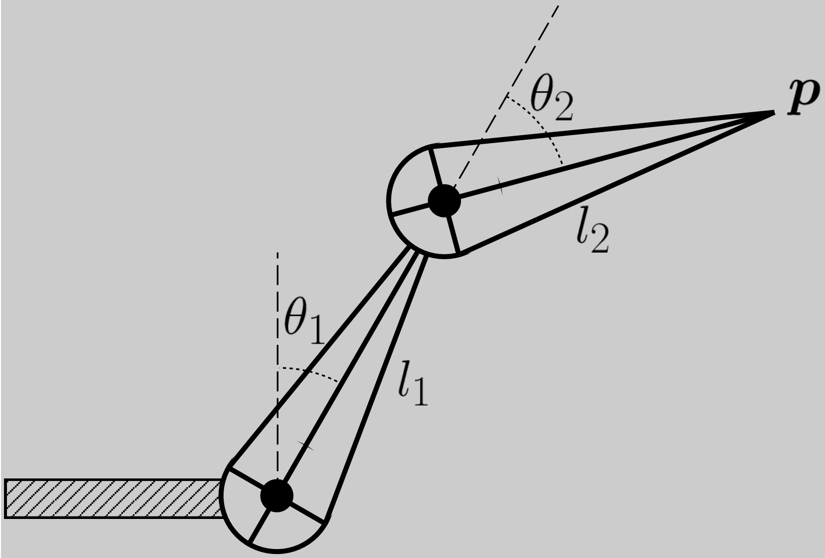
$$\frac{\partial p_z}{\partial \theta_1} = -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2)$$

$$\frac{\partial p_x}{\partial \theta_1} = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$\frac{\partial p_z}{\partial \theta_2} = - l_2 \sin(\theta_1 + \theta_2)$$

$$\frac{\partial p_x}{\partial \theta_2} = + l_2 \cos(\theta_1 + \theta_2)$$

Simple System: A Two Segment Arm



Direction in Config. Space

$$\theta_1 = c_1\theta_*$$

$$\theta_2 = c_2\theta_*$$

$$\frac{\partial p_z}{\partial \theta_*} = c_1\frac{\partial p_z}{\partial \theta_1} + c_2\frac{\partial p_z}{\partial \theta_2}$$

The Jacobian (of p w.r.t. θ)

$$J_{ij} = \frac{\partial p_i}{\partial \theta_j}$$

Example for two segment arm

$$J = \begin{bmatrix} \frac{\partial p_z}{\partial \theta_1} & \frac{\partial p_z}{\partial \theta_2} \\ \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} \end{bmatrix}$$

The Jacobian (of *p* w.r.t. θ)

$$J = \begin{bmatrix} \frac{\partial p_z}{\partial \theta_1} & \frac{\partial p_z}{\partial \theta_2} \\ \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{p}}{\partial \theta_*} = J \cdot \begin{bmatrix} \frac{\partial \theta_1}{\partial \theta_*} \\ \frac{\partial \theta_2}{\partial \theta_*} \end{bmatrix} = J \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

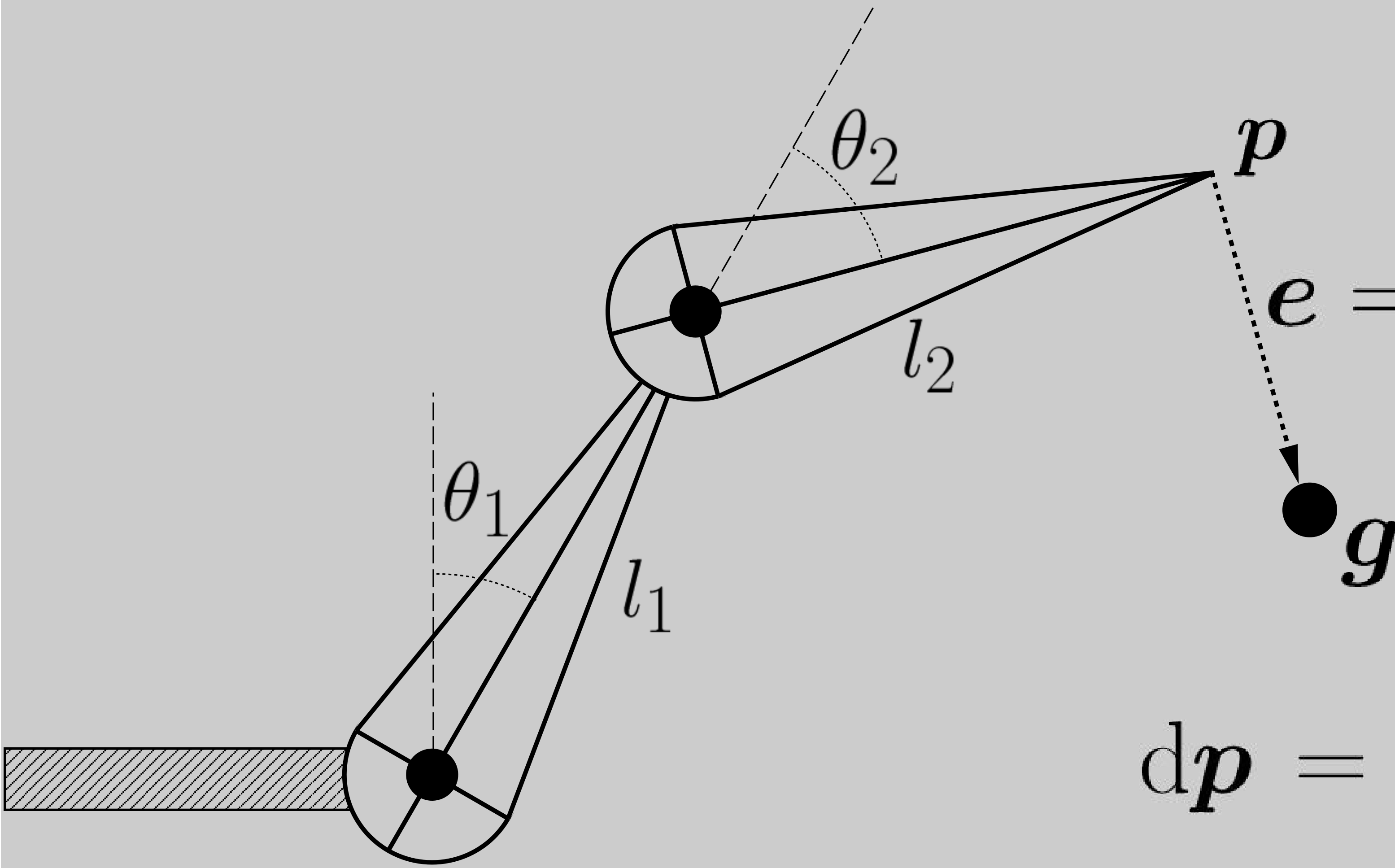
Solving for c_1 and c_2

$$\boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \qquad \mathrm{d}\boldsymbol{p} = \begin{bmatrix} \mathrm{d}p_z \\ \mathrm{d}p_x \end{bmatrix}$$

$$\mathrm{d}\boldsymbol{p} = J \cdot \boldsymbol{c}$$

$$\boldsymbol{c} = J^{-1} \cdot \mathrm{d}\boldsymbol{p}$$

Solving for c_1 and c_2



$$\boldsymbol{e} = \mathrm{d}\boldsymbol{p}$$

$$\mathrm{d}\boldsymbol{p} = J \cdot \boldsymbol{c}$$

$$\boldsymbol{c} = J^{-1} \cdot \mathrm{d}\boldsymbol{p}$$

Is the Jacobian invertible?

Problems...

Jacobian may (will) not be invertible

Option #1: Use pseudo inverse (SVD)

Option #2: Use iterative method

Jacobian is not constant

$$J = \begin{bmatrix} \frac{\partial p_z}{\partial \theta_1} & \frac{\partial p_z}{\partial \theta_2} \\ \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} \end{bmatrix} = J(\theta)$$

Non−linear optimization...
but problem is well behaved (mostly)

More Complex Systems

More complex joints (prism and ball)

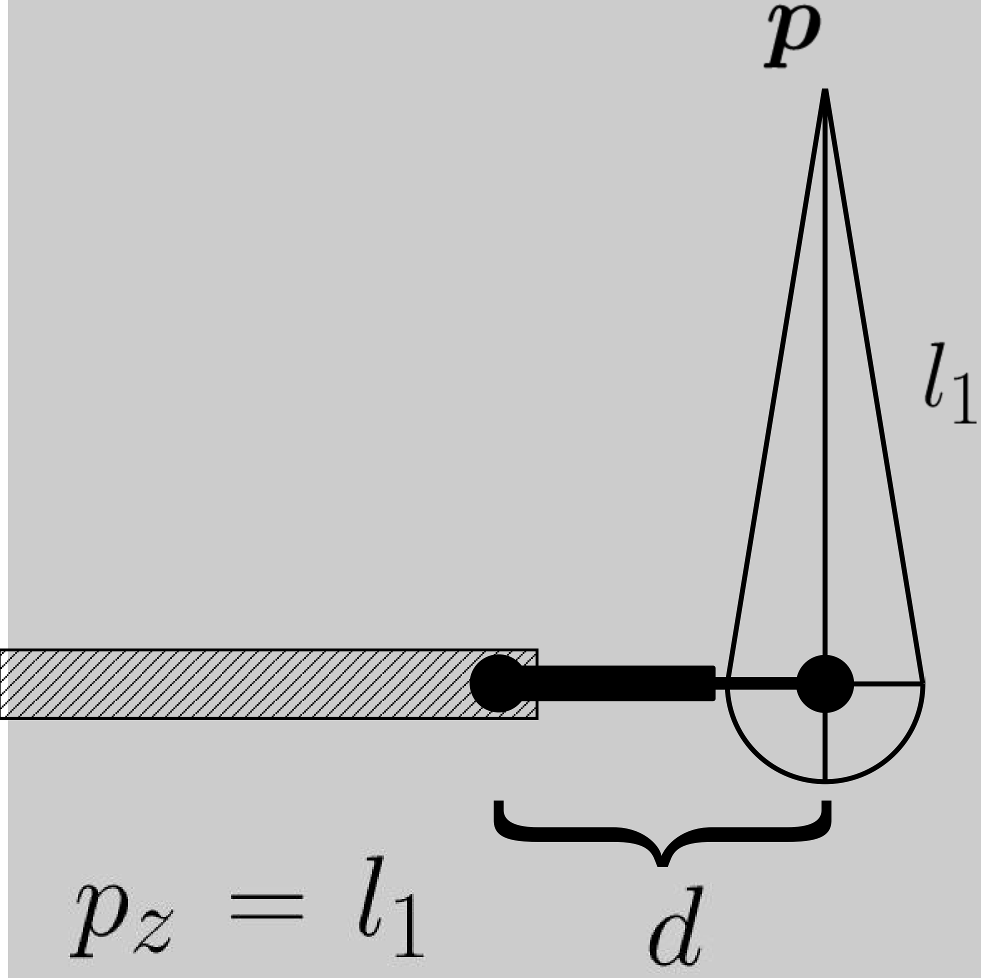
More links

Other criteria (COM over sup. poly.)

Hard constraints (joint limits)

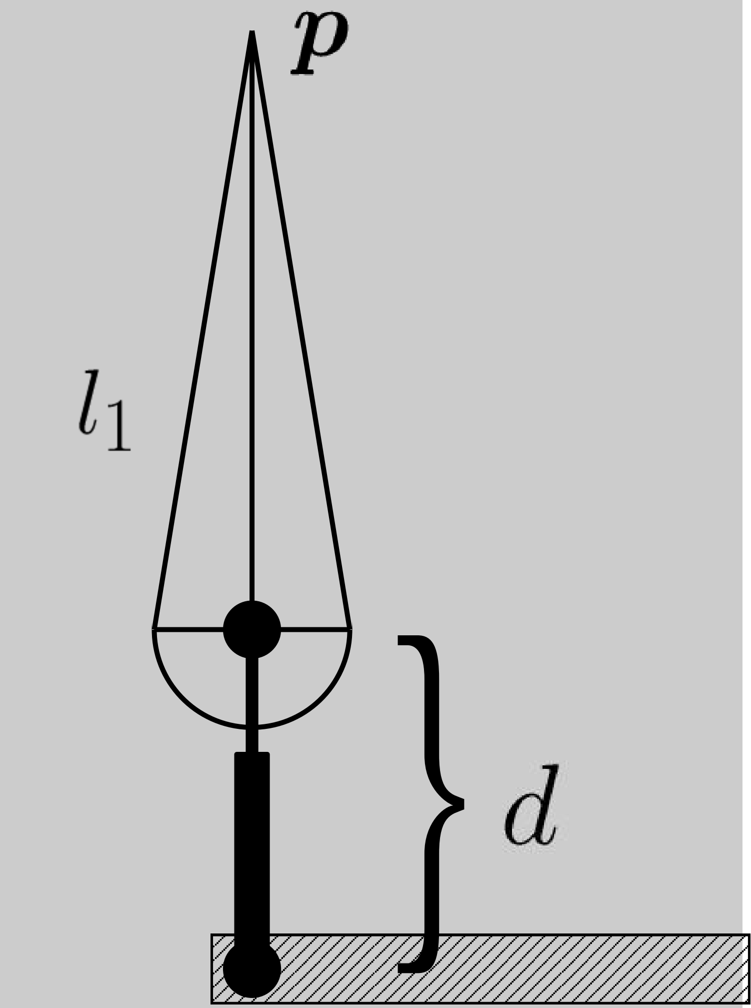
Multiple chains

Prism Joints



$$p_z = l_1$$

$$p_x = d$$

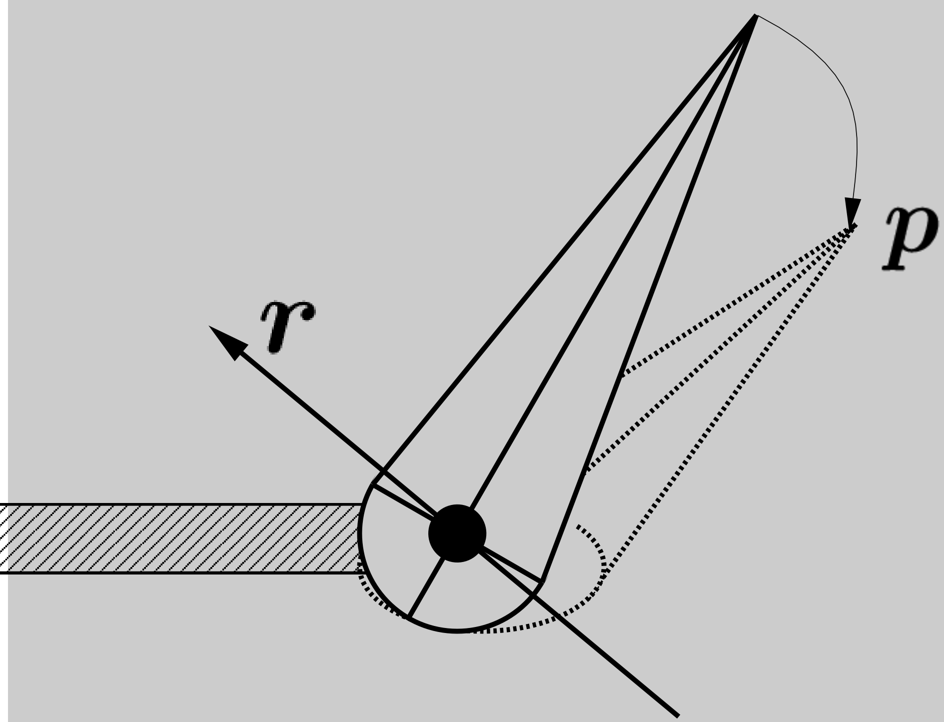


$$p_z = l_1 + d$$

$$p_x = 0$$

Ball Joints

$$\begin{aligned} \boldsymbol{p} &= \hat{\boldsymbol{r}}(\hat{\boldsymbol{r}} \cdot \boldsymbol{x}) \\ &+ \sin(||\boldsymbol{r}||)(\hat{\boldsymbol{r}} \times \boldsymbol{x}) \\ &- \cos(||\boldsymbol{r}||)(\hat{\boldsymbol{r}} \times (\hat{\boldsymbol{r}} \times \boldsymbol{x})) \end{aligned}$$



Ball Joints (moving axis)

$$\mathrm{d}\boldsymbol{p} = [\mathrm{d}\boldsymbol{r}]\cdot e^{[\boldsymbol{r}]}\cdot\boldsymbol{x} = [\mathrm{d}\boldsymbol{r}]\cdot\boldsymbol{p} = \underbrace{-[\boldsymbol{p}]}\cdot\mathrm{d}\boldsymbol{r}$$

That is the Jacobian for this joint

$$[\boldsymbol{r}] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$$

$$[\boldsymbol{r}]\cdot\boldsymbol{x} = \boldsymbol{r}\times\boldsymbol{x}$$

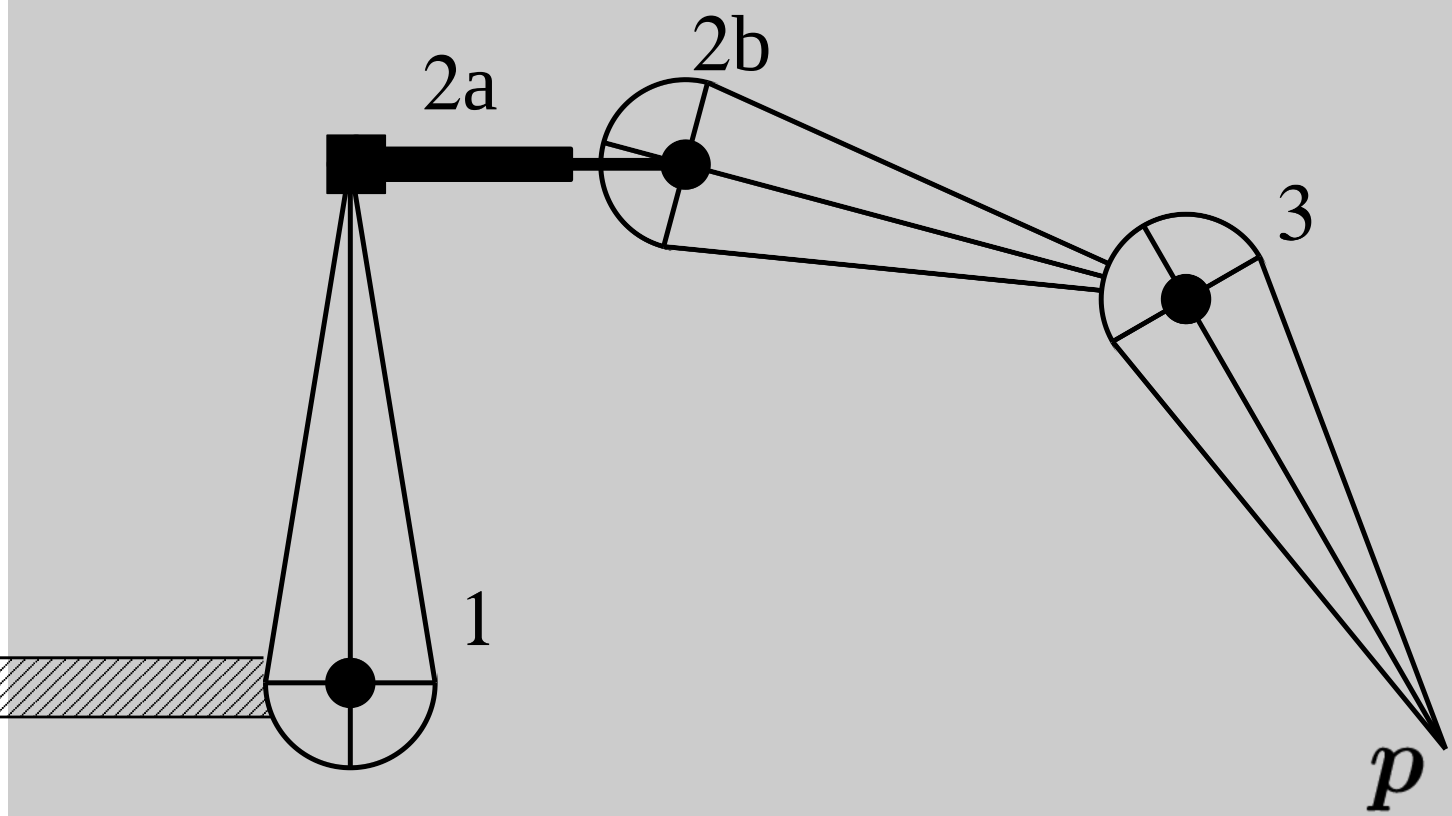
Ball Joints (fixed axis)

$$\mathrm{d}\boldsymbol{p} = (\mathrm{d}\theta)[\hat{\boldsymbol{r}}] \cdot \boldsymbol{x} = \underbrace{-[\boldsymbol{x}] \cdot}\hat{\boldsymbol{r}}\mathrm{d}\theta$$

That is the Jacobian for this joint

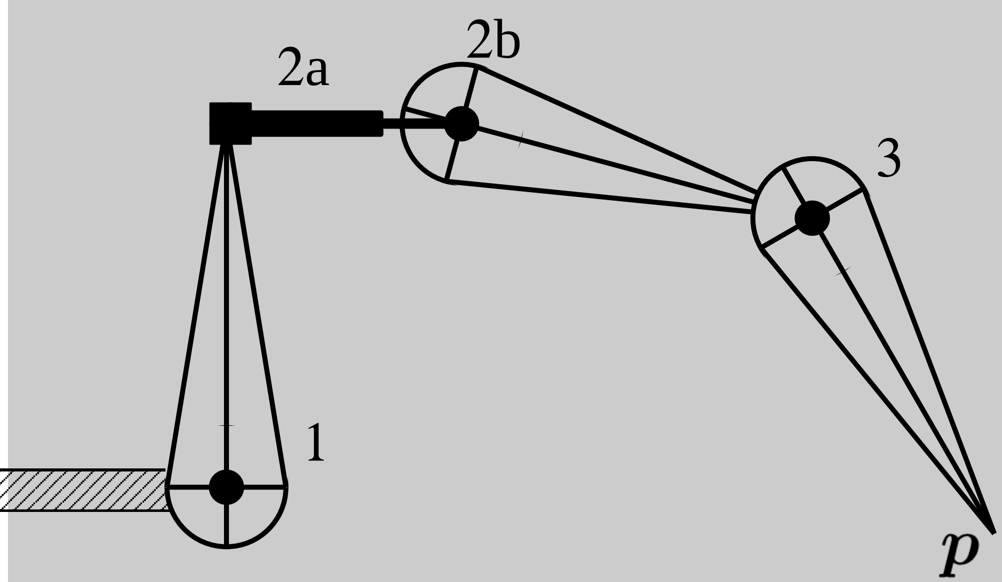
Many Links/Joints

We need a generic method of building Jacobian



Many Links/Joints

$$\tilde{J} = [J_3\ J_{2\mathrm{b}}\ J_{2\mathrm{a}}\ J_{1\mathrm{b}}]$$



$$\boldsymbol{d} = \begin{bmatrix} d_3 \\ d_{2\mathrm{b}} \\ d_{2\mathrm{a}} \\ d_{1\mathrm{b}} \end{bmatrix}$$

$$\mathrm{d}\boldsymbol{p} \neq \tilde{J} \cdot \mathrm{d}\boldsymbol{d}$$

Many Links/Joints

Transformation from body to parent

$$X_{(i-1)\leftarrow i} = \begin{bmatrix} R_{(i-1)\leftarrow i} & \boldsymbol{t}_{(i-1)\leftarrow i} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Rotation Portion
(May include scale as well)

Translation Portion

Many Links/Joints

Transformation from body to world

$$X_{0\leftarrow i} = \prod_{j=1}^{i} X_{(j-1)\leftarrow j} = X_{0\leftarrow 1} \cdot X_{1\leftarrow 2} \cdots$$

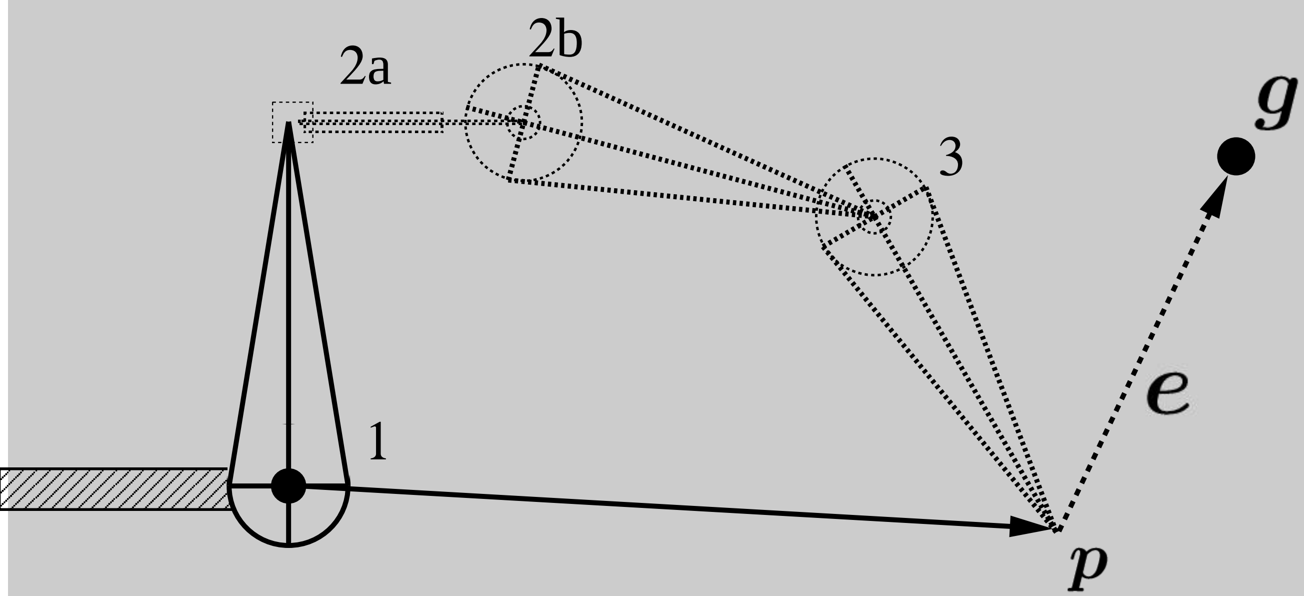
Rotation from body to world

$$R_{0\leftarrow i} = \prod_{j=1}^{i} R_{(j-1)\leftarrow j} = R_{0\leftarrow 1} \cdot R_{1\leftarrow 2} \cdots$$

Many Links/Joints

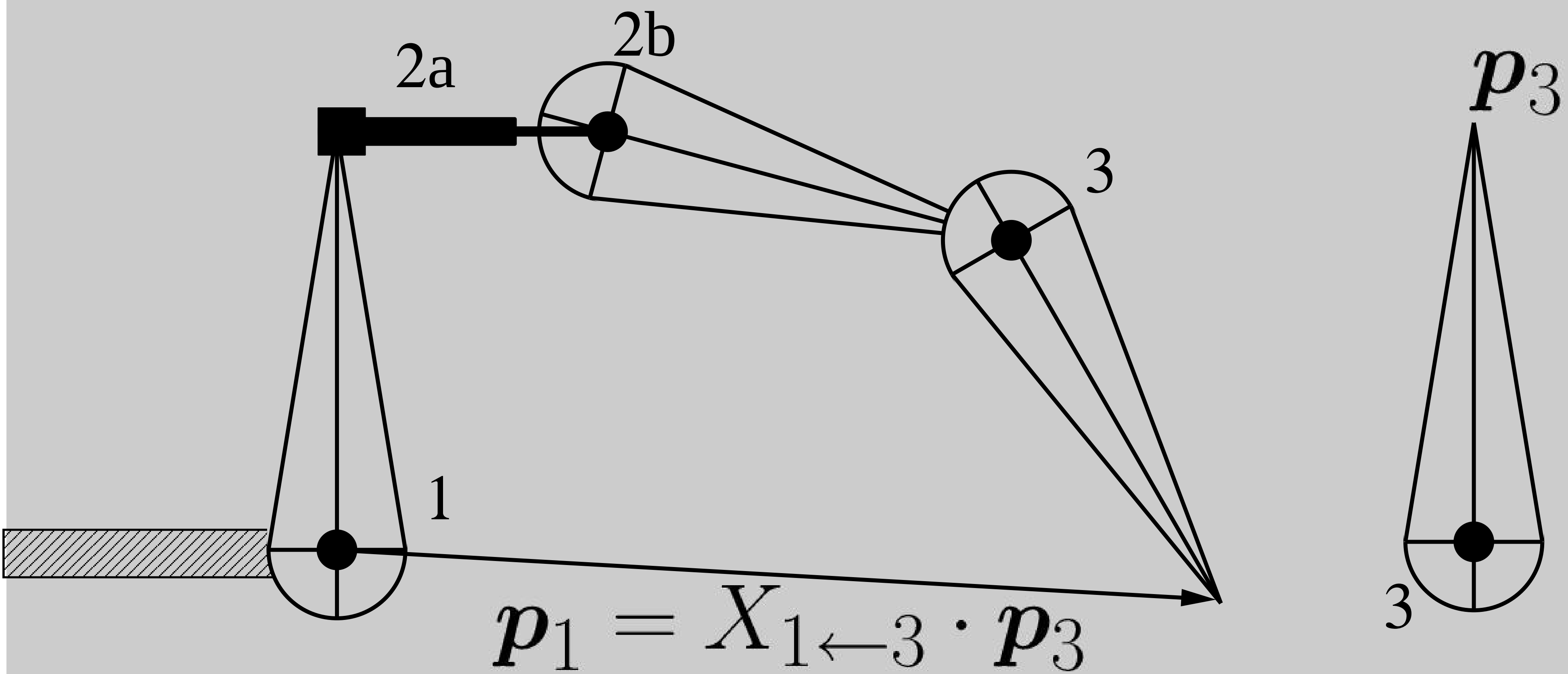
Jacobian is function of theta <u>and</u> error

$$J(\theta) = J(\theta, \boldsymbol{e})$$



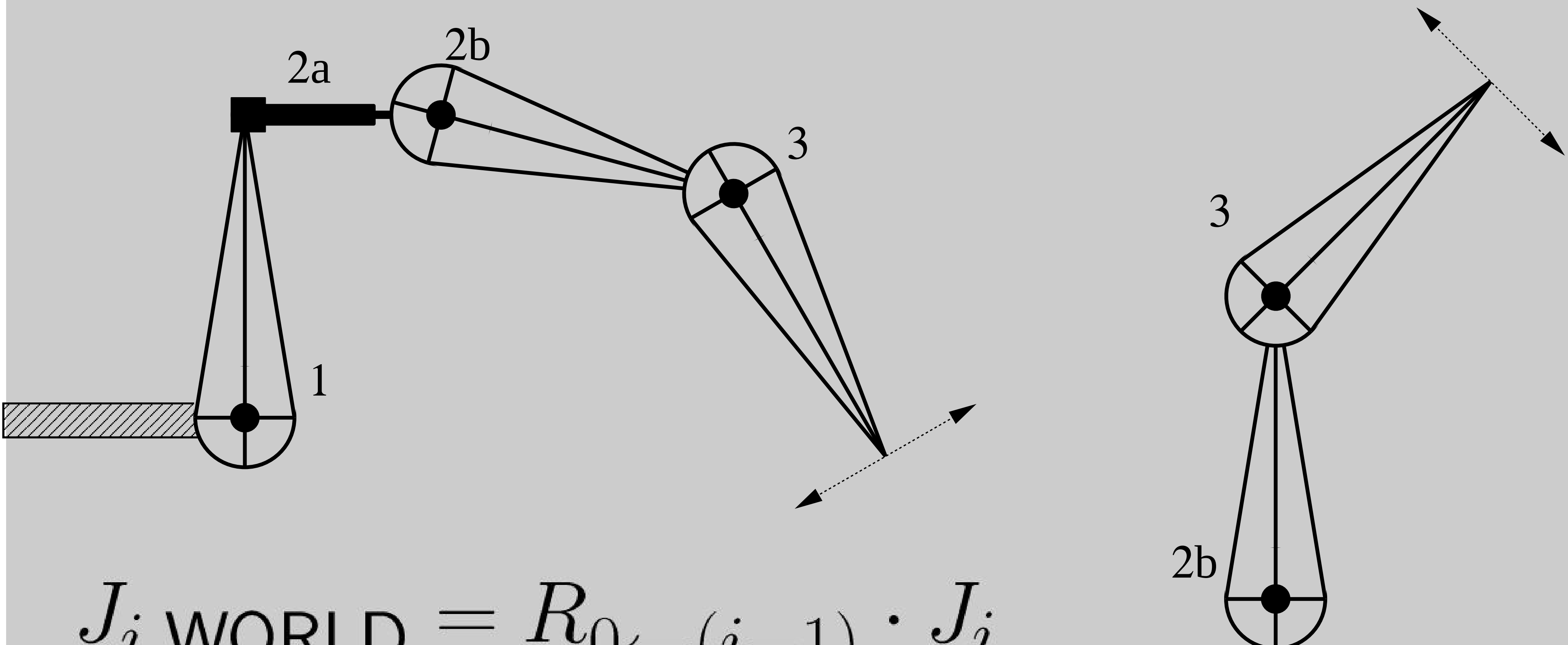
Many Links/Joints

Jacobian is function of theta <u>and</u> error



Many Links/Joints

Need to transform Jacobians to common coordinate system (WORLD)



$$J_{i,\mathrm{WORLD}} = R_{0\leftarrow(i-1)} \cdot J_i$$

Many Links/Joints

$$J = \begin{bmatrix} R_{0\leftarrow 2\mathrm{b}}\cdot & J_3(\theta_3, \boldsymbol{p_3}) \\ R_{0\leftarrow 2\mathrm{a}}\cdot & J_{2\mathrm{b}}(\theta_{2\mathrm{b}}, X_{2\mathrm{b}\leftarrow 3}\cdot \boldsymbol{p_3}) \\ R_{0\leftarrow 1}\cdot & J_{2\mathrm{a}}(\theta_{2\mathrm{a}}, X_{2\mathrm{a}\leftarrow 3}\cdot \boldsymbol{p_3}) \\ & J_1(\theta_1, X_{1\leftarrow 3}\cdot \boldsymbol{p_3}) \end{bmatrix}^{\mathsf{T}}$$

$$\boldsymbol{d} = \begin{bmatrix} d_3 \\ d_{2\mathrm{b}} \\ d_{2\mathrm{a}} \\ d_{1\mathrm{b}} \end{bmatrix}$$

Note: Each row in the above should be transposed....

$$\mathrm{d}\boldsymbol{p} = J \cdot \mathrm{d}\boldsymbol{d}$$

Other criteria (COM over sup. poly.)

Hard constraints (joint limits)

Multiple chains