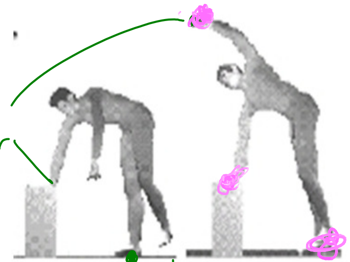
Inverse Kinematics



[Fanuc]

[Ronan Boulic]

- position "end effectors"
- convenient user interface
- issues: underconstrained
  solution might not exist.

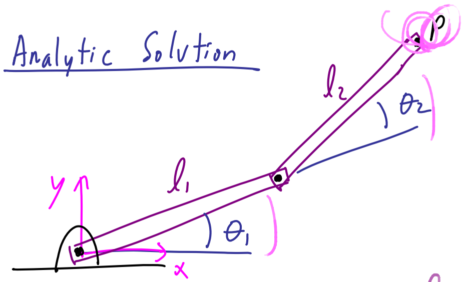
joint coords — cartesian coords

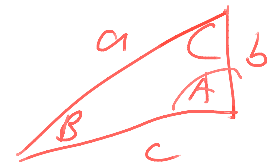
$$\vec{\theta} \underset{IK}{\overset{FK}{\rightleftarrows}} \vec{p}$$

Solutions

1. Analytic
2. Cyclic Coordinate Descent
3. Gradient Descent
4. Gauss Newton
5. Data Driven Methods

1 Analytic Solution





$$a^2 = b^2 + c^2 - 2bc\cos A$$

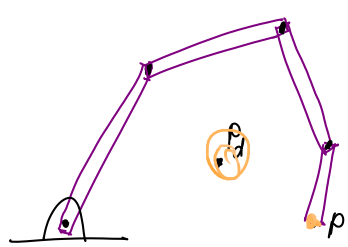
$$p_x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$p_y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

solve for θ_1, θ_2:

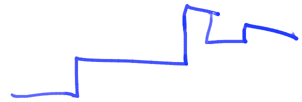
$$\theta_2 = \cos^{-1}\left(\frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

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

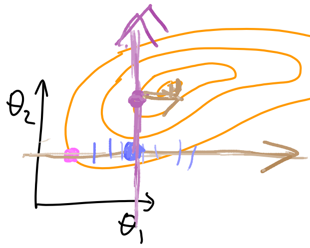
2, 3, 4, 5 IK as Optimization



minimize $f(\theta) = \| p_d - p(\theta) \|^2$



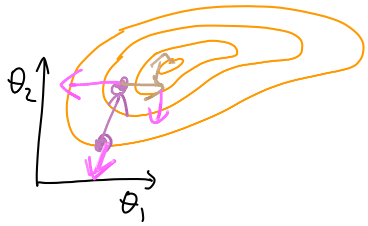
Building Block: Optimization



Coordinate Descent

for each direction in turn:
- which way is downhill
- take a step

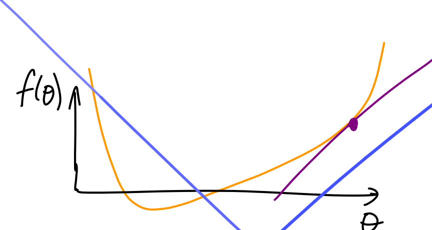
Gradient Descent



$$\theta_{n+1} = \theta_n - \alpha \nabla f(\theta)$$

α small

Newton's Method



$$f(\theta) = f(\theta_0) + (\theta - \theta_0)\frac{df}{d\theta} + \frac{(\theta-\theta_0)^2}{2}\frac{d^2 f}{d\theta^2} + \dots$$

Building Block: Solving Linear Systems

① $\begin{array}{cc} & 6 \\ 3 & [\;A\;] \end{array} [x] = [b]$ e.g. 3 eqns, 6 unknowns — underconstrained

$$\underbrace{[A^T]}_{6\times 3}\underbrace{[A]}_{3\times 6}[x] = [A^T][b]$$

$$(A^TA)\,x = A^Tb$$

$$x = \underbrace{(A^TA)^{-1}A^T}\,b$$

pseudoinverse of a matrix

→ minimizes $\|x\|^2$

Common solution:
solution is:

② $[A][x] = [b]$ eg. 3 unknowns, 6 equations

③ $Ax = b$

Pseudoinverse — Moore Penrose Inverse

$Ax = b$ $[A][x] = [b]$

$$A^TAx = A^Tb$$

$$[A^T][A][x] = [A^T][b]$$

$$[A^TA][x] = [A^T][b]$$

$$x = \underbrace{(A^TA)^{-1}A^T}_{A^+ \text{ pseudoinverse}}\,b$$

left pseudoinverse

$$A^+A = I$$

$$= (A^TA)^{-1}A^T A$$

$$= \underbrace{A^{-1}\underbrace{A^{T^{-1}}A^T}}A = I$$

right pseudoinverse

$$X = \underbrace{A^T(A\,A^T)^{-1}}_{A^+_{right}}\,b$$

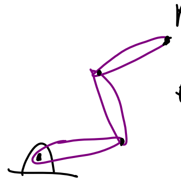
Building Block: Jacobian

- derivative of one vector quantity w.r.t. another vector quantity
- e.g. $\vec{P}(\vec{\theta})$

$$\frac{\partial \vec{P}}{\partial \vec{\theta}} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_1} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \end{bmatrix}$$

$$\frac{\Delta P}{\Delta t} = J \frac{\Delta \theta}{\Delta t}$$

$$\vec{V} = J \vec{\omega}$$



$P = \begin{bmatrix} x \\ y \end{bmatrix}$

$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \theta} \end{bmatrix} \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta \theta_3 \end{bmatrix}$$

Computing the Jacobian

Analytic Method

eg: 2 link robot

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

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

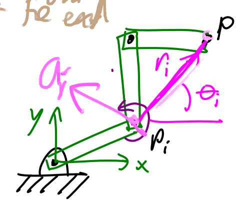
$$\frac{\partial \vec{P}}{\partial \vec{\theta}} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix}$$

Geometric Method

$$\frac{\partial P}{\partial \theta_i} = \vec{a}_i \times \vec{r}_i$$

vector from joint to end

joint axis



Finite Differences

$P = f(\theta_1, \theta_2, \theta_3)$

$$\frac{\partial P}{\partial \theta_2} =$$

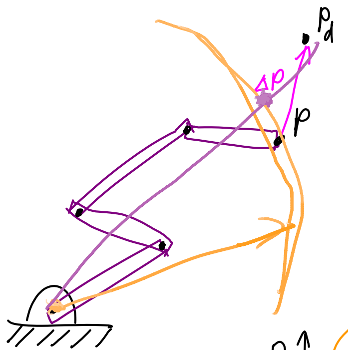
② Cyclic Coordinate Descent

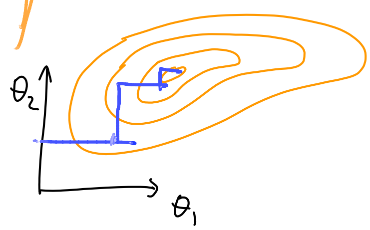
while $\|p - p_d\| > \epsilon$

for each joint

[swing until p is as close to p as possible]

analytic solution





③ Gradient Descent, "Jacobian Transpose"

$$f(p) = \frac{1}{2} \|p_d - p\|^2$$

$$\nabla f = \frac{\partial f}{\partial \theta} = \frac{\partial f}{\partial p} \frac{\partial p}{\partial \theta}$$ chain rule

$$\nabla f = -(p_d - p)^T \frac{\partial p}{\partial \theta}$$

$$[\ \nabla f\ ] = [\Delta x\ \Delta y\ \Delta z]\left[\ \frac{\partial p}{\partial \theta}\ \right] 3$$

$$[\nabla f] = \left[\frac{\partial p}{\partial \theta}\right]^T \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

$$\nabla f = J^T \Delta p$$

$$\theta_{n+1} = \theta_n - \alpha \nabla f(\theta)$$

$$\theta_{n+1} = \theta_n + \alpha J^T \cdot \Delta p$$

④ Gauss Newton, "pseudo inverse"

$$\Delta P \simeq \underbrace{\frac{\partial P}{\partial \theta}}_{J} \cdot \Delta\theta$$

$$J\Delta\theta = \Delta P$$

$$\Delta\theta = \underbrace{J^{+}}_{\text{right pseudo inverse}} \Delta P$$

$\alpha = 1$ or less

while $\|P - P_d\|^2 > \epsilon$

$$\theta_{n+1} = \theta_n + \alpha \underbrace{J^{+}\Delta P}_{\Delta\theta}$$

valid example pose

⑤ Data Driven Methods (machine learning) or regression or function approximation

$$\underset{\theta}{\text{minimize}}\, f(\theta) = \underbrace{w_1 \|P_d - P(\theta)\|^2}_{\text{meet IK constraint}} + \underbrace{w_2 \|\theta^* - \theta\|^2}_{\text{stay close to known valid poses}}$$

free parameters

eg.: "style based inverse kinematics" paper

Relative Merits

1. Analytic
   - closed form solution, no iteration
   - only for fully constrained IK
2. Cyclic Coordinate Descent
   - commonly used, fast, simple
   - choose joint order, no control over "style" of solution
3. Jacobian Transpose
   - simple, fast, commonly used
   - need to determine step size α
   - no user control over style
4. Pseudo inverse
   - more expensive
   - can add user control
5. Data Driven Methods
   - "smartest" ~~c~~
   - allows for user control over style
   - still more expensive than 2 3