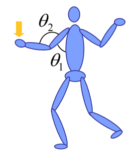
Inverse Kinematics

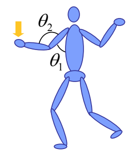
Jehee Lee
Seoul National University

Forward and Inverse Kinematics



$$(\mathbf{p}, \mathbf{q}) = \mathrm{F}(\theta_i)$$

Forward Kinematics

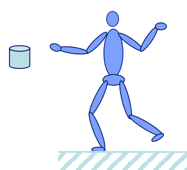


$$\theta_i = \mathrm{F}^{-1}(\mathbf{p}, \mathbf{q})$$

Inverse Kinematics

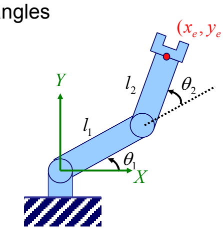
Why Inverse Kinematics ?

- Environmental interactions
  - Pick up an object or place feet on the ground
  - Hard to do with forward kinematics
- The pose of the character is described in the ***joint angle space***
- The environmental interaction is described in the ***work (Cartesian) space***

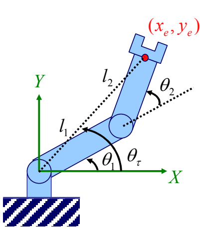


Inverse Kinematics: A Simple Example

- A simple robot arm in 2-dimensional space
  - Two revolute joints
  - The position of the end-effector is known
  - Compute joint angles



Analytic Solution for A Simple Example



$$\cos(\theta_\tau) = \frac{x_e}{\sqrt{x_e^2 + y_e^2}}$$

$$\theta_\tau = \cos^{-1}\left(\frac{x_e}{\sqrt{x_e^2 + y_e^2}}\right)$$

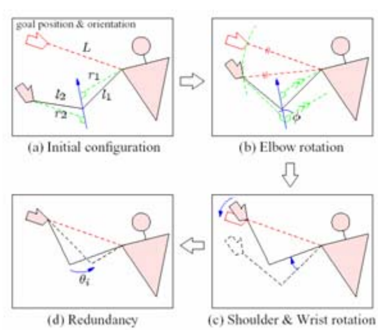
$$\cos(\theta_\tau - \theta_1) = \frac{l_1^2 + x_e^2 + y_e^2 - l_2^2}{2 l_1 \sqrt{x_e^2 + y_e^2}}$$

$$\theta_1 = \theta_\tau - \cos^{-1}\left(\frac{l_1^2 + x_e^2 + y_e^2 - l_2^2}{2 l_1 \sqrt{x_e^2 + y_e^2}}\right)$$

$$\cos(\pi - \theta_2) = \frac{l_1^2 + l_2^2 - x_e^2 - y_e^2}{2 l_1 l_2}$$

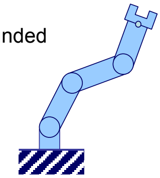
$$\theta_2 = \pi - \cos^{-1}\left(\frac{l_1^2 + l_2^2 - x_e^2 - y_e^2}{2 l_1 l_2}\right)$$

Redundancy in Human Arms



Why so difficult to get a closed-form solution ?

- Redundancies
  - Multiple solutions (# of unknowns > # of equations)
- Joint limit
  - The range of each unknown is bounded
- Reachable workspace
  - No solution, or
  - A unique solution, or
  - Multiple solutions
- Multiple goals
  - Four limbs may have constraints simultaneously
  - Intermediate links can also be constrained

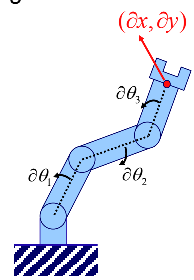


Iterative Methods

- Iteratively step all joints to the goal
  - Consider infinitesimal changes

compute $(\theta_1, \theta_2, \theta_3)$ from (x, y)

compute $(\partial\theta_1, \partial\theta_2, \partial\theta_3)$ from $(\partial x, \partial y)$



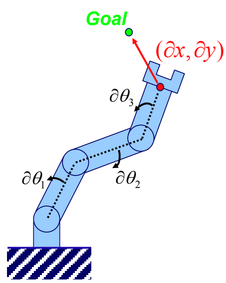
Jacobians of Forward Kinematics Map

- Forward Kinematics Map

$$(x, y) = (F_x, F_y) = F(\theta_1, \theta_2, \theta_3)$$

- Jacobian

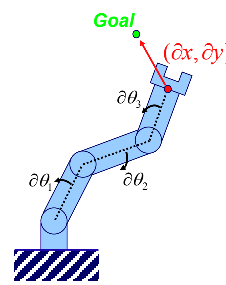
$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \mathbf{J} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = \begin{pmatrix} \frac{\partial F_x}{\partial \theta_1} & \frac{\partial F_x}{\partial \theta_2} & \frac{\partial F_x}{\partial \theta_3} \\ \frac{\partial F_y}{\partial \theta_1} & \frac{\partial F_y}{\partial \theta_2} & \frac{\partial F_y}{\partial \theta_3} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}$$



Jacobian of Forward Kinematics Map

- If the inverse of the Jacobian can be computed

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}_{t+1} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}_t + \Delta t \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}_t = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}_t + \Delta t \mathbf{J}^{-1} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}_t$$



A System of Linear Equations

$$\mathbf{J\theta} = \mathbf{x}$$

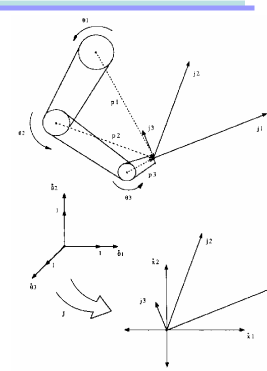
- # of unkowns = the dimension of $\theta = n$
- # of equations = the dimension of $\boldsymbol{x} = m$
- $\boldsymbol{J}$ is a (m x n) matrix

- The solution is unique if $m=n$
- The linear system is under-specified if $m<n$
- The linear system is over-specified if $m>n$

the pseudoinverse of Jacobian Is required

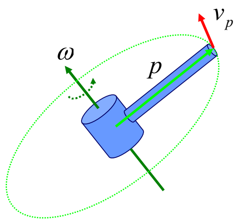
A Geometric Interpretation of Jacobian

- You don't have to differentiate the forward kinematics map to compute Jacobian
- The colume vectors (j1, j2, j3) of ***J*** are perpendicular to the corresponding vectors (p1, p2, p3) from the joint to the end-effector



Linear and Angular Velocities in 3D

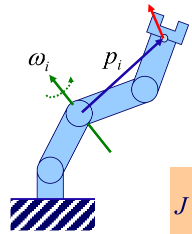
- Velocity of a point due to rotating joints



$$v_p = \omega \times p$$
$$\omega_p = \omega$$

Jacobian in 3D

- A serial chain with *n* revolute joints



$$\begin{pmatrix} v_{end-effector} \\ \omega_{end-effector} \end{pmatrix} = J \begin{pmatrix} \partial\theta_1 \\ \partial\theta_2 \\ \vdots \\ \partial\theta_n \end{pmatrix}$$

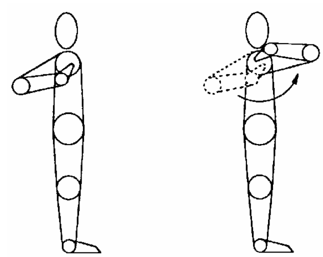
$$J = \begin{pmatrix} \omega_1 \times p_1 & \omega_2 \times p_2 & \cdots & \omega_n \times p_n \\ \omega_1 & \omega_2 & \cdots & \omega_n \end{pmatrix}$$

Redundancy

- A single solution must be chosen from multiple solutions
  - **“Closest” to the current configuration**
    - Pseudo inverse minimizes joint angle rates (locally)
  - **Move outermost links the most**
    - The outermost link sweeps a smallest region (visual change)
  - **Minimum time**
    - Dynamics involves
  - **Secondary goal**
    - Additional constraints
  - **Natural looking**
    - Biomechanical experiments

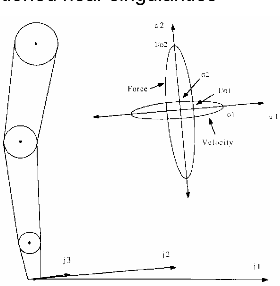
IK can be ill-conditioned

- An small change in the position of the end-effector could require substantial changes of joint angles



Singularity

- Singularities cause the rank of the Jacobian to change
- Often ill-conditioned near singularities



Iterative Method Using Jacobian

- Jacobian should be computed at every time step
- Simple Euler integration can be used
- Singularity could incur numerical instability

Non-Linear Optimization

- Non-linear programming
  - Numerical method for finding the (local) minimum of a non-linear function
- An efficient way of solving IK
- The path may not look natural
- Multiple goals can easily be handled
- We need to define
  - ***Objective function*** : Place hands at specific positions
  - ***Constraints*** : Joint limits

Objective Function

- Position $G(\theta) = \left\| \mathbf{p}(\theta) - \mathbf{p}_{goal} \right\|^2$
- Orientation $G(\theta) = \left\| \log\left(\mathbf{q}_{goal}^{-1}\mathbf{q}(\theta)\right) \right\|^2$
- Direction $G(\theta) = \left\| \mathbf{v}(\theta) - \mathbf{v}_{goal} \right\|^2$

Constrained Optimization

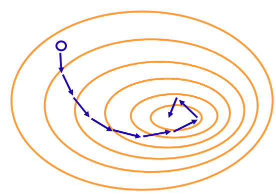
- Formulation

$$\begin{array}{ll} \text{Minimize} & G(\theta) \\ \text{subject to} & l_i \le \theta_i \le u_i \end{array}$$

- Use a standard numerical technique
  - Gradient decent
  - Conjugate gradient
  - Sequential quadratic programming

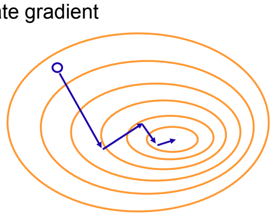
Basic Idea of Numerical Techniques

- Pseudoinverse + Euler integration
  - Constant time step



Basic Idea of Numerical Techniques

- Direction selection + Line minimization
  - Least squares
  - Gradient
  - Conjugate gradient

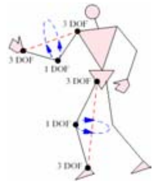


Quick and Dirty

- Cyclic coordinate decent
  - Try to solve the problem with subset of joints
  - If it fails, add a joint
- Jacobian transpose method
  - Use the transpose of Jacobian, instead of pseudoinverse
  - Do not guarantee least-squares

Hybrid Approach

- Arms and legs allow analytic solutions
  - Redundancy can be parameterized by elbow circles
- Reduced-coordinate formulation
  - We can replace 7 dof for each limb by a single elbow circle parameter
  - Hybrid formulation (numerical + analytic)
  - The numerical optimization can be done with fewer number of parameters



Data-Driven Approach

- IK is underdetermined
  - Many possible poses satisfy constraints
  - Some poses more likely than others
  - The likelihood of poses depends on the body shape and style of the individual person
- Data-driven IK
  - Drive the likelihood function from human motion data
  - data-driven-ik.avi

Summary

- Very simple structure allows an analytic solution
- Most of complex articulated figures requires a numerical solution
- May not always get the “right” answer
  - Need to tweak the solution later