Computer Animation

Lecture 2.

Basics of Character Animation

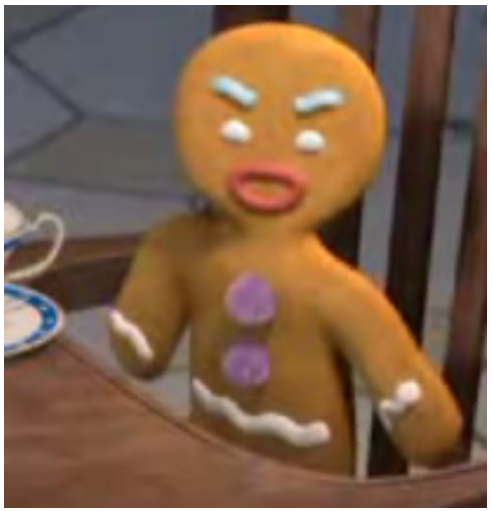
Taku Komura

Overview

- Character Animation
  - Posture representation
    - Hierarchical structure of the body
    - Joint types
      - Translational, hinge, universal, gimbal, free
    - Euler angles
      - Gimbal lock
    - Quaternions
- Creating the animation (Keyframe animation)
  - Interpolation
  - Inverse Kinematics
    - Analytical
    - CCD
    - Pseudo Inverse

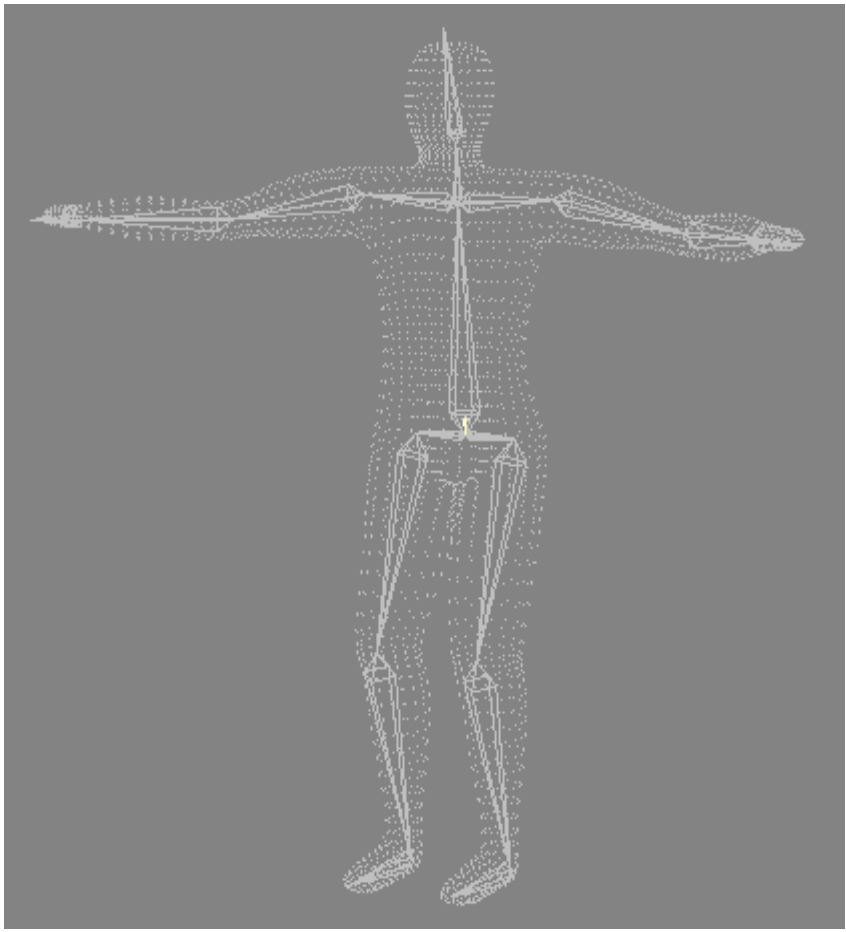
Characters include

- Human models
- Virtual characters
- Animal models



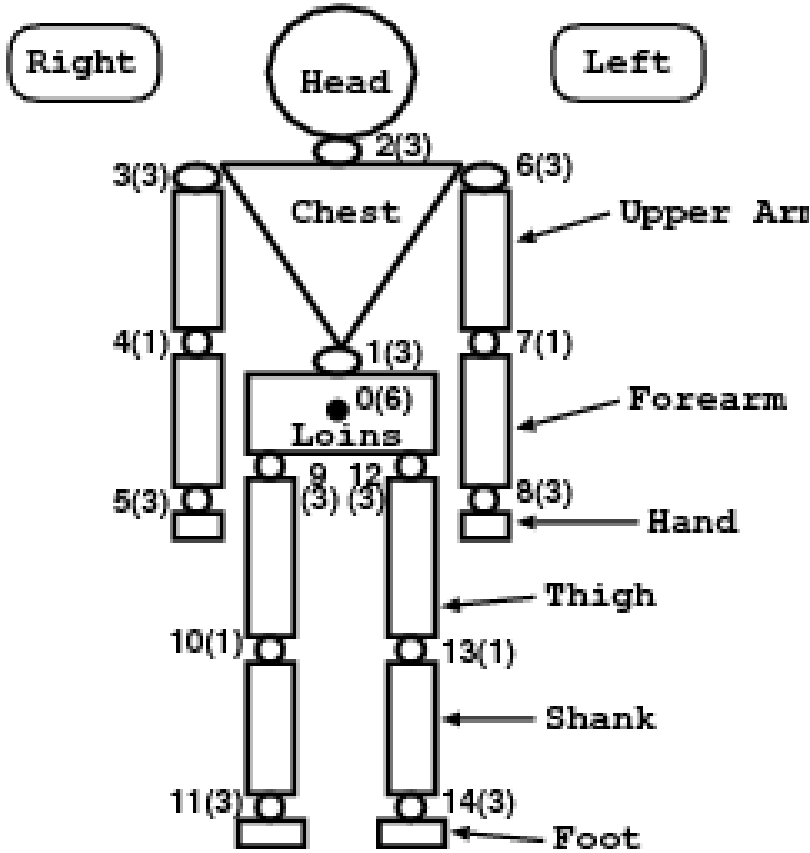
Controlling the skeleton

- We control the skeleton of characters
- The skin follows the movement of the skeleton



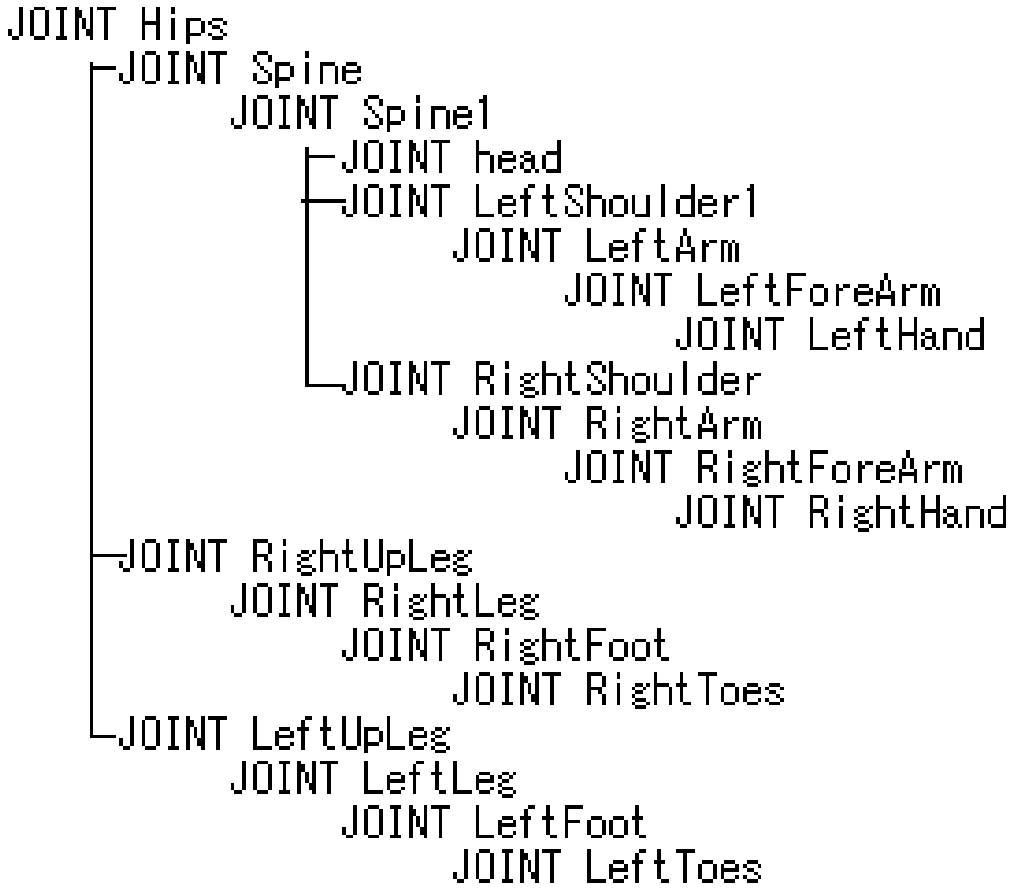
Representation of postures

- The body has a hierarchical structure
- Many types of joints



Hierarchical structure of the body

- The position of the joints lower in the hierarchy are affected by those above it
- Each joints can have 1 to 6 degrees of freedom
- For rotational joints, usually it is 1, 2, or 3
- The “Root” of the body has 3 degrees of freedom for the translation

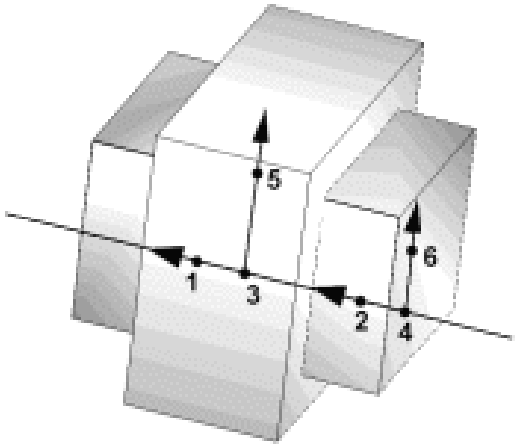


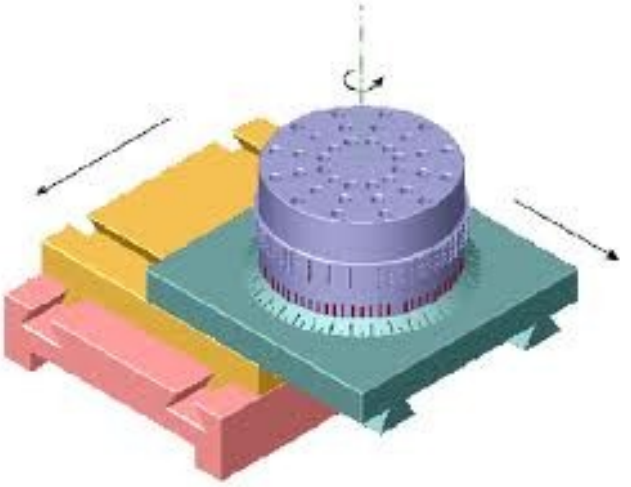
Joints

- The Degrees of Freedom (DOF) is defined for various joints
- There are several kinds of joints
  - Translational joint (1,2,3 DOF)
  - hinge joints (1 DOF)
  - Universal joint (2 DOF)
  - Gimbal joint (3 DOF)
  - Free joint (3DOF)

Translational joint

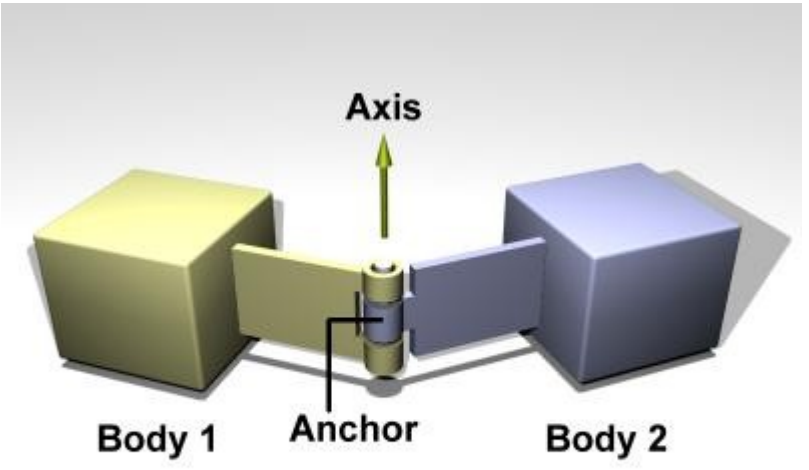
- A sliding joint
- Can be 1,2 or 3 DOF





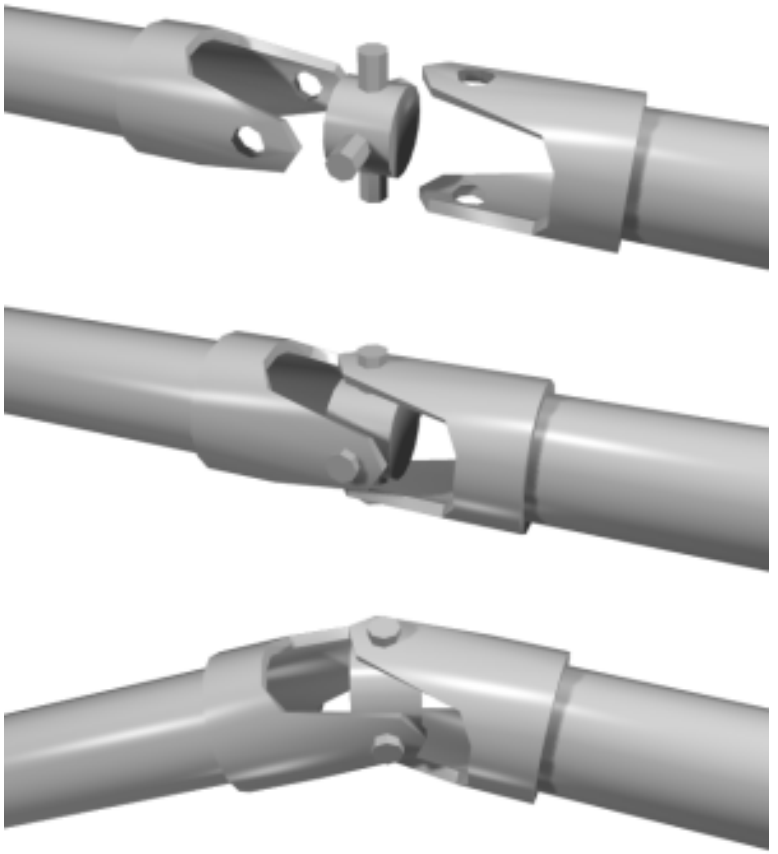
Hinge Joint

- A 1 DOF rotational joint
- Can be defined by the axis of rotation
- **Knee, elbow**



Universal Joint

- 2DOF
- Rotation around 2 axes perpendicular to each other
- **Wrist joint**

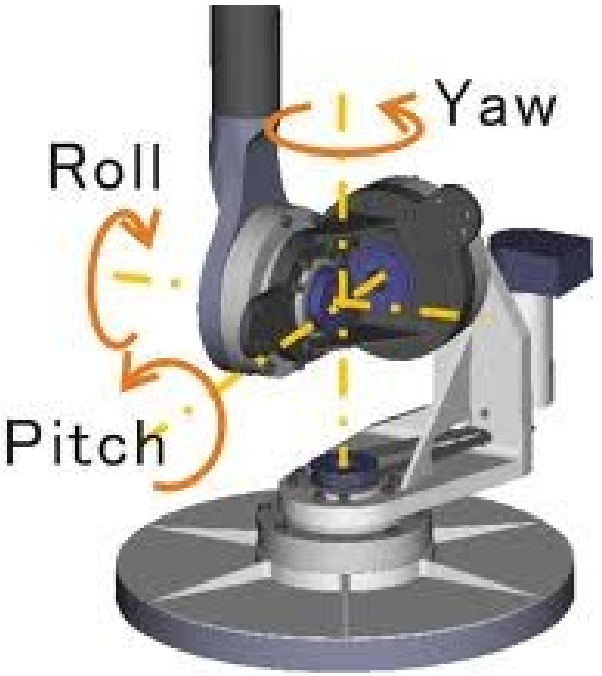


3DOF rotational joints

- Shoulder, hip, neck
- Two ways to represent the rotations
  - Gimbal joint (Euler Angles)
  - Free joint (Quaternions)

Gimbal joint : Euler Angles

- 3DOF joints
- Comes from Robotics
- 3DOF joints in robots were designed by connecting three motors pointing different axes

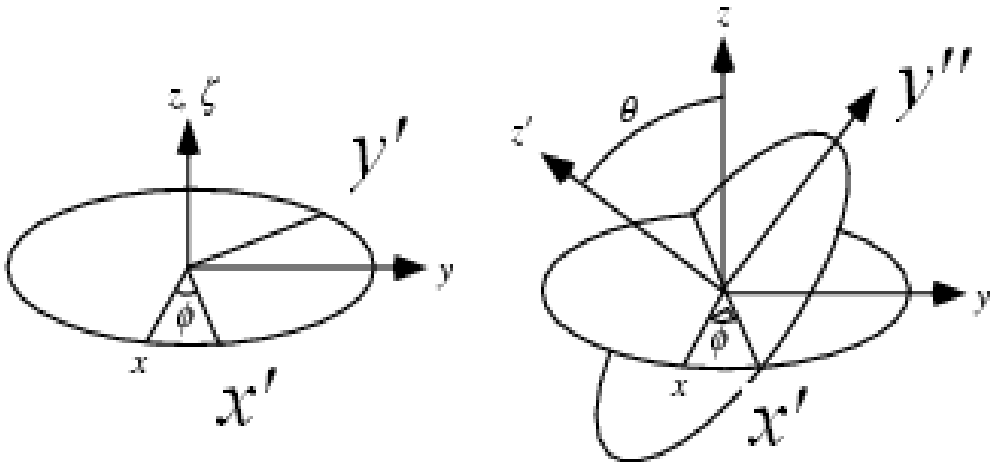


Gimbal joint : Euler Angles

- Rotation defined by the three axes and the angle of rotation around them
- the rotation order has to be specified such as

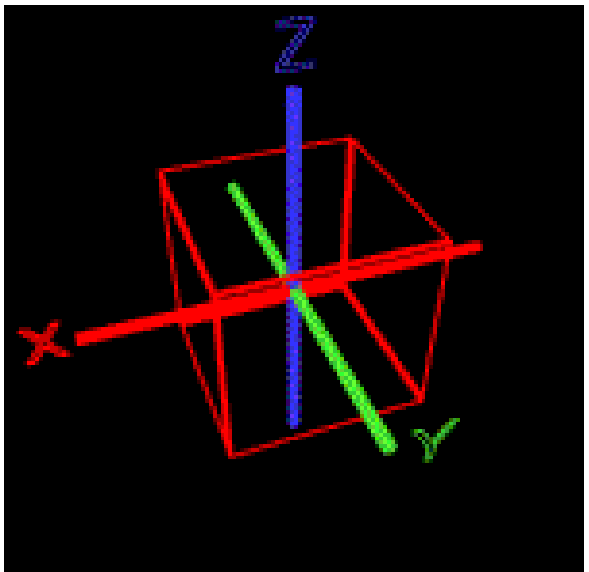
  X-Y-Z, Z-X-Y, Y-Z-X, etc

The one below is Z-X



Problem: Gimbal Lock

- Two rotational axis of an object pointing in the same direction - 1DOF is lost
  - For example for rotation defined in the order of X-Y-Z
  - Gimbal lock occurs when rotating Y for 90 degrees.
  - X and Z axis get pointed down the same axis



Free joint

- A ball joint
- 3DOF
- Do not have to worry about gimbal lock

Free joint : Quaternion

- Do not have to worry about gimbal lock
- The rotation is represented by a vector of four components (w, x, y, z)

Free joint : Quaternion

- Do not have to worry about gimbal lock
- The rotation is represented by a vector of four components (w, x, y, z)

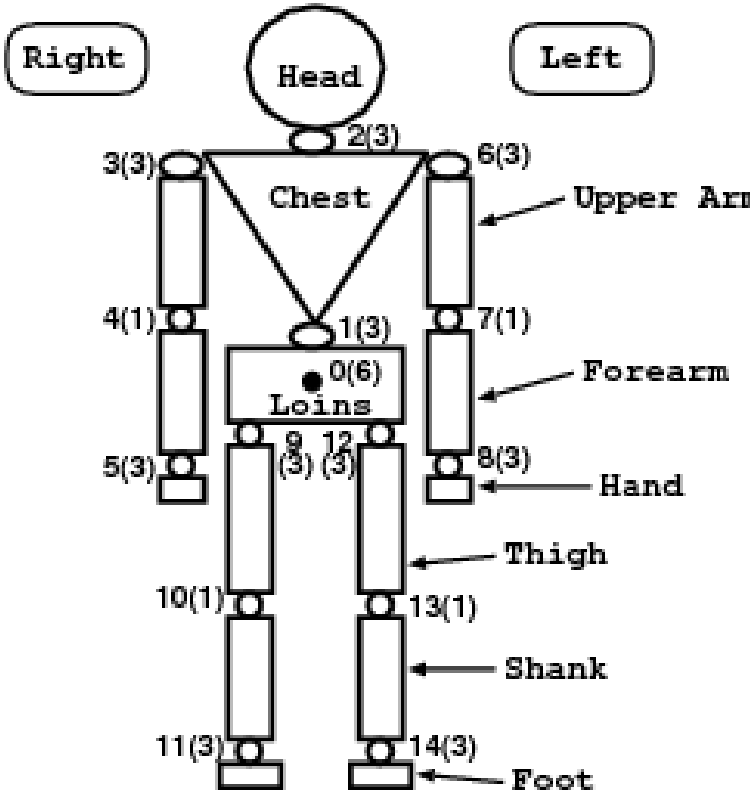
A rotation about the unit vector $\boldsymbol{u}$ by an angle makes a quaternion

$$(\cos(\theta/2),\ u_x \sin(\theta/2),\ u_y \sin(\theta/2),\ u_z \sin(\theta/2))$$

Animation of the whole body

I have explained about each joints

Now let me explain about how to define the posture of the whole body

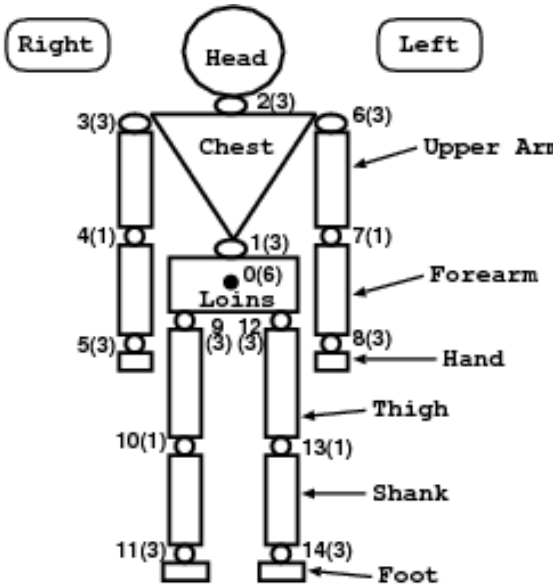


Generalized coordinates

A vector to specify the posture of the body

$$q = (q_1, q_2, q_3, q_4, q_5, q_6, q_7, \ldots, q_n)$$

Usually, the first three numbers : location of the root
The next three numbers : orientation of root
The rest: the joint angles of the body

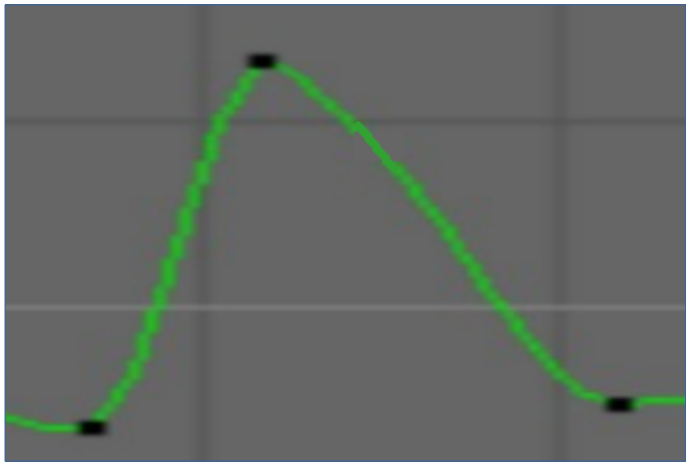


Overview

- Character Animation
  - Posture representation
    - Hierarchical structure of the body
    - Joint types
      - Translational, hinge, universal, gimbal, free
    - Euler angles
      - Gimbal lock
    - Quaternions
- **Creating the animation (Keyframe animation)**
  - Interpolation
  - Inverse Kinematics
    - Analytical
    - CCD
    - Pseudo Inverse

Keyframe Animation

- The keyframe postures are designed by the animator
- The inbetween motion is created by interpolation



Keyframe Design

- Each postures are created by the user interface
  - Forward Kinematics (FK, Virtual Track Ball)
    - The user clicks the segment and rotates it around the joint origin
    - The movement of the mouse is mapped to the rotation of the joint

https://www.youtube.com/watch?v=pGuR4xyi7MQ

  - Inverse Kinematics (IK)
    - The user clicks the segment and drags it in the 3D coordinate
    - The motion of the mouse is mapped to the translation in 3D coordinate
    - The movement of the segment is achieved by moving each joint of the body

http://www.youtube.com/watch?v=--5hWniftyI

Keyframe animation by Poser

- Poser is a commercial software to generate human animation
- There is another free software called MikuMikuDance
- Another Blender plugin called “Manuel Bastioni Labs”

Interpolation

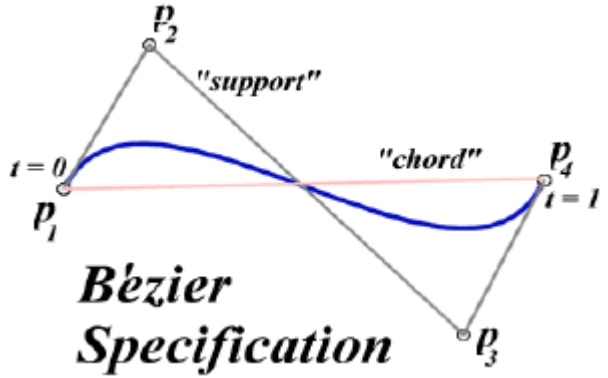
The generalized coordinates can be interpolated by

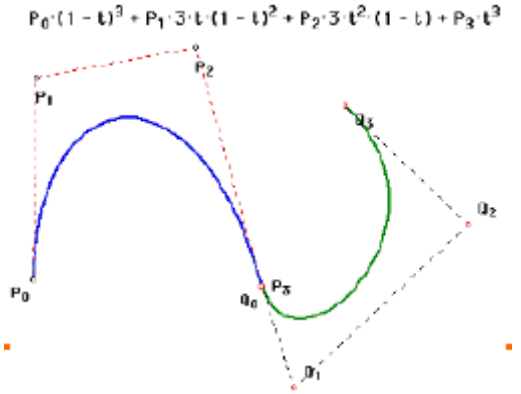
- Linear interpolation

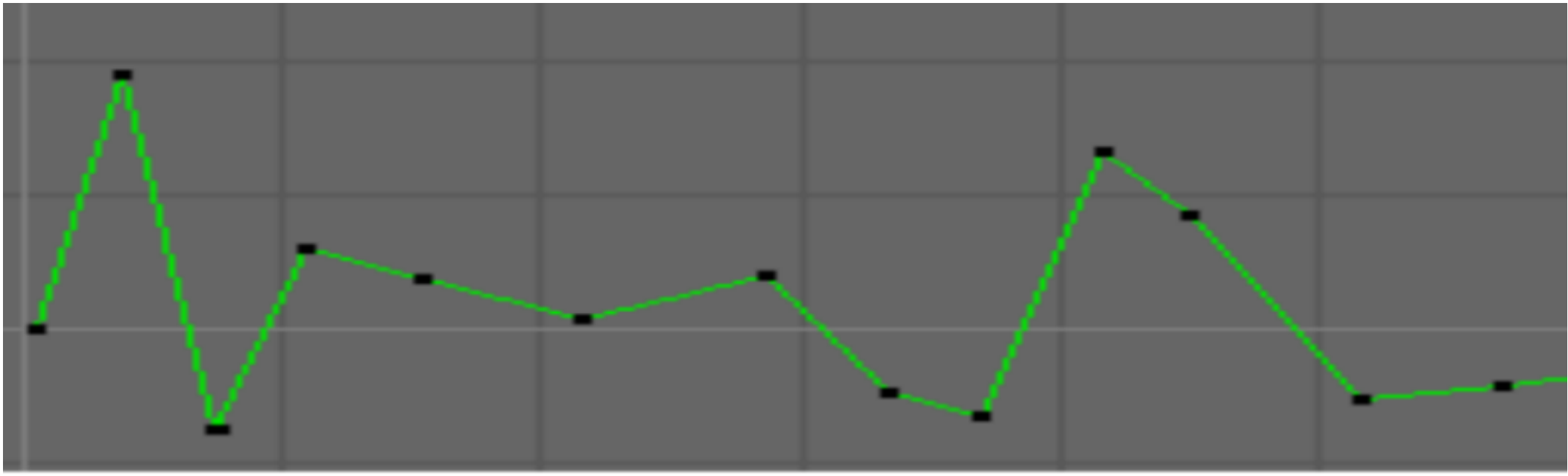
$$q(t) = (1-t)q^{i} + tq^{i+1}$$

- high-order polynomials (e.g. Bsplines, Bezier)

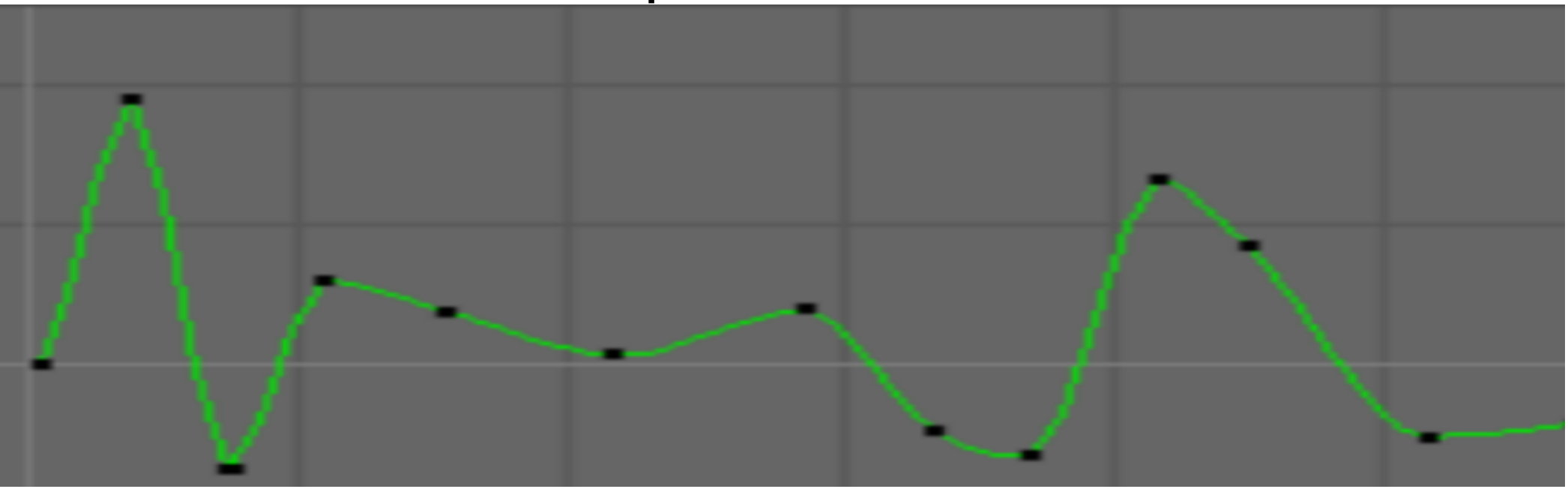
  Bezier : 2 end points, two points to control the tangent vector







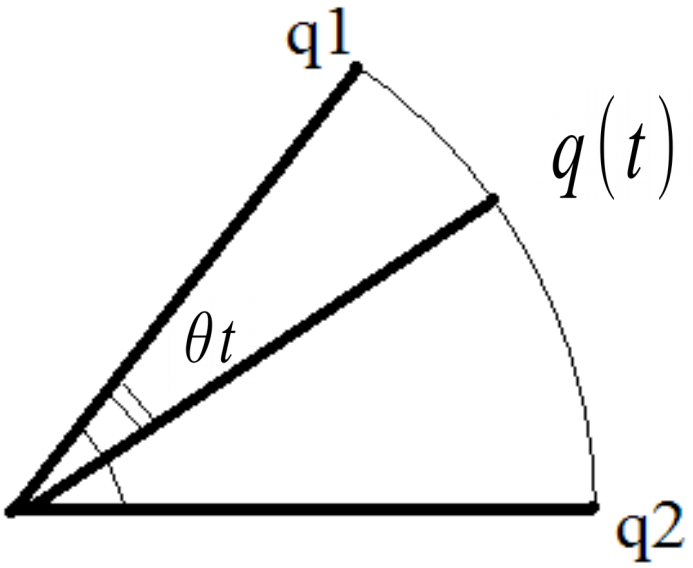
Linear interpolation



Bsplines

Interpolation of Quaternions

- Interpolation of two rotations (SLERP)
- Changing the orientation from q1 to q2 by rotating around a single axis ***u***
- θ angle of rotation around ***u*** to change from q1 to q2



$$\theta = \arccos\left(q_1 \cdot q_2\right)$$

$$q(t) = \frac{\sin\theta(1-t)}{\sin\theta} q_1 + \frac{\sin\theta t}{\sin\theta} q_2$$

Iyan 3D - Make your own Animation Movie on your iOS Device

https://youtu.be/VbQFrJGRCZw

Problems with interpolating the generalized coordinates

Problem:

- Important constraints might not be satisfied
  - The feet on the ground can slide

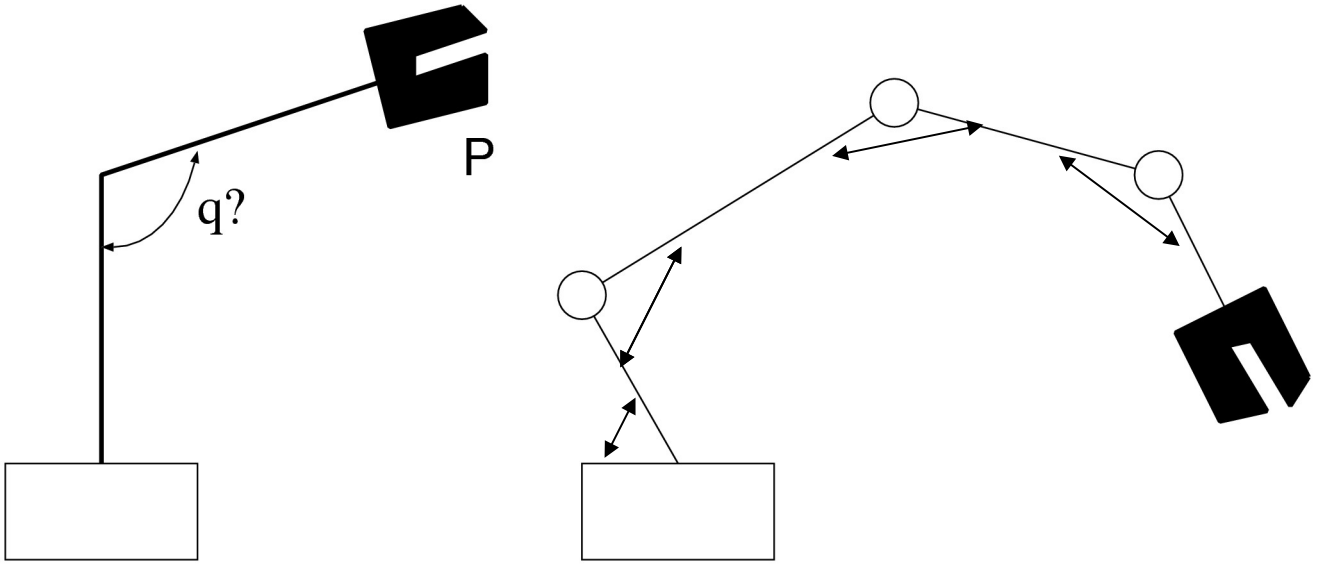
- Solution

  Specify the position of the joints and use inverse kinematics to calculate the joint angles

http://www.youtube.com/watch?v=--5hWniftyI

Inverse Kinematics Problem

- Calculate the joint angles from the position of the end effector



Inverse Kinematics

- Forward Kinematics: calculating the joint positions from the joint angles
- **Inverse Kinematics : Calculate the joint angles based on the joint positions**

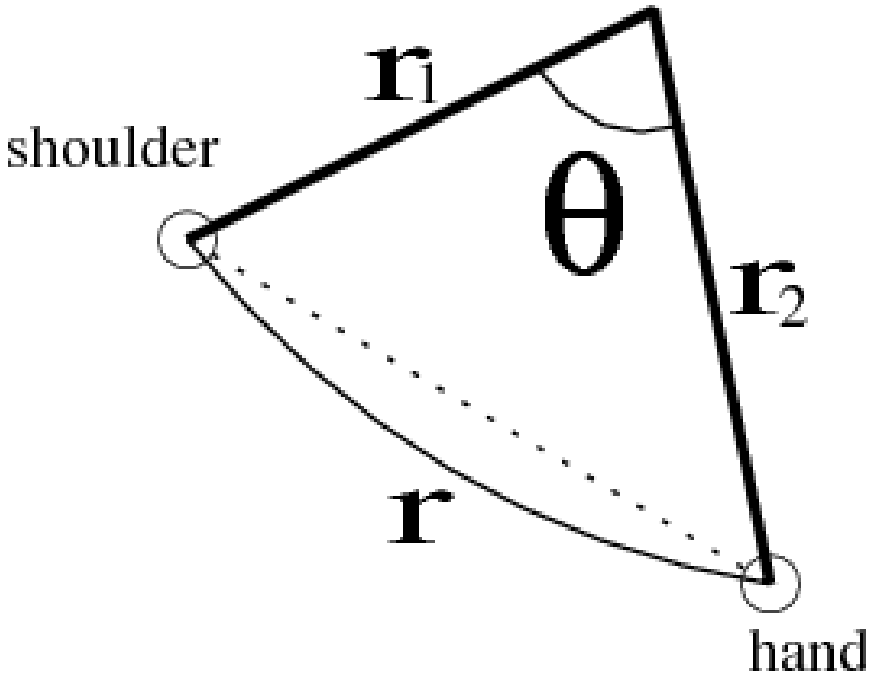
  - Many different approaches
    - Analytical approaches (analytical solution exists)
    - CCD (Cyclic Coordinate Descent)
    - Jacobian-based methods (compute by optimization)

  - Often used in robotics

Analytical Approaches

- Using an analytical solver for calculating the joint angles
- e.g. suppose the positions of the wrist and shoulder are given, calculate the elbow angle

$$\theta = \arccos \frac{r_1^2 + r_2^2 - r^2}{2 r_1 r_2}$$



Cyclic-Coordinate Descent

- Moving the joints closest to the end effectors first and minimize the distance between the end effector and the target
- Move up the hierarchy and move the next joint to minimize the distance between the end effector and the target
- Repeat the process until the base is reached
- Move to the first joint again and repeat the same process until the end effector reaches the target or max iterations reached
  - This is needed to handle cases that the target is not reachable

http://thewanderingtech.50webs.com/Flash/IK%20Comparison%20Application/IKCompare.html

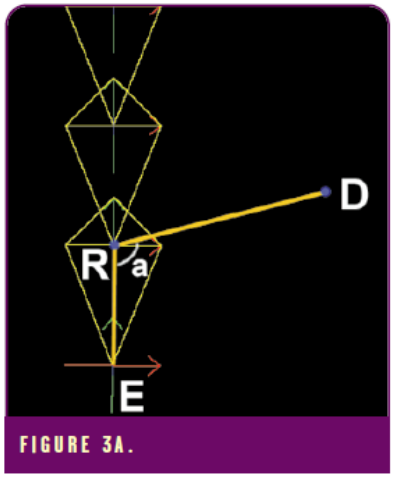


FIGURE 3A.

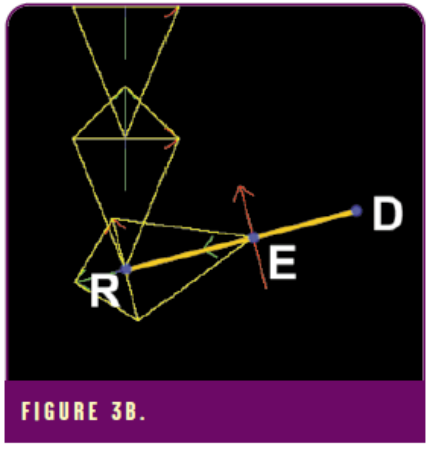


FIGURE 3B.

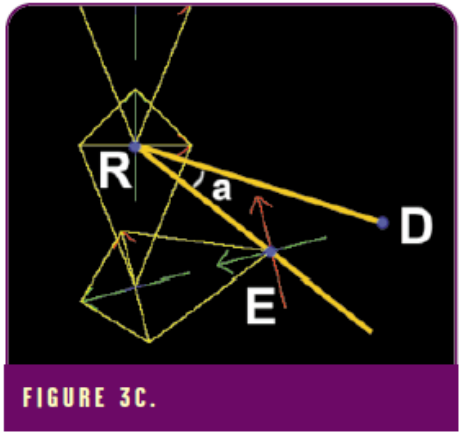


FIGURE 3C.

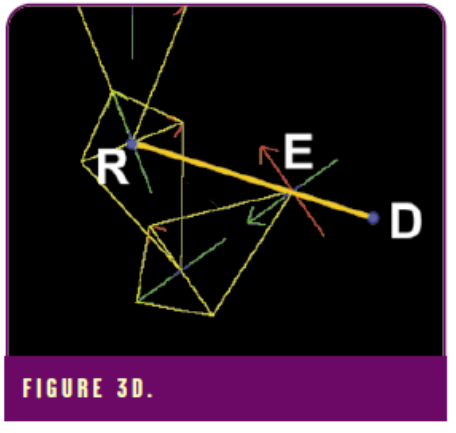


FIGURE 3D.

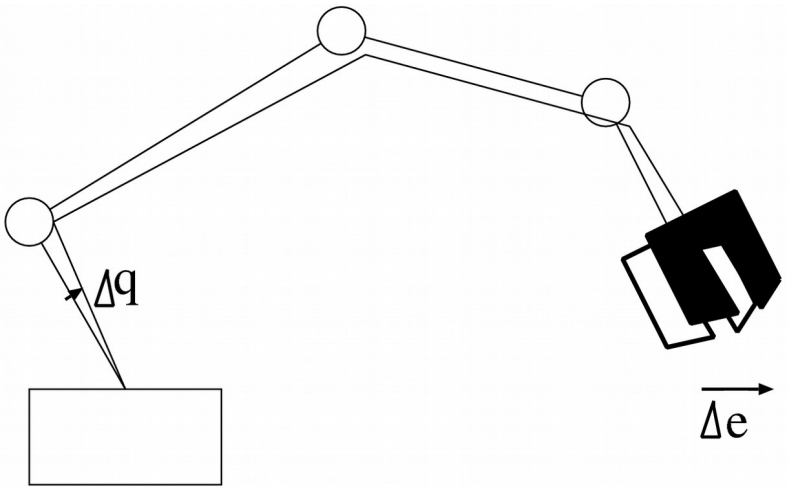
Pseudo-inverse method

- The relationship between the end effector position and the joint angles are non-linear

$$\mathbf{e} = f(\mathbf{q})$$

- But for small movements, their relation can be considered linear

$$\mathbf{\Delta e} = \mathbf{J}\,\mathbf{\Delta q}$$



Jacobian matrix

Correlates the movement of the end effector Δe with movements of the joints $\mathbf{\Delta q_i}$

Each column describing how much $\mathbf{\Delta e}$ changes when the $\mathbf{\Delta}$qi is changed

$$\Delta e = \mathbf{J}\Delta q$$

$$\text{Jacobian} \quad J = \frac{\partial e}{\partial q} \approx \begin{pmatrix} \frac{\Delta e_x}{\Delta q_1} & \cdots & \frac{\Delta e_x}{\Delta q_n} \\ \frac{\Delta e_y}{\Delta q_1} & \cdots & \frac{\Delta e_y}{\Delta q_n} \\ \frac{\Delta e_z}{\Delta q_1} & & \frac{\Delta e_z}{\Delta q_n} \end{pmatrix}$$

Pseudo-inverse method

- The relationship between the end effector position and the joint angles are non-linear

$$\mathbf{e} = f(\mathbf{q})$$

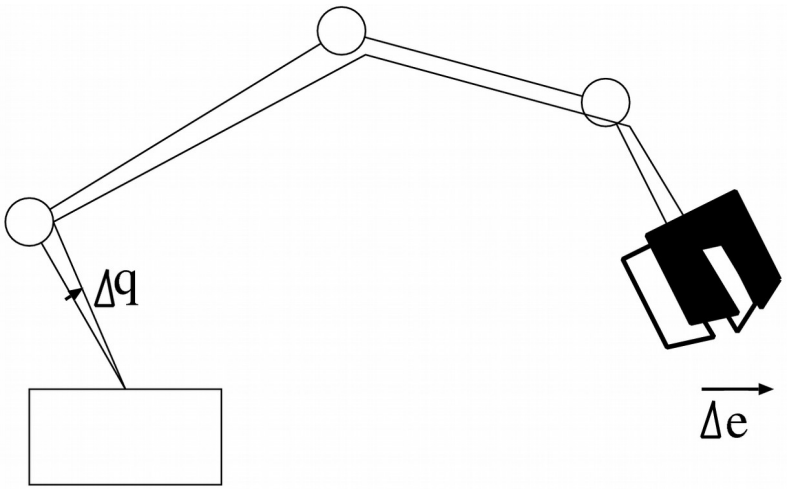
- But for small movements, their relation can be considered linear

$$\Delta\mathbf{e} = \mathbf{J}\,\Delta\mathbf{q}$$

This is Forward mapping

Inverse mapping?

$$\Delta\mathbf{e} \longrightarrow \Delta\mathbf{q}$$



Pseudo Inverse Matrix

- The minimal joint angle movements that achieves the end effector movement e can be computed by the pseudo inverse matrix

$$\text{pseudo inverse} \quad J^{+} = \left(J^{T} J \right)^{-1} J^{T}$$

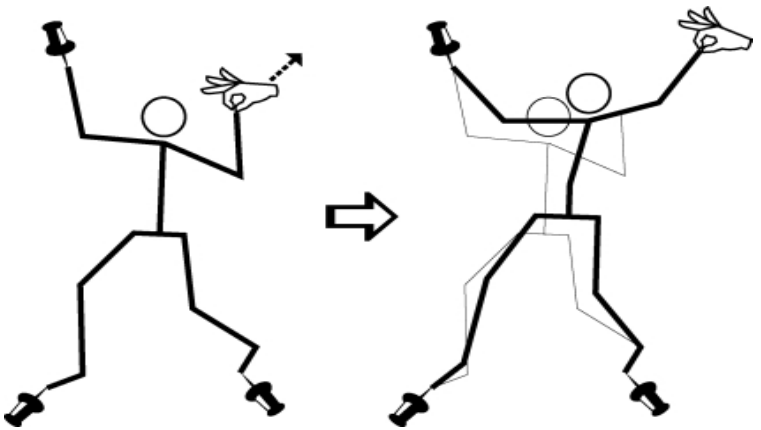
$$\Delta q = J^{+} \Delta e$$

Pseudo-inverse method

Iteratively updating the generalized coordinates so that the position constraints are satisfied

e: end effector position

g: target location



while (Δe is too far from **g**) {

compute the Jacobian matrix **J**

$$\Delta \mathbf{e} = \mathbf{J}\Delta \mathbf{q}$$

compute the pseudoinverse of the Jacobian matrix $\mathbf{J}^{+}$

compute change in joint DOFs: $\Delta \mathbf{q} = \mathbf{J}^{+}\Delta \mathbf{e}$

apply the change to DOFs, move a small step of

$$\mathbf{q} = \mathbf{q} + \Delta \mathbf{q}$$

}

Derivation

- Minimizing $g(\dot{q}) = \frac{1}{2}\dot{q}^T W \dot{q}$
- While satisfying $v_e = J\dot{q}$
- -> Minimize

$$g(\dot{q}, \lambda) = \frac{1}{2}\dot{q}^T W \dot{q} + \lambda^T (v_e - J\dot{q}),$$

Lagrange multipliers

Derivation

- Can solve by $\left(\frac{\partial g}{\partial \dot{\boldsymbol{q}}}\right)^T = \mathbf{0}$ $\left(\frac{\partial g}{\partial \boldsymbol{\lambda}}\right)^T = \mathbf{0}.$

$$\boldsymbol{W}\dot{\boldsymbol{q}} - \boldsymbol{J}^T\boldsymbol{\lambda} = \mathbf{0}$$

$$\dot{q} = W^{-1}J^T\lambda \longrightarrow v_e = J\dot{q}$$

$$v_e = JW^{-1}J^T\lambda;$$

$$\boldsymbol{W}\dot{\boldsymbol{q}} - \boldsymbol{J}^T\boldsymbol{\lambda} = \mathbf{0} \leftarrow \boldsymbol{\lambda} = (\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{v}_e$$

$$\dot{q} = W^{-1}J^T(JW^{-1}J^T)^{-1}v_e.$$

If W=I, we get J+

Singularity Problem

- There can be postures that the end effector cannot be moved to some directions
  - i.e. When all the joints are fully extended
- The system becomes unstable

http://thewanderingtech.50webs.com/Flash/IK%20Comparison%20Application/IKCompare.html

- Solution: Damped least squares

(imposing soft constraints)

$$J^{\star} = J^{T}(JJ^{T} + k^{2}I)^{-1}$$

Damped Least Squares

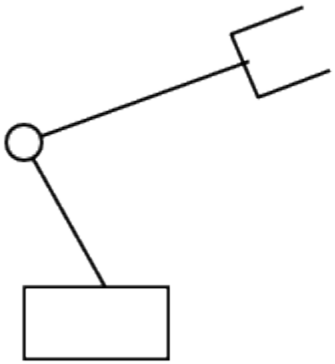
Instead of minimizing

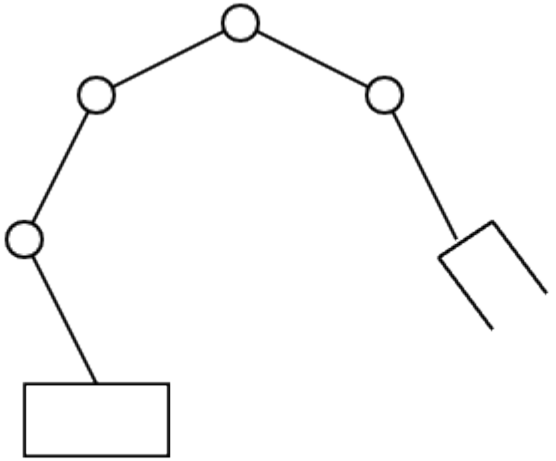
$$g(\dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}} + \boldsymbol{\lambda}^T(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}}),$$

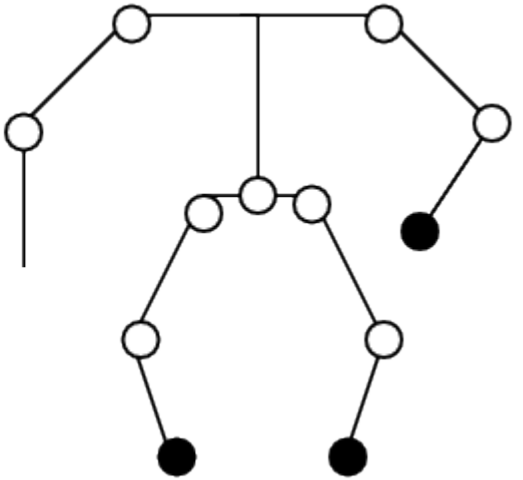
minimize

$$g''(\dot{\boldsymbol{q}}) = \frac{1}{2}k^2\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}} + \frac{1}{2}(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}})^T(\boldsymbol{v}_e - \boldsymbol{J}\dot{\boldsymbol{q}})$$

Which method will be good for controlling each of the below?







Cons and Pros of IK

- Analytical : Fast, but for most cases have no analytical solution
- CCD : simple, fast, easy to implement, can take into account the limit of the joint angles, may have oscillation problems
- Pseudo Inverse:
  - Used in robotics often, can handle any topological structure, multiple constraints
  - Can incorporate physics
  - Singularity problems (unstable when the limb is fully extended) -> damped least squares

Summary

- Representation of the Posture
  - Euler angles, generalized coordinates, quaternions
- Character Animation by Interpolation
- Inverse Kinematics
  - Analytical
  - CCD
  - Pseudo inverse

Links, Readings

- Poser http://poser.smithmicro.com/poser.html
- Manuel Bastioni Labs http://www.manuelbastioni.com/manuellab.php
- MikuMikuDance
  - http://www.geocities.jp/higuchuu4/index_e.htm
- About CCD http://graphics.cs.cmu.edu/nsp/course/15-464/Fall09/assignments/asst2/jlander_gamedev_nov98.pdf
- IK introduction
  - http://www.math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf
- A robotics textbook (can be downloaded from within the university, see chapter 3)
  - http://link.springer.com/content/pdf/10.1007%2F978-1-84628-642-1.pdf