Animation & Simulation

He Wang (王鹤)

Interpolation-Based Animation

- 3D shape interpolation
  - Matching topology
  - Star-shaped polyhedral
  - Axial slices
  - Advanced 2D interpolation
  - Advanced 3D interpolation
  - Map to sphere

Interpolation-Based Animation

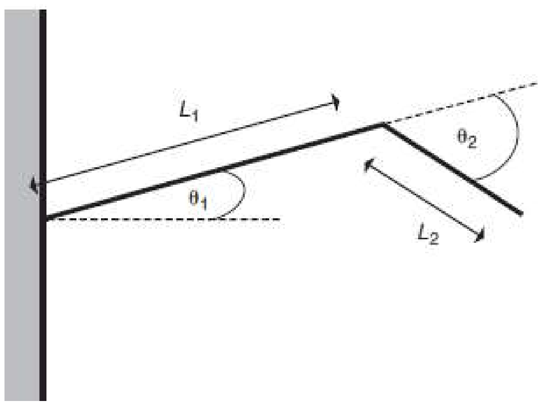
- 2D morphing
  - Morphing one image to another
  - Coordinate grid
  - Feature-based
- Forward Kinematics
  - Hierarchical Modelling
  - Data structure

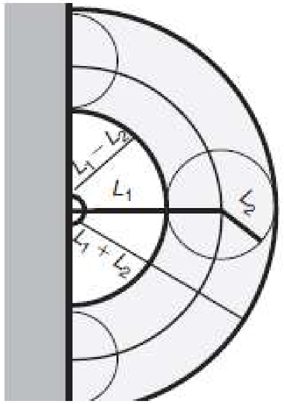
Kinematic Linkages

- Inverse Kinematics
  - Desired position (maybe orientation too) of end-effectors are given
  - Compute all joint angles
  - There might be 0, 1 or many solutions

Kinematic Linkages

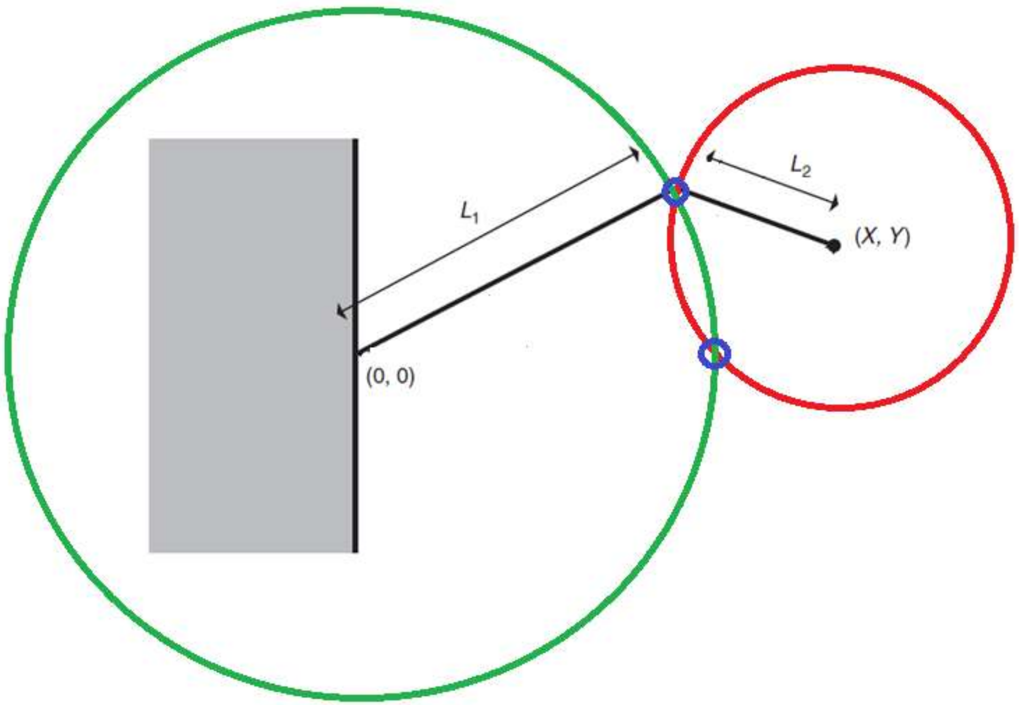
- Inverse Kinematics
  - Analytical solution (simple mechanisms)





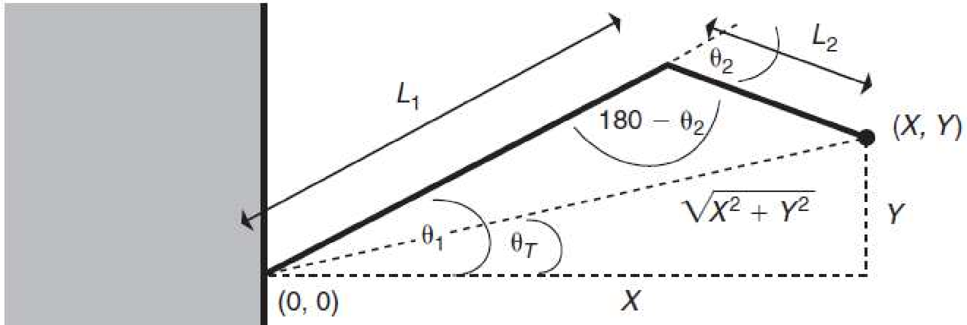
Kinematic Linkages

- Inverse Kinematics
  - Analytical solution (simple mechanisms)



Kinematic Linkages

- Inverse Kinematics
  - Analytical solution (simple mechanisms)



$$\mathrm{acos}(\theta_T) = \frac{X}{\sqrt{X^2 + Y^2}}$$

$$\theta_T = \mathrm{acos}\left(\frac{X}{\sqrt{X^2 + Y^2}}\right)$$

$$\cos(\theta_1 - \theta_T) = \frac{L_1^2 + X^2 + Y^2 - L_2^2}{2L_1\sqrt{X^2 + Y^2}} \qquad \text{(cosine rule)}$$

$$\theta_1 = \mathrm{acos}\left(\frac{L_1^2 + X^2 + Y^2 - L_2^2}{2L_1\sqrt{X^2 + Y^2}}\right) + \theta_T$$

$$\cos(180 - \theta_2) = -\cos(\theta_2) = \frac{L_1^2 + L_2^2 - (X^2 + Y^2)}{2L_1L_2} \qquad \text{(cosine rule)}$$

$$\theta_2 = \mathrm{acos}\left(\frac{L_1^2 + L_2^2 - X^2 + Y^2}{2L_1L_2}\right)$$

Kinematic Linkages

- Inverse Kinematics
    - Many mechanism are too complex for analytic solutions
    - Numerical solutions
        - Motions are incrementally constructed
        - At each time step, joint angles are adjusted to move the end-effector toward the desired location
        - Many ways, but concern the partial derivatives of joint angles, Jacobian

Kinematic Linkages

- Inverse Kinematics
  - Jacobian

$$
\begin{aligned}
y_1 &= f_1(x_1, x_2, x_3, x_4, x_5, x_6) \\
y_2 &= f_2(x_1, x_2, x_3, x_4, x_5, x_6) \\
y_3 &= f_3(x_1, x_2, x_3, x_4, x_5, x_6) \\
y_4 &= f_4(x_1, x_2, x_3, x_4, x_5, x_6) \\
y_5 &= f_5(x_1, x_2, x_3, x_4, x_5, x_6) \\
y_6 &= f_6(x_1, x_2, x_3, x_4, x_5, x_6)
\end{aligned}
$$

$$dy_i = \frac{\partial f_i}{\partial x_1} dx_1 + \frac{\partial f_i}{\partial x_2} dx_2 + \frac{\partial f_i}{\partial x_3} dx_3 + \frac{\partial f_i}{\partial x_4} dx_4 + \frac{\partial f_i}{\partial x_5} dx_5 + \frac{\partial f_i}{\partial x_6} dx_6$$

$$dY = \frac{\partial F}{\partial X} dX$$

Kinematic Linkages

- Inverse Kinematics
  - Jacobian

$$\mathbf{J} = \left[\frac{\partial \mathbf{f}}{\partial x_1} \quad \cdots \quad \frac{\partial \mathbf{f}}{\partial x_n} \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$\dot{Y} = J(X)\dot{X}$$

Kinematic Linkages

- Inverse Kinematics
  - Jacobian
    - Say Y is the position and orientation of the end-effector

$$Y = [p_x \ p_y \ p_z \ \alpha_x \ \alpha_y \ \alpha_z]^T$$

then $V = \dot{Y} = J(\theta)\dot{\theta}$ Jacobian relates the joint angle velocities to the end-effector's Linear and rotational velocities

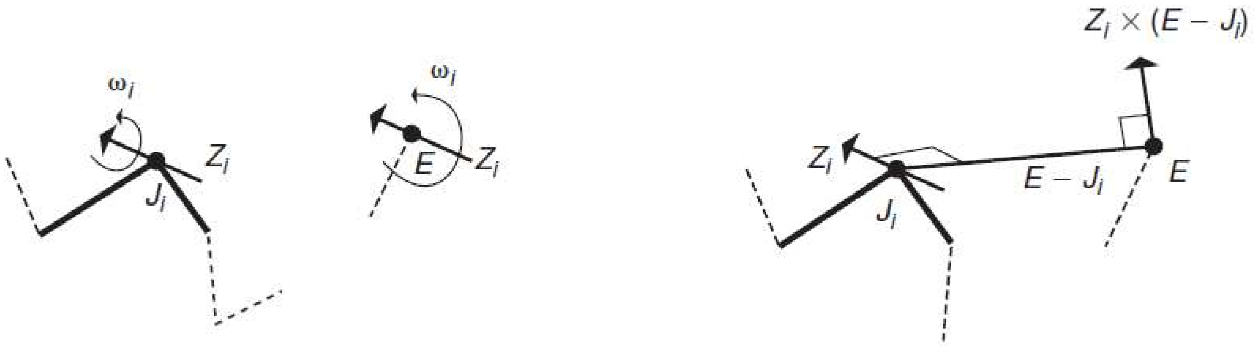
$$V = [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^T$$

$$\dot{\theta} = \left[\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3 \ \ldots \ \dot{\theta}_n\right]^T$$

$$J = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} & \cdots & \frac{\partial p_x}{\partial \theta_n} \\ \frac{\partial p_y}{\partial \theta_1} & \frac{\partial p_y}{\partial \theta_2} & \cdots & \frac{\partial p_y}{\partial \theta_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \alpha_z}{\partial \theta_1} & \frac{\partial \alpha_z}{\partial \theta_2} & \cdots & \frac{\partial \alpha_z}{\partial \theta_n} \end{bmatrix}$$

Kinematic Linkages

- Inverse Kinematics
  - Jacobian
    - The desired angular and linear velocities are computed by finding the difference between the current configuration of the end effector and the desired configuration



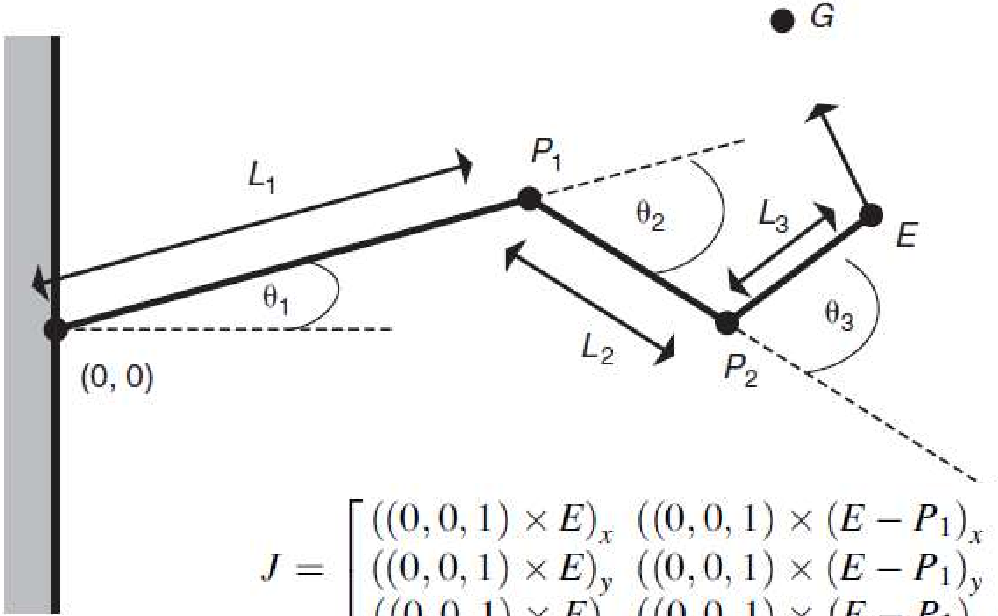
a) Angular velocity, ω_i

b) Linear velocity, $Z_i \times (E - J_i)$

E — end effector
J_i — i th joint
Z_i — i th joint axis
ω_i — angular velocity of i th joint

Kinematic Linkages

- Inverse Kinematics
  - A simple problem



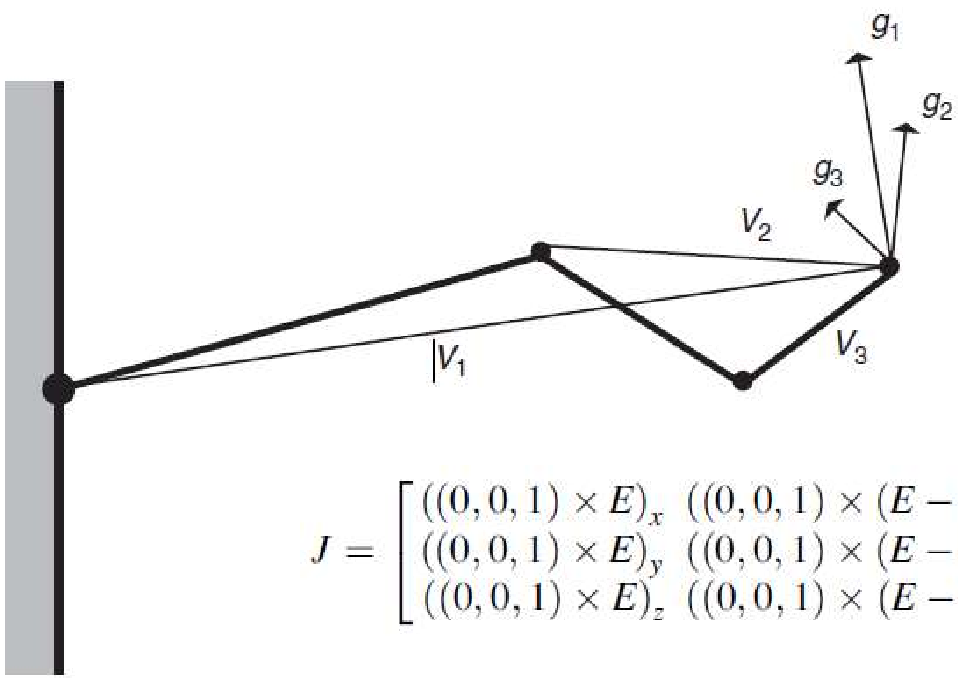
$$V = \begin{bmatrix} (G-E)_x \\ (G-E)_y \\ (G-E)_z \end{bmatrix}$$

$$J = \begin{bmatrix} ((0,0,1)\times E)_x & ((0,0,1)\times (E-P_1)_x & ((0,0,1)\times (E-P_2)_x \\ ((0,0,1)\times E)_y & ((0,0,1)\times (E-P_1)_y & ((0,0,1)\times (E-P_2)_y \\ ((0,0,1)\times E)_z & ((0,0,1)\times (E-P_1)_z & ((0,0,1)\times (E-P_2)_z \end{bmatrix}$$

Kinematic Linkages

- Inverse Kinematics
  - A simple problem

$$V = \begin{bmatrix} (G-E)_x \\ (G-E)_y \\ (G-E)_z \end{bmatrix}$$



$$V = J\dot{\theta}$$

$$J^{-1}V = \dot{\theta}$$ This is what we want

$$J = \begin{bmatrix} ((0,0,1) \times E)_x & ((0,0,1) \times (E - P_1)_x & ((0,0,1) \times (E - P_2)_x \\ ((0,0,1) \times E)_y & ((0,0,1) \times (E - P_1)_y & ((0,0,1) \times (E - P_2)_y \\ ((0,0,1) \times E)_z & ((0,0,1) \times (E - P_1)_z & ((0,0,1) \times (E - P_2)_z \end{bmatrix}$$

Kinematic Linkages

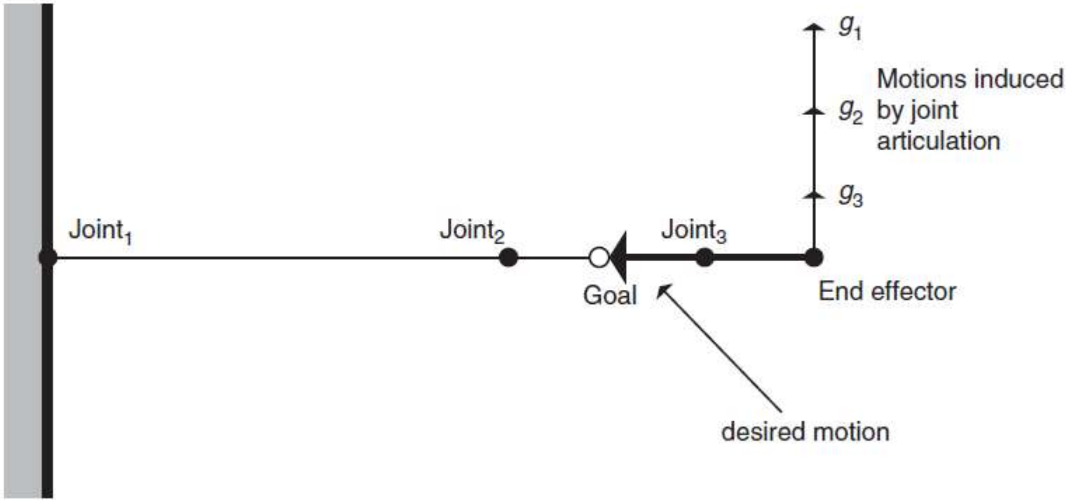
- Inverse Kinematics
  - Numerical solution
    - What if J^{-1} does not exist? (what matrix is not inversible?)

$$V = J\dot{\theta}$$

$$J^{-1}V = \dot{\theta}$$

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - What if J^{-1} does not exist? (what matrix is not inversible?)
    - When J is singular



Even adding small perturbation will not work well because of large $\dot{\theta}$

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Things become better when redundancy exists (more DoF then constraints)
    - J is not square thus there are infinite number of solutions
    - If J is column independent, use pseudoinverse $(J^T J)^{-1} J^T$

$$
\begin{aligned}
V &= J\dot{\theta} \\
J^T V &= J^T J\dot{\theta} \\
(J^T J)^{-1} J^T V &= (J^T J)^{-1} J^T J\dot{\theta} \\
J^+ V &= \dot{\theta}
\end{aligned}
$$

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Things become better when redundancy exists (more DoF then constraints)
    - J is not square thus there are infinite number of solutions
    - If J is column independent, use $(J^T J)^{-1} J^T$
    - If J is row independent, use pseudoinverse $J^+ = J^T (J J^T)^{-1}$

$$J^+ V = \dot{\theta}$$
$$J^T (J J^T)^{-1} V = \dot{\theta}$$

$$\beta = (J J^T)^{-1} V$$
$$(J J^T)\beta = V \leftarrow \text{LU decomposition}$$

$$J^T \beta = \dot{\theta}$$

Kinematic Linkages

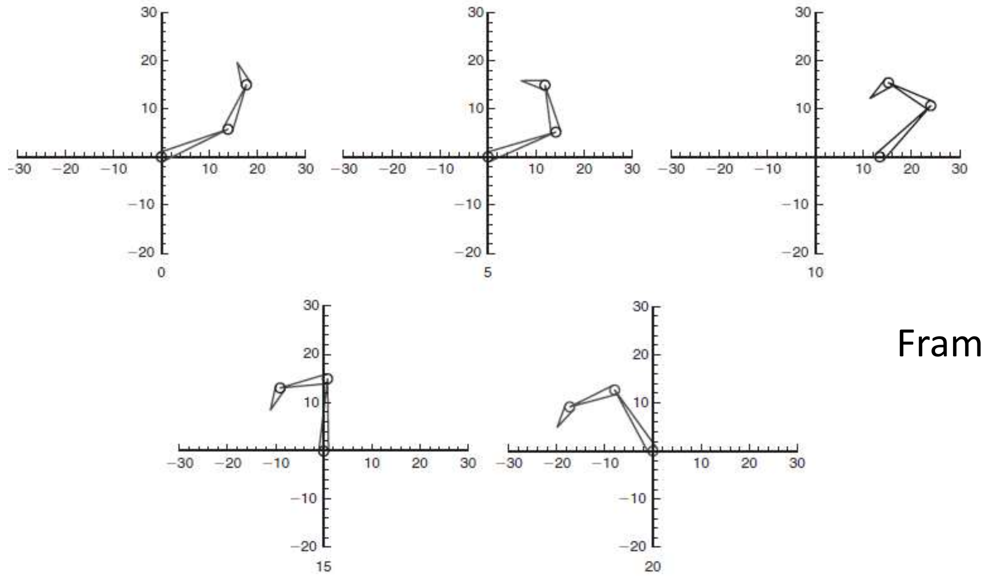
- Inverse Kinematics
  - Numerical solution
    - LU Decomposition

Ax = b
Let A = LU, LU decomposition
Then LUx = b
Let Lc = b so Ux = c
Solve Lc = b
Then solve Ux = c

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - J only works for the current instantaneous configuration
    - So a series of small steps are needed



Frame 0, 5, 10, 15, 20

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Damped IK

$$\dot{\theta} = J^T(JJ^T + \lambda^2 I)^{-1}V$$

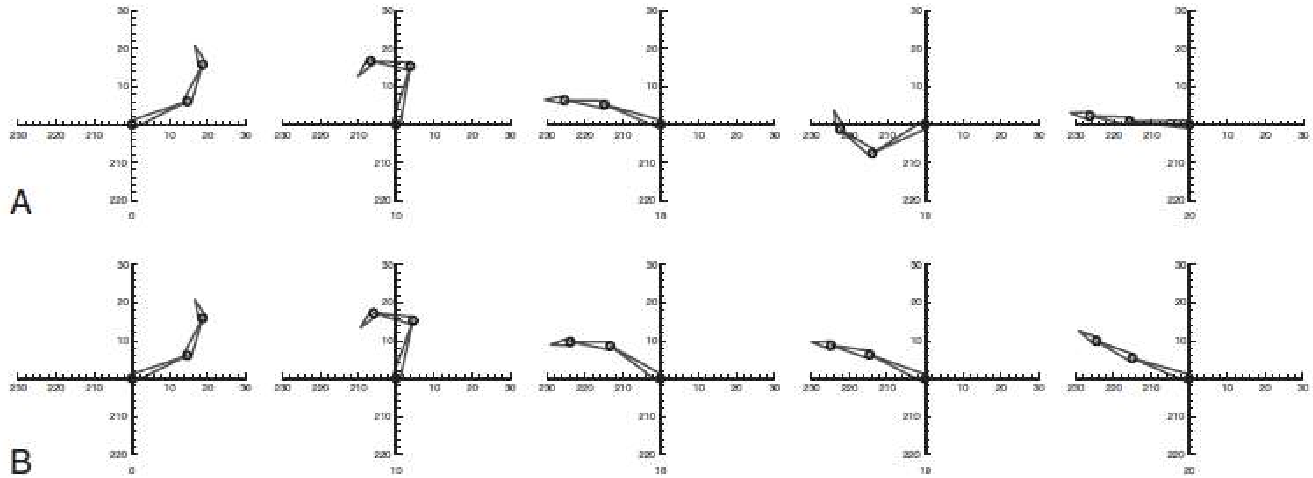
When J is nearly Singular (when does it happen?)
How does adding a (scaled) Identify Matrix help?

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Damped IK

$$\dot{\theta} = J^T (JJ^T + \lambda^2 I)^{-1} V$$

Damped IK, user provides lambda



Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Adding control

$$\dot{\theta} = (J^{+}J - I)z$$

Introducing z does not change the angular velocity

$$V = J\dot{\theta}$$
$$V = J(J^{+}J - I)z$$
$$V = (JJ^{+}J - I)z$$
$$V = 0z$$
$$V = 0$$

gain

$$z = \alpha_i(\theta_i - \theta_{ci})^2$$

current desired

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Adding control

$$\dot{\theta} = (J^{+}J - I)z$$

gain

$$z = \alpha_i(\theta_i - \theta_{ci})^2$$

current desired

$$\dot{\theta} = J^{+}V + (J^{+}J - I)z$$

$$\dot{\theta} = J^{+}V + J^{+}Jz - Iz$$

$$\dot{\theta} = J^{+}(V + Jz) - z$$

$$\dot{\theta} = J^{T}(JJ^{T})^{-1}(V + Jz) - z$$

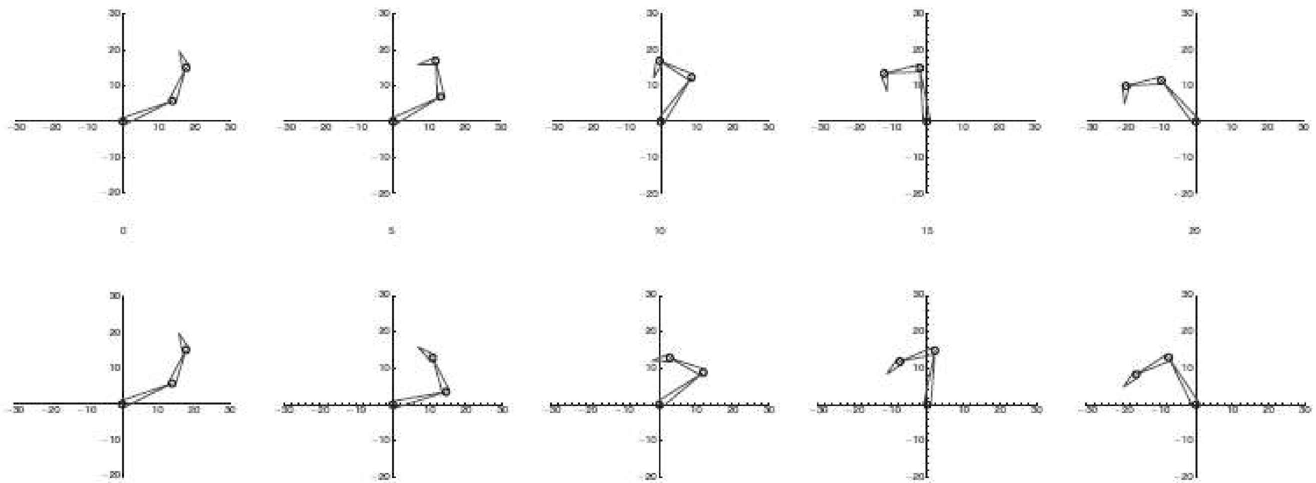
$$\dot{\theta} = J^{T}\left[(JJ^{T})^{-1}(V + Jz)\right] - z$$

$$\dot{\theta} = J^{T}\beta - z$$

$$V + Jz = (JJ^{T})\beta$$

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Adding control $\dot{\theta} = (J^{+}J - I)z$

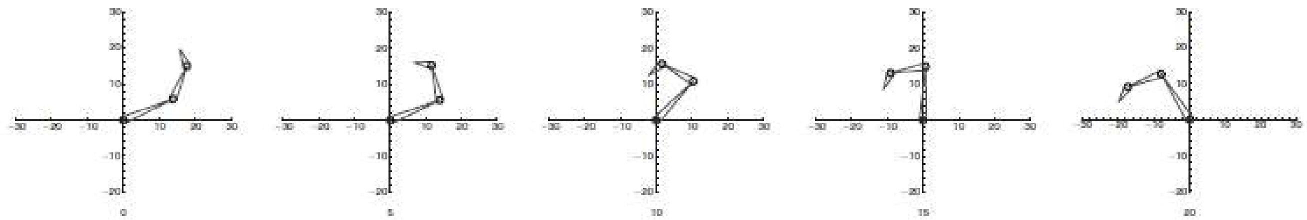


[0.1, 0.5, 0.1]

[0.1, 0.1, 0.5]

Kinematic Linkages

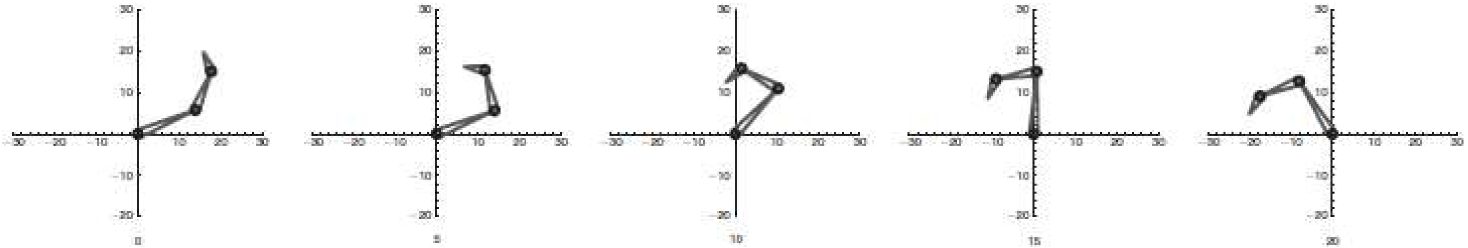
- Inverse Kinematics
  - Numerical solution
    - Alternative Jacobian
      - Pull the goal towards EF instead
      - Using goal position instead of the EF position in the pseudoinverse



$$V = \begin{bmatrix} (G-E)_x \\ (G-E)_y \\ (G-E)_z \end{bmatrix}$$

Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Use transpose instead of inverse $\dot{\theta} = \alpha J^T V$
      - Some small-weight instantaneous change vector can still drag the EF away
      - One more parameter to tune



Kinematic Linkages

- Inverse Kinematics
  - Numerical solution
    - Procedural method: cyclic coordinate descent
      - Process each joint a time
      - Rank joints by importance/contribution, go for the most important one