Inverse Kinematics

Overview

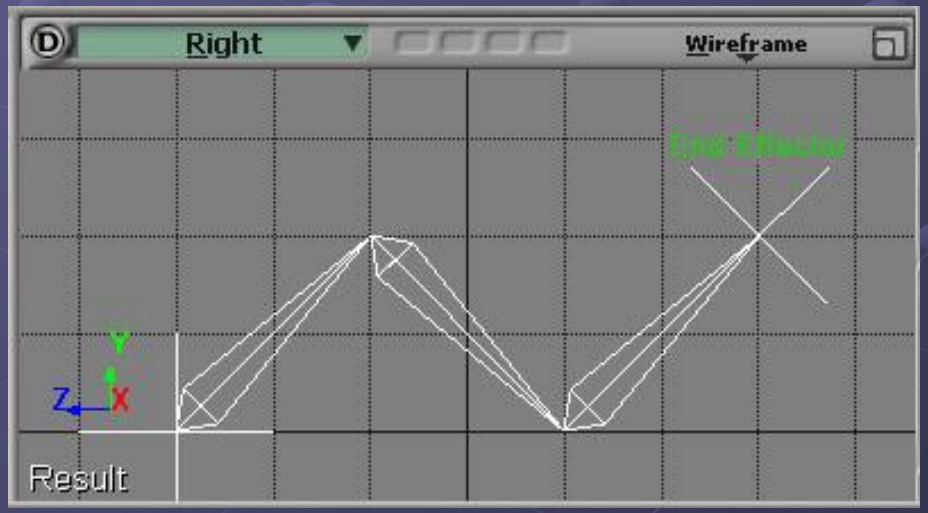
- What is IK?
- Important concepts in IK
- The role of IK in game development

What is IK?

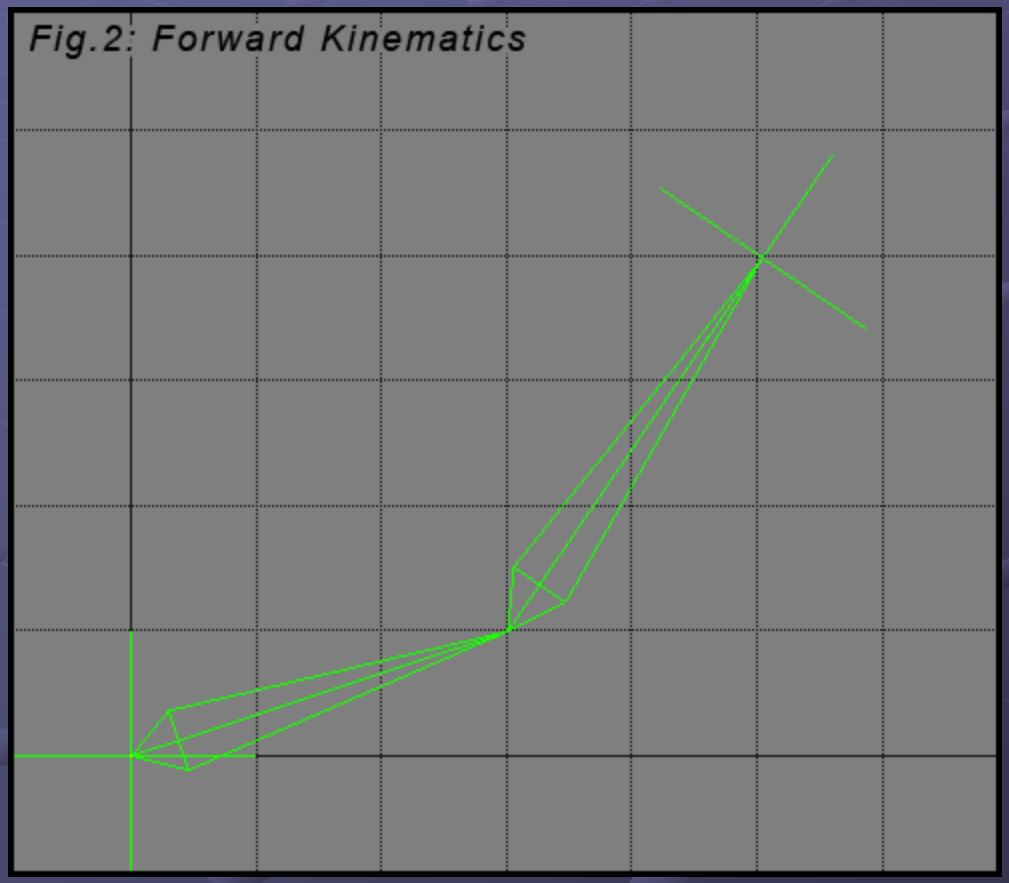
Forward Kinematics (FK)

Incrementally manipulating each of the component parts of a flexible, jointed object to achieve an overall, desired pose.

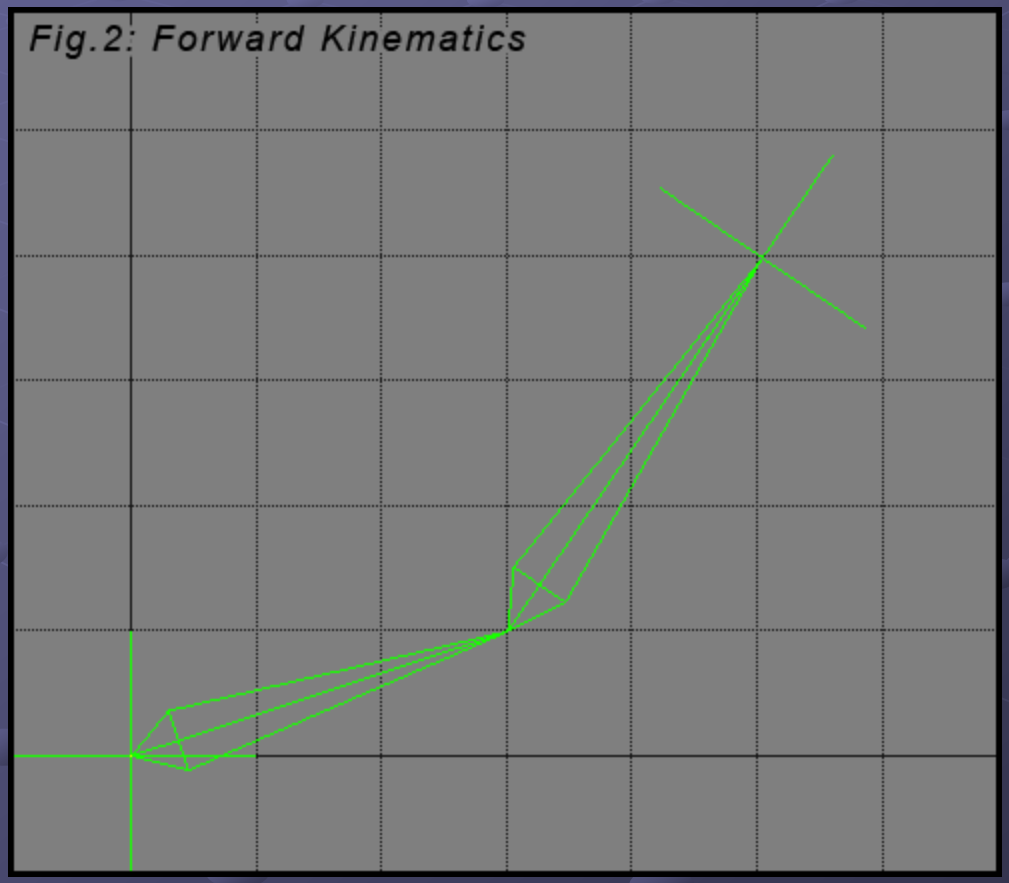
Mathematically is concerned with finding the position of the end effector, given the angle of the joints and the length of each articulated segment.



FK Equation



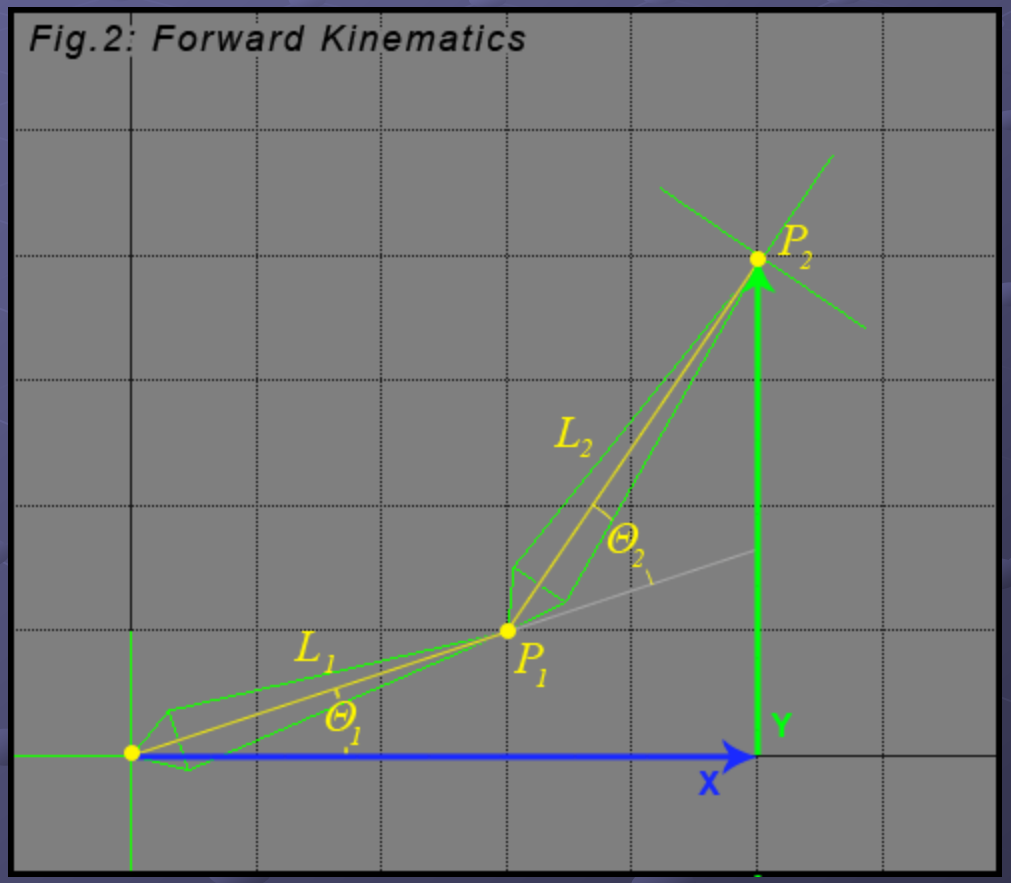
FK Equation



FK Equation

Expression for P1:

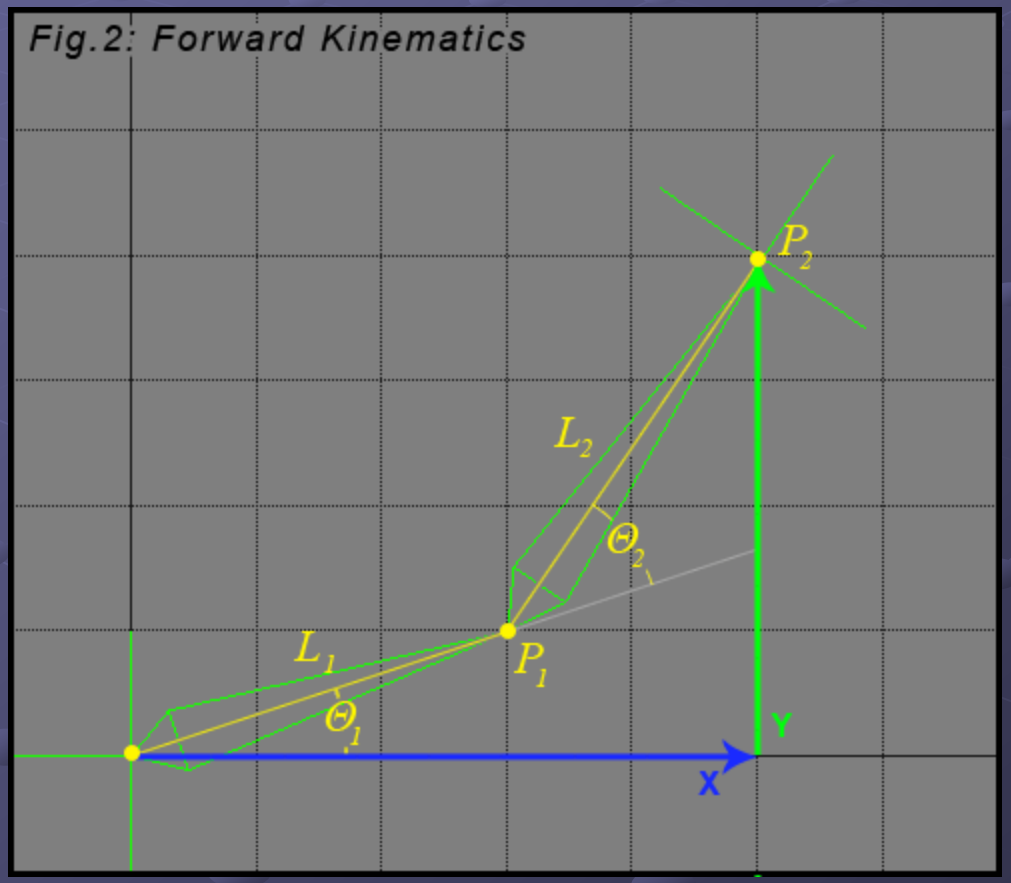
$$Px_1 = L_1 \times \cos(\Theta_1)$$
$$Py_1 = L_1 \times \sin(\Theta_1)$$



FK Equation

Expression for P2:

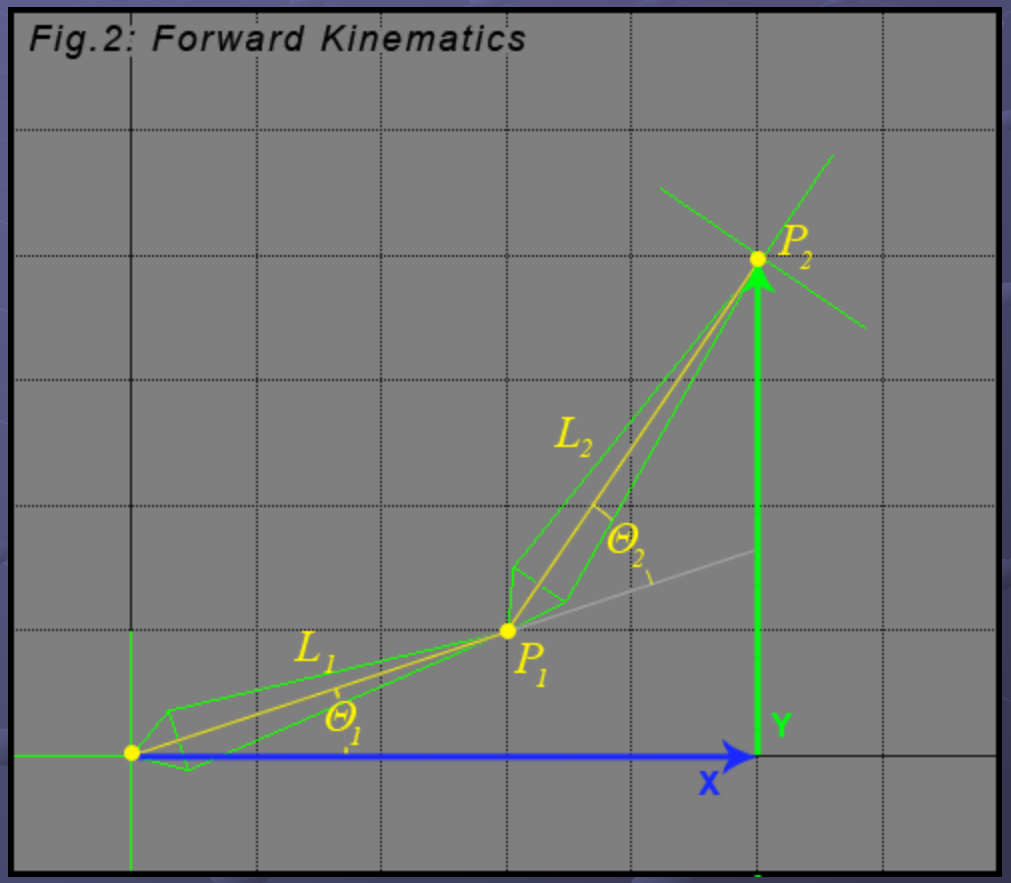
$$Px_2 = Px_1 + (L_2 \times \cos(\Theta_1 + \Theta_2)$$
$$Py_2 = Py_1 + (L_2 \times \sin(\Theta_1 + \Theta_2)$$



FK Equation

Expanded, FK Solution for P2:

$$Px_2 = L_1 \times \cos(\Theta_1) + (L_2 \times \cos(\Theta_1 + \Theta_2)$$
$$Py_2 = L_1 \times \sin(\Theta_1) + (L_2 \times \sin(\Theta_1 + \Theta_2)$$

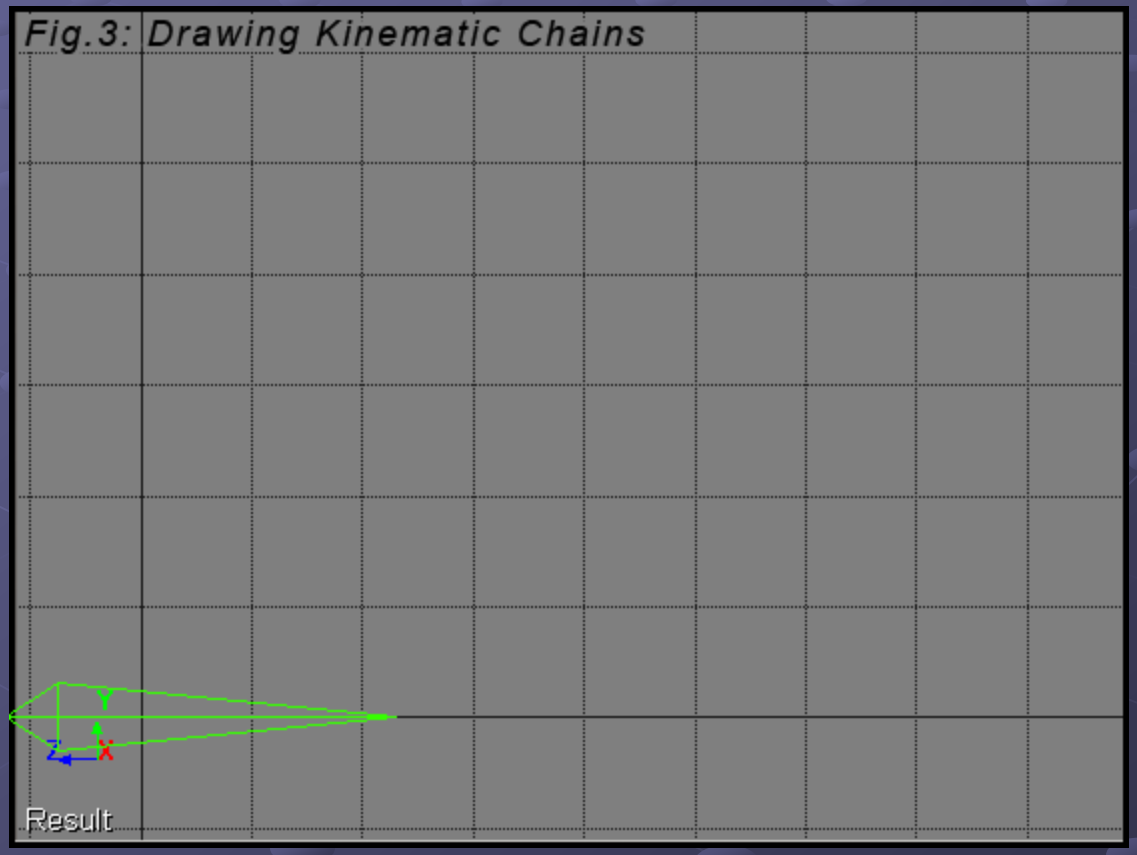


Drawing Kinematic Chains

- Drawing kinematic chains requires that the links are drawn from the outermost link to the innermost.
- The positioning of each link requires translations and rotations from each other link prior to it.

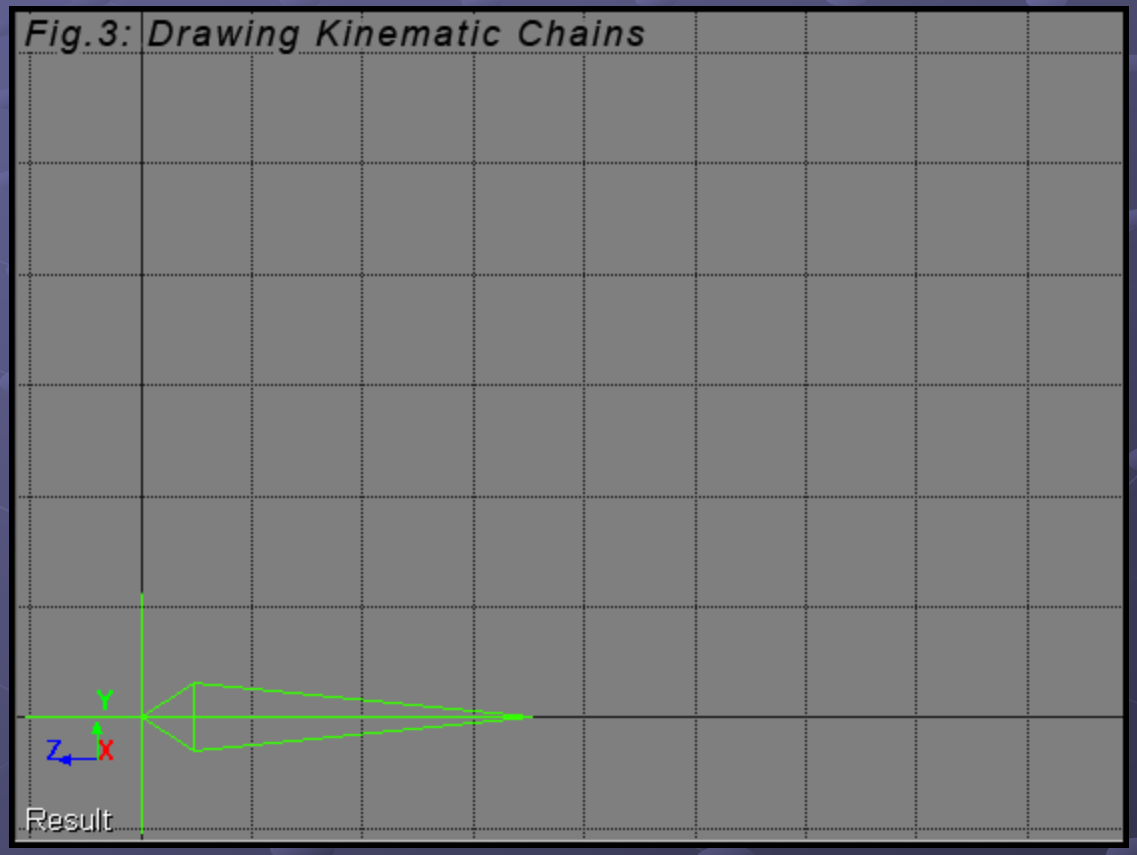
Drawing Kinematic Chains

- Starting with the effector's object:



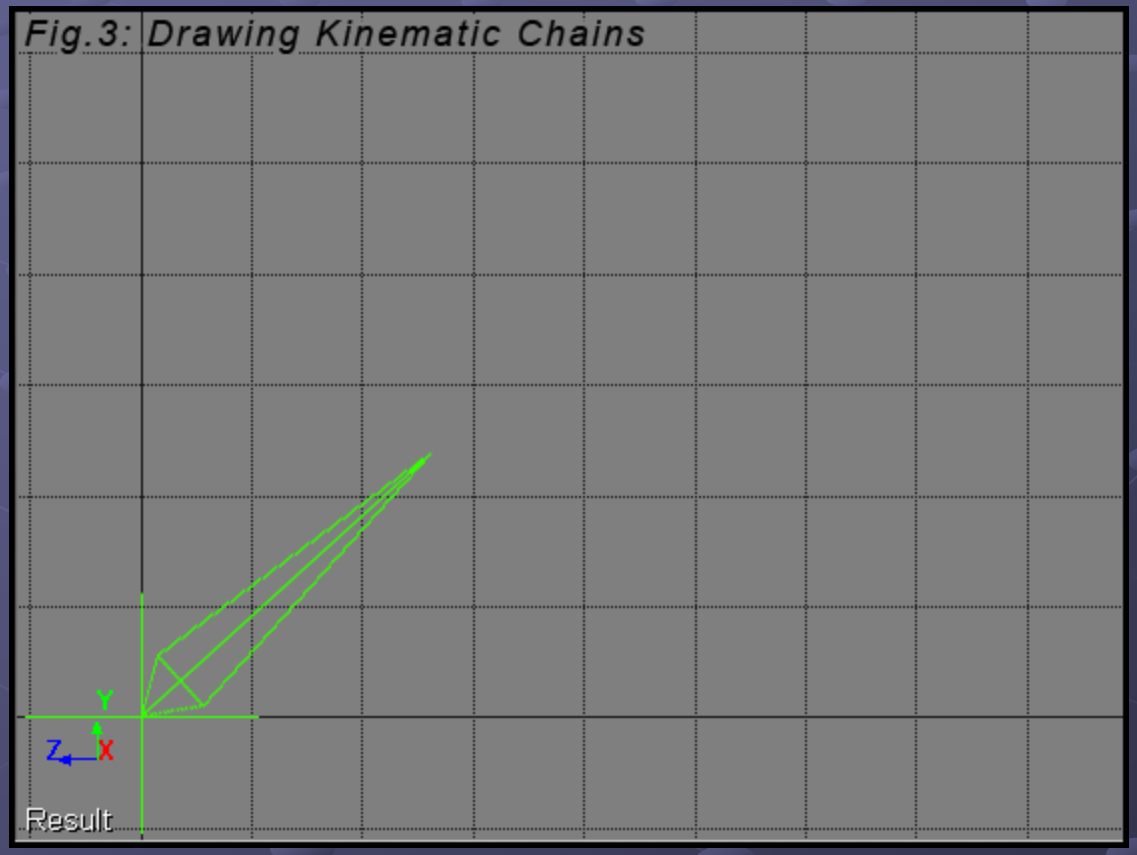
Drawing Kinematic Chains

- Starting with the effector's object:
  - Translate by length



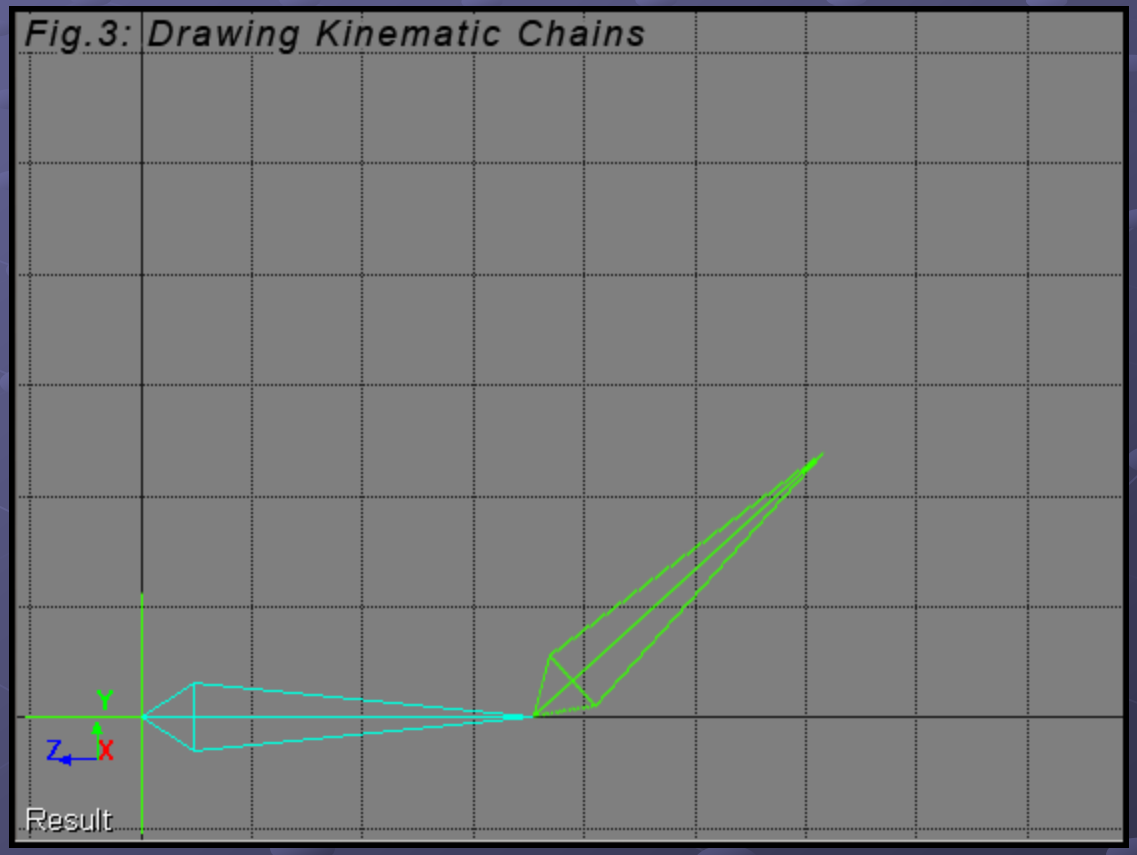
Drawing Kinematic Chains

- Starting with the effector's object:
  - Translate by length
  - Rotate by angle



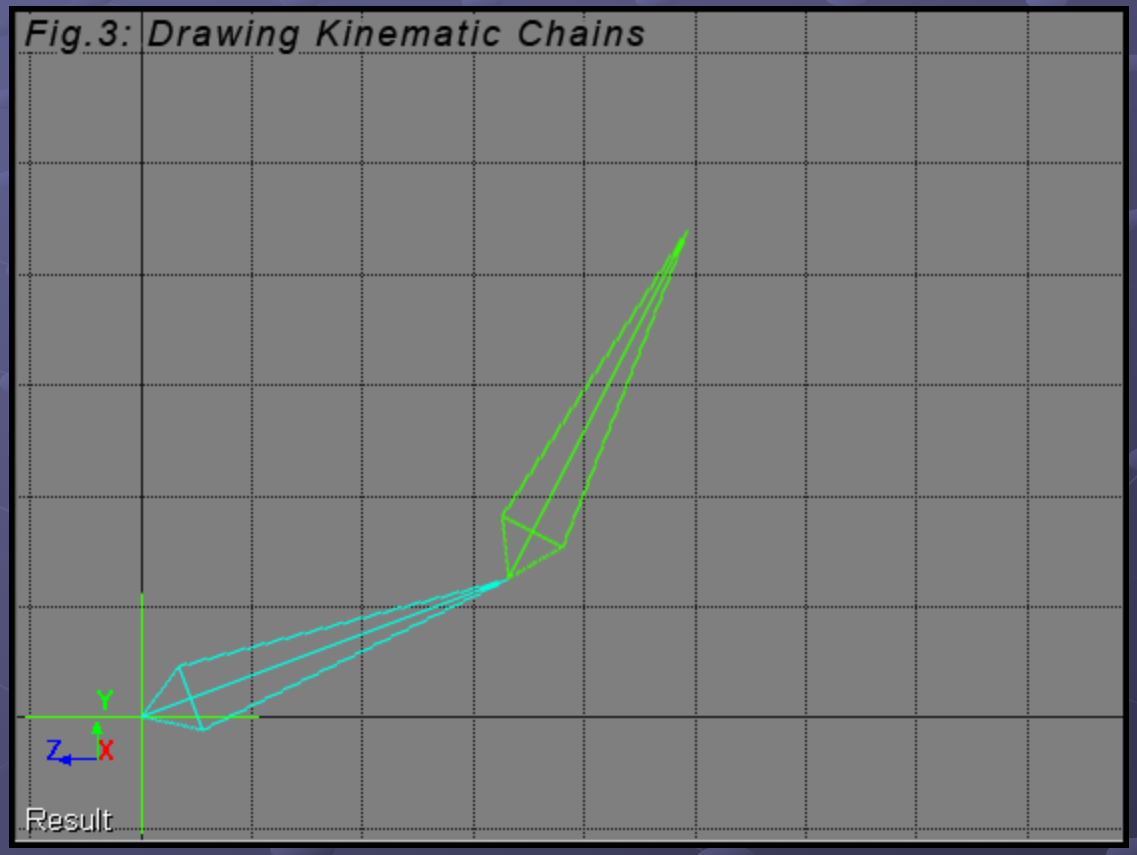
Drawing Kinematic Chains

- Starting with the effector's object:
  - Now translate by the length of the next link…



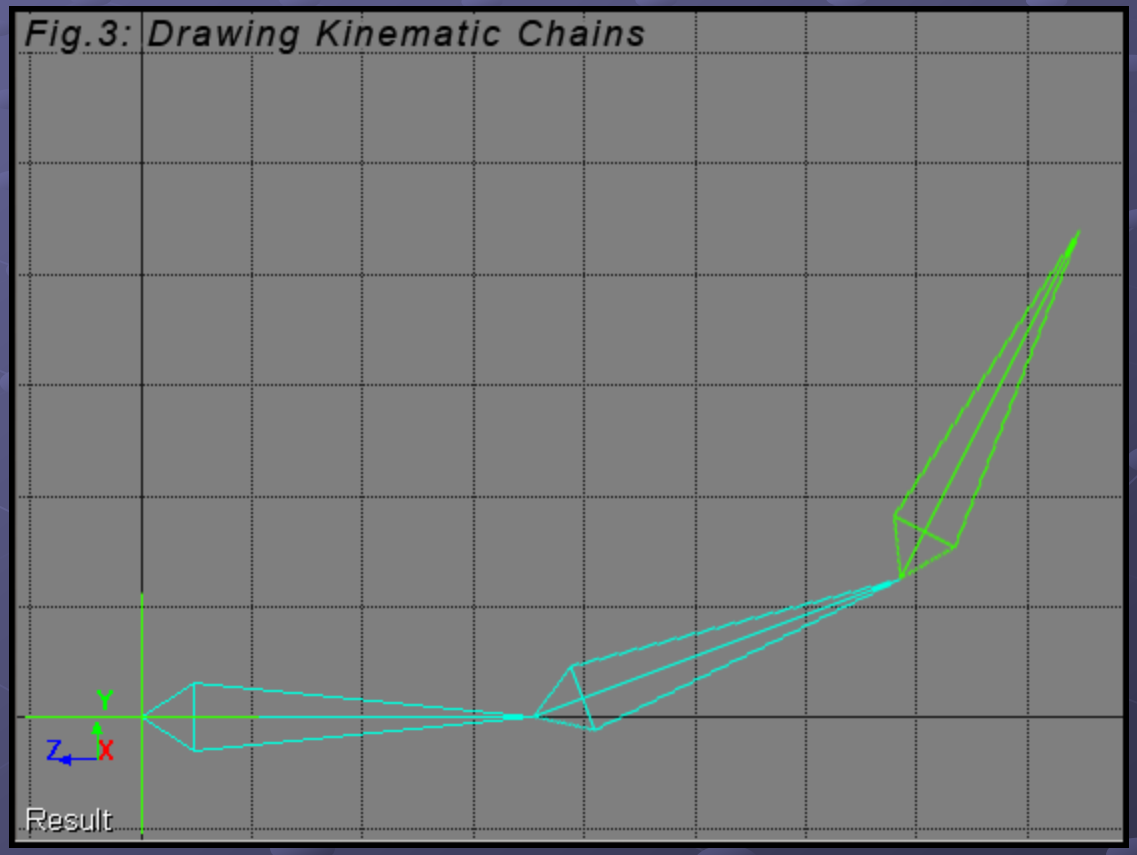
Drawing Kinematic Chains

- Starting with the effector's object:
  - Now translate by the length of the next link…
  - … and rotate the entire chain by the angle of that link



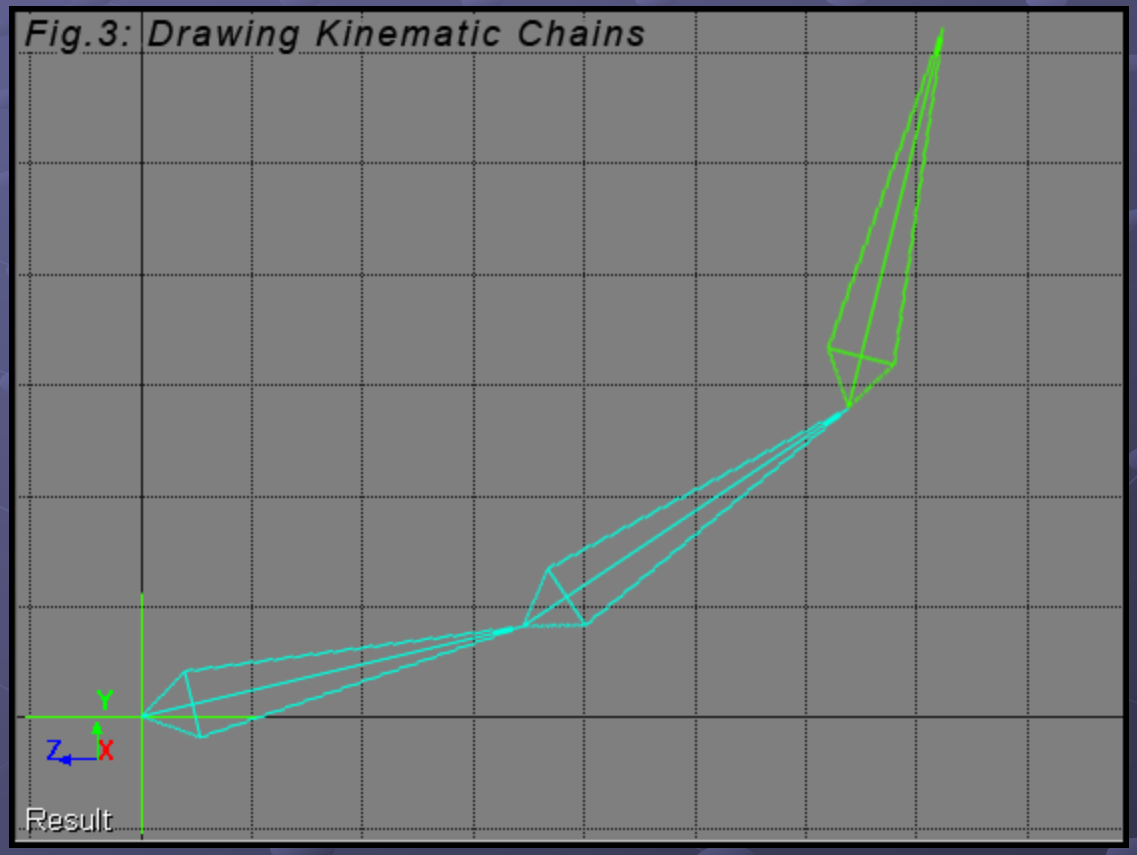
Drawing Kinematic Chains

- Starting with the effector's object:
  - Translate again by the length of the next link in the chain…



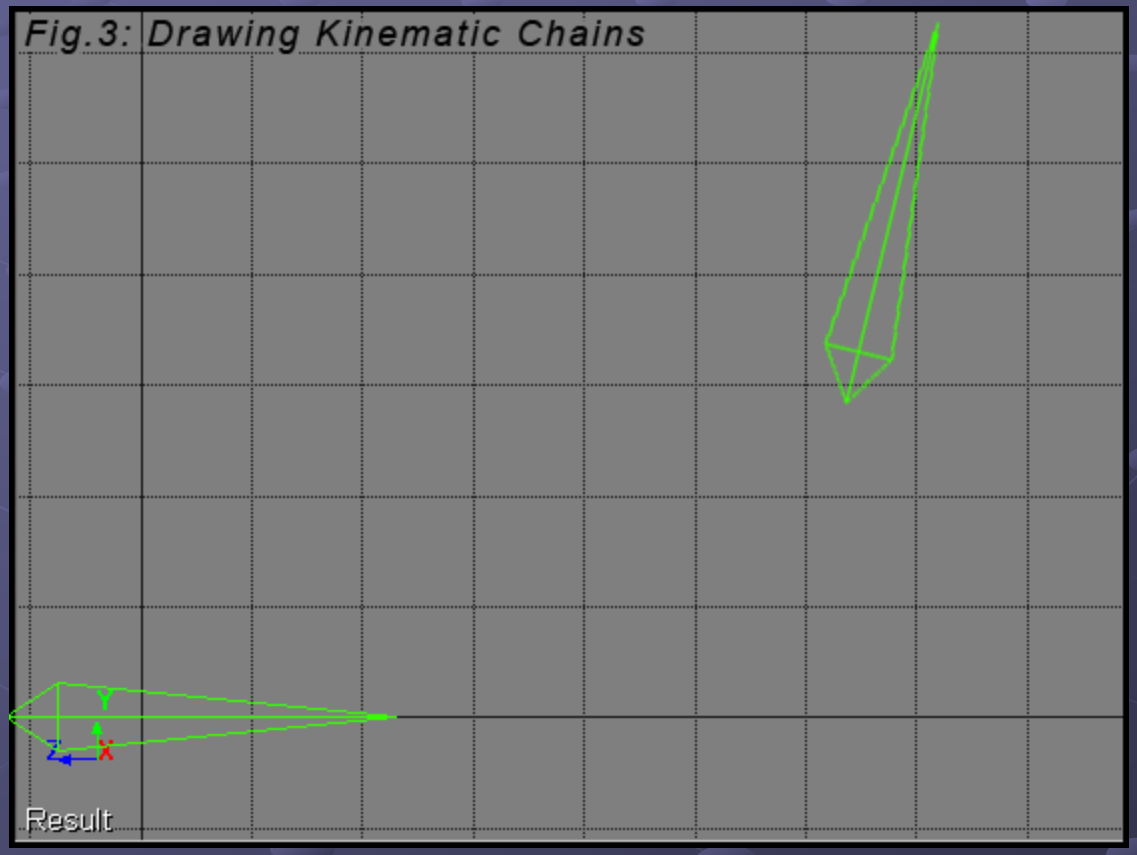
Drawing Kinematic Chains

- Starting with the effector's object:
  - Translate again by the length of the next link in the chain…
  - … and rotate the entire chain by that link's angle



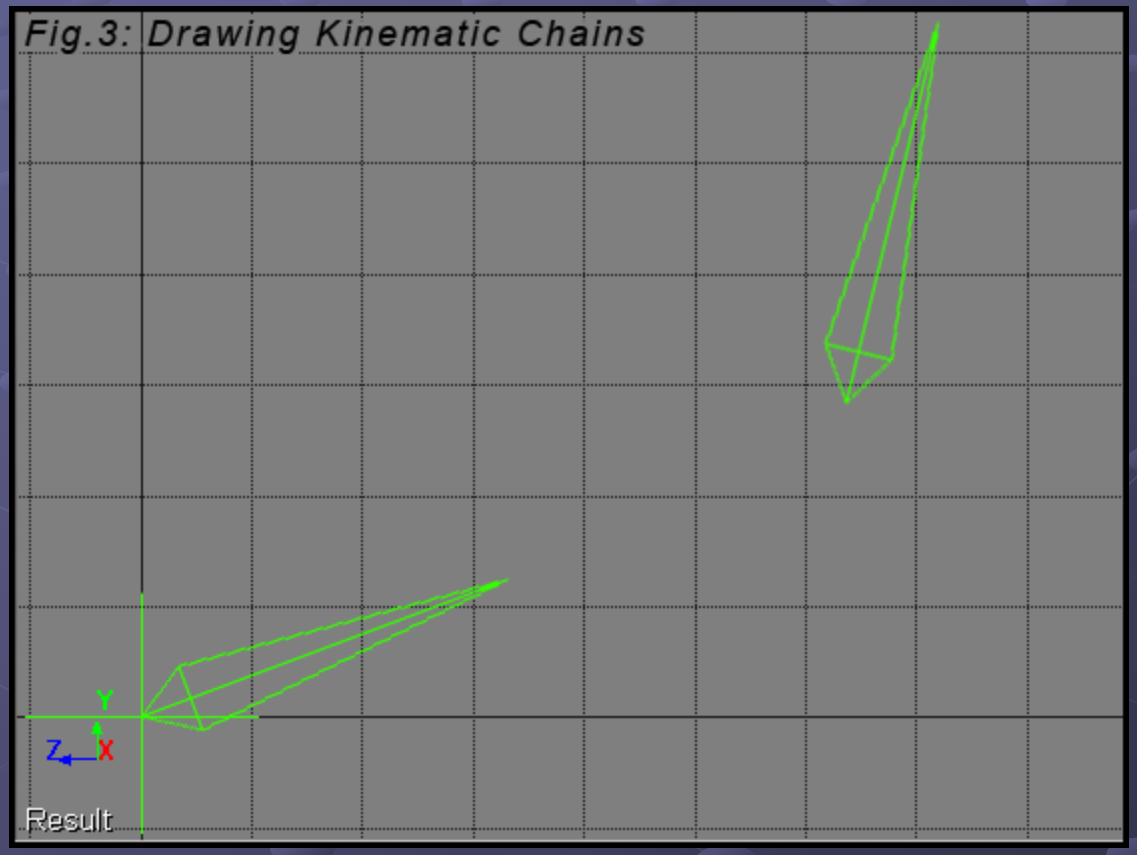
Drawing Kinematic Chains

- Starting with the NEXT link's object:



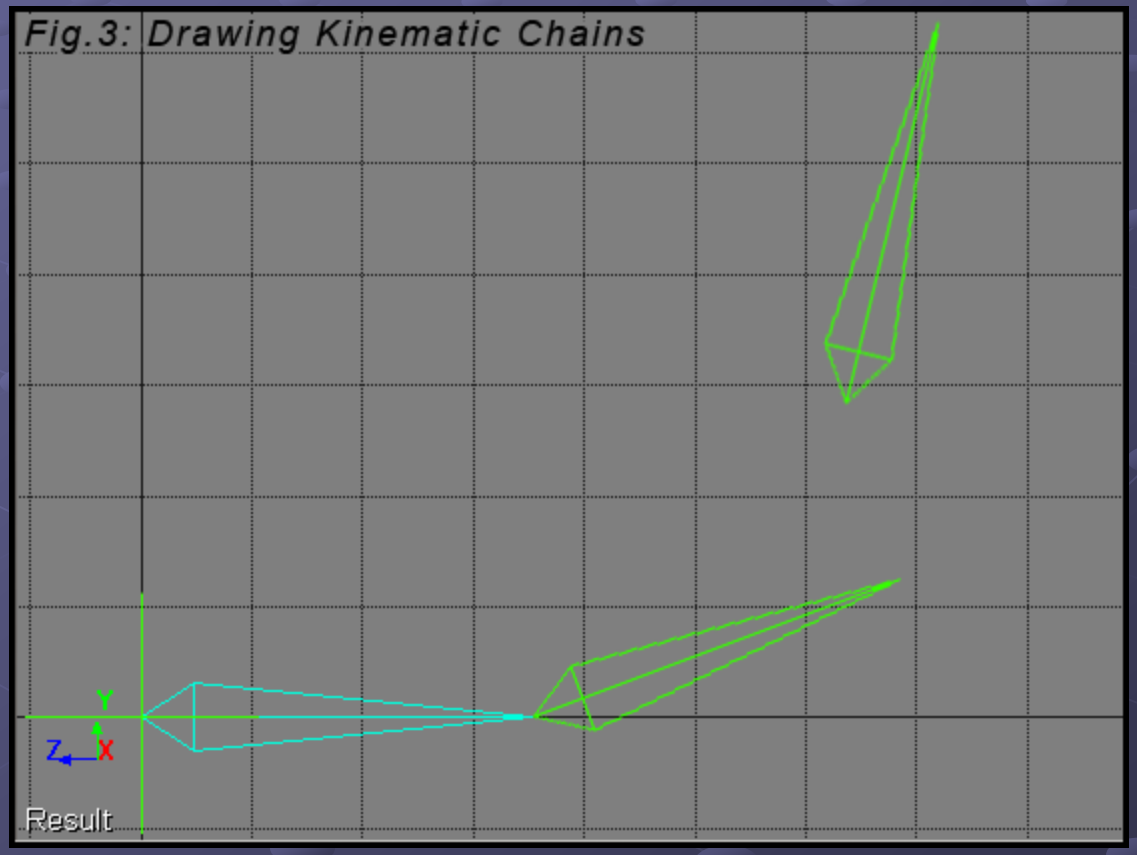
Drawing Kinematic Chains

- Starting with the NEXT link's object:
  - Translate by the object's length…
  - Rotate by the object's angle.



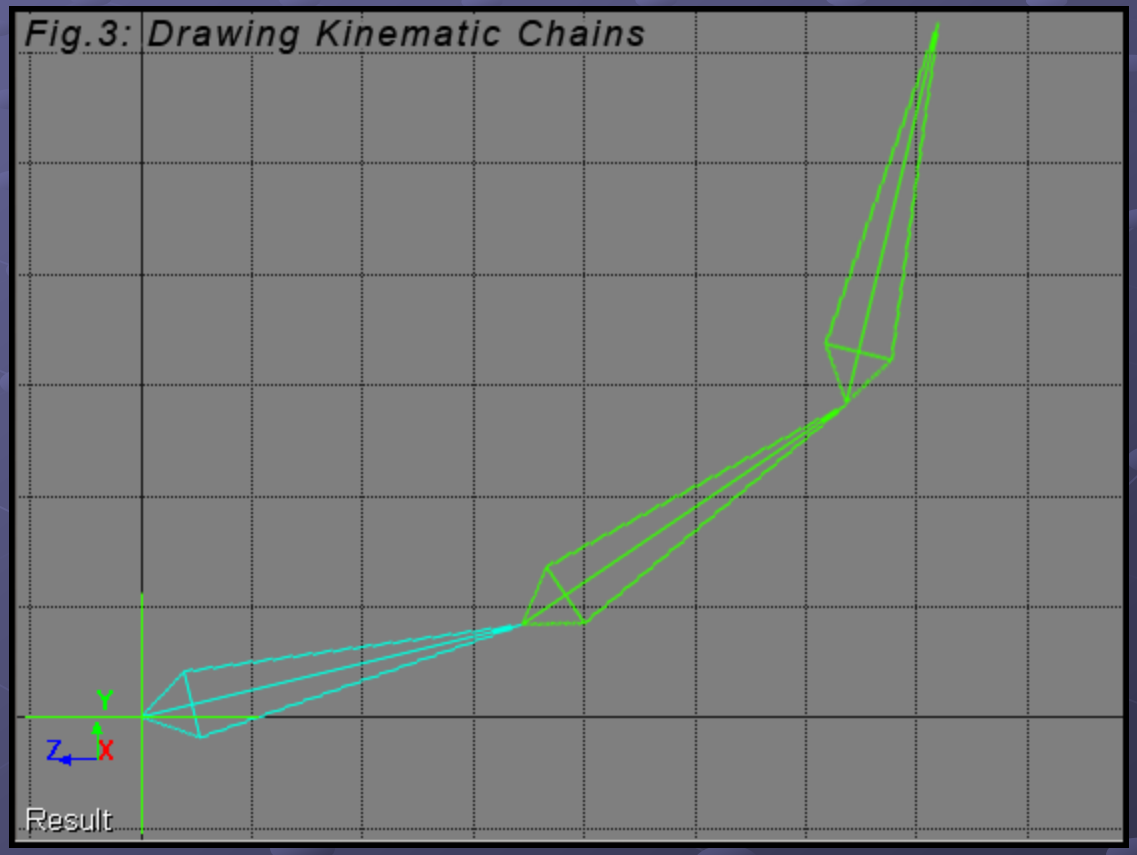
Drawing Kinematic Chains

- Starting with the NEXT link's object:
  - Translate by the next object's length…



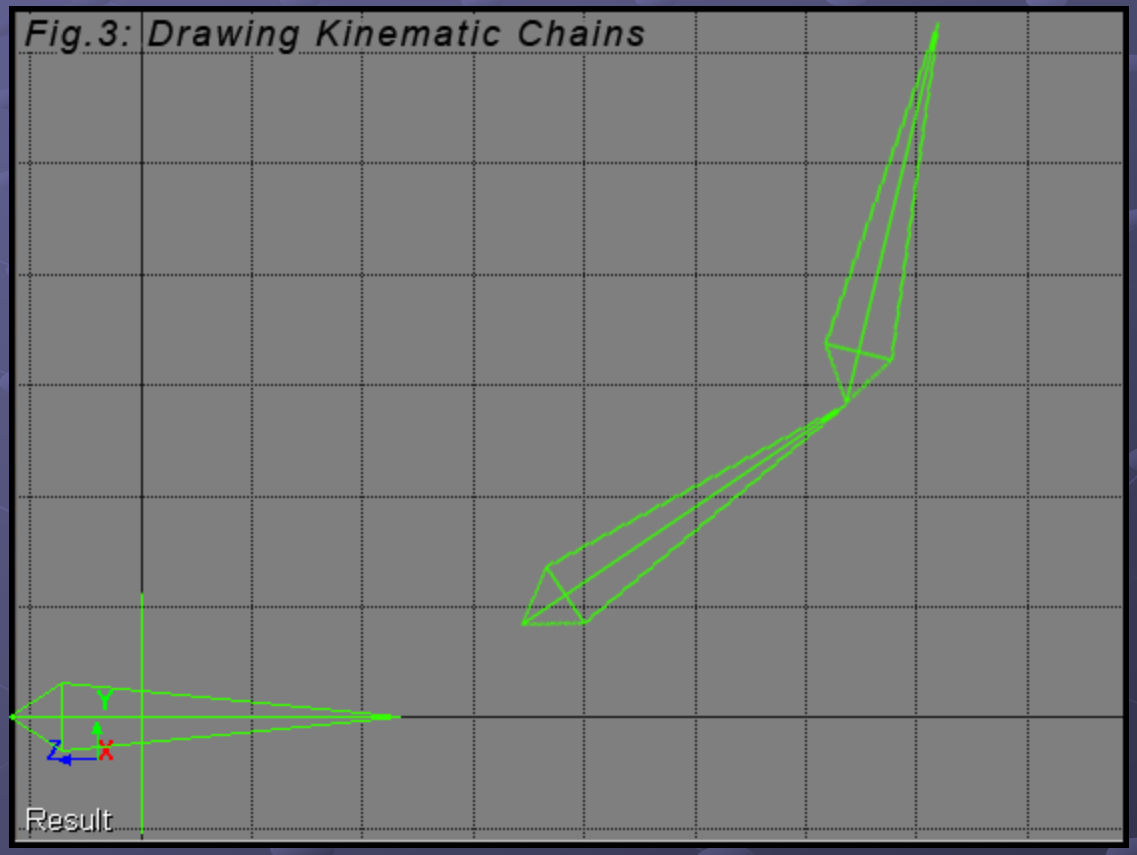
Drawing Kinematic Chains

- Starting with the NEXT link's object:
  - Translate by the next object's length…
  - And rotate the entire chain by that object's angle.



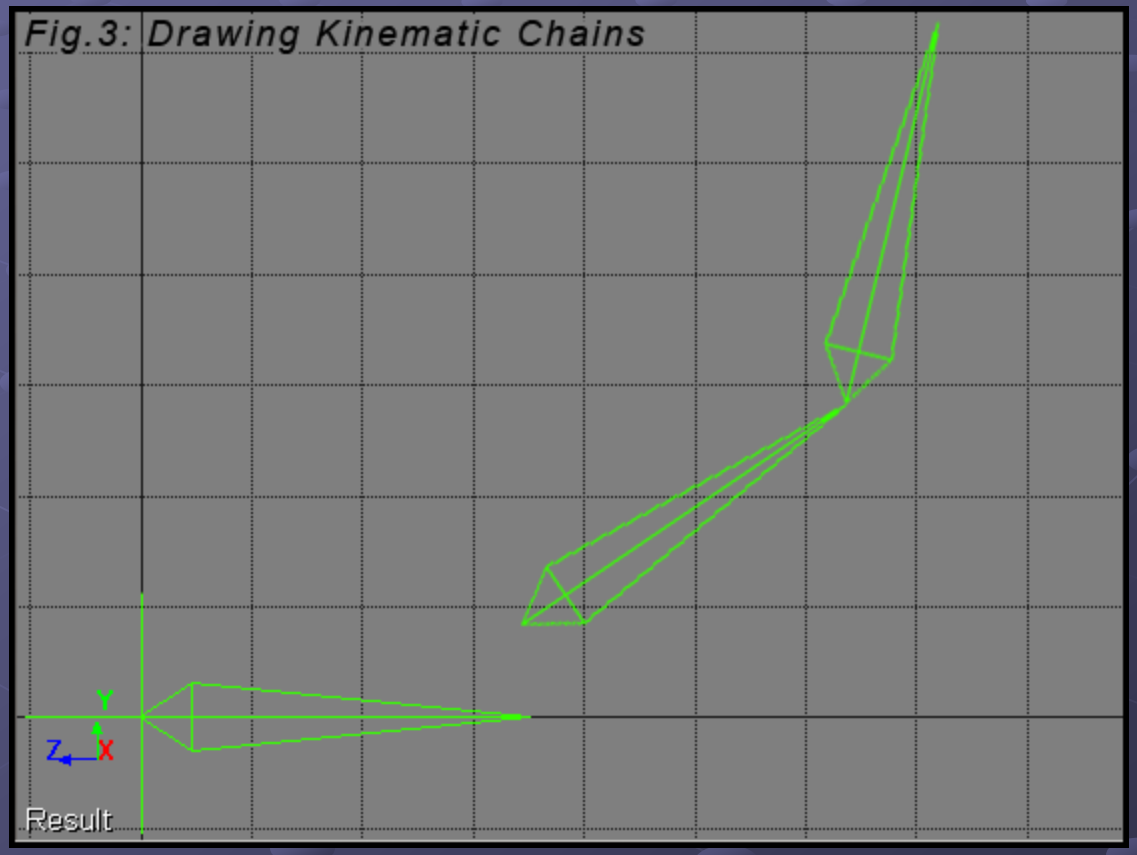
Drawing Kinematic Chains

- So on, and so forth…
  - Place the next link



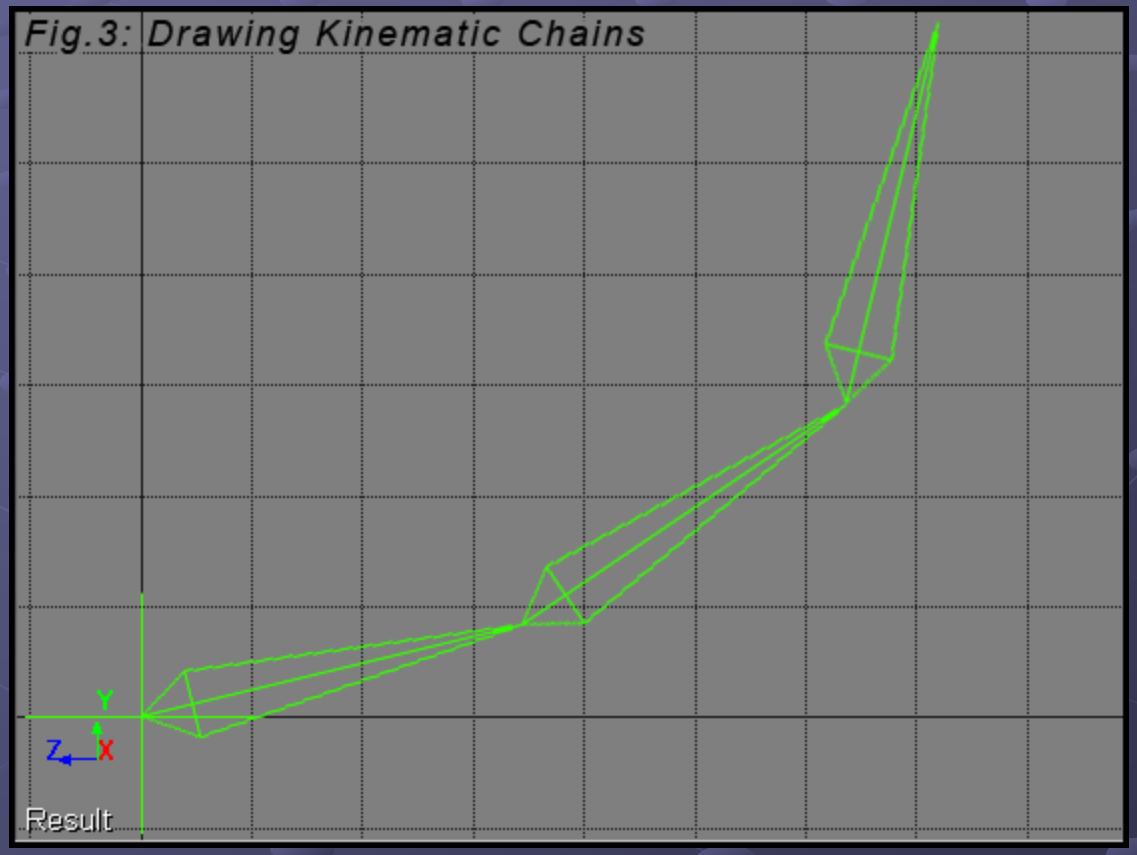
Drawing Kinematic Chains

- So on, and so forth…
  - Place the next link
  - translate



Drawing Kinematic Chains

- So on, and so forth until chain is complete.
  - Place the next link
  - Translate
  - Rotate



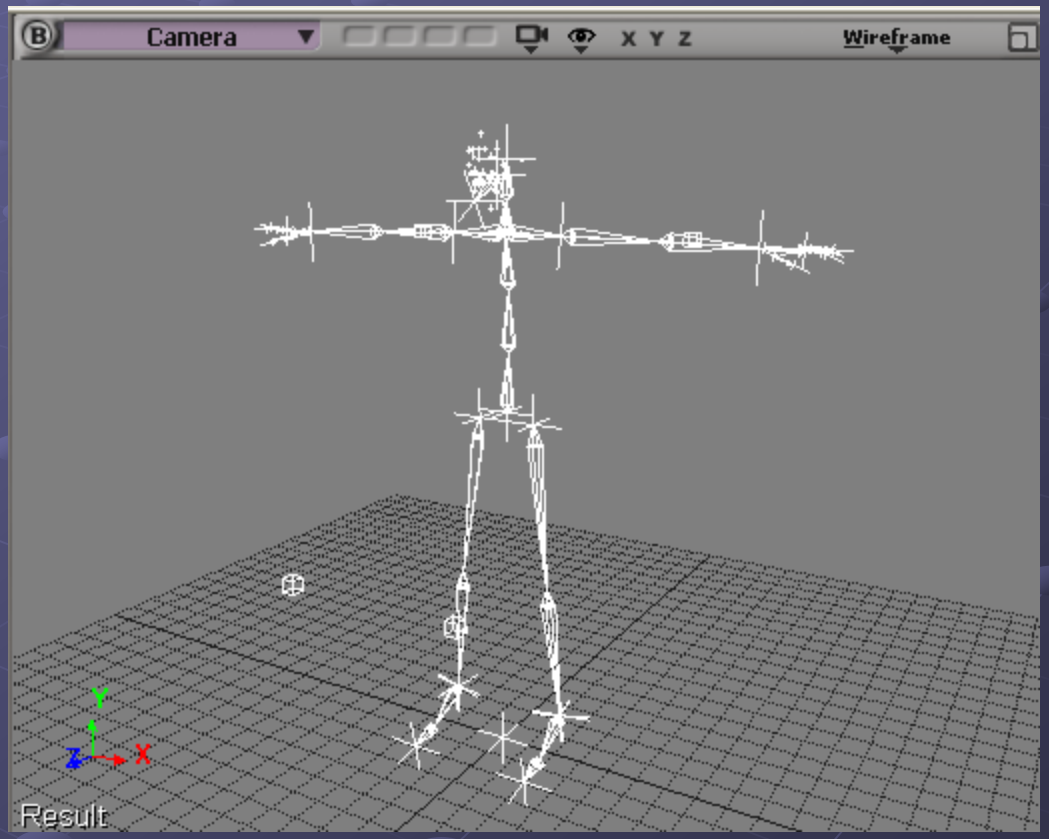
Inverse Kinematics

Now that we know generally what Forward Kinematics are, how do we define Inverse Kinematics?

Inverse Kinematics

Forward Kinematics (FK)

Mathematically determining the position and angle of joints in a series of flexible, jointed objects after determining the position and orientation of the end effector.



Inverse Kinematics

What is IK used for?

- Originally used in industrial robotics for assembly plants
- In game design, IK is typically used most often in character animation

Types of IK Solutions

- Closed Form / Analytical
  - Calculates the angle from the root to the effector, and allows us to determine immediately if a solution is even available. Then using trig identities, we can develop an equation that can determine any number of possible solutions. Consider the example below for a two link chain:

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

$$\theta_1 = \tan^{-1}\left(\frac{-l_2 \sin\theta_2 x + (l_1 + l_2 \cos\theta_2) y}{l_2 \sin\theta_2 y + (l_1 + l_2 \cos\theta_2) x}\right)$$

- Relatively simple solution for smaller problems.
- However, as the chain increases in the number of elements adding new degrees of freedom, the problem quickly becomes very complex.

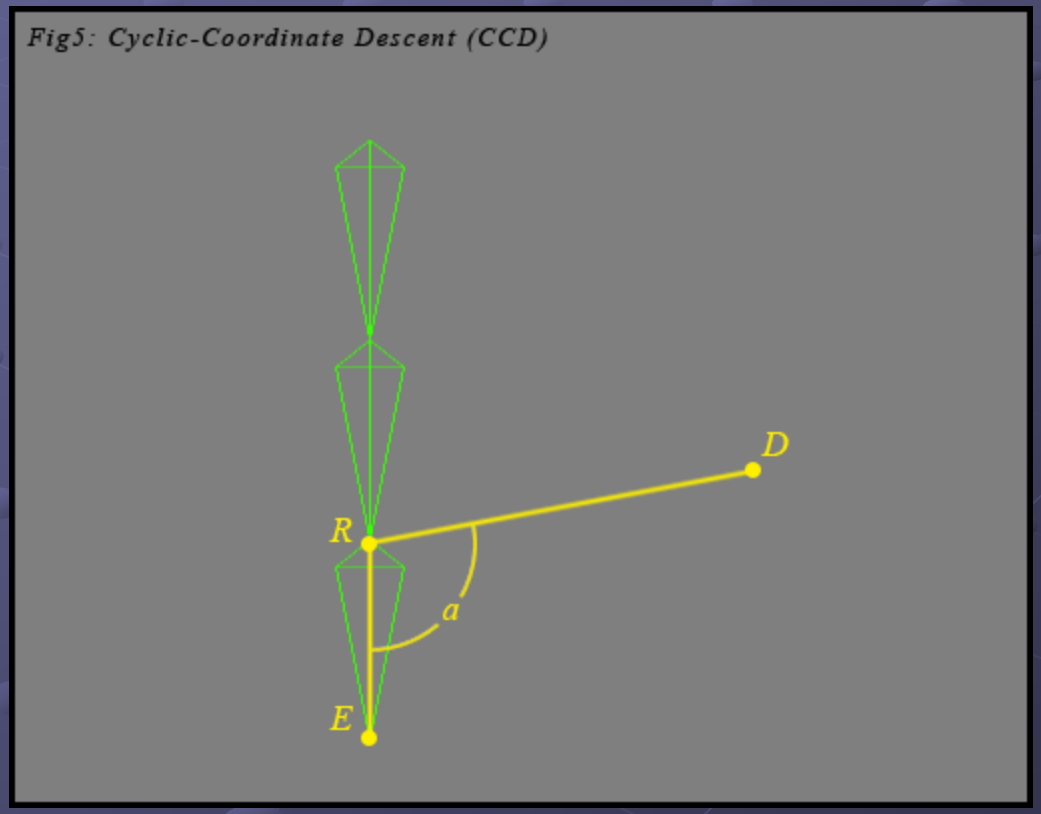
Types of IK Solutions

- Cyclic-Coordinate Descent
  - First defined by Chris Welmen, borrowing from work by Li-Chun Tommy Wang and Chih Cheng Chen in a paper titled IEEE Transactions on Robotics and Automation
  - Minimizes the possibility of system error by adjusting each joint angle one at a time

Types of IK Solutions

Cyclic-Coordinate Descent

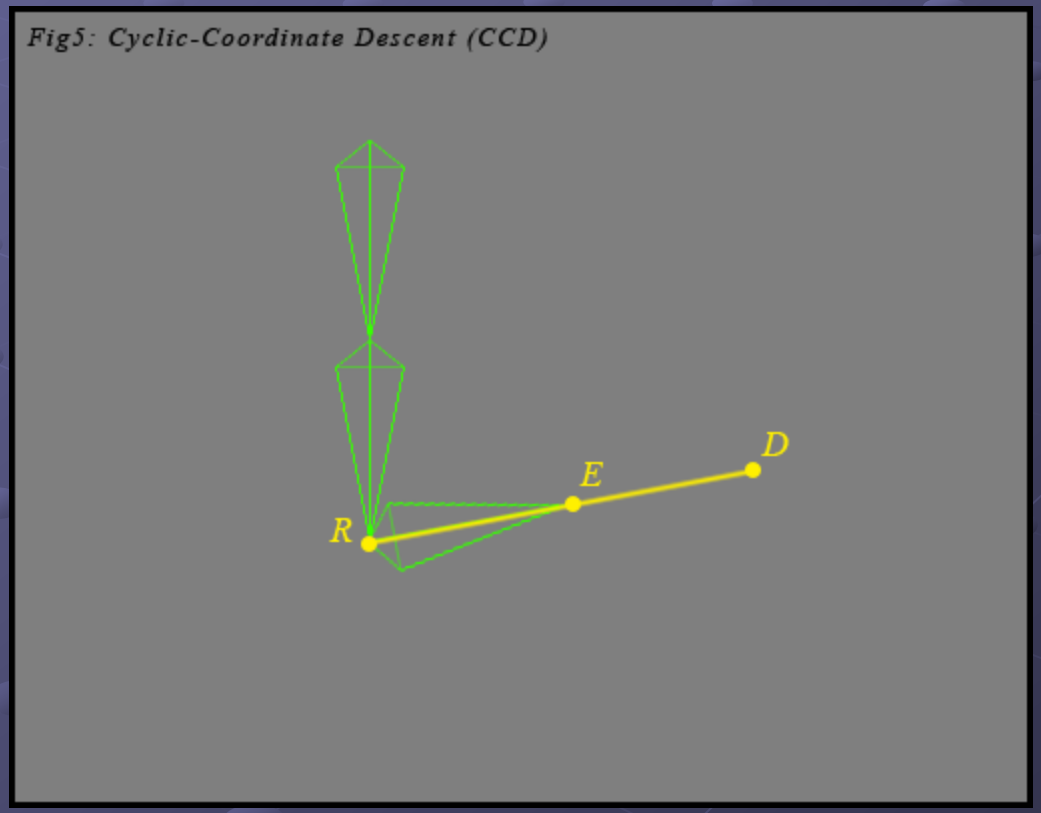
- Starting with the root of our effector, R, to our current endpoint, E.
- Next, we draw a vector from R to our desired endpoint, D
- The inverse cosine of the dot product gives us the angle between the vectors: cos(a) = RD ● RE



Types of IK Solutions

Cyclic-Coordinate Descent

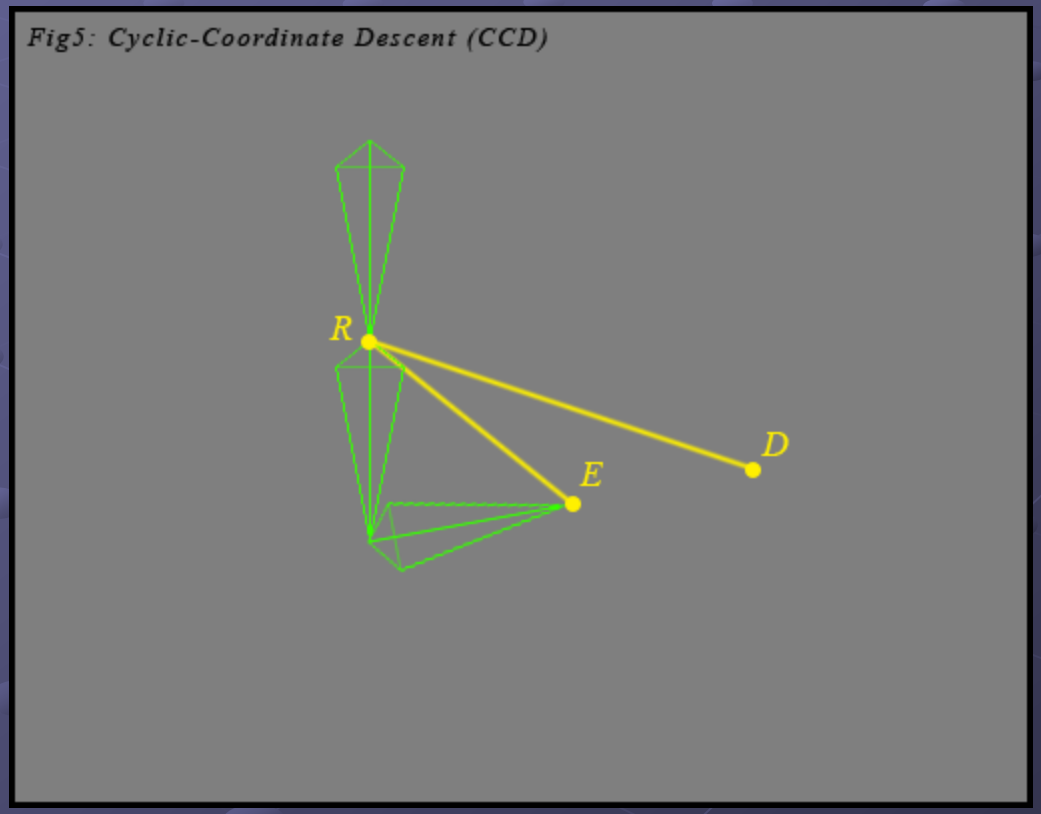
Rotate our link so that RE falls on RD



Types of IK Solutions

Cyclic-Coordinate Descent

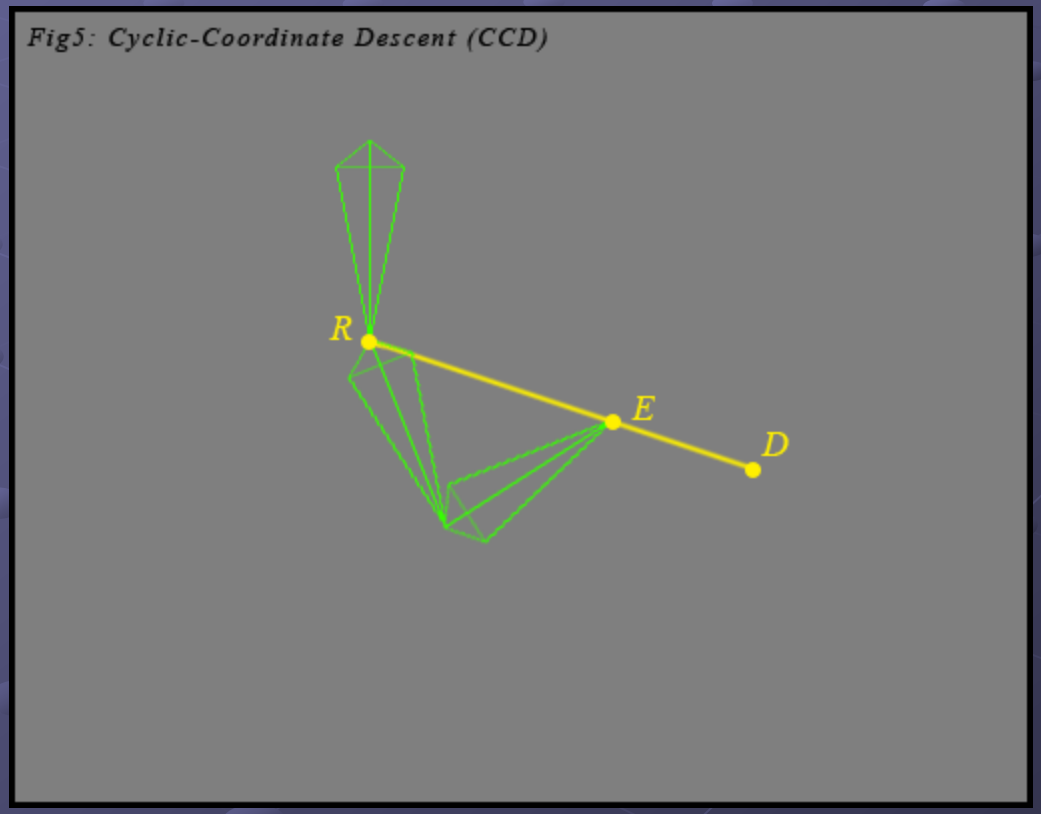
Move one link up the chain, and repeat the process



Types of IK Solutions

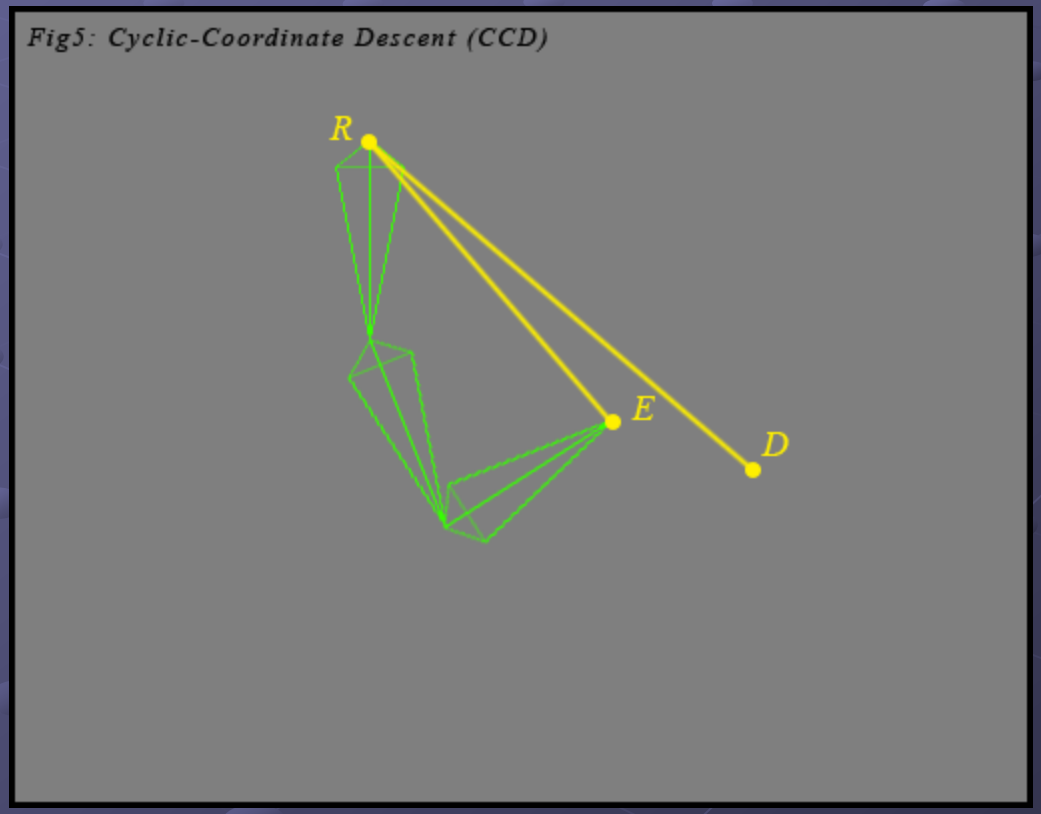
Cyclic-Coordinate Descent

The process is basically repeated until the root joint is reached. Then the process begins all over again starting with the end effector, and will continue until we are close enough to D for an acceptable solution.



Types of IK Solutions

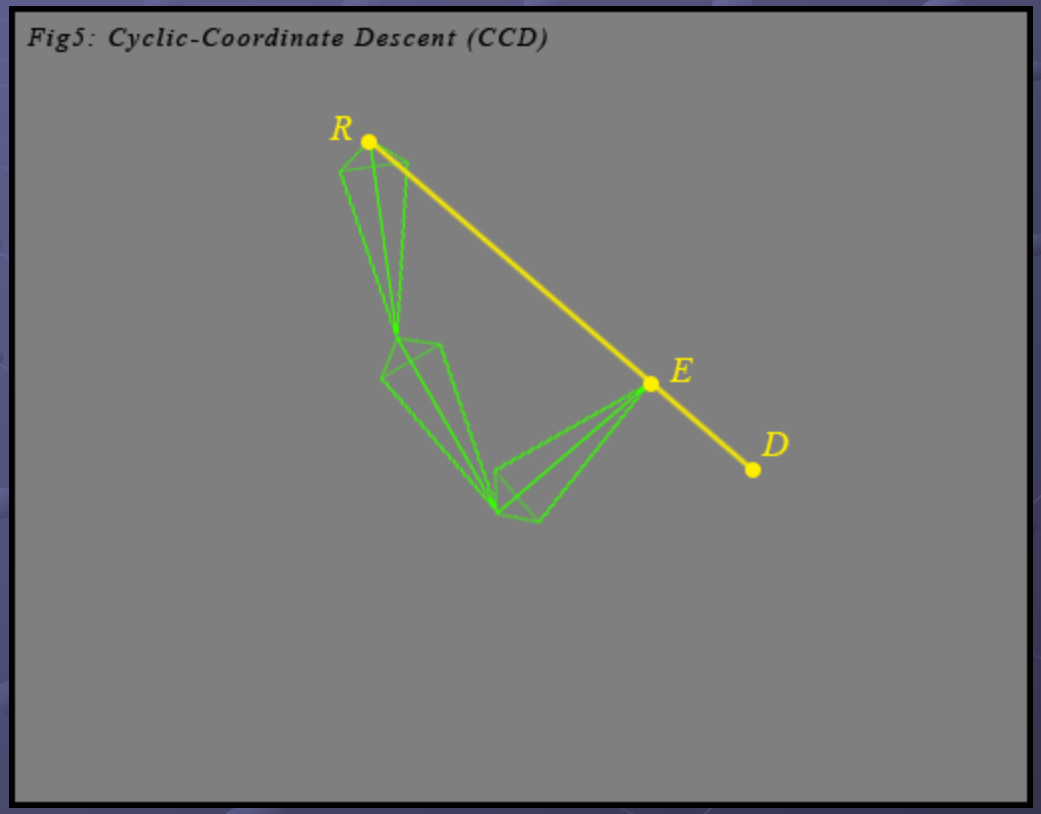
Cyclic-Coordinate Descent



Types of IK Solutions

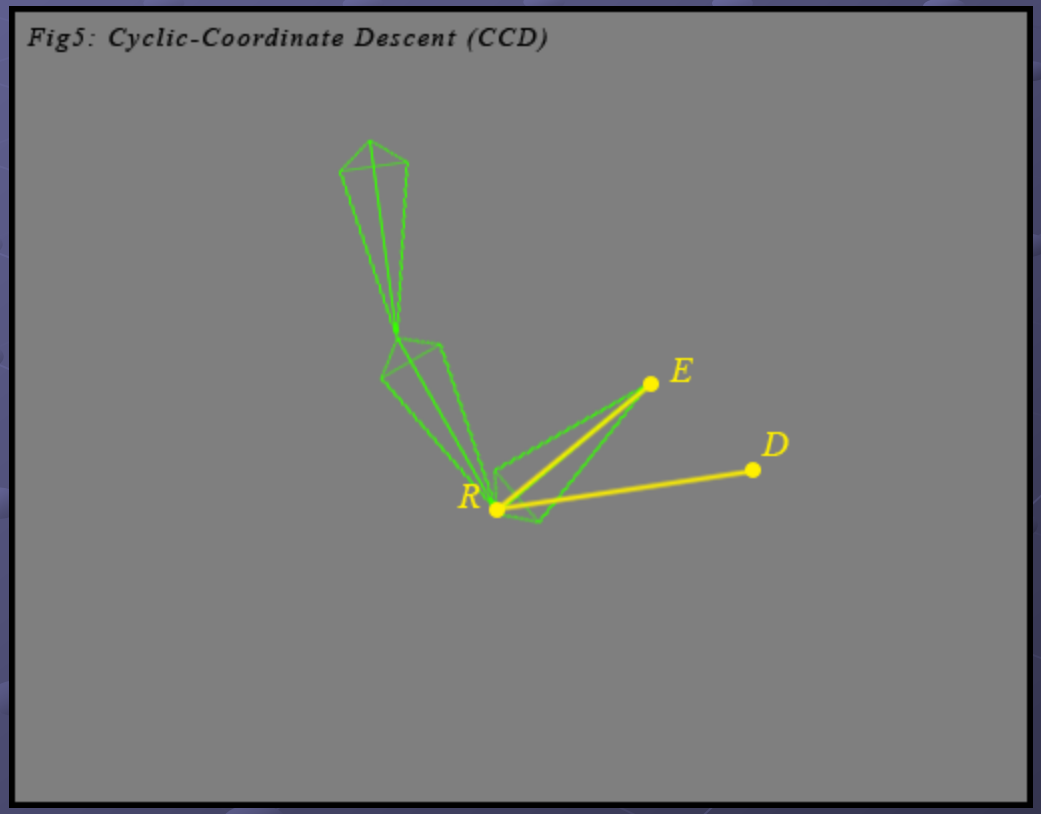
Cyclic-Coordinate Descent

We've reached the root. Repeat the process



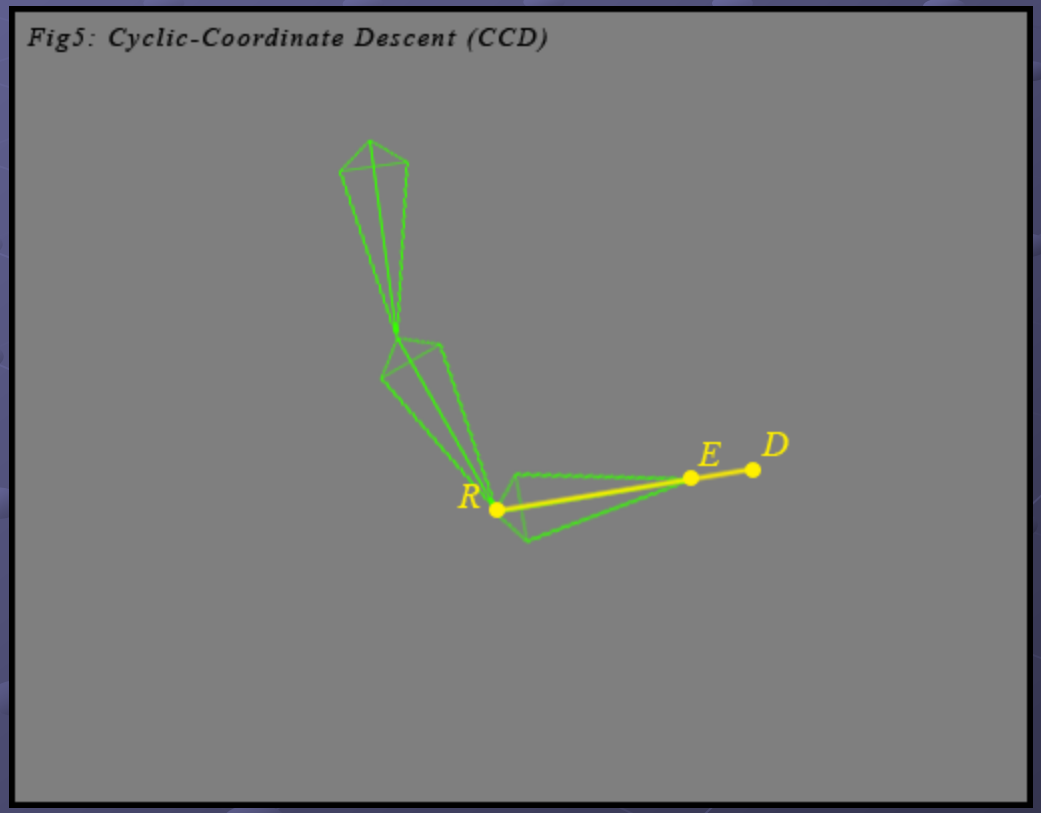
Types of IK Solutions

Cyclic-Coordinate Descent



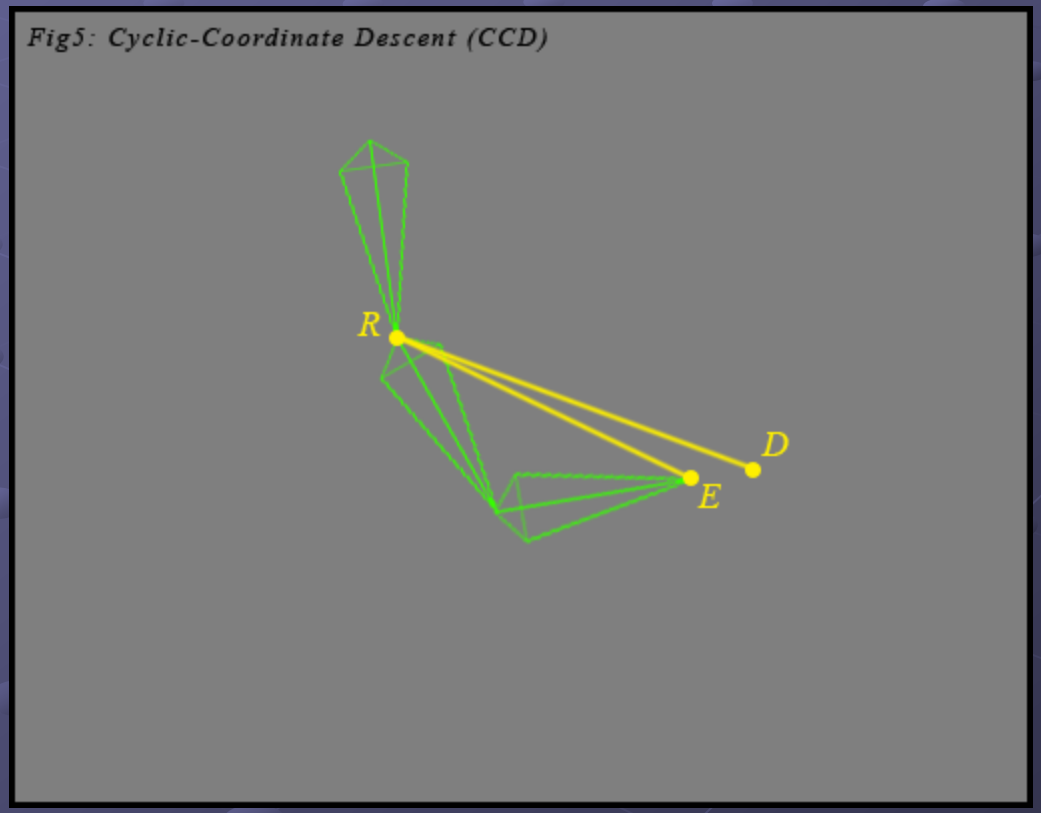
Types of IK Solutions

Cyclic-Coordinate Descent



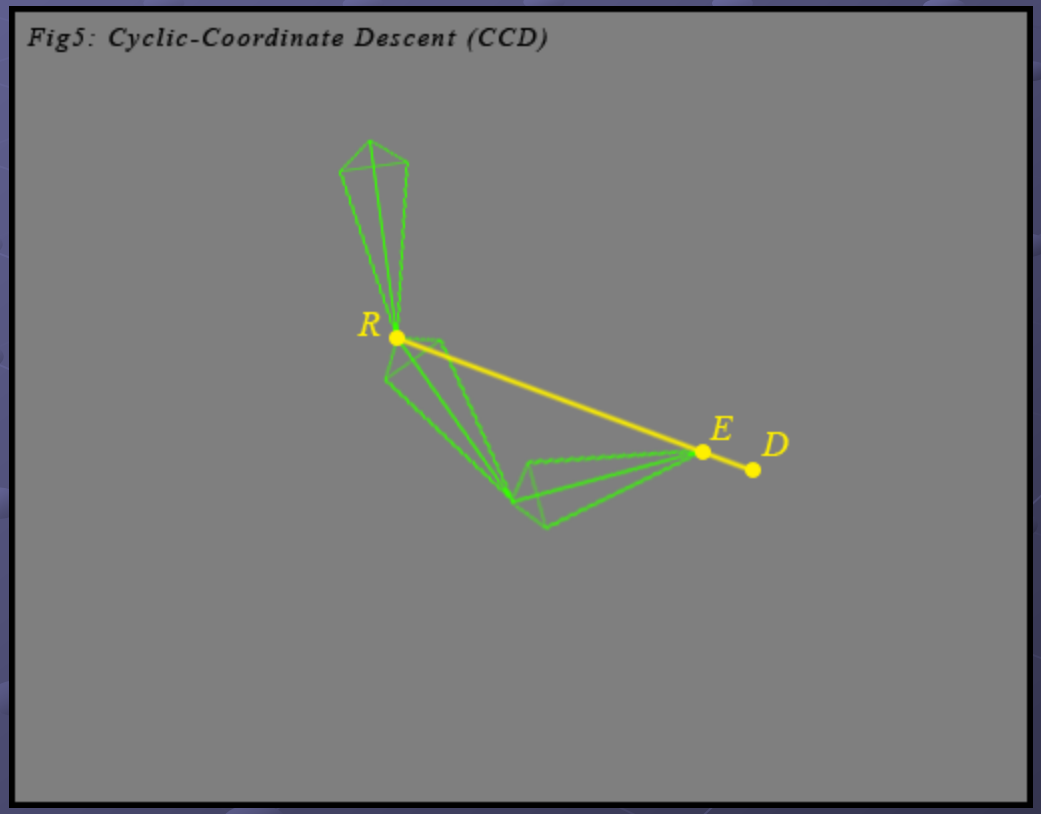
Types of IK Solutions

Cyclic-Coordinate Descent



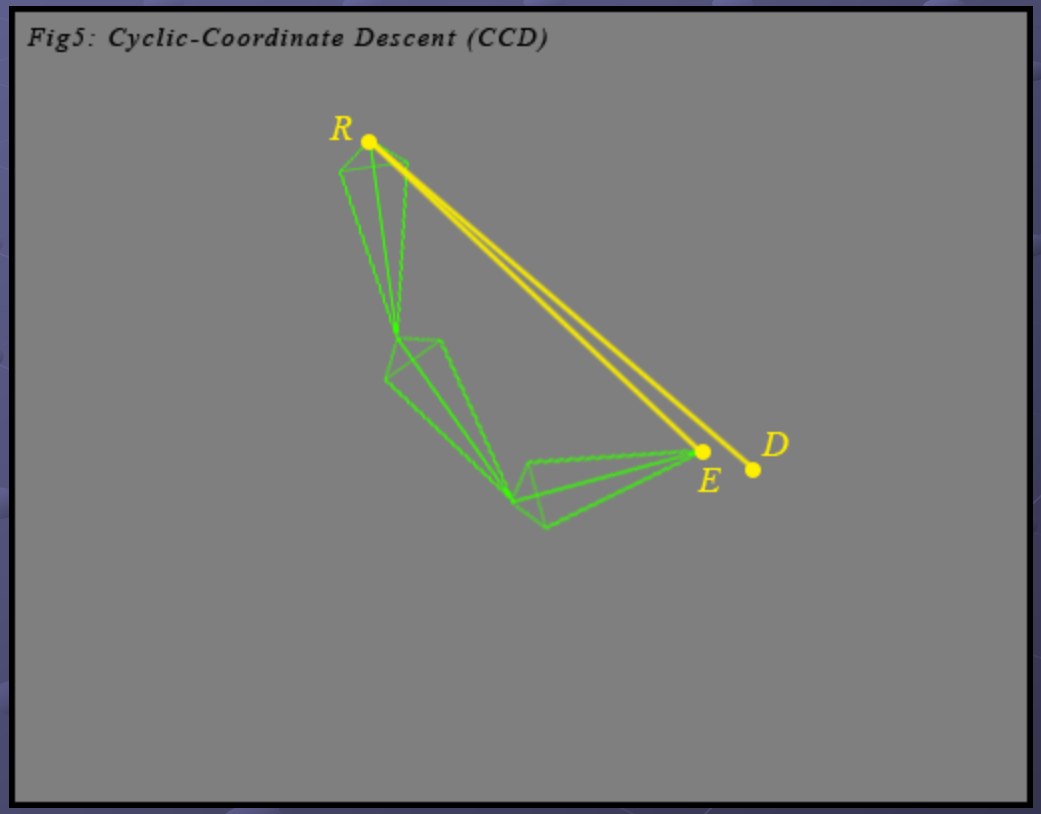
Types of IK Solutions

Cyclic-Coordinate Descent



Types of IK Solutions

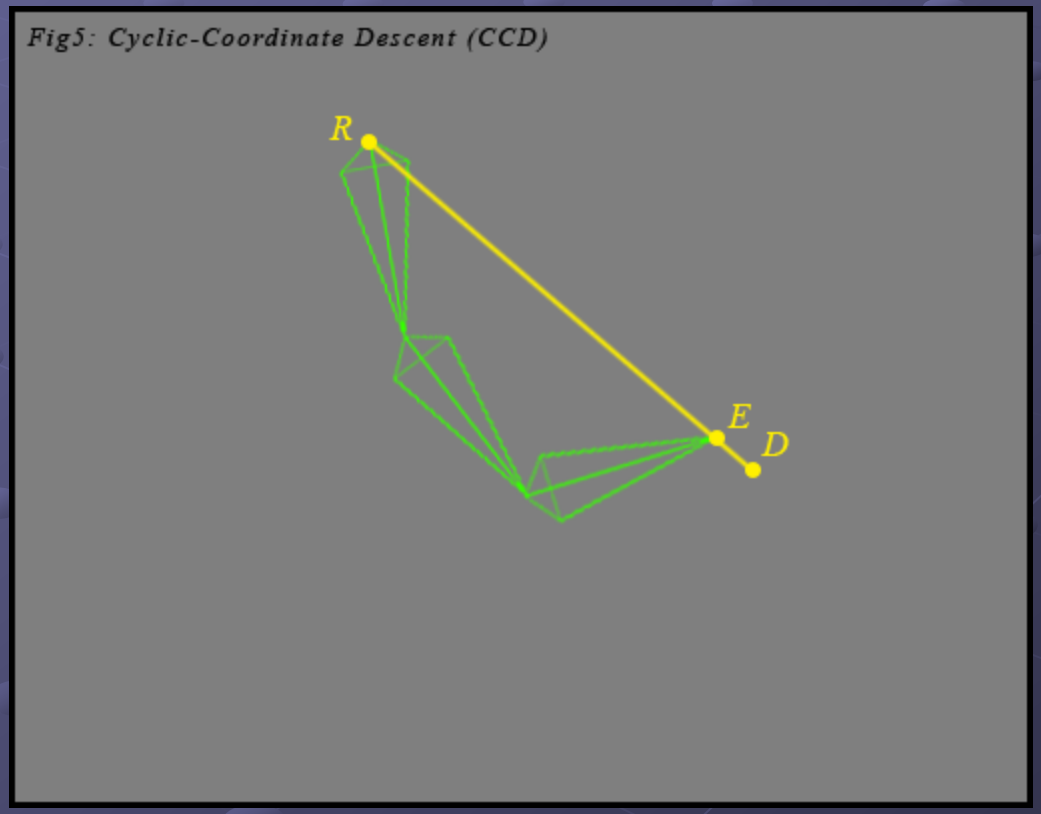
Cyclic-Coordinate Descent



Types of IK Solutions

Cyclic-Coordinate Descent

We've reached the root again. Repeat the process until solution reached.



Using IK in Game Development

Examples of IK in action:

- Character Animation Demo (Softimage XSI 5.0)
- Real-Time calculations: E3 2003 Demo Footage of Half-Life 2