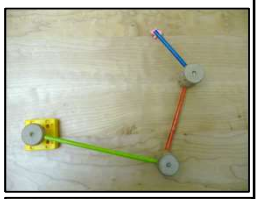
1

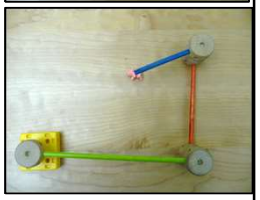
Inverse Kinematics

Oregon State University

Mike Bailey

mjb@cs.oregonstate.edu



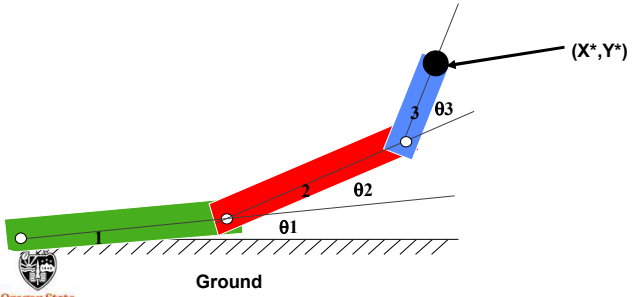


2

Inverse Kinematics

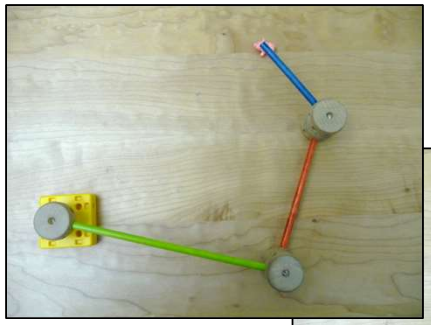
Forward Kinematics solves the problem "if I know the link transformation parameters, where are the links?".

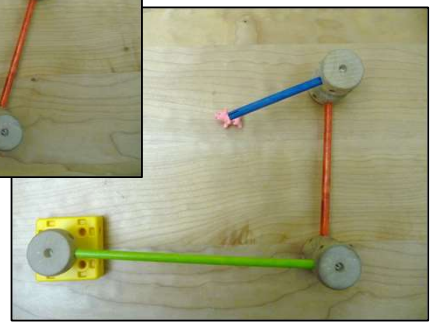
Inverse Kinematics *(IK)* solves the problem "If I know where I want the links to be (X*,Y*), what link transformation parameters will put them there?"



3

Inverse Kinematics (IK): Things Need to Move – What Parameters Will Make Them Do That?

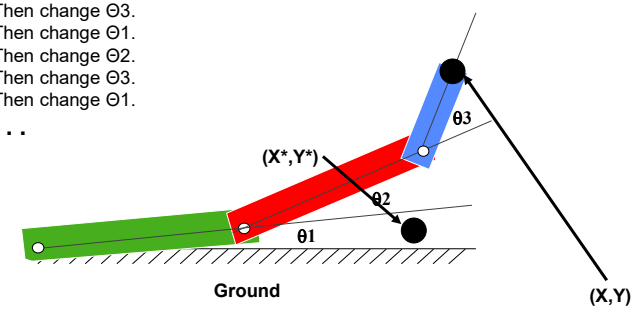




4

Cyclic Coordinate Descent (CCD) Method

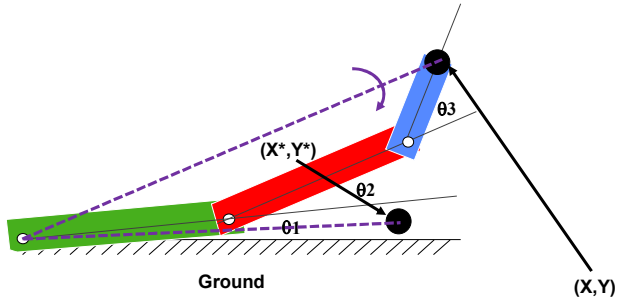
The idea is to change Θ1 so that (X,Y) are as close to (X*,Y*) as possible.
Then change Θ2.
Then change Θ3.
Then change Θ1.
Then change Θ2.
Then change Θ3.
Then change Θ1.
. . .



5

Changing Θ1

Holding Θ2 and Θ3 constant, rotate Θ1 towards (X*,Y*) so that the dashed purple lines line up.



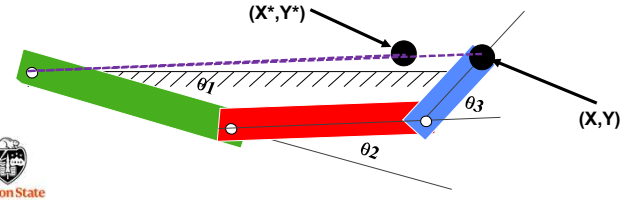
Oregon State University
Computer Graphics

mjb – July 30, 2021

5

6

Changing Θ1



Oregon State University
Computer Graphics

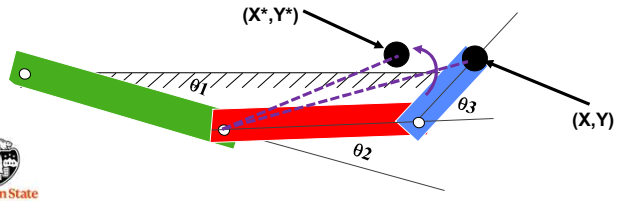
mjb – July 30, 2021

6

7

Changing Θ2

Holding Θ1 and Θ3 constant, rotate Θ2 towards (X*,Y*) so that the dashed purple lines line up.



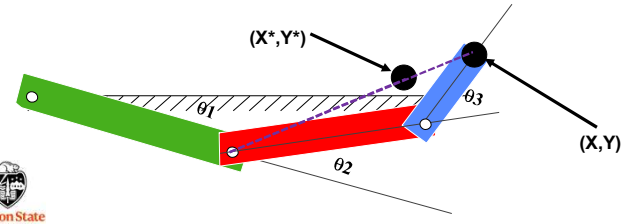
Oregon State University
Computer Graphics

mjb – July 30, 2021

7

8

Changing Θ2



Oregon State University
Computer Graphics

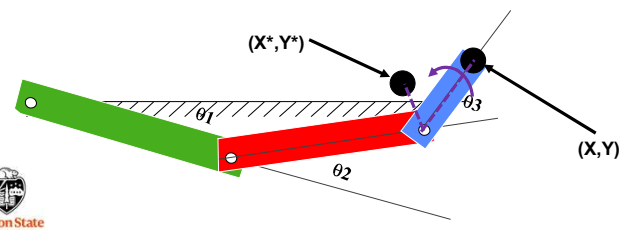
mjb – July 30, 2021

8

9

Changing Θ3

Holding Θ1 and Θ2 constant, rotate Θ3 towards (X*,Y*) so that the dashed purple lines line up.



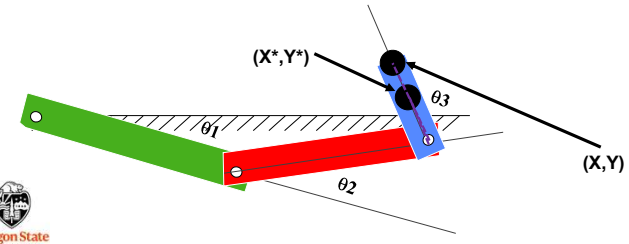
Oregon State University
Computer Graphics

mjb – July 30, 2021

9

10

Changing Θ3



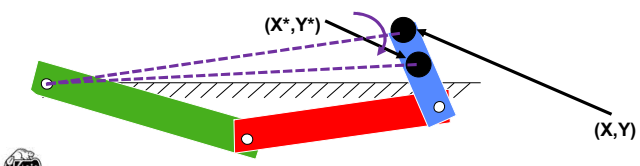
Oregon State University
Computer Graphics

mjb – July 30, 2021

10

11

Now, do it again -- Changing Θ1



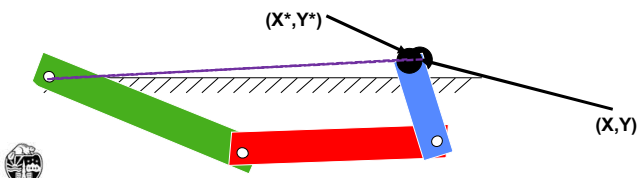
Oregon State University
Computer Graphics

mjb – July 30, 2021

11

12

Now, do it again -- Changing Θ1



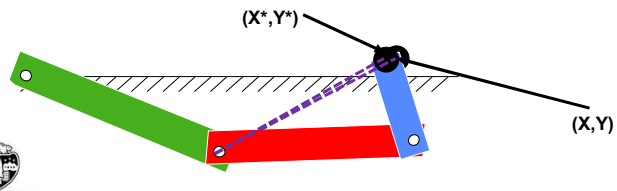
Oregon State University
Computer Graphics

mjb – July 30, 2021

12

13

Now, do it again -- Changing Θ2



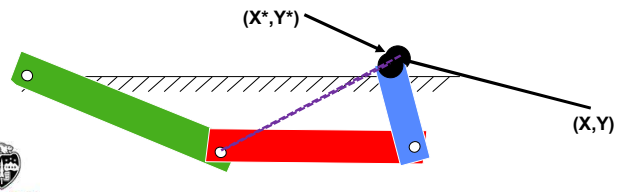
Oregon State University
Computer Graphics

mjb – July 30, 2021

13

14

Now, do it again -- Changing Θ2



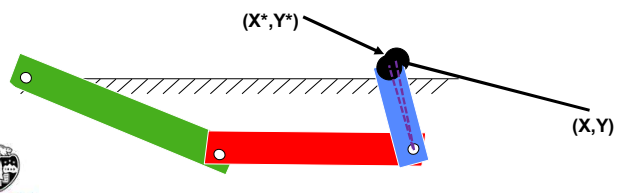
Oregon State University
Computer Graphics

mjb – July 30, 2021

14

15

Now, do it again -- Changing Θ3



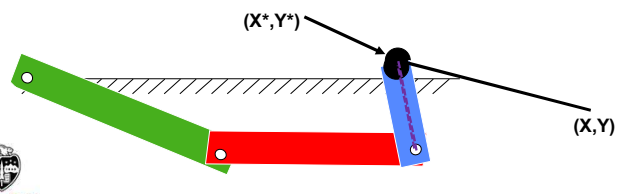
Oregon State University
Computer Graphics

mjb – July 30, 2021

15

16

Now, do it again -- Changing Θ3

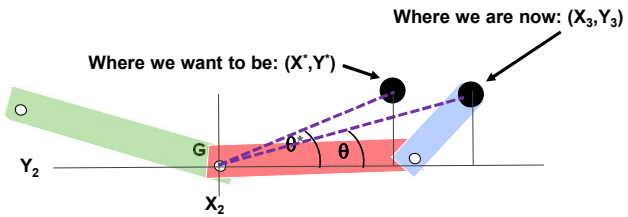


Oregon State University
Computer Graphics

mjb – July 30, 2021

16

Computing how much to change a rotation by (in this example, we are changing θ_2)



Use **the C/C++ atan2( ) function:**

$\theta^* = \text{atan2}(\ Y^*-Y_2, X^*-X_2\ );$

$\theta = \text{atan2}(\ Y_3-Y_2\ , X_3-X_2\ );$

$\Delta\theta_2 = \theta^* - \theta$

Do not use **the C/C++ atan( ) function:**

$\theta^* = \text{atan}(\ (Y^*-Y_2)\ /\ (X^*-X_2)\ );$

$\theta\ = \text{atan}(\ (Y_3-Y_2)\ /\ (X_3-X_2)\ );$

$\Delta\theta_2 = \theta^* - \theta$

Oregon State University
Computer Graphics