6.141:
Robotics systems and science
Lecture 15: Forward and Inverse Kinematics

Lecture Notes Prepared by Daniela Rus
EECS/MIT

Spring 2012

Reading: Chapter3, Craig: Robotics

http://courses.csail.mit.edu/6.141/

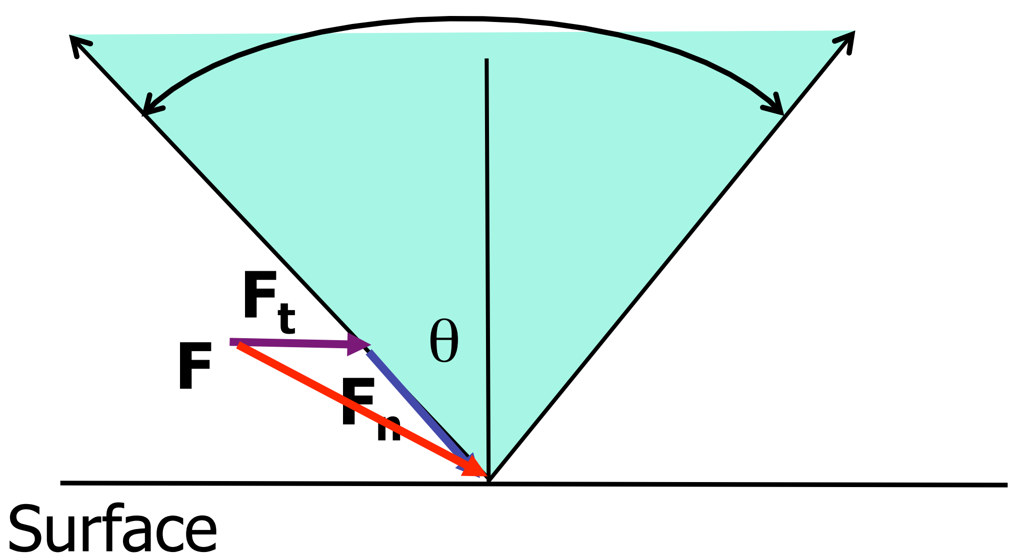
Challenge: Build a Shelter on Mars

What is coming up in RSS

- Last time: grasping
- Today in class: grasping, FK, IK
- Today in lab: Lab 7, Lab 6 presentations
- Monday: Arthur Licata on Robot Ethics

Point Contact with Friction

- Consider a point contact exerting force at some angle θ to the surface normal. What happens?

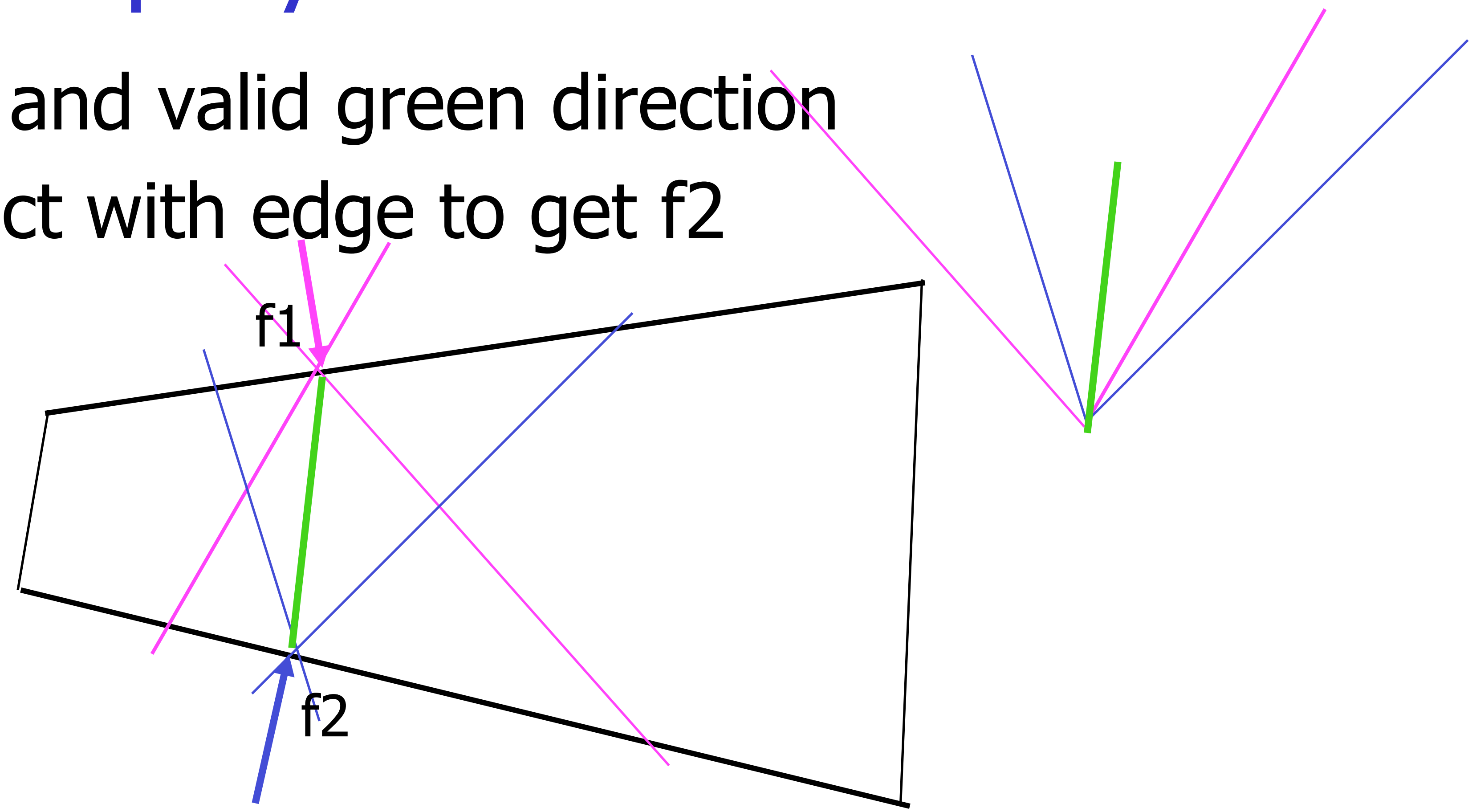


$\theta_{crit} = \tan^{-1} \mu$

- Produces a of force directions

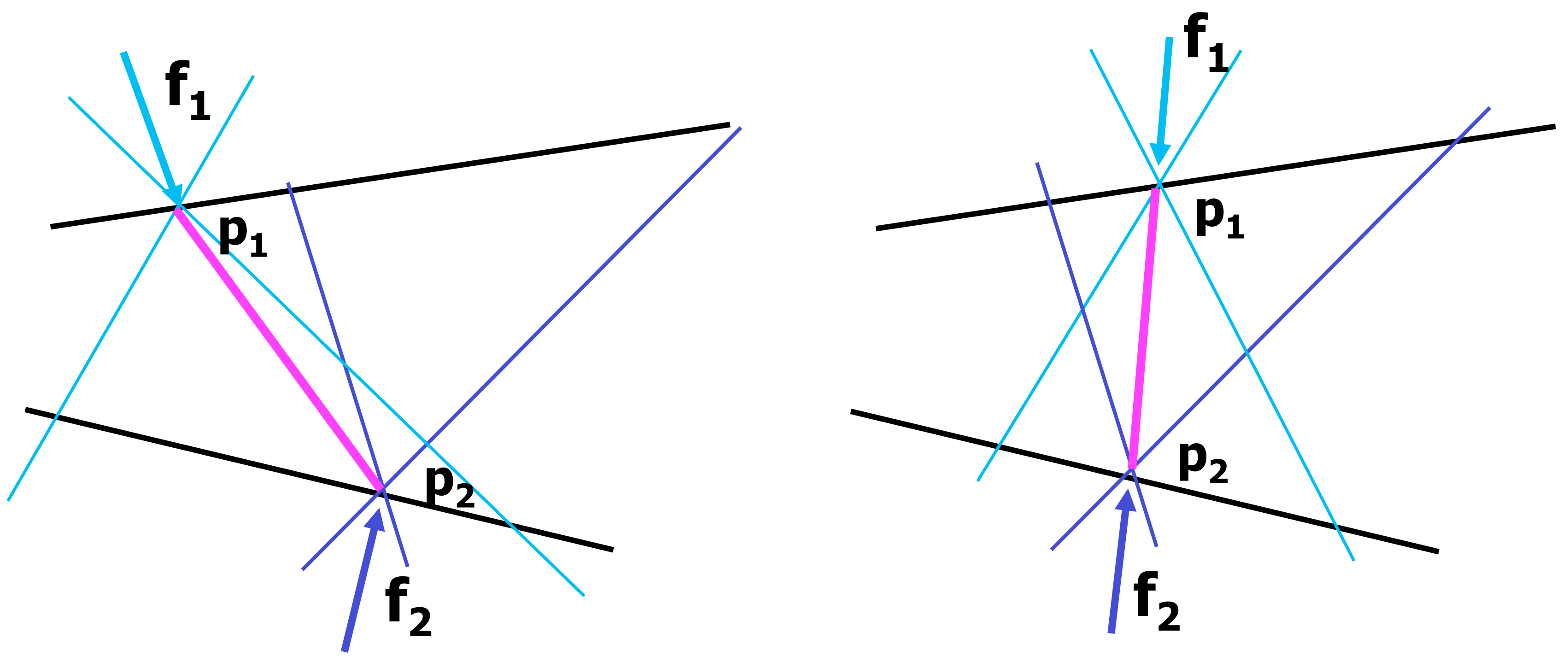
Grasp Synthesis with Friction

- Pick f1 and valid green direction
- Intersect with edge to get f2



Grasp Analysis With Friction

Consider forces $\mathbf{f}_1$, $\mathbf{f}_2$ at frictional contacts $\mathbf{p}_1$, $\mathbf{p}_2$



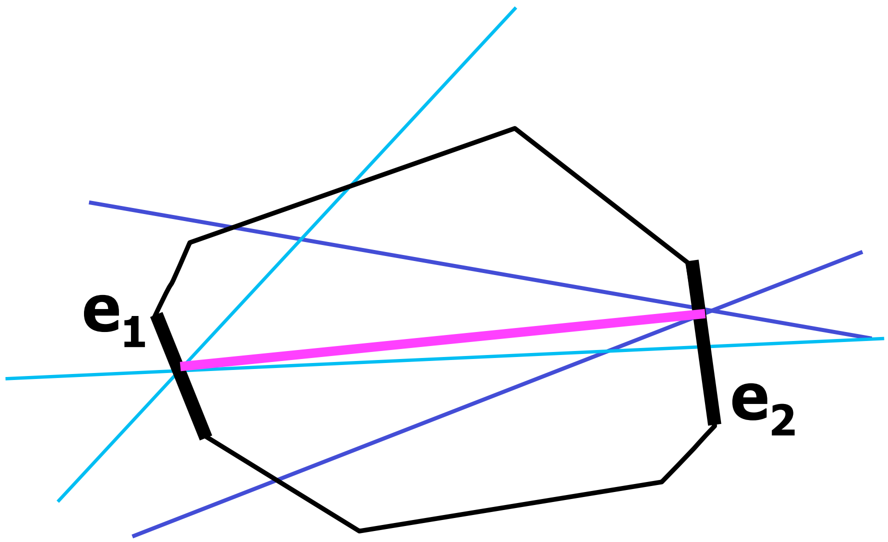
When can $\mathbf{f}_1$, $\mathbf{f}_2$ oppose one another without sliding?

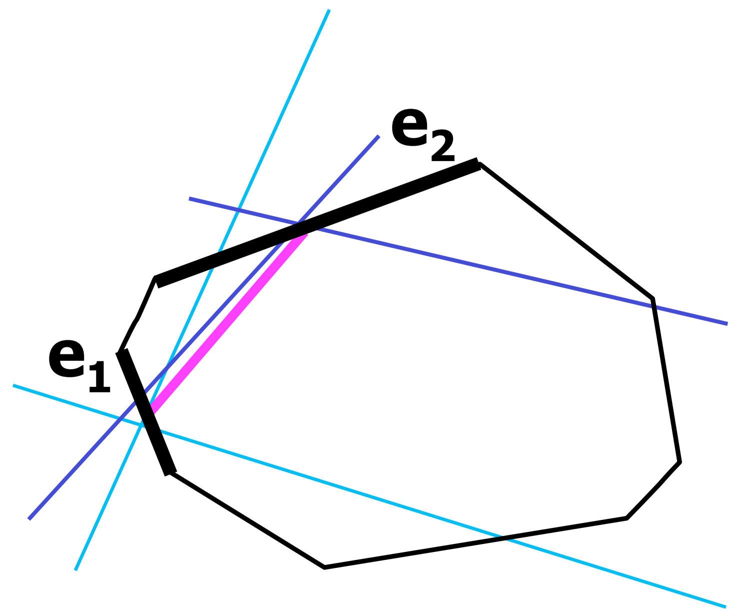
- Each force must
- Point $\mathbf{p}_1$ (resp. $\mathbf{p}_2$) must

Grasp Synthesis With Friction

Choose a *compatible* pair of edges $\mathbf{e_1}$, $\mathbf{e_2}$

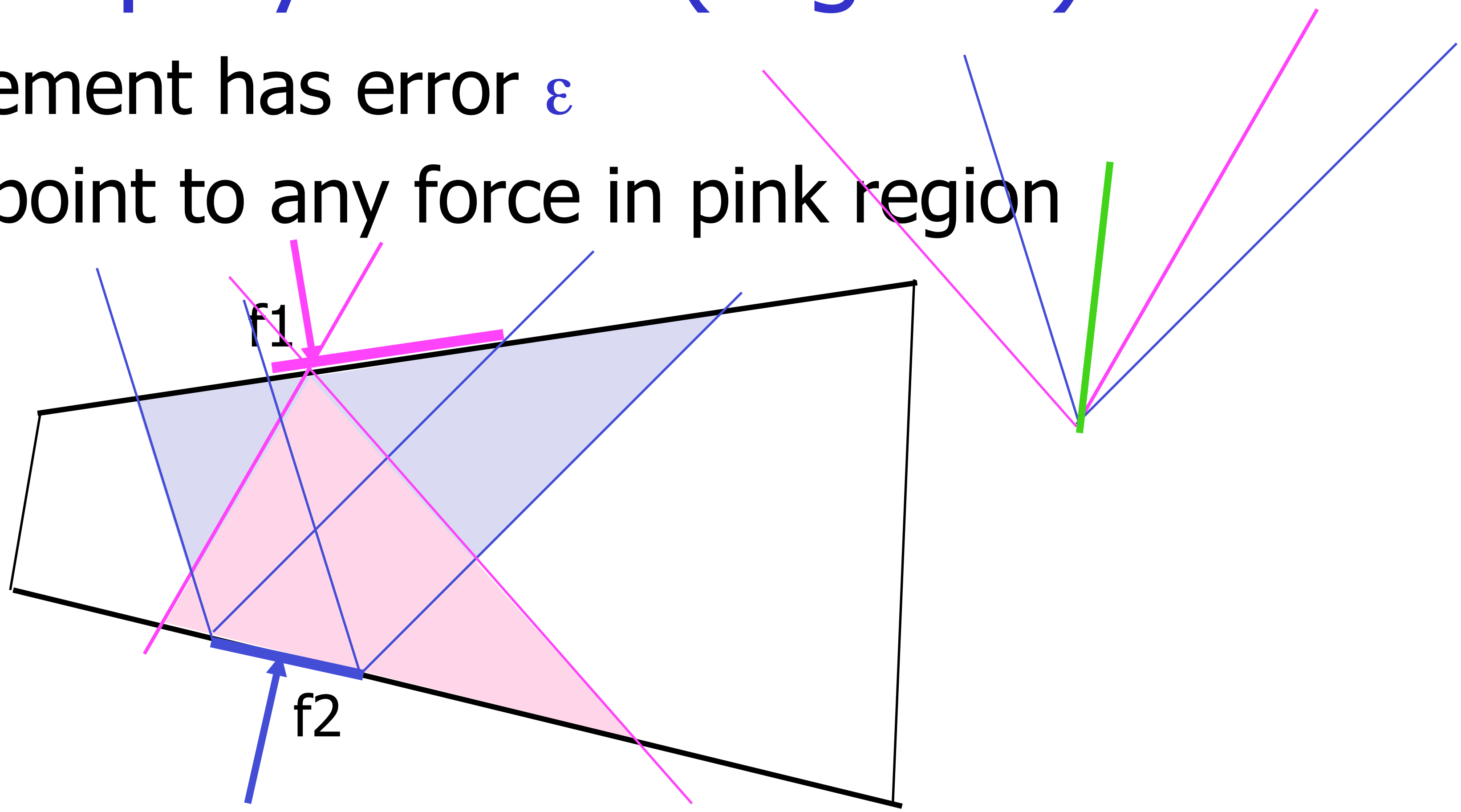
Intuition? Using what data? How to choose?





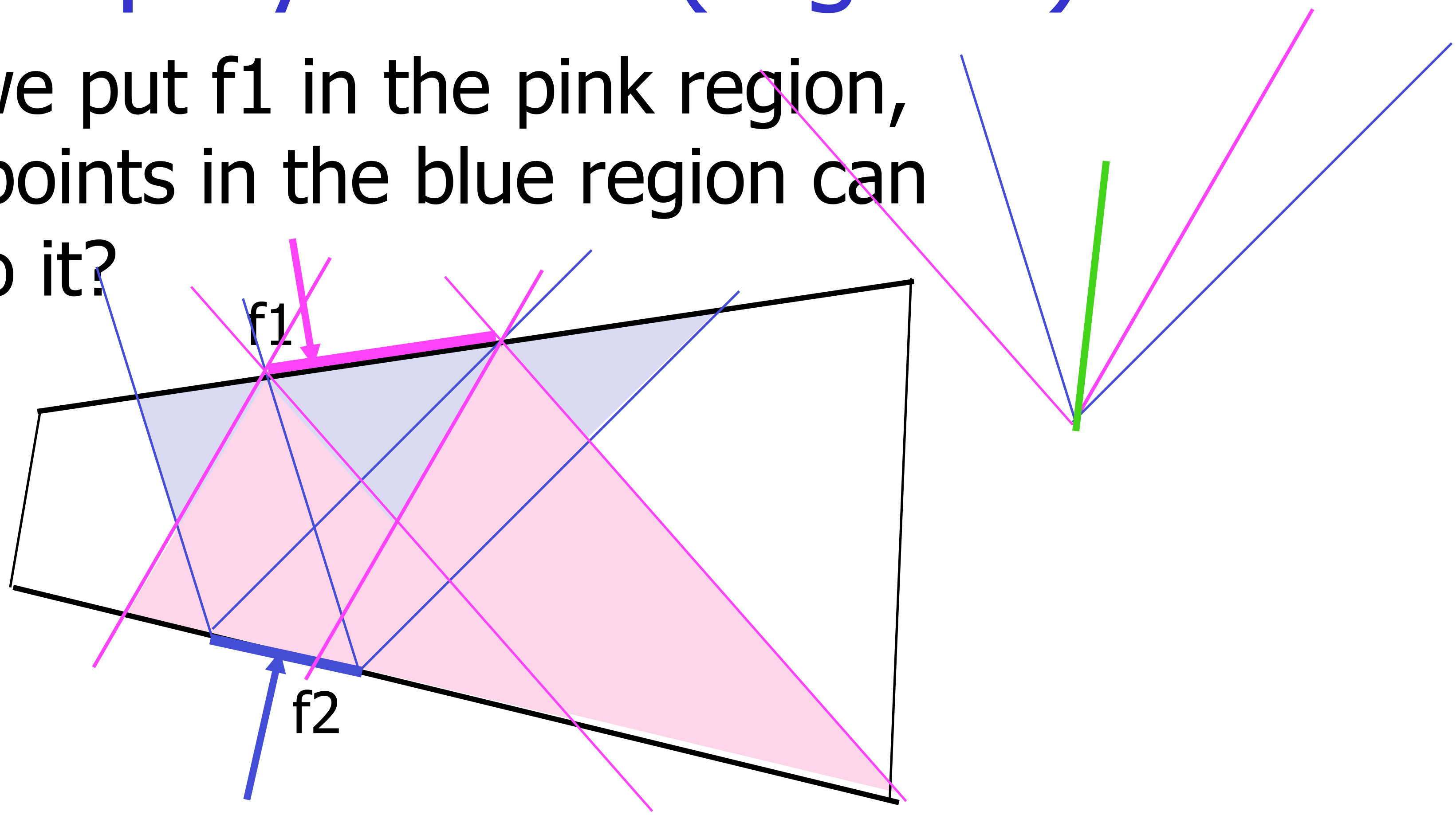
Grasp Synthesis (regions)

- f2 placement has error ε
- f2 can point to any force in pink region



Grasp Synthesis (regions)

- But if we put f1 in the pink region, which points in the blue region can point to it?



Grasp Synthesis (friction)

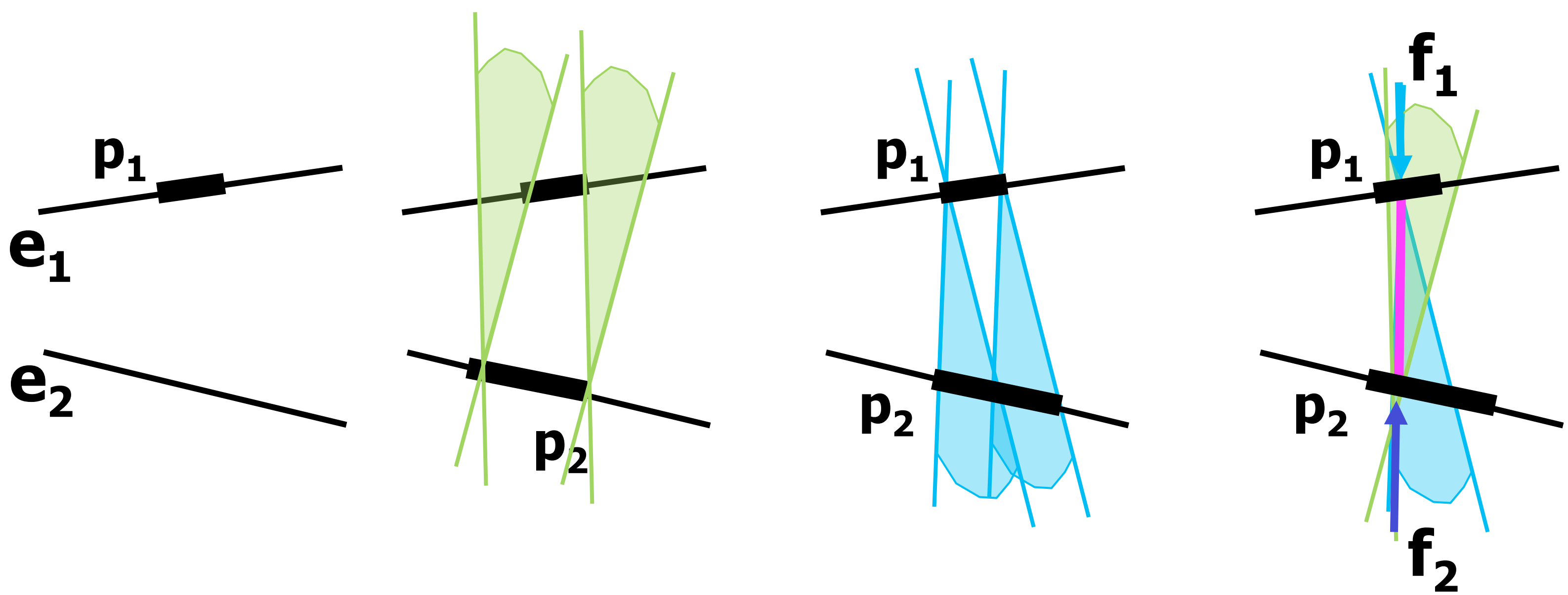
- 2 Finger Forces have to be within friction cones to stick
- 2 Finger Forces have to point at each other
- So…
- We need to find 2 edges with overlapping friction cones

Grasp Synthesis With Friction

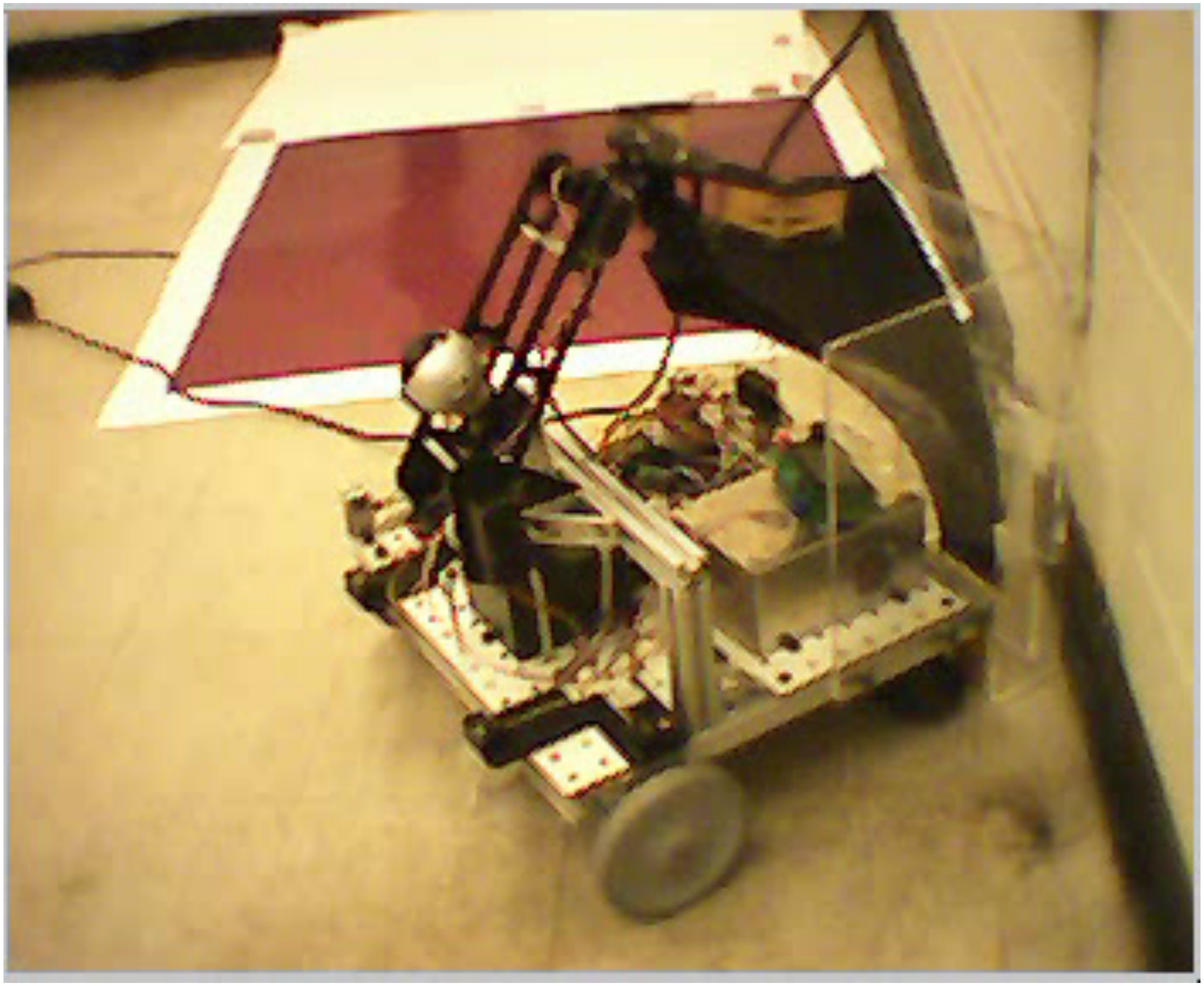
Choose target region for contact point $\mathbf{p_1}$

Determine feasible target region for contact $\mathbf{p_2}$

Orient and scale $\mathbf{f_1}$, $\mathbf{f_2}$ so as to cancel along $\overline{\mathbf{p_1 p_2}}$

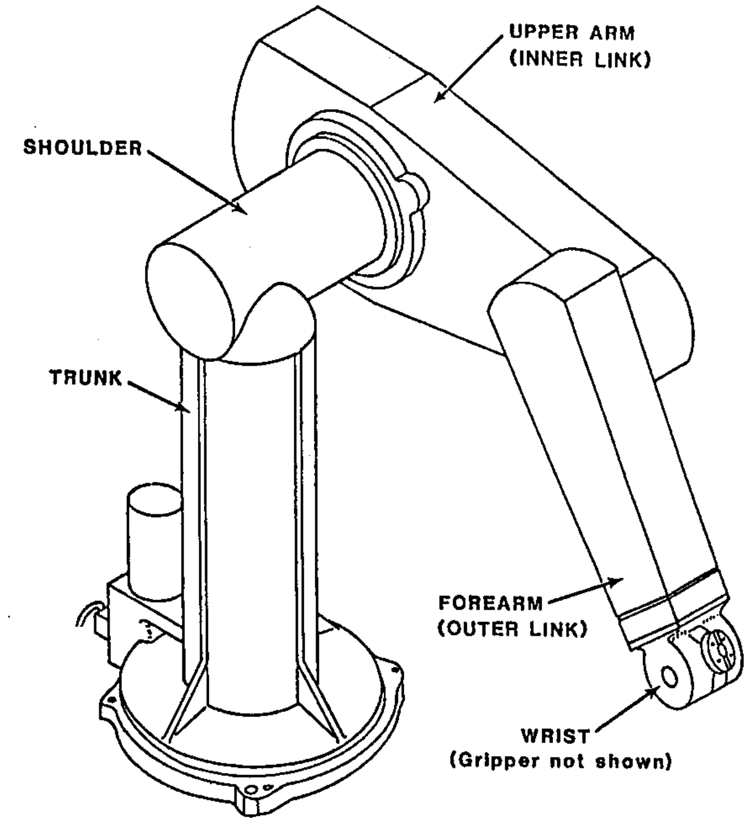


Example: 6.141 robot



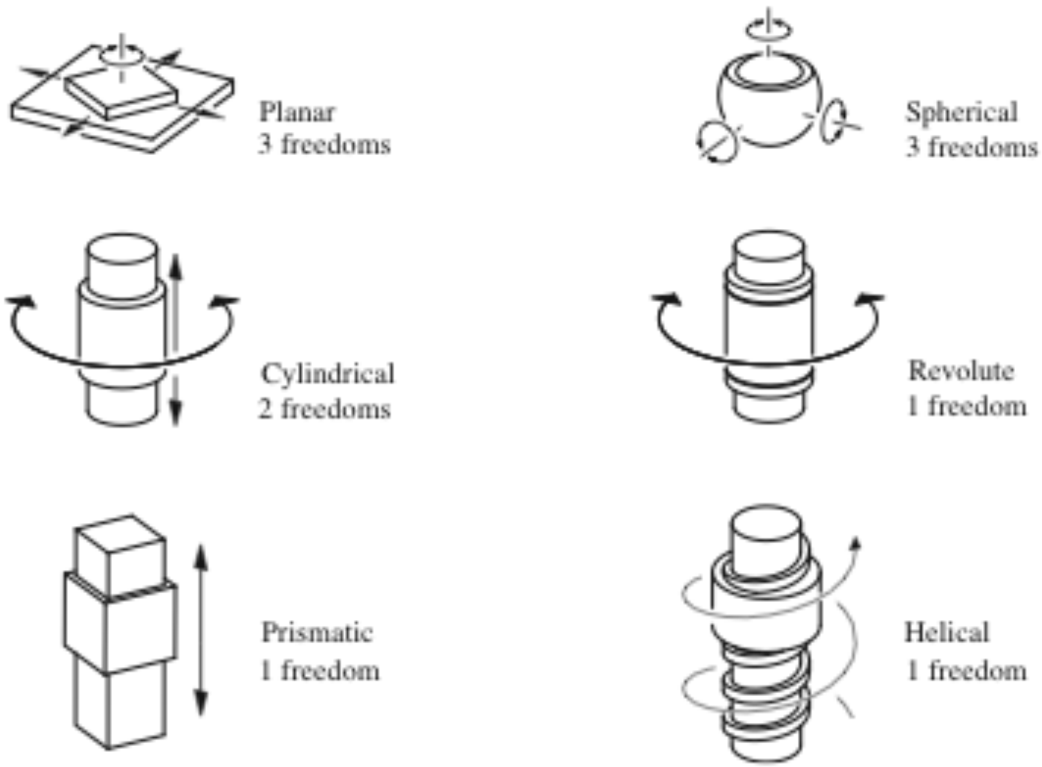
Arm Control to Reach

- Mechanism design
- Forward kinematics
- Inverse kinematics



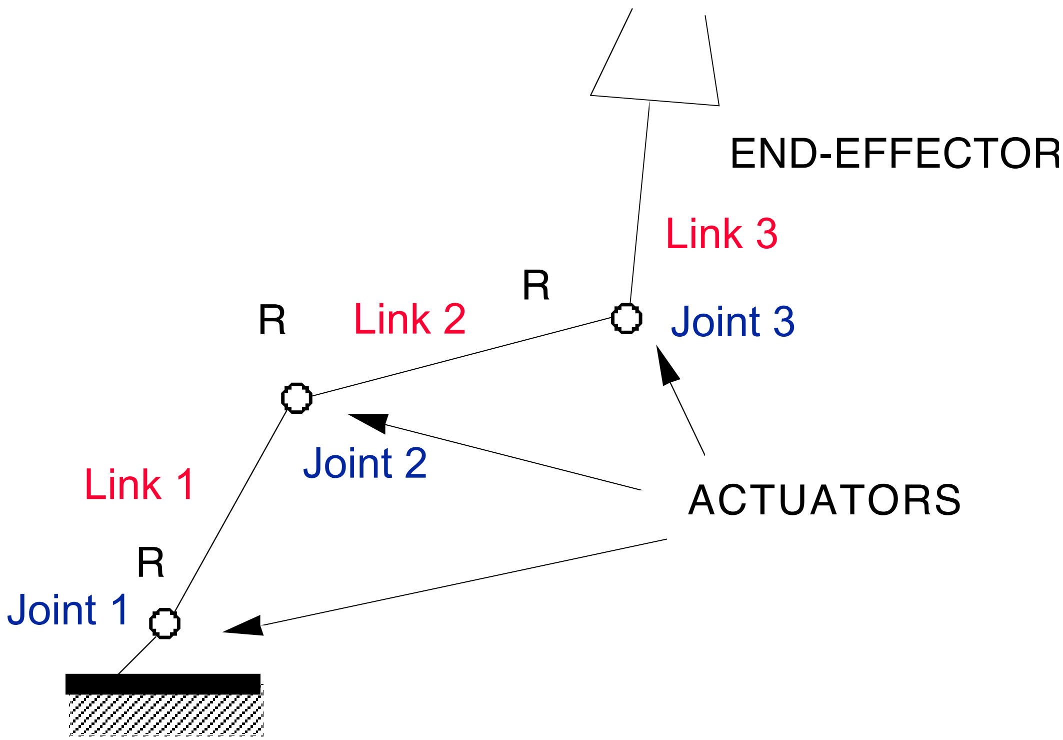
Kinematic Mechanisms

Link: rigid body
Joint: constraint
on two links
Kinematic mechanism:
links and joints



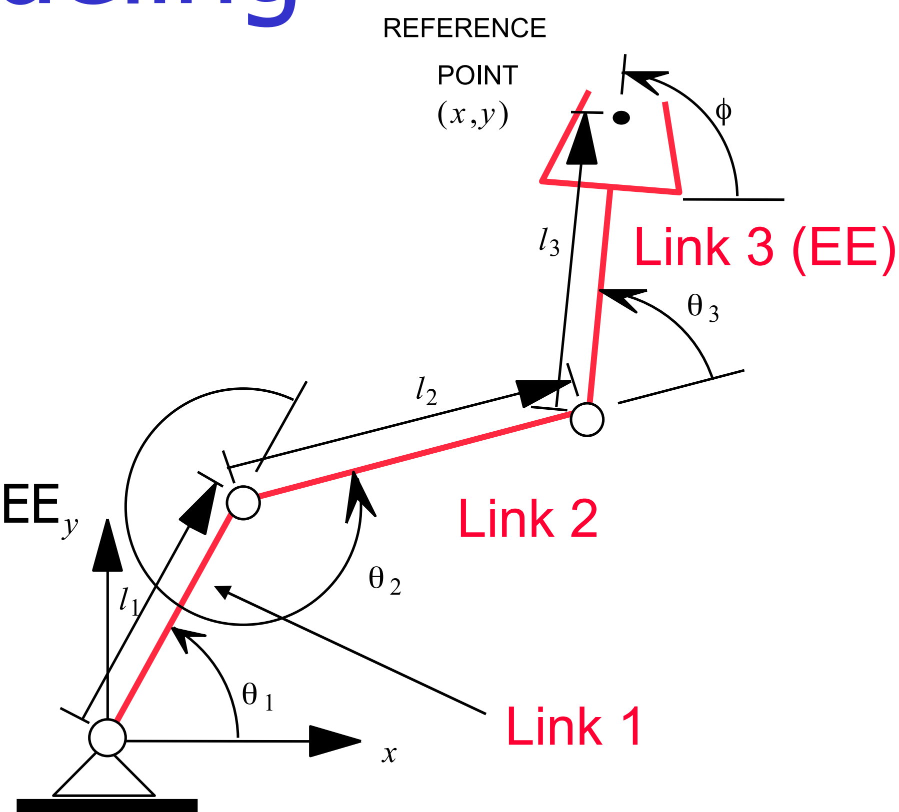
The Planar 3-R manipulator

- Planar kinematic chain
- All joints are revolute



Kinematic modeling

- Link
- Actuated joint
- End effector (EE)
  - Reference point on EE



- Joint coordinates

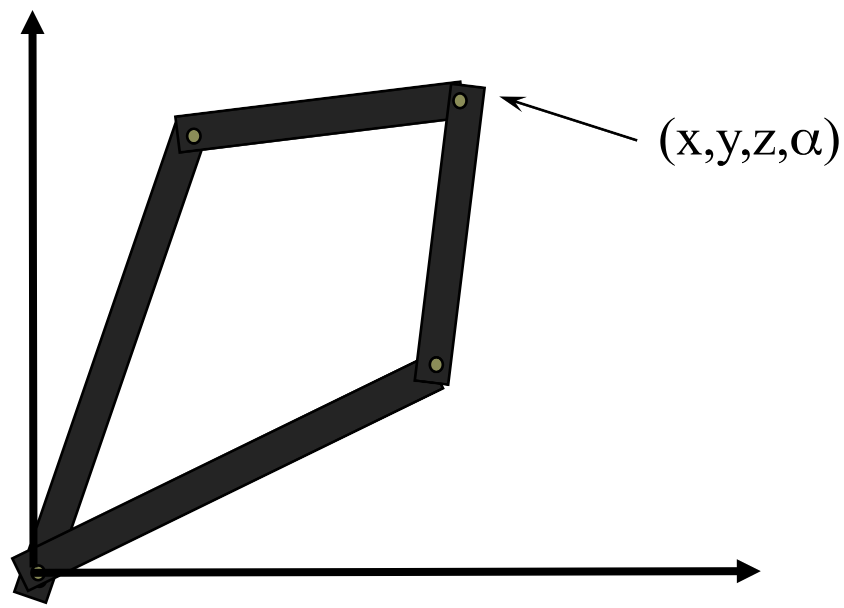
  θ_1, θ_2, θ_3
- End effector coordinates

  x, y, ϕ
- Link lengths (l_i)

Kinematic transformations

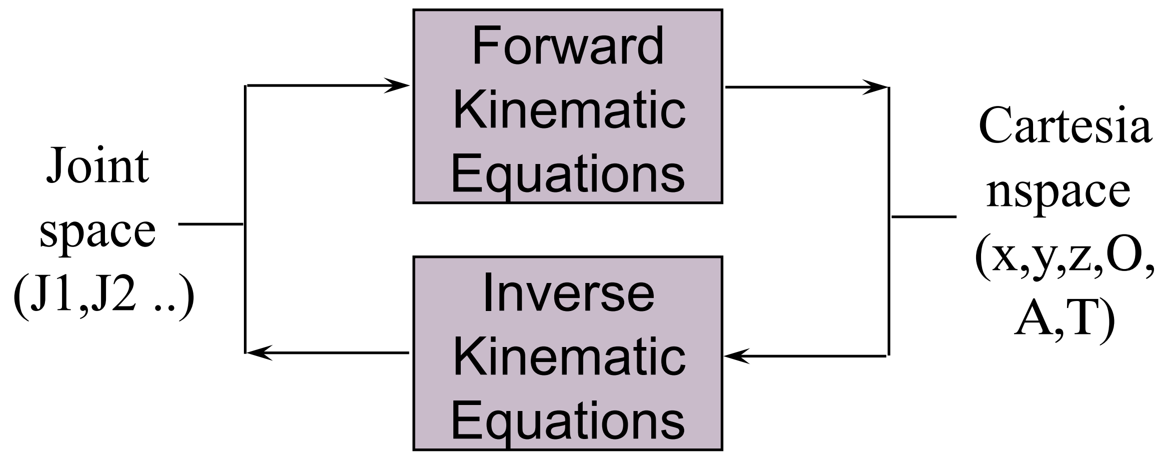
- Direct kinematics
  - Joint coordinates to end effector coordinates
    - Sensors are located at the joints. DK algorithm is used to figure out where the robot is in 3-D space.
    - Robot "thinks" in joint coordinates. Programmer/ engineer thinks in "world coordinates" or end effector coordinates.
- Inverse kinematics
  - End effector coordinates to joint coordinates
    - Given a desired position and orientation of the EE, we want to be able to get the robot to move to the desired goal. IK algorithm used to obtain the joint coordinates.
    - Essential for control.

Inverse kinematics has multiple solutions



Which is the correct robot pose ?

Kinematics Summary

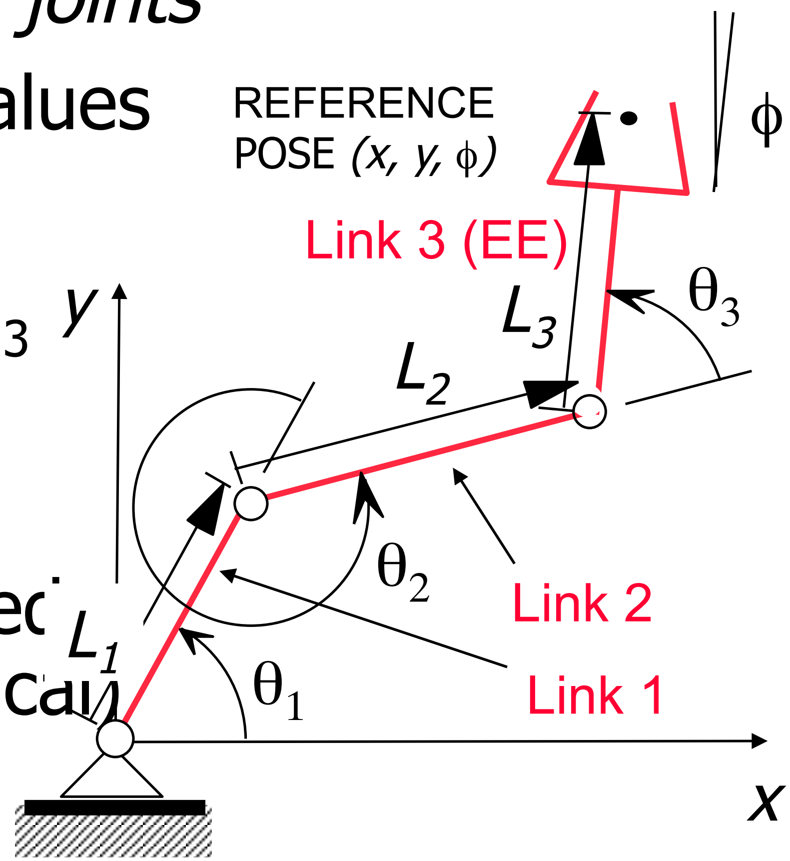


Robot kinematic calculations deal with the relationship between joint positions and an external fixed Cartesian coordinate frame.

Dynamics, force, momentum etc. are not considered.

Forward and Inverse Kinematics

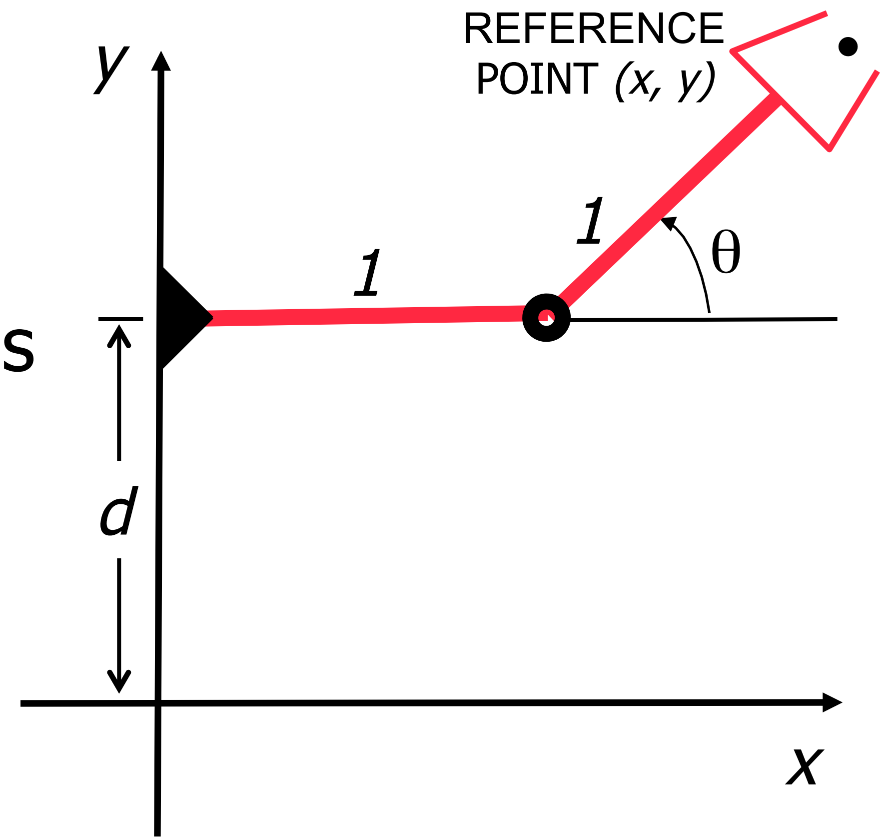
- So far, have cast computations in Cartesian space
- But manipulators controlled in *configuration space:*
  - Rigid *links* constrained by *joints*
  - For now, focus on joint values
- Example 3-link mechanism:
  - Joint coordinates θ_1, θ_2, θ_3
  - Link lengths L_1 , L_2 , L_3
- End effector coordinates
  - "Reference pose" describec by *x*, *y*, and ϕ (w.r.t. vertical)
- How can we relate EE to configuration variables?



Forward Kinematics

- Given mechanism description and joint values, express end effector pose in Cartesian coordinates
  - Example: two-link arm with one sliding, one rotating joint
- Configuration variables:
  - Joint coordinates *d*, θ
  - Link lengths (both 1)
- End effector coordinates
  - "Reference point" (*x*, y)
- Challenge: express as

$$x = x(d, \theta) =$$
$$y = y(d, \theta) =$$



Inverse Kinematics

- Given end effector pose in Cartesian coordinates, identify the joint values that yield specified pose
- Challenge: solve for joint values in terms of pose

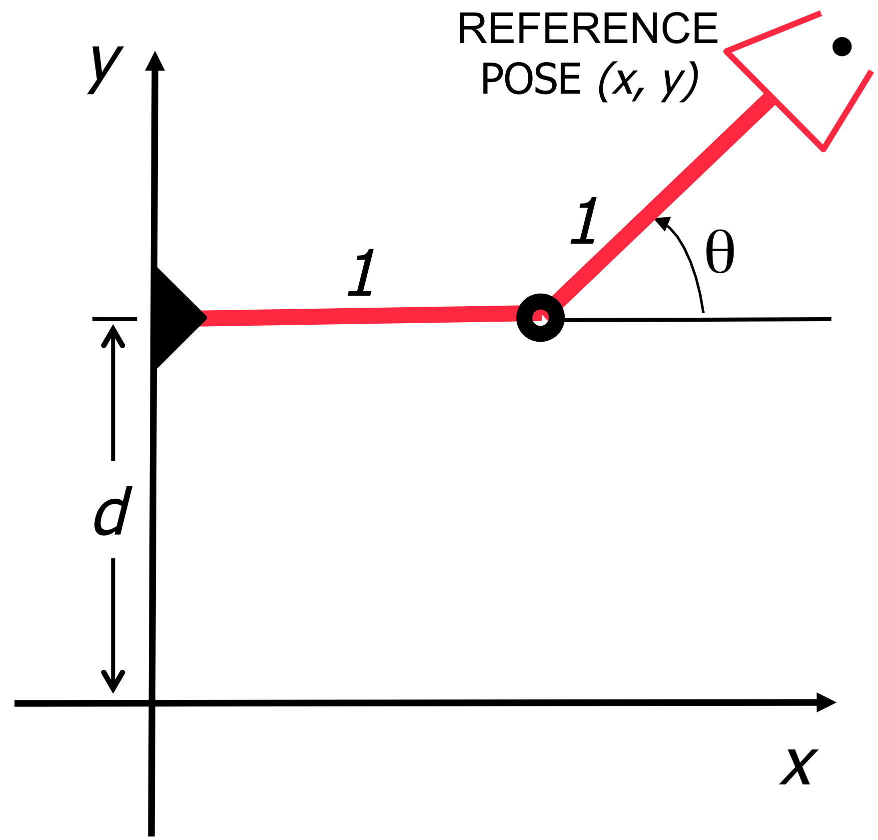
$\theta = \theta\,(x, y)$

=

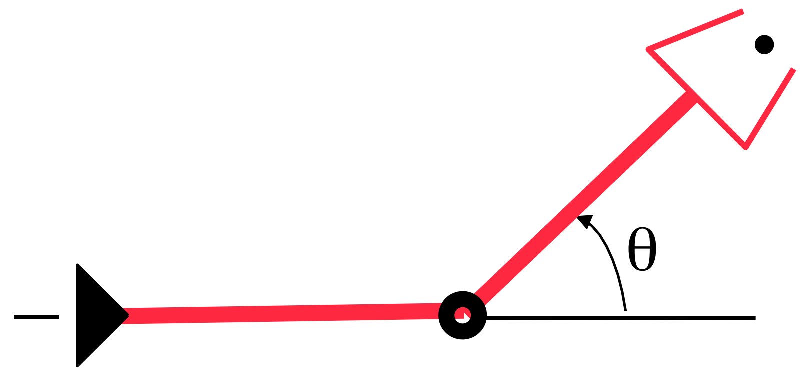
$d = d\,(x, y)$

=

Hints:

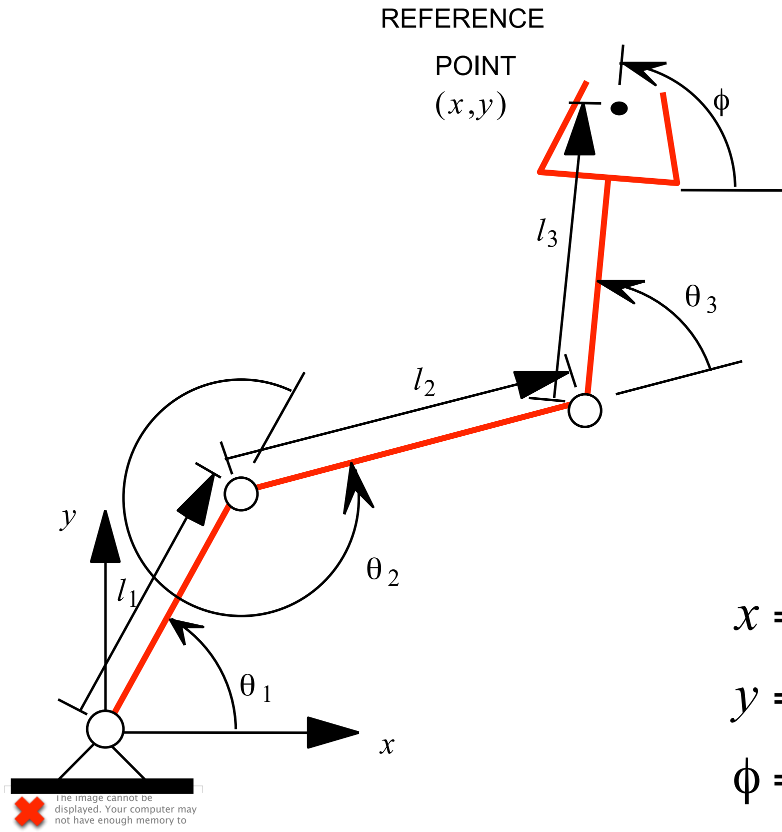


Why is IK difficult?



- Nonlinear
  - Revolute joints → inverse trigonometry
- Multi-valued
  - Often multiple solutions for a single Cartesian pose
- Discontinuities and singularities
  - Can lose one or more DOFs in some configurations
- Possibly over-constrained (no exact solution)
  - Use of approximation and iterative algorithms
- Dynamics
  - In reality, want to apply forces and torques (while respecting physical constraints), not just move arm!

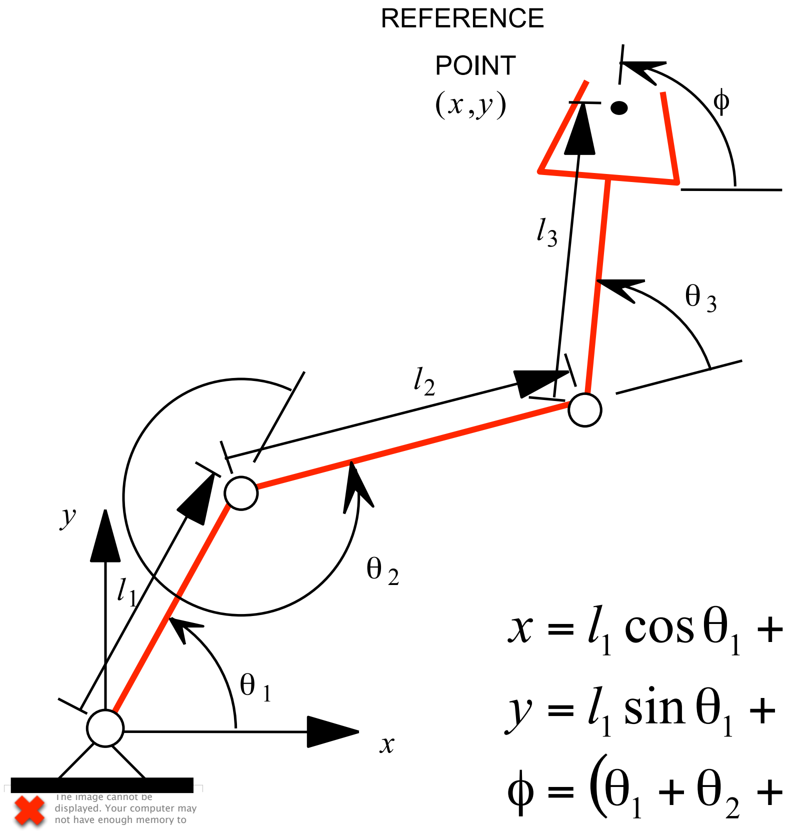
Direct kinematics



- Transform joint coordinates to end effector coordinates

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$$
$$\phi = (\theta_1 + \theta_2 + \theta_3)$$

Inverse kinematics



- Transform end effector coordinates to joint coordinates

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$$
$$\phi = (\theta_1 + \theta_2 + \theta_3)$$

Given *x*, *y*, ϕ, solve for θ_1, θ_2, θ_3

A Trigonometric Digression

$$\sin\theta = -\sin(-\theta) = -\cos(\theta + \frac{\pi}{2}) = \cos(\theta - \frac{\pi}{2})$$

$$\cos\theta = \cos(-\theta) = \sin(\theta + \frac{\pi}{2}) = -\sin(\theta - \frac{\pi}{2})$$

$$\cos^2\theta + \sin^2\theta = 1$$

$$\sin\theta = \pm\sqrt{1 - \cos^2\theta}$$

$$\sin(\alpha + \beta) = \sin\alpha \cos\beta + \cos\alpha \sin\beta$$

$$\cos(\alpha + \beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta$$

$$\sin(\alpha - \beta) = \sin\alpha \cos\beta - \cos\alpha \sin\beta$$

$$\cos(\alpha - \beta) = \cos\alpha \cos\beta + \sin\alpha \sin\beta$$

$\cos\theta = (1-t^2) / (1+t^2)$

$\sin\theta = 2t / (1 + t^2)$

Arctangents

By definition $\tan\theta = \dfrac{\sin\theta}{\cos\theta}$ **, but tan has a period of π, whereas**

sin and cos have a period of 2π. If it is known that

$$x = r\cos\theta,\ \ y = r\sin\theta,\ \ r > 0$$

then recover with

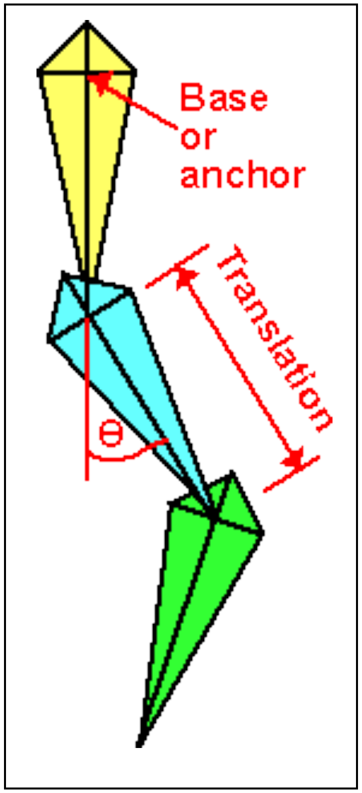
$$\theta = \arctan\left(\frac{y}{x}\right) = \arctan\left(\frac{r\sin\theta}{r\cos\theta}\right)$$

But, this only gives solution in first and fourth quadrants. The signs of x and y actually uniquely determine the quadrant, so use: $\theta \leftarrow \mathrm{atan2}(y,x)$

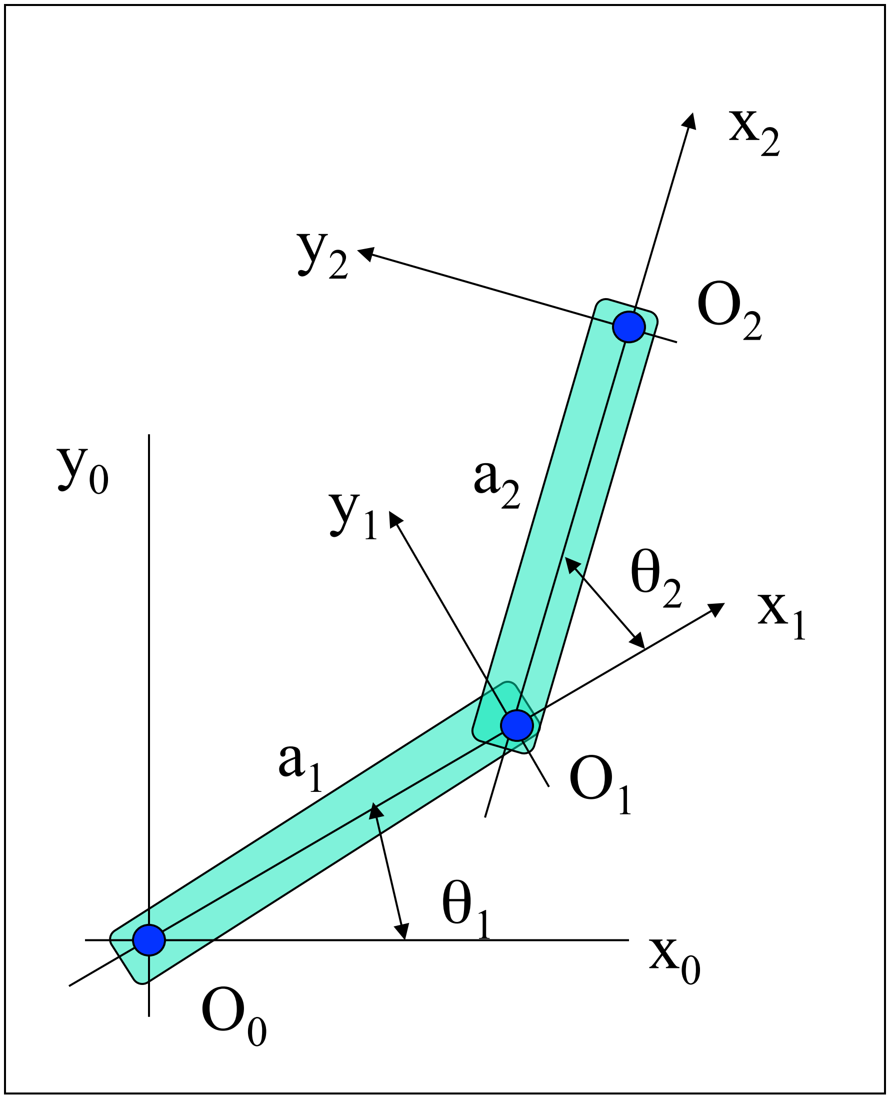
and then $\tan\theta = \dfrac{y}{x},\ \ \sin\theta = \dfrac{y}{\sqrt{x^2+y^2}},\ \ \cos\theta = \dfrac{x}{\sqrt{x^2+y^2}}$

Forward Kinematics

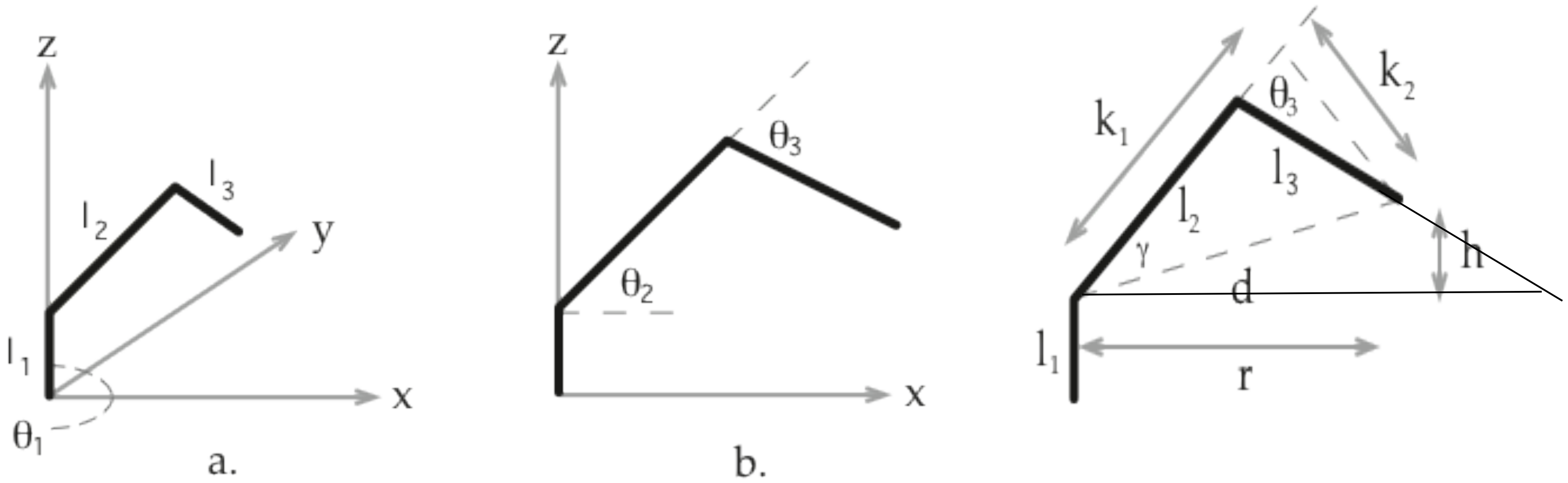
- Traverse kinematic tree and propagate transformations downward
  - Use stack
  - Compose parent transformation with child's
  - Pop stack when leaf is reached
- High DOF models are tedious to control this way



Planar Example



Example: 3DOF Revolute Arm



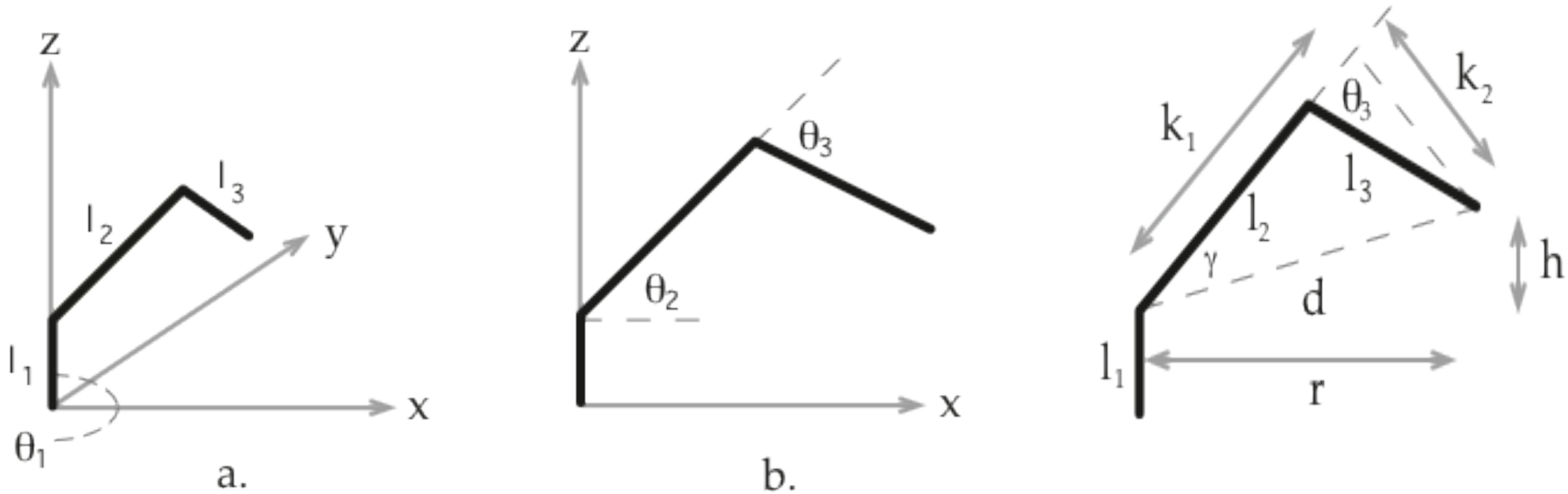
Three revolute joints. Write s_1 **and** c_1 **for** $\sin\theta_1$ **and** $\cos\theta_1$ **respectively.**

Downward elbow angle is $\theta_{3-2} = \theta_3 - \theta_2$ **write:**

$$s_{3-2} = \sin\theta_{3-2} = \sin(\theta_3 - \theta_2)$$

$$c_{3-2} = \cos\theta_{3-2} = \cos(\theta_3 - \theta_2)$$

Forward Kinematics



$$r = \sqrt{x^2 + y^2} = l_2 c_2 + l_3 c_{3-2}$$

$$h = z - l_1 = l_2 s_2 - l_3 s_{3-2}$$

$$x = [l_2 c_2 + l_3 c_{3-2}] c_1$$

$$y = [l_2 c_2 + l_3 c_{3-2}] s_1$$

$$z = l_1 + l_2 s_2 - l_3 s_{3-2}$$

Solution for Planar Example

$$x^2 + y^2 = a_1^2 + a_2^2 - 2a_1a_2\cos(\pi - \theta_2)$$

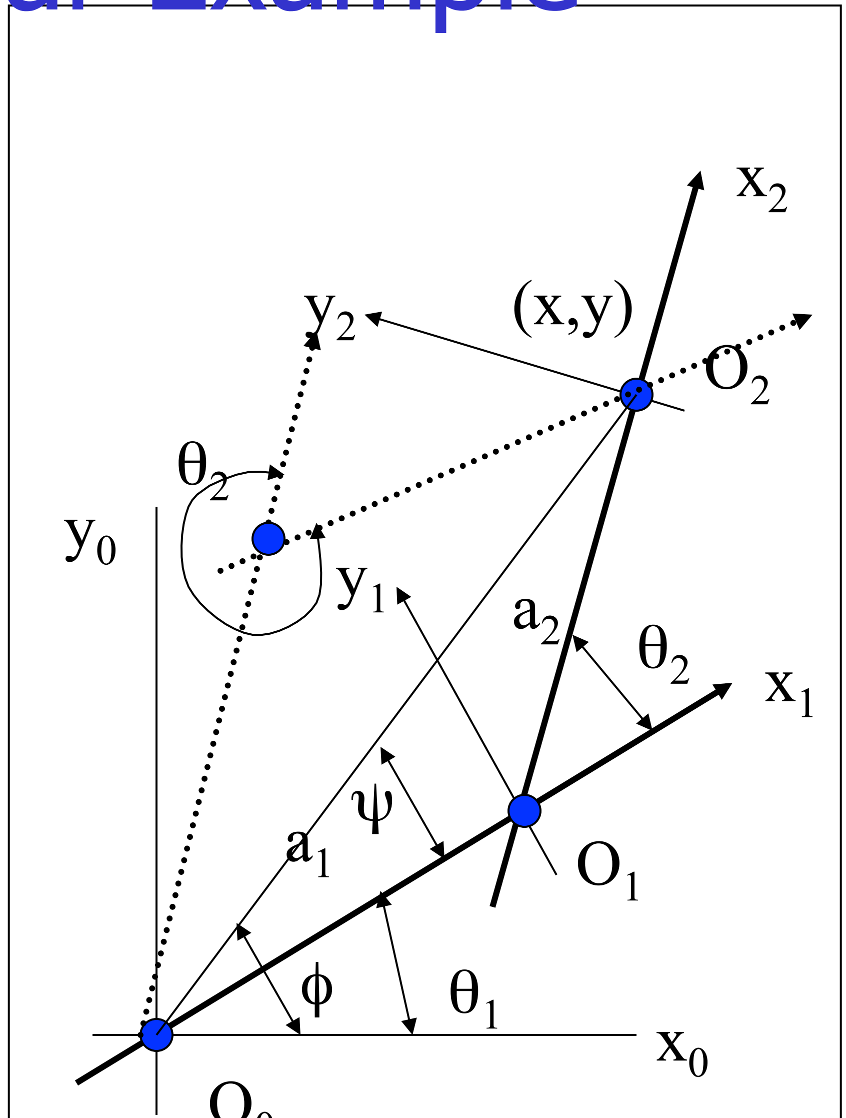
$$\cos\theta_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

for greater accuracy

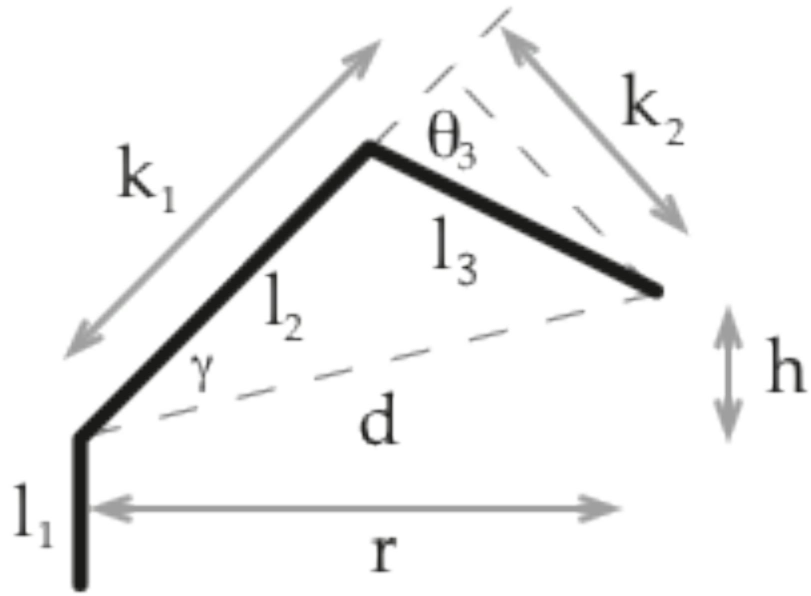
$$\tan^2\frac{\theta_2}{2} = \frac{1-\cos\theta}{1+\cos\theta} = \frac{2a_1a_2 - x^2 - y^2 + a_1^2 + a_2^2}{2a_1a_2 + x^2 + y^2 - a_1^2 - a_2^2}$$

$$= \frac{\left(a_1^2 + a_2^2\right)^2 - \left(x^2 + y^2\right)}{\left(x^2 + y^2\right) - \left(a_1^2 - a_2^2\right)^2}$$

$$\theta_2 = \pm 2\tan^{-1}\sqrt{\frac{\left(a_1^2 + a_2^2\right)^2 - \left(x^2 + y^2\right)}{\left(x^2 + y^2\right) - \left(a_1^2 - a_2^2\right)^2}}$$



Inverse Kinematics of 3DOF Manipulator



$$r = \sqrt{x^2 + y^2} = l_2 c_2 + l_3 c_{3-2}$$

$$h = z - l_1 = l_2 s_2 - l_3 s_{3-2}$$

$$x^2 + y^2 + (z - l_1)^2 = r^2 + h^2$$

$$= l_2^2 c_2^2 + l_3^2 c_{3-2}^2 + 2 l_2 l_3 c_2 [c_3 c_2 + s_3 s_2] + l_2^2 s_2^2 + l_3^2 s_{3-2}^2 - 2 l_2 l_3 s_2 [s_3 c_2 - c_3 s_2]$$

$$= l_2^2 + l_3^2 + 2 l_2 l_3 [c_2^2 c_3 + s_2^2 c_3]$$

$$= l_2^2 + l_3^2 + 2 l_2 l_3 c_3$$

$$c_3 = \frac{x^2 + y^2 + (z - l_1)^2 - l_2^2 - l_3^2}{2 l_2 l_3}$$

Solving for the Elbow

Can choose an "up elbow" solution with $s_3 = +\sqrt{1 - c_3^2}$

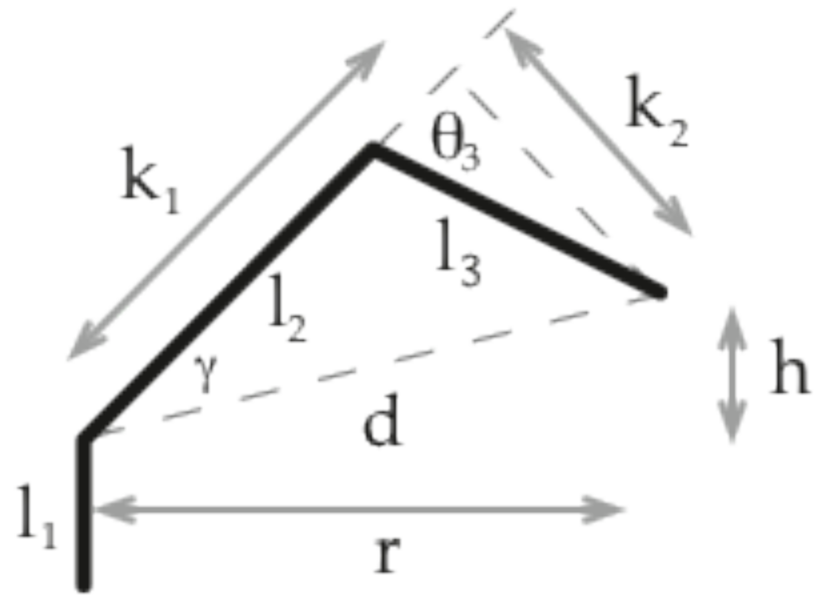
or a "down elbow" solution with $s_3 = -\sqrt{1 - c_3^2}$

where $c_3 = \dfrac{x^2 + y^2 + (z - l_1)^2 - l_2^2 - l_3^2}{2 l_2 l_3}$

In either case can then recover θ_3 **with**

$$\theta_3 = \operatorname{atan2}(s_3, c_3)$$

Pushing On...



$$r = l_2 c_2 + l_3 c_{3-2} = l_2 c_2 + l_3 c_3 c_2 + l_3 s_3 s_2 = k_1 c_2 + k_2 s_2$$
$$h = l_2 s_2 - l_3 s_{3-2} = l_2 s_2 - l_3 s_3 c_2 + l_3 c_3 s_2 = k_1 s_2 - k_2 c_2$$

where

$$k_1 = l_2 + l_3 c_3$$
$$k_2 = l_3 s_3$$

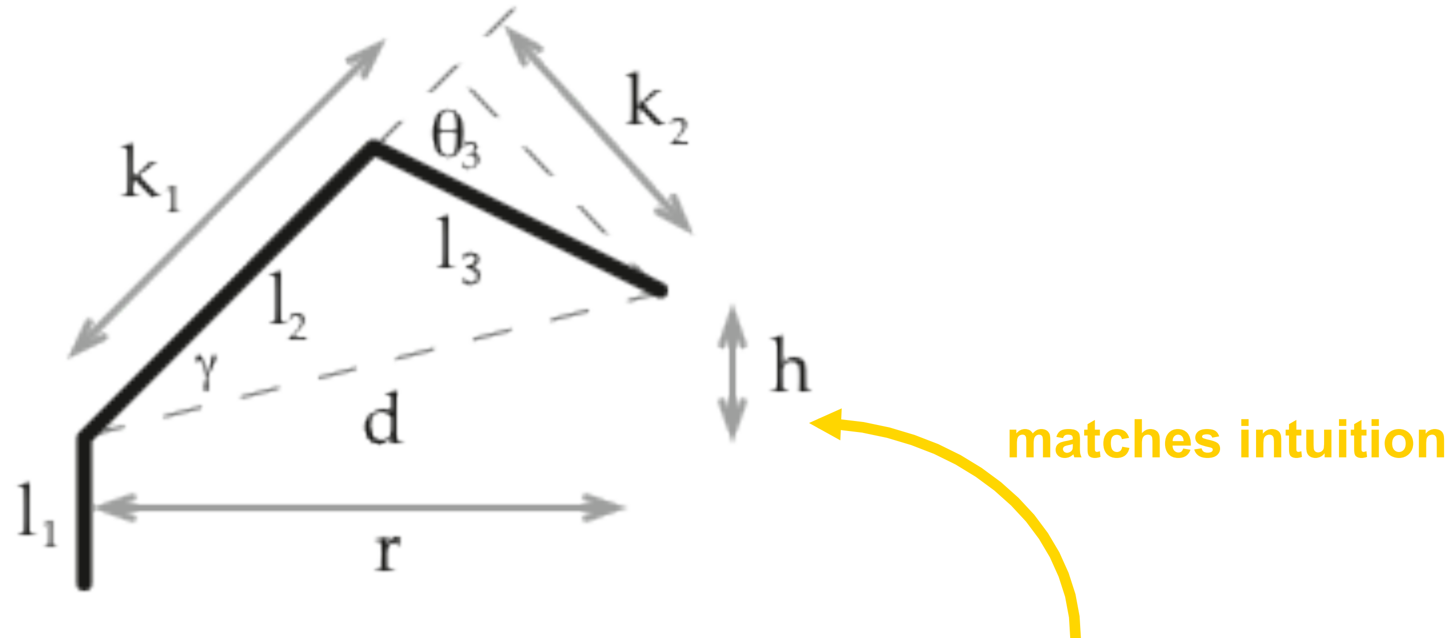
$$d = +\sqrt{k_1^2 + k_2^2}$$
$$\gamma = \text{atan2}(k_2, k_1)$$

and so

$$k_1 = d \cos\gamma$$
$$k_2 = d \sin\gamma$$

substitute in...

And on…



$$r = d\cos\gamma\cos\theta_2 + d\sin\gamma\sin\theta_2 = d\cos(\theta_2 - \gamma)$$

$$h = d\cos\gamma\sin\theta_2 - d\sin\gamma\cos\theta_2 = d\sin(\theta_2 - \gamma)$$

$$\theta_2 - \gamma = \mathrm{atan2}(\frac{h}{d}, \frac{r}{d}) = \mathrm{atan2}(h, r)$$ **since** $d > 0$

Final Inverse Kinematics

$$\theta_1 \leftarrow \operatorname{atan2}(y, x)$$

$$c_3 \leftarrow \frac{x^2 + y^2 + (z - l_1)^2 - l_2^2 - l_3^2}{2 l_2 l_3}$$

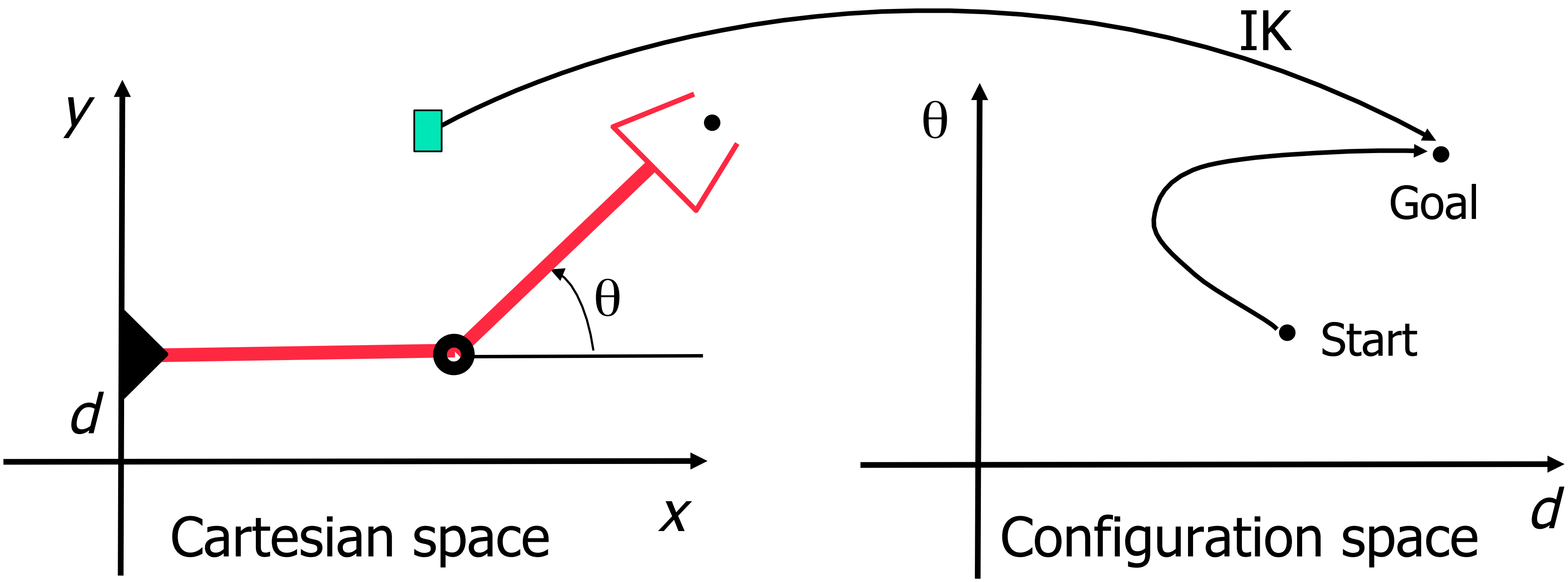
$$s_3 \leftarrow +\sqrt{1 - c_3^2}$$

$$\theta_3 \leftarrow \operatorname{atan2}(s_3, c_3)$$

$$\theta_2 \leftarrow \operatorname{atan2}(z - l_1, \sqrt{x^2 + y^2}) - \operatorname{atan2}(l_3 s_3, l_2 + l_3 c_3)$$

Putting it All Together: Grasping

- Input workspace, obstacles, and manipuland:
  - Determine a feasible grasp (set of contact points)
  - Use IK to solve for target end-effector pose in c-space
  - Plan a collision-free reach to the computed pose
  - Control end-effector along desired trajectory

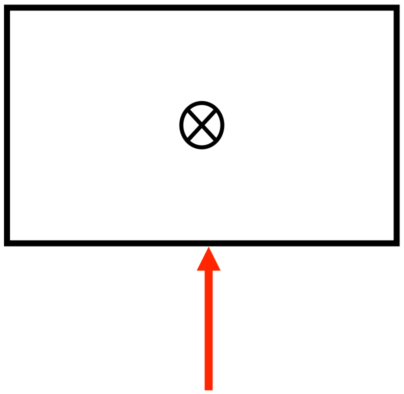


What have we swept under the rug?

- Sensing
  - Shape, pose of target object, accessibility of surfaces
  - Classification of material type from sensor data
  - Freespace through which grasping action will occur
- Prior knowledge
  - Estimate of μ, mass, moments given material type
  - Internal, articulated, even active degrees of freedom
- Uncertainty & compliance
  - Tolerate noise inherent in sensing and actuation
  - Ensure that slight sensing, actuation errors won't cause damage
  - Handle soft fingers making contact over a finite area (not a point)
- Dynamics
  - All of the above factors may be changing in real time

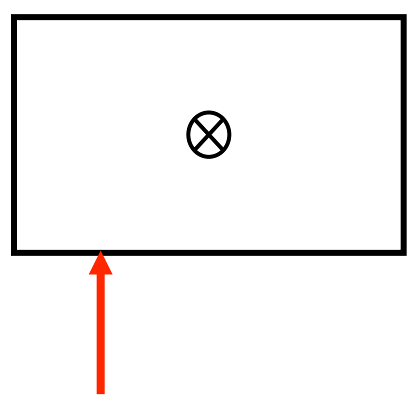
Pushing

- Straight-line motion



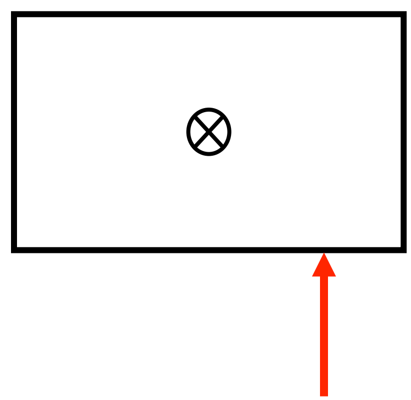
Pushing

- Clockwise rotation



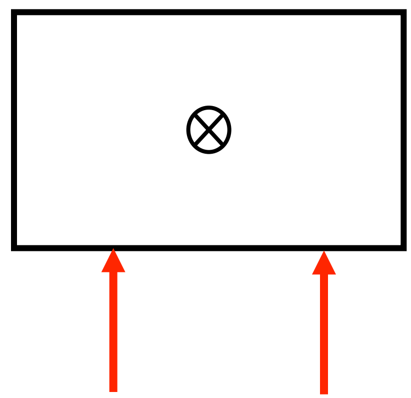
Pushing

- Counter-clockwise rotation



Pushing

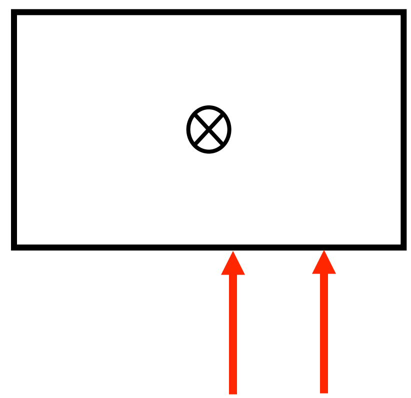
- Robust translation



What if we do not know where the center of mass is?

Pushing

- Robust translation



Push and sense: if clockwise rotation, move right
if counterclockwise rotation move left