ME 328: Medical Robotics
Winter 2019

Lecture 2:
Kinematics of medical robotics

Allison Okamura
Stanford University

kinematics

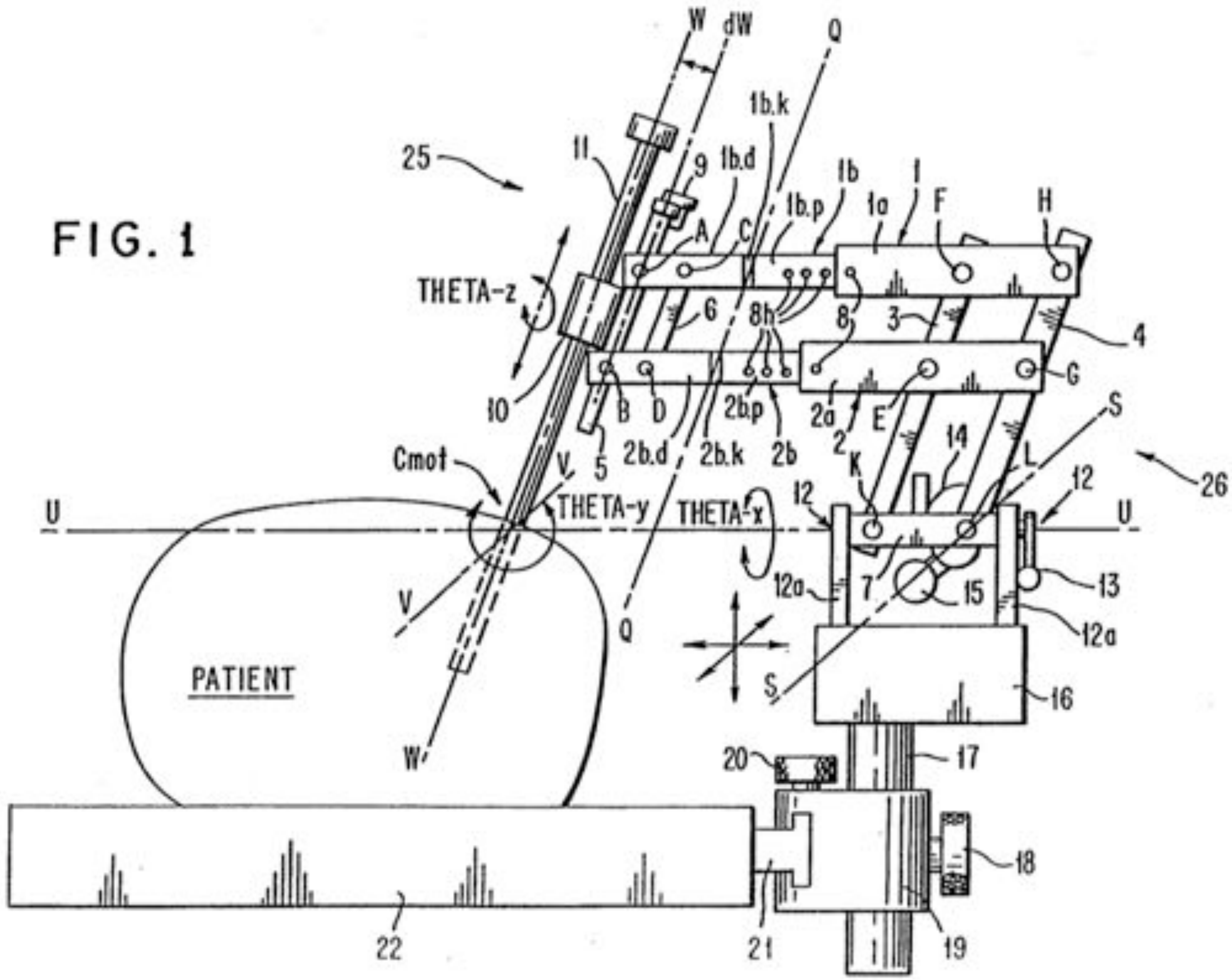
- The study of movement
- The branch of classical mechanics that describes the motion of objects without consideration of the forces that cause it
- Why do you need it?
  - Determine endpoint position and/or joint positions
  - Calculate mechanism velocities, accelerations, etc.
  - Calculate force-torque relationships

degrees of freedom

- Number of independent position variables needed to in order to locate all parts of a mechanism
- DOF of motion
- DOF of sensing
- DOF of actuation
- The DOF of a mechanism does not always correspond to number of joints

a remote center of motion (RCM) robot

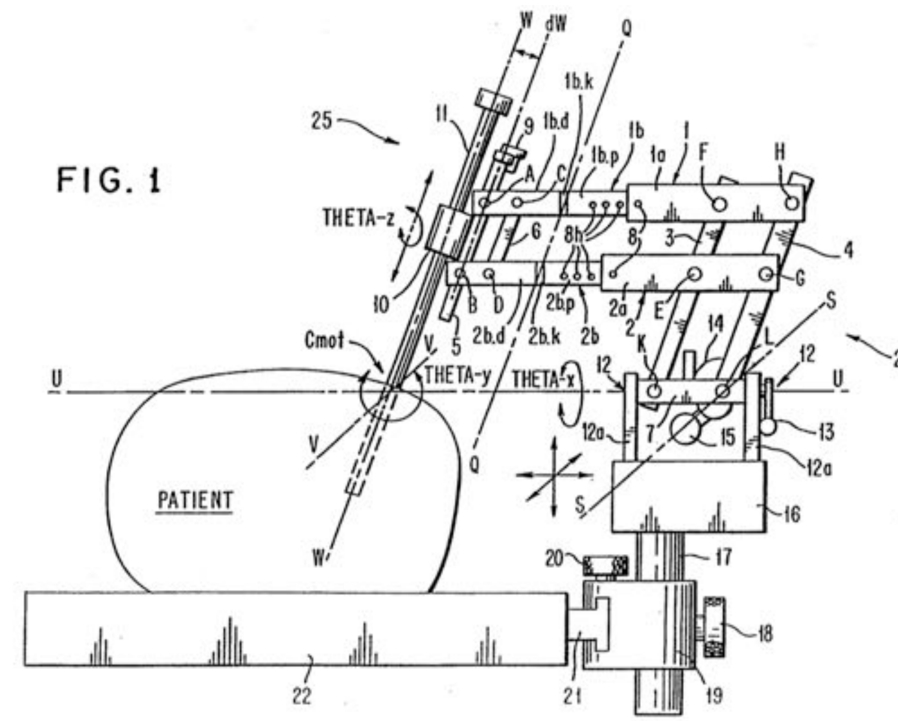
FIG. 1

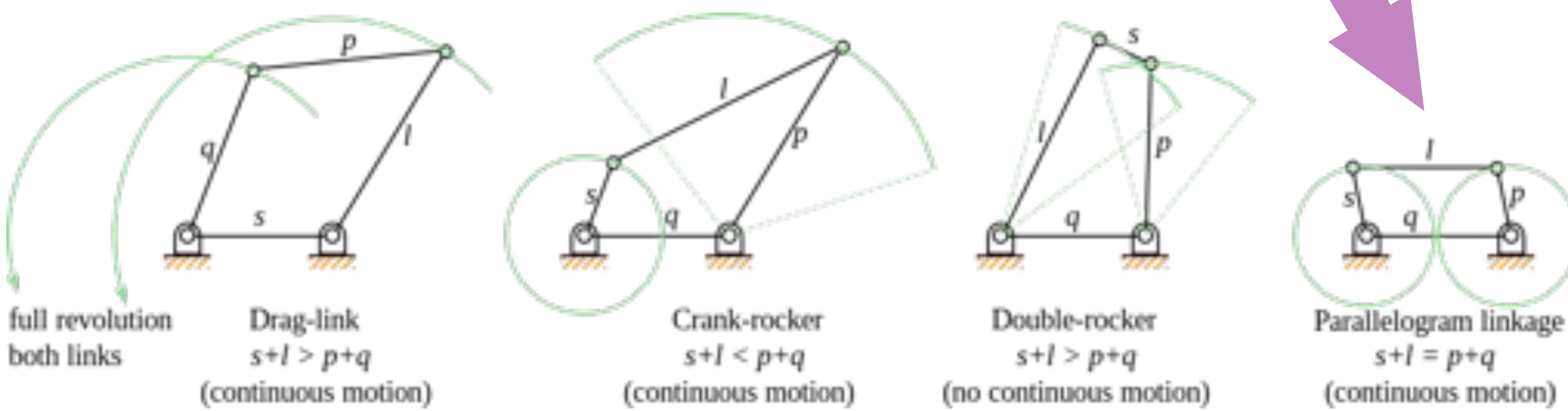


US patent 5397323 (Taylor, et al.)

four-bar linkage

- commonly used 1-DOF mechanism
- relationship between input link angle and output link angle can be computed from geometry





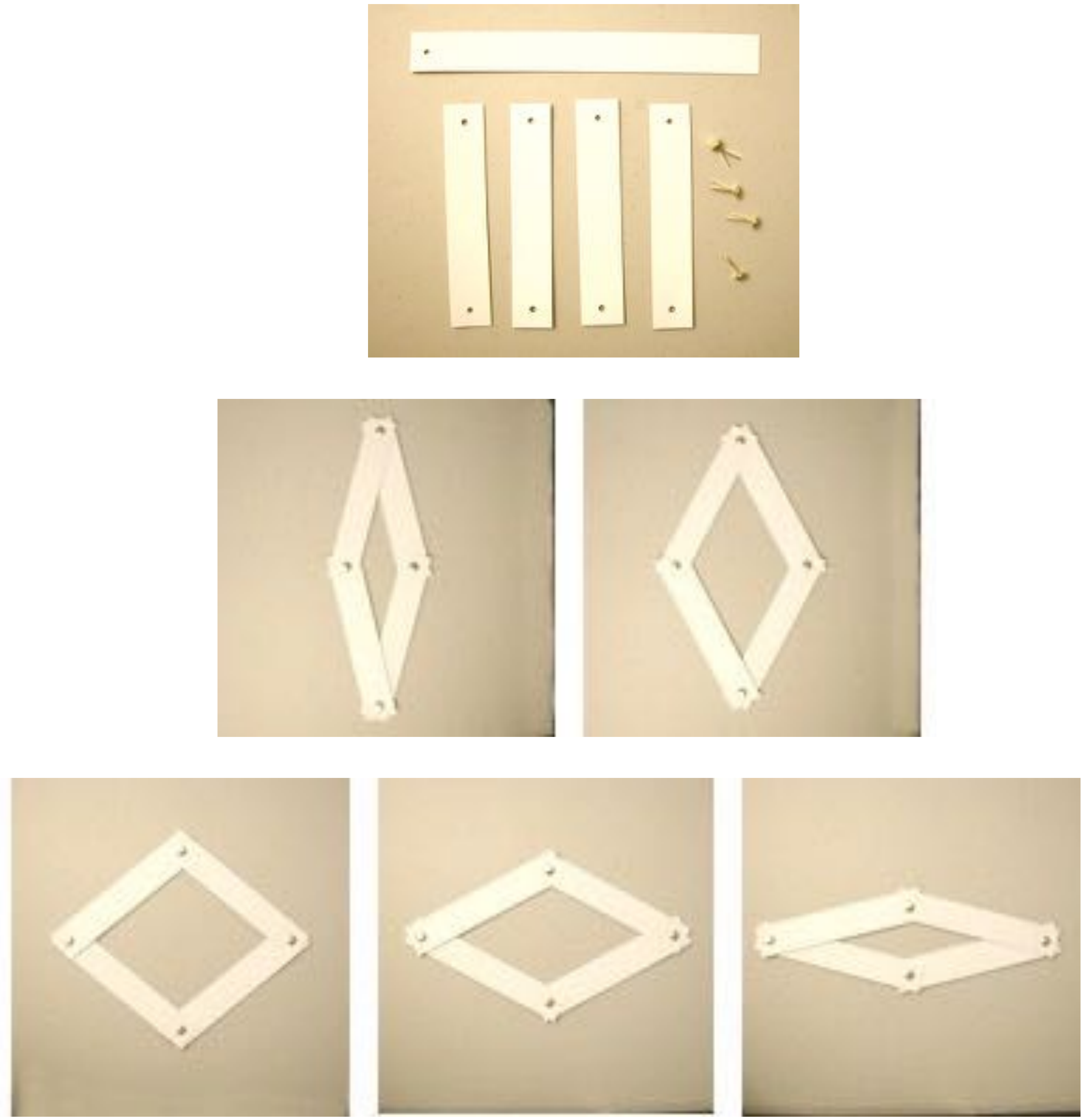
Types of four-bar linkages, *s* = shortest link, *l* = longest link

wikipedia.org

it might help to prototype

round head
paper fasteners

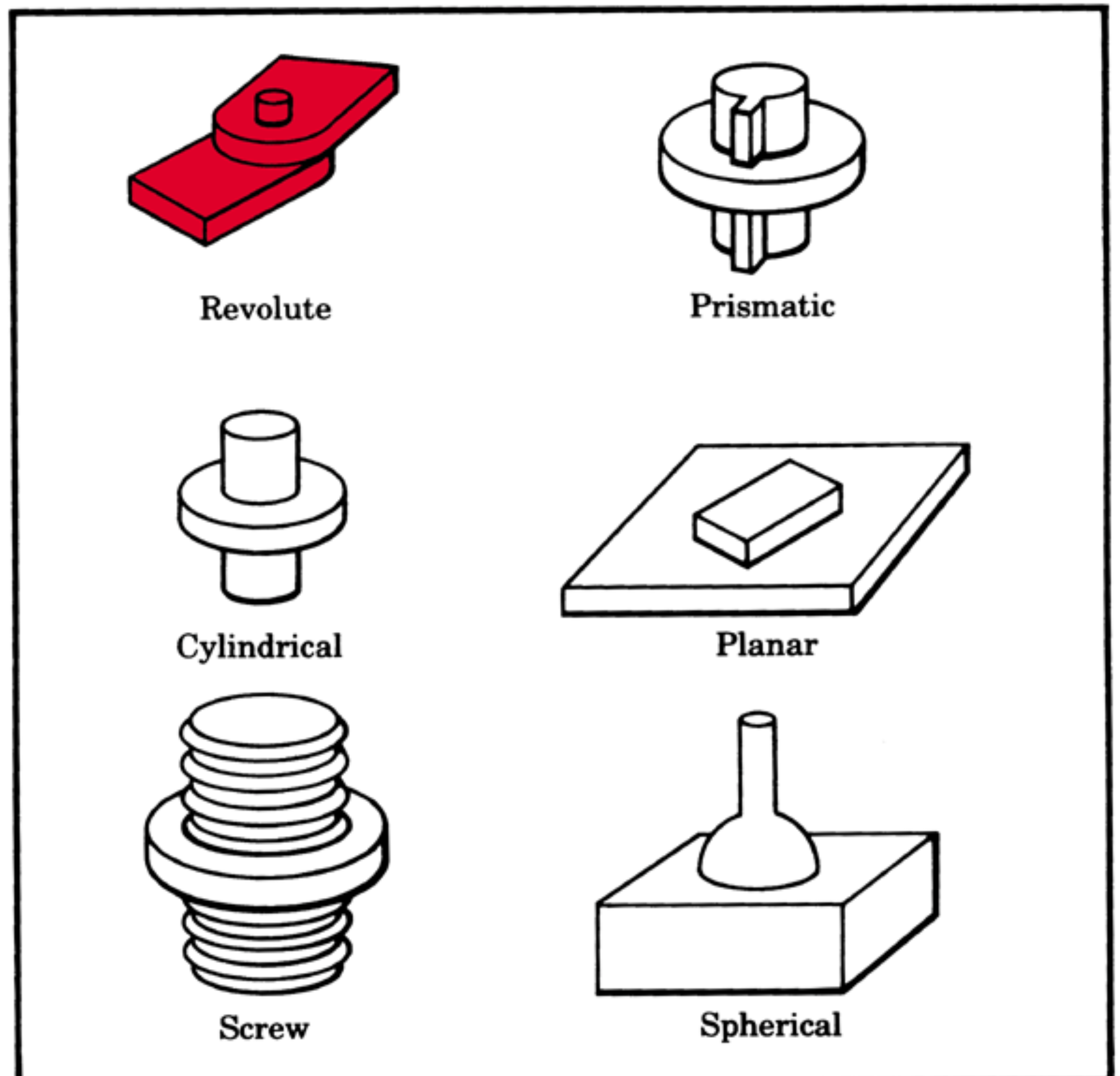
officedepot.com



www.rogersconnection.com/triangles

joints

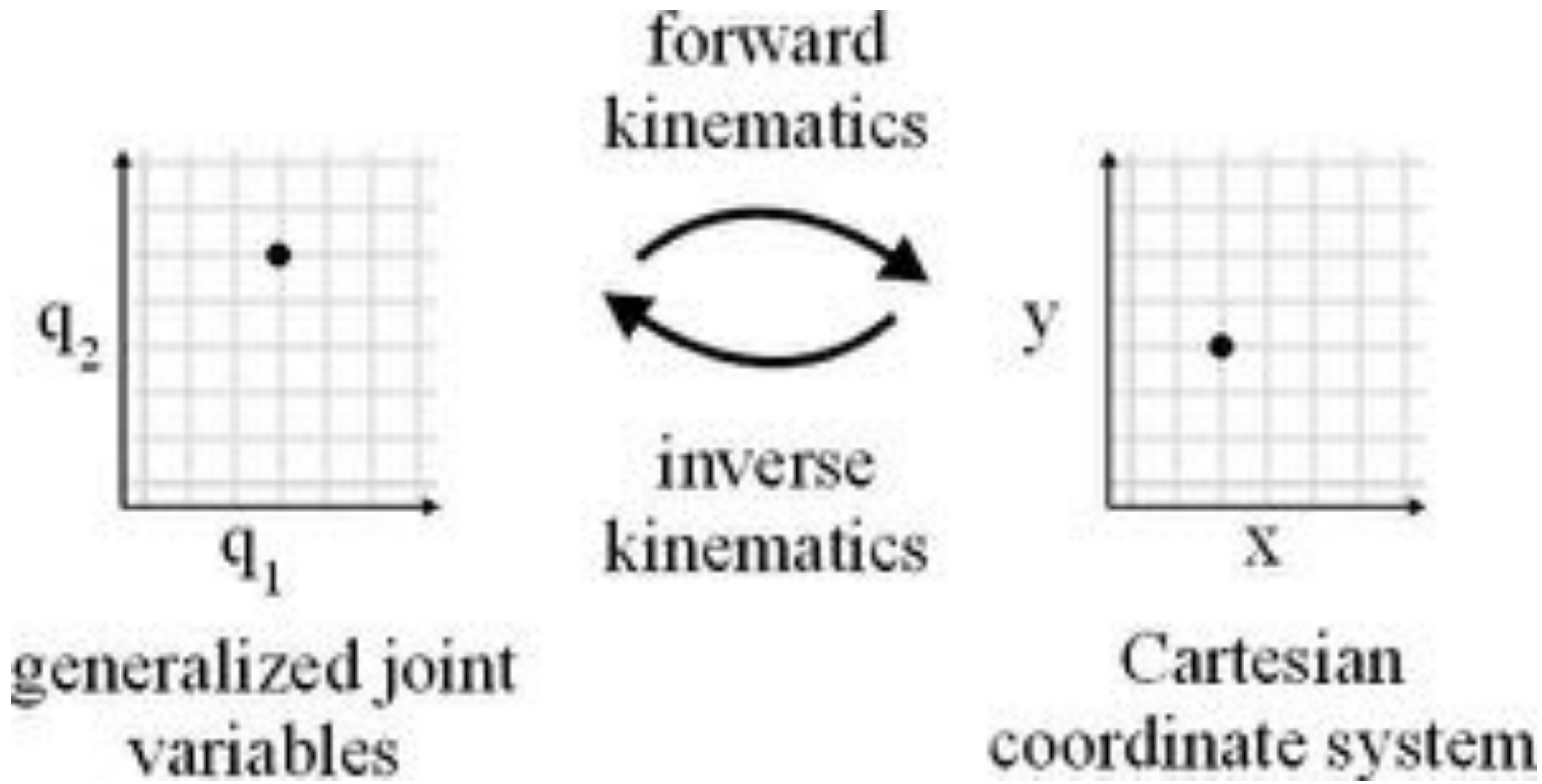
- Think of a manipulator/ interface as a set of bodies connected by a chain of joints
- **Revolute** is the most common joint for robots



From Craig, p. 69

kinematics for robots

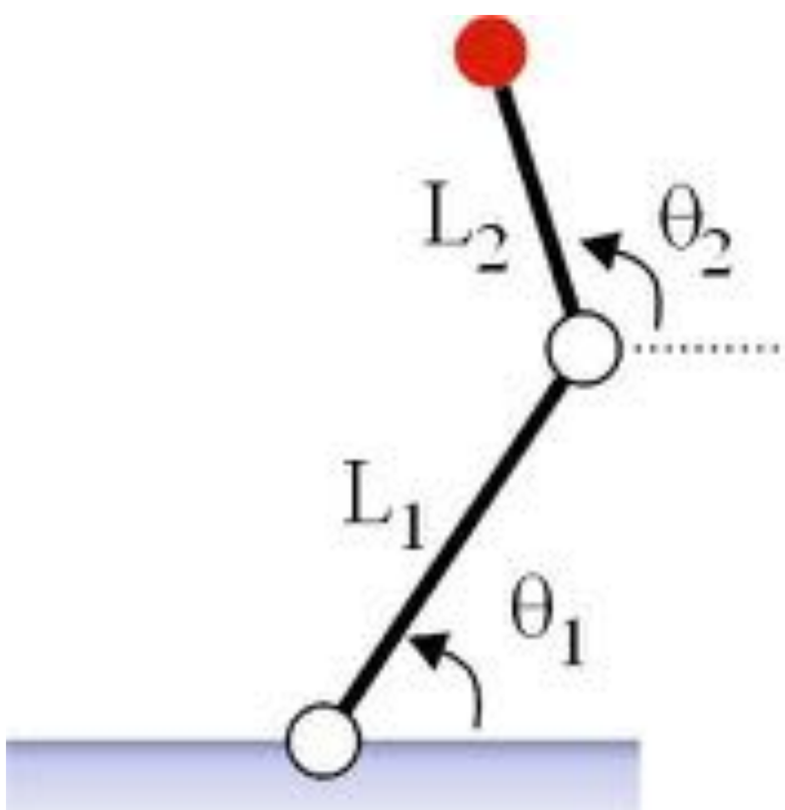
- Allows you to move between ***Joint Space*** and ***Cartesian Space***



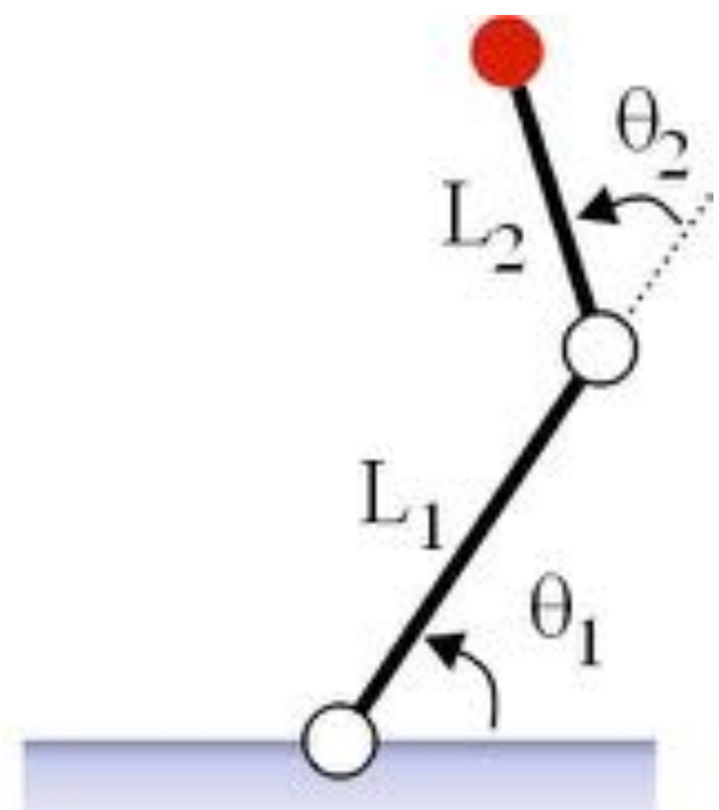
- Forward kinematics: based on joint angles, calculate end-effector position

joint variables

Be careful how you define joint positions



Absolute



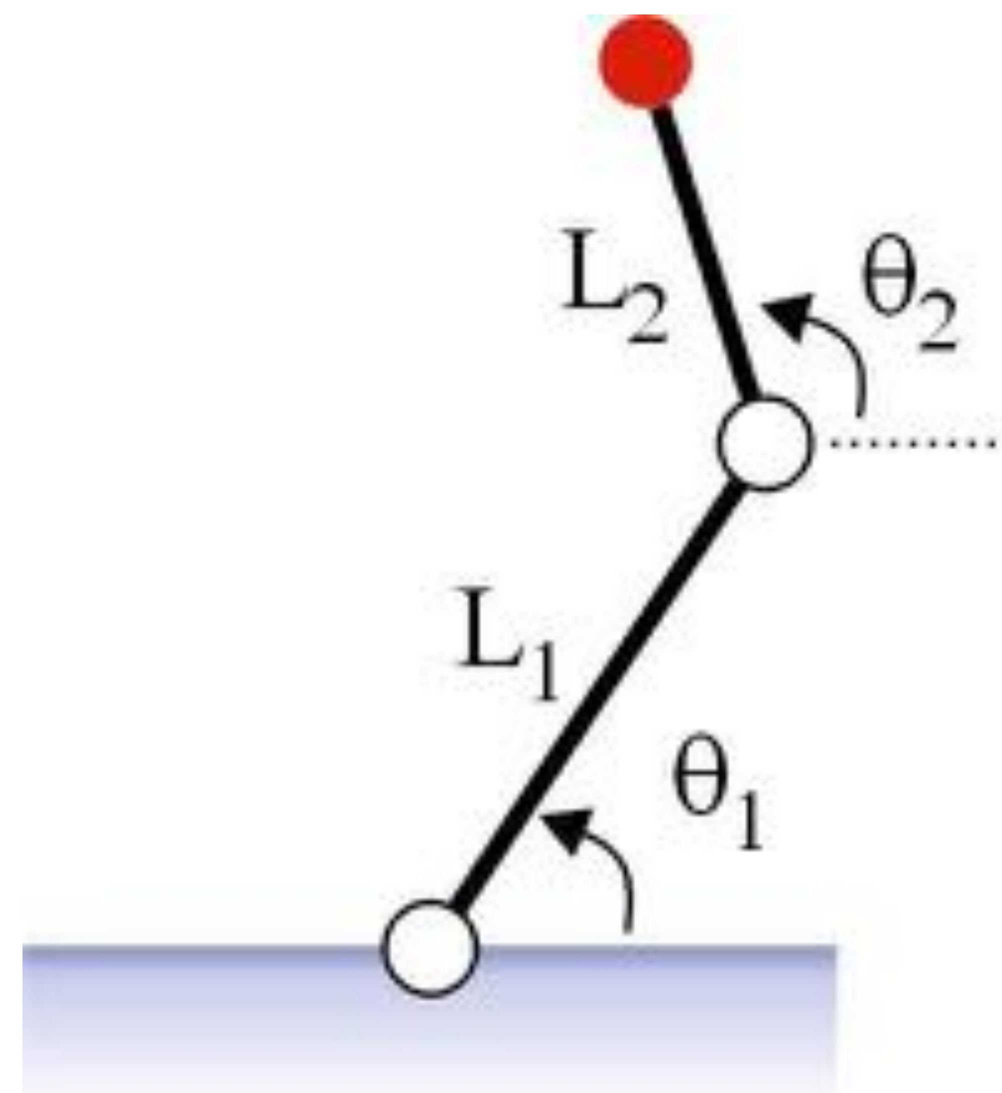
Relative

absolute forward kinematics

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_2)$$

$$y = L_1\sin(\theta_1) + L_2\sin(\theta_2)$$

(Sometimes done this way for robots)

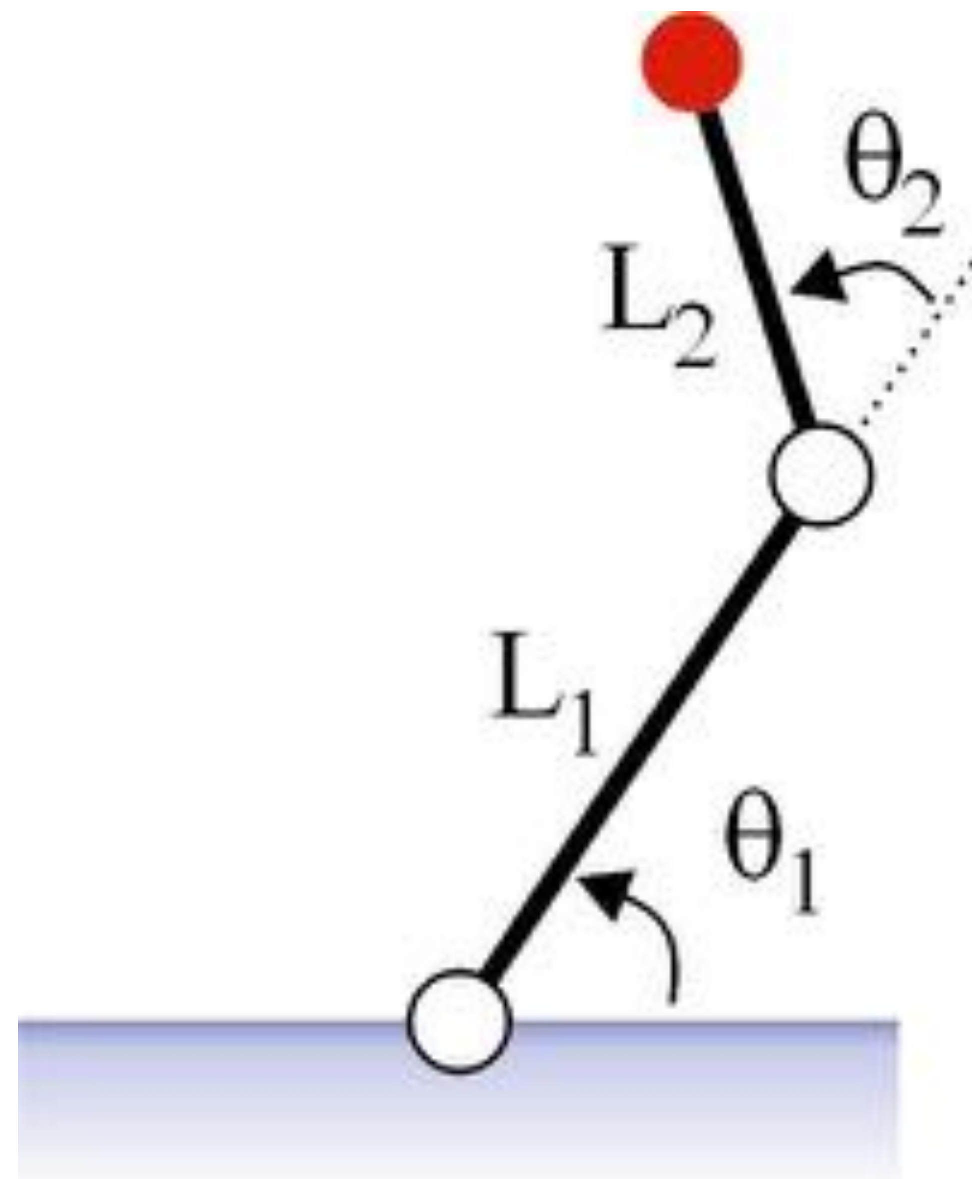


relative forward kinematics

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1+\theta_2)$$

$$y = L_1\sin(\theta_1) + L_2\sin(\theta_1+\theta_2)$$

(Often done this way for robots)

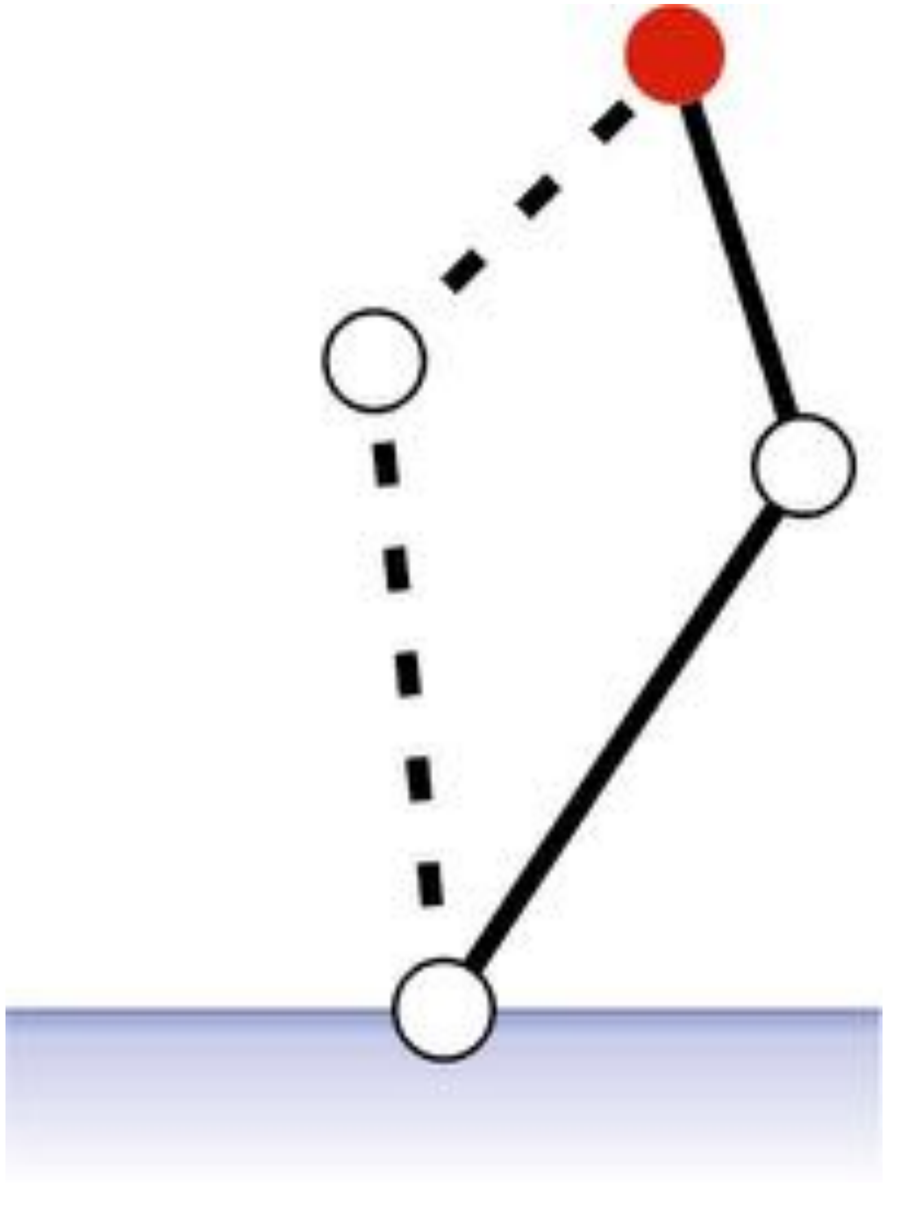


Inverse Kinematics

- Using the end-effector position, calculate the joint angles necessary to achieve that position
- Not used often for input devices or for robot control
  - But useful for planning
- There can be:
  - No solution (workspace issue)
  - One solution
  - More than one solution

example

- Two possible solutions
- Two approaches:
  - algebraic method (using transformation matrices)
  - geometric method
- Our devices will be simple enough that you can just use geometry



computing velocity

- Recall that the forward kinematics tells us the end-effector *position* based on joint positions
- How do we calculate velocity?
- Use a matrix called the ***Jacobian***

$$\dot{\mathbf{x}} = J\dot{\theta}$$

formulating the Jacobian

Use the chain rule:

$$\dot{x} = \partial x / \partial\theta_1 \, \dot{\theta}_1 + \partial x / \partial\theta_2 \, \dot{\theta}_2$$

$$\dot{y} = \partial y / \partial\theta_1 \, \dot{\theta}_1 + \partial y / \partial\theta_2 \, \dot{\theta}_2$$

Take partial derivatives:

$$\partial x / \partial\theta_1 = -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2)$$

$$\partial x / \partial\theta_2 = -L_2 \sin(\theta_1 + \theta_2)$$

$$\partial y / \partial\theta_1 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)$$

$$\partial y / \partial\theta_2 = L_2 \cos(\theta_1 + \theta_2)$$

assemble in a matrix

$$\dot{\mathbf{x}} = J\dot{\theta}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \partial x / \partial \theta_1 & \partial x / \partial \theta_2 \\ \partial y / \partial \theta_1 & \partial y / \partial \theta_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Singularities

- Many devices will have configurations at which the Jacobian is singular
- This means that the device has lost one or more degrees of freedom in Cartesian Space
- Two kinds:
  - Workspace boundary
  - Workspace interior

Singularity Math

- If the matrix is invertible, then it is non-singular.

$$\dot{\theta} = J^{-1}\dot{\mathbf{x}}$$

- Can check invertibility of J by taking the determinant of J. If the determinant is equal to 0, then J is singular.
- Can use this method to check which values of θ will cause singularities.

Calculating Singularities

Simplify text: $\sin(\theta_1+\theta_2)=s_{12}$

$$\det\left(J(\theta)\right) = \begin{vmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{vmatrix}$$

$$= (-L_1 s_1 - L_2 s_{12}) L_2 c_{12} + (L_1 c_1 + L_2 c_{12}) L_2 s_{12}$$

For what values of θ_1 and θ_2 does this equal zero?

even more useful….

The Jacobian can be used to relate ***joint torques*** to ***end-effector forces***:

$$\boldsymbol{\tau} = J^T \mathbf{F}$$

Why is this important for robots?

how do you get this equation?

The **Principle of virtual work**

states that changing the coordinate frame does not change the total work of a system

$$\mathbf{F} \cdot \delta\mathbf{x} = \tau \cdot \delta\theta$$

$$\mathbf{F}^T \delta\mathbf{x} = \tau^T \delta\theta$$

$$\mathbf{F}^T J \delta\theta = \tau^T \delta\theta$$

$$\mathbf{F}^T J = \tau^T$$

$$J^T \mathbf{F} = \tau$$

suggested references

- Introduction to robotics : mechanics and control
  John J. Craig
- Robot modeling and control
  Mark W. Spong, Seth Hutchinson, M. Vidyasagar
- A mathematical introduction to robotic manipulation
  Richard M. Murray, Zexiang Li, S. Shankar Sastry
- Springer handbook of robotics
  B. Siciliano, Oussama Khatib (eds.)
  http://site.ebrary.com/lib/stanford/docDetail.action?docID=10284823

Assignment 1

Problem 0: Commentary on seminar

Problem 1: Read/skim papers, answer questions

Problem 2: Kinematics of the Phantom Omni

Problem 3: Kinematics of a remote-center-of-motion (RCM) robot

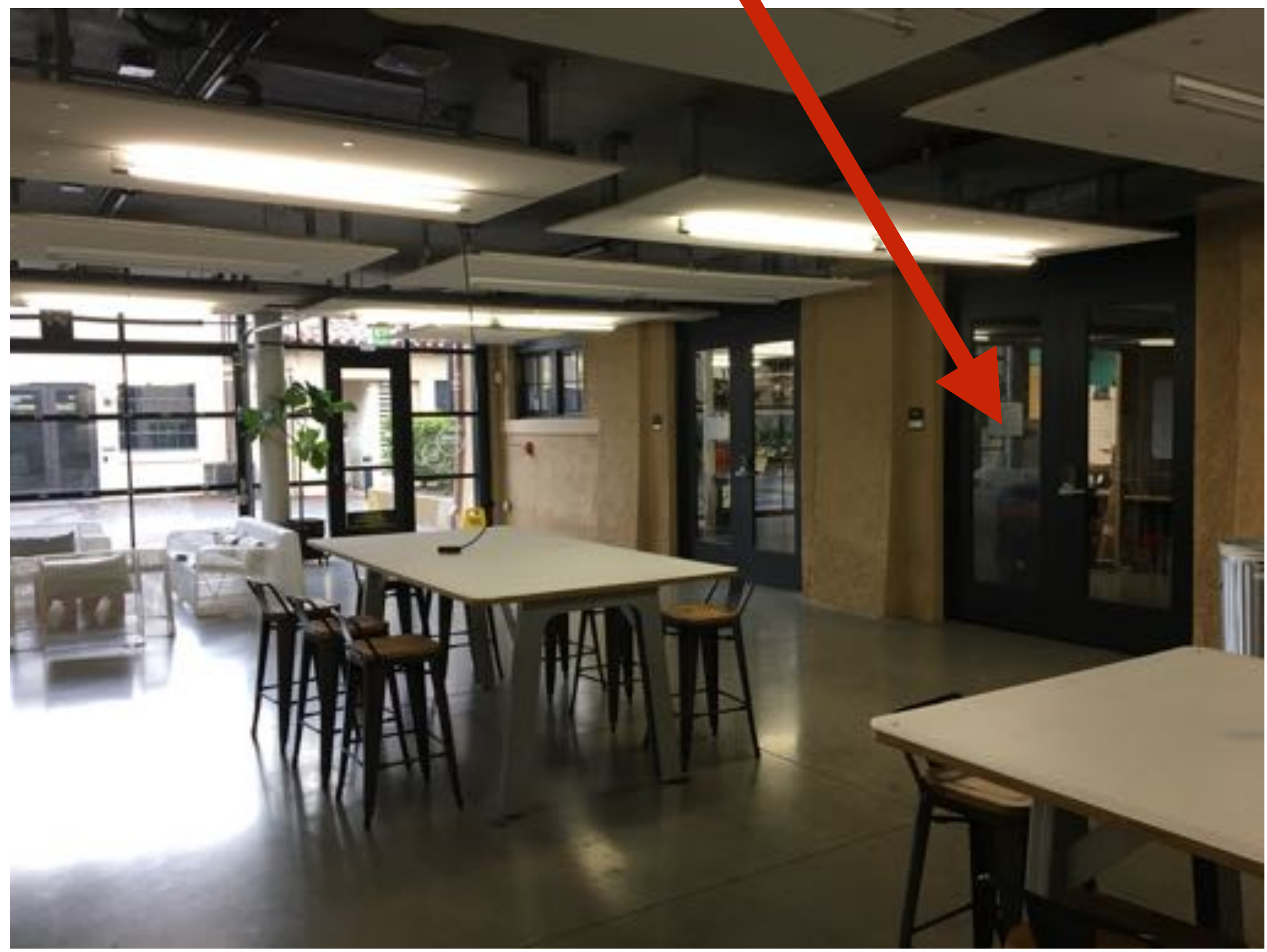
Posted later today, due Wednesday Jan. 16 at 4 pm

The keypad code for 550-108 is...

To do

- Fill out the survey (handout), return in class or to the box outside Allison's office (today!)
- Sign up on piazza: https://piazza.com/stanford/winter2019/me328
- Remember: Seminar on Friday **in 320-105**, come early!
- Visit us in office hours (in/around 550-108):
  Lisa: Fridays 2:30-4:30 pm
  Cole: Mondays 3-4 pm, 6-7 pm
  Allison: Tuesdays 12-1:30 pm

550-108



in box