6.141:
Robotics systems and science
Lecture 15:
Grasping and Manipulation

Lecture Notes Prepared by Daniela Rus

EECS/MIT

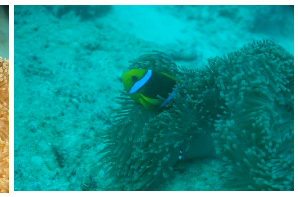
Spring 2009

Reading: Chapter3, Craig: Robotics

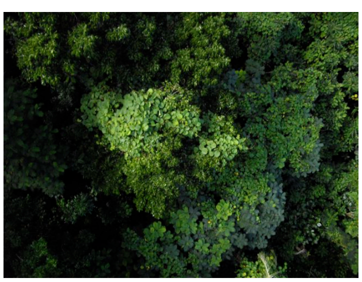
http://courses.csail.mit.edu/6.141/

Challenge: Build a Shelter on Mars

What the robot did over Spring break



What the other robot did

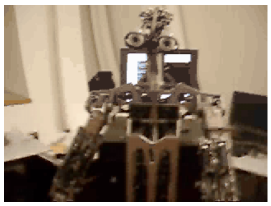


Last 2 modules were about

- High-level planning
- Localization

Today

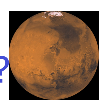
- Robot manipulation and grasping
- Applications: industrial assembly, home robots, surgery, construction, exploration, etc.
- Reading: chapters 3, 6

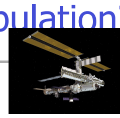


What is Manipulation?

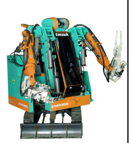
- Hayes, K.C. and Hayes, C.

What is Robot Manipulation?





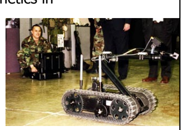
Space - in-orbit, repair and maintenance, planetary exploration anthropomorphic design facilitates collaboration with humans



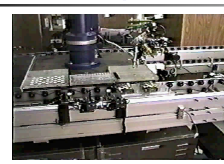
Home - basic science - manufacturing, logistics, automated warehousing and distribution, computational models of cognitive systems, learning, human interfaces

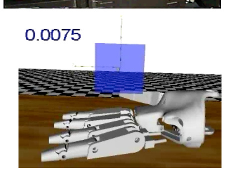
Assistive - clinical applications, "aging-in-place," physical and cognitive prosthetics in assisted-living facilities

Military - supply chain and logistics support, re-fueling, bomb disposal



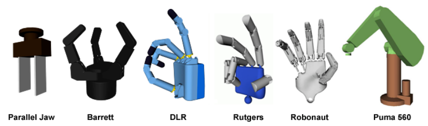
Grasping and Manipulation Examples; why is this hard?



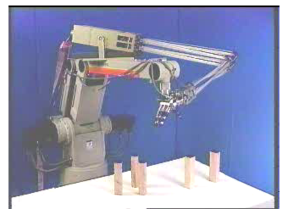
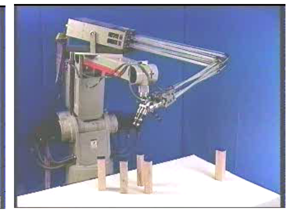


Robot Hands

- End-effectors are the part of the robot that usually does manipulation
- Many designs…



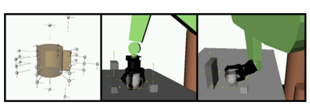
Problems

How does the robot reach for the object?
How does the robot grab the object?
How does the robot move the object?

Grasping

- Using end-effectors (fingers) to immobilize something relative to the hand
- Issues:



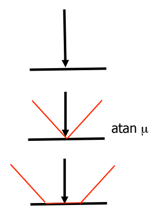
- What contacts?
- Where to place the contact points?
- What grasp properties?

Grasp Types

- Force closure: fingers resist any external force
- Torque closure: fingers resist any external torque
- Equilibrium: the contact forces can balance the object weight and external forces

Finger types

- Point contact with friction
- Hardfinger Contact
- Softfinger Contact

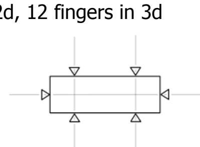


Issues in Grasp Design

- Existence: given an object and constraints determine if closure exist
- Analysis: given an object and contacts determine if closure applies
- Synthesis: given an object, find contacts that result in closure

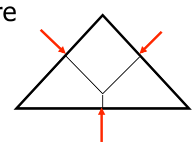
Existence

- Given an object, does it have a force-closure grasp?
- Theorem1 (Mishra, Schwartz, Sharir): for any bounded object that is not a surface of revolution a force closure grasp exists
- Theorem2 (Mishra, Schwartz, Sharir): at most 6 fingers in 2d, 12 fingers in 3d



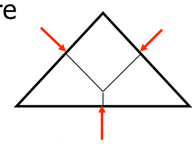
Grasp Analysis (no friction)

- Force-direction closure

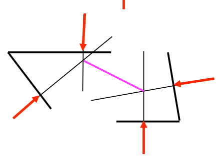


Grasp Analysis (no friction)

- Force-direction closure



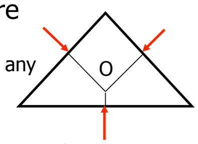
- Torque closure



Grasp Analysis (no friction)

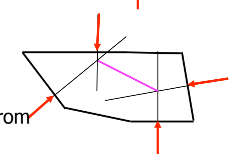
- Force-direction closure

Translate forces to O;
they compose to generate any
desired resultant force



- Torque closure

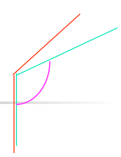
Translate forces to intersection
Points; they can be adjusted to
point at each other and away from
each other to generate torque



How do we turn this into an algorithm for grasping?

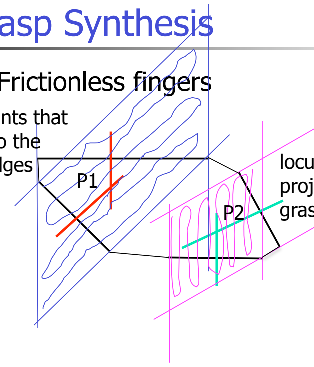
- Locus of A1
- Locus of A2
- Legal directions between A1 and A2
- Then
- Pick a line
- Convert to A1, A2,
- Project to get grasping points

Grasp Synthesis



- Frictionless fingers

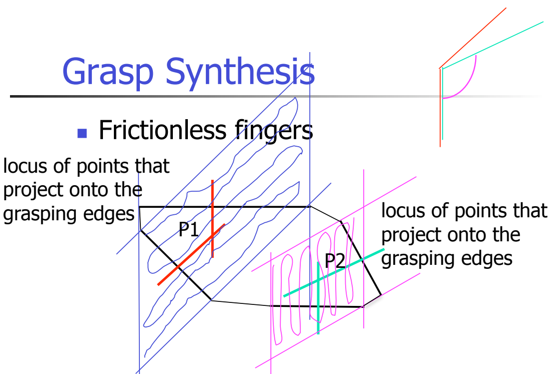
locus of points that
project onto the
grasping edges



locus of points that
project onto the
grasping edges

Grasp Synthesis

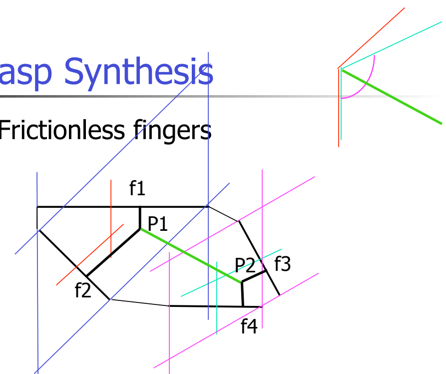
- Frictionless fingers



Pick P1 in blue region and P2 in pink region so that the line P1P2 has direction contained in the intersecting normal cones

Grasp Synthesis

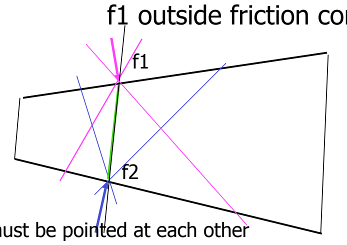
- Frictionless fingers



Project P1 and P2 to form grasping points f1, f2, f3, f4

Grasp Analysis (friction)

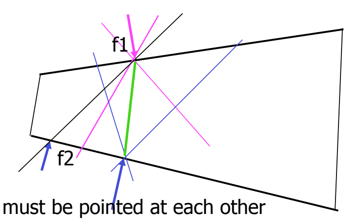
- With friction: f1 within friction cone--stick & f1 outside friction cone--slide



Forces must be pointed at each other
If blue force is anywhere within pink cone pink force can be pointed at it; can the blue force be pointed at pink force?

Grasp Analysis (friction)

- With friction (stick vs slide)



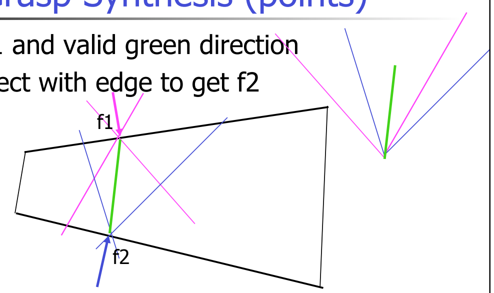
Forces must be pointed at each other
If blue force is outside pink cone pink force can not be pointed at it because it will start slipping

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

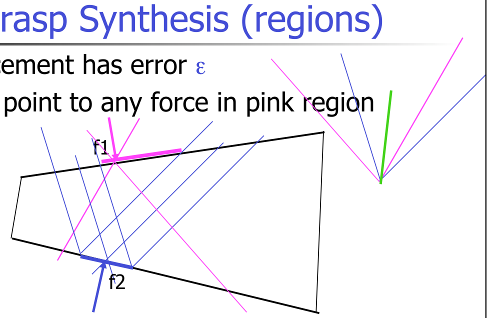
Grasp Synthesis (points)

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



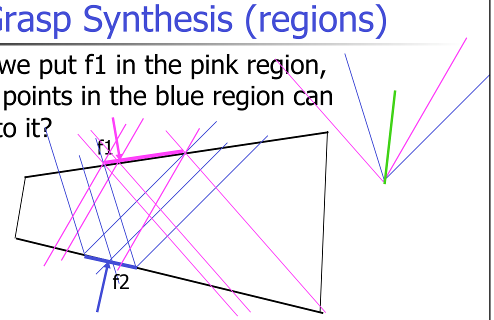
Grasp Synthesis (regions)

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

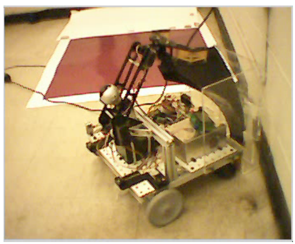


Grasp Synthesis (regions)

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

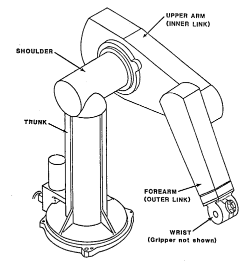


Example: 6.141 robot



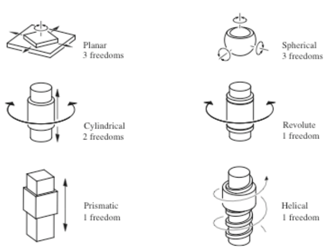
Arm Control to Reach

- Mechanism design
- Forward kinematics
- Inverse kinematics



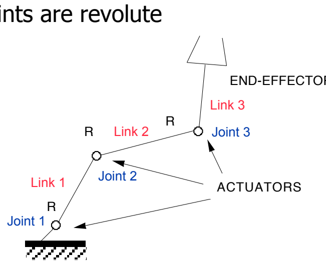
Kinematic Mechanisms

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



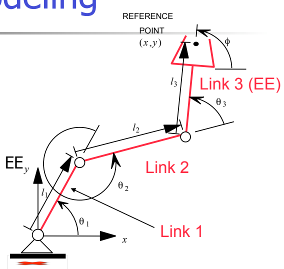
The Planar 3-R manipulator

- Planar kinematic chain
- All joints are revolute



Kinematic modeling

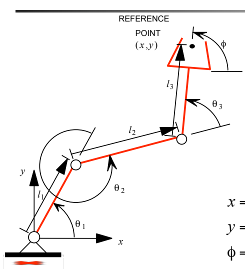
- Link
- Actuated joint
- End effector (EE)
  - Reference point on EE
- Joint coordinates
  - $\theta_1, \theta_2, \theta_3$
- End effector coordinates
  - x, y, ϕ
- Link lengths (l_i)



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.

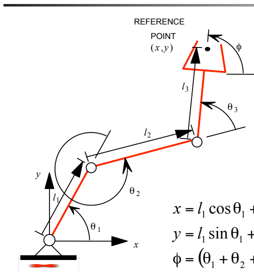
Direct kinematics



- 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

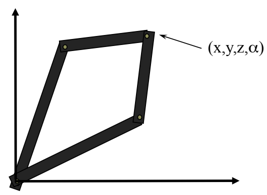


- Transform end effector coordinates to joint coordinates

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

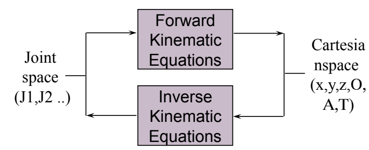
Given x, y, ϕ, solve for θ_1, θ_2, θ_3

Inverse kinematics has multiple solutions



Which is the correct robot pose ?

Kinematics Summary

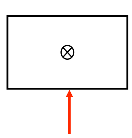


Robot kinematic calculations deal with the relationship between joint positions and an external fixed Cartesian coordinate frame.

Dynamics, force, momentum etc. are not considered.

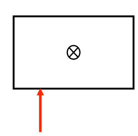
Pushing

- Straight-line motion



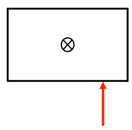
Pushing

- Clockwise rotation



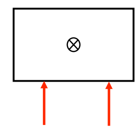
Pushing

- Counter-clockwise rotation



Pushing

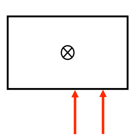
- Robust translation



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

Grasping and manipulation summary

- Reaching: forward and inverse kinematics
- Grasping: analysis and synthesis of closure grasps
- Manipulation: prehensile and non-prehensile