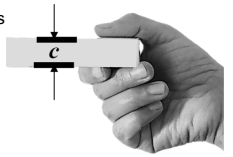
WICHITA STATE UNIVERSITY

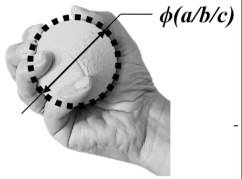
CS898AC Final Project: Dexterous Grasping Design

By Hui Li, Hongsheng He

Key Points of Dexterous Grasping

- Multiple manipulators or fingers cooperate together to grasp and manipulate objects.
- placing fingertips on the "right" spots of an object
  - Decide a proper grasping pose based on features of the object
  - Find the proper contacts point based on the selected grasping pose
- applying sufficient force and maintaining grasp stability.
  - Evaluate the weight, texture (friction) of object
  - Decide the force needed





WICHITA STATE UNIVERSITY

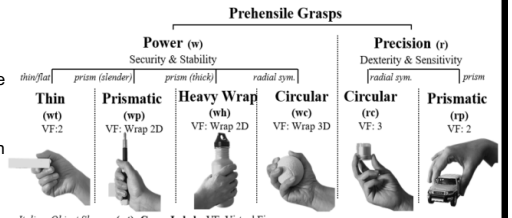
Goal of the project

- Design a proper grasp pose for each specific object.
- Decide the contact points
- Use inverse kinematics to test your results
- Simulate grasping process using Pybullet

WICHITA STATE UNIVERSITY

Grasp Poses Design

- There are many ways to grasp an object
  - There are no correct ways, only proper ways
  - Depend on features of objects and tasks
- Six most commonly used grasp pose are selected.
- Everyone will be assigned 4 objects and use given grasp poses based on the specific objects to pick up the objects
  - At least 2 grasp poses should be used
- References: http://grasp.xief.net/documents/THMS_taxonomy.pdf

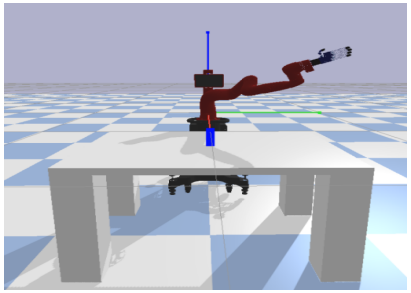


Italics: Object Shapes; **(wt): Grasp Labels;** VF: Virtual Fingers

WICHITA STATE UNIVERSITY

Contact Points

- Its hard to find the position and orientation in 3D space
- Use grasp simulation to explore the points
- Reach the palm to the object
- Try to pick up the object using selected grasp pose
- The Contact points will be print out
- Record the points of a successfully grasp
- Use: project_control_joint.py to achieve



Test Results With Inverse Kinematics

- Use: project_lk.py to test the results
- Adjust position and orientations of the hand using output from last step
- Try to pick up the object
  - If success, record the points
  - If fail, adjust the points using project_control_joint.py

```
############################################################ Simulation ##########################################################################################
# positions and orientations of hand joints, get from project_joint_control.py
thumbP = [1.0400433894335506, 0.013382494970968481, -0.04141455265676014]
thumbOrien = [0.3587517326156799, 0.7082855262317822, -0.4484997840336797 4, 0.4104590232172300 5]
indexP = [1.0390123451674059, 0.042578321409903 6, -0.0390986558452992]
indexOrien = [0.2403952017171509 8, 0.6778864151007256, -0.3320160927972921 6, 0.6102831058917473]
midP = [1.0405245909033144, 0.040746663759736 96, -0.0604410589054777]
midOrien = [0.2382840454420143 4, 0.6790490126394149, -0.3296484743121422 4, 0.6111182887927437]
ringP = [1.0255919695183234, 0.038991427407085555, -0.0801741136524998 36]
ringOrien = [0.2338644376254366 8, 0.6816579814873749, -0.3342022114908143, 0.6130060045360655]
pinkyP = [1.0033085619082292, 0.033602065890243324, -0.0976641571539278 6]
pinkyOrien = [0.1846309953328157, 0.7049581935504861, -0.2698121995027802 4, 0.6294018731366479]

currentP = [0]*65
k = 0
while 1:
        k = k + 1

        # move palm to target postion
        i=0
        while 1:
                i+=1
                p.stepSimulation()
                currentP = palmP([1.15, -0.07, 0.25], p.getQuaternionFromEuler([0, -math.pi*1.5, 0]))
                time.sleep(0.03)
                if(i==150):
                        break

        i=0
        while 1:
                i+=1
                p.stepSimulation()
                currentP = palmP([1.15, -0.05, -0.15], p.getQuaternionFromEuler([0, -math.pi*1.5, 0]))
                time.sleep(0.03)
                if(i==80):
                        break
```


Thank you & Questions?