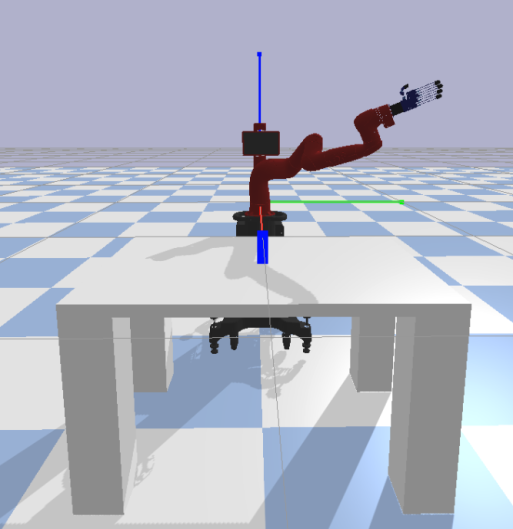
|  |  |
| --- | --- |
| Image result for Wichita State logo | **Introduction to Intelligent Robotics**  **Fall, 2018 / CS898 AC**  Dr. Hongsheng He, Department of Electrical Engineering and Computer Science |

**Final Project Requirements**

Vision Guided Grasping

# Background

The essence of dexterous manipulation is that multiple manipulators or fingers cooperate together to grasp and manipulate objects, such as placing fingertips on the “right” spots of an object, applying sufficient force and maintaining grasp stability.

The overall objective of this project is to design human-like grasping gestures to pick up the object, find the ‘right’ spots on the object for specific grasping gestures and use inverse kinematics to test your results. Human grasping has been studied in some detail and the relevant documentation shall be provided for reference.

Grasping task can be accomplished using few different tools. For this project, it is recommended to use Pybullet to simulate the grasping process. A pre-configured development environment shall be provided to reduce setup effort.

# Project Objective

The objective of this project is to design human-like grasping gestures and use the gestures to pick up specific objects. Specifically,

1. Design a proper grasp pose for each specific object. Please use Ref [1] for some types of grasping.
2. Decide the contact points (position and orientation) for each finger and thumb on the object.
3. Use inverse kinematics to test your results by picking up the object in simulation environments.

# Procedure

1. Decide a proper grasp pose by selecting one of the simple grasp taxonomies as shown in figure 2. Please refer to the “task\_allocation\_roster.xlsx” for assigned objects and grasping. Each of you only need to select one of the four objects (e.g., 004-007) to grasp and implement one of the grasping strategies (e.g., Thin, Prismatic, Heavy Wrap, Circular, Prismatic) in the given grasping types (either power or precision).
2. Design the selected grasp pose by controlling joint positions of AR10 robotic hand using the functions in project\_joint\_control.py. Record the positions and orientations of the palm, thumb and fingers of a success grasping.
3. Test your design with inverse kinematic using project\_IK.py. Use the positions and orientations recorded in step 2 as input and try to pick up the object.
4. If step 3 successes, record the positions and orientations in a table.

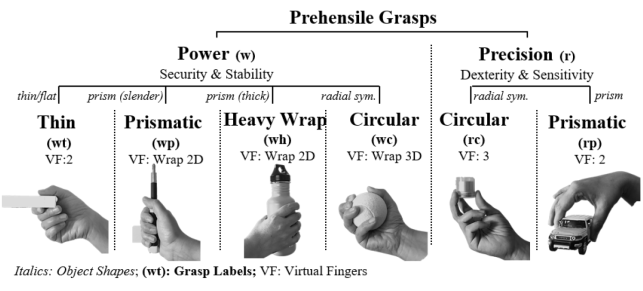


Figure 2.

# Deliverables, Evaluation Criteria and Timelines

Following are the expected deliverables and the weightage:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Deliverable | Grade Weightage | Due Date |
| 2 | Project Report in academic paper format (Abstract, Introduction, Methodology, Results, Discussion, Conclusion, References) | 20% | 12/04/2019 |
| 3 | Demonstration or simulation. | 5% | Video recording + codes |

# Note:

1. Each student will be assigned with 4 different objects in /random\_urdfs folder, check roster.xlsx for the assignment.
2. The PyBullet API uses quaternions to represent orientations. Since quaternions are not very intuitive for people, there are two APIs to convert between quaternions and Euler angles. You can use getEulerFromQuaternion([x,y,z]) function to convert Euler angles [3] to quaternions. For example, euler angle [0, math.pi\*0.5, 0]) will rotate 90 degrees around Y axis.

# References

[1] Human Grasp Taxonomy: <http://grasp.xief.net/documents/THMS_taxonomy.pdf>

[2] Pybullet Guide: <https://docs.google.com/document/d/10sXEhzFRSnvFcl3XxNGhnD4N2SedqwdAvK3dsihxVUA/edit#heading=h.2ye70wns7io3>

[3] Euler angles: <https://en.wikipedia.org/wiki/Euler_angles>

# Development Environment

**OS:** Ubuntu 16.04, Mac OS, or Windows

**Development:** Pybullet