Indian Institute of Technology, Gandhinagar

ME 692-V

Introduction to Robot Grasping

Semester–I, Academic Year 2023-24

Assignment - 2

By
Kush Patel
[20110131]

Q: Take a circle of given radius. Assume 3 pointed finger contact (assume contact model if required). Based on quality functions studied in class plus all the quality functions given for self reading, find best finger placement given by each quality index.
Comment your understanding about the answers generated in each index.

The provided script in *Grasp_Quality.py* is an implementation of a Grasp Quality Evaluator for a multi-fingered robot hand that grasps a circular object. It calculates and evaluates 4 grasp quality metrics: Q_MSV(G) (Minimum Singular Value), Q_EV(G) (Eigenvalue Volume), Q_GI(G) (Grasp Index), and Grasp Polygon Regularity Index (Q_GPRI). The objective is to find the best finger placement based on each of these quality indices.

- **Assumptions** :
    - *Finger Configuration:* The script assumes a 3-fingered robot hand for grasping a circular object. The initial finger positions are specified as [0.0, 0.0, 0.0], which corresponds to placing three fingers at 0 degree.
    - *Object Geometry:* The grasp quality metrics are evaluated for a circular object with a given radius. The radius of the circular object is set to 1.0.
    - *Degrees of Freedom (m):* The script assumes that there are 6 degrees of freedom for the robot hand. This parameter can be adjusted for 2D or 3D scenarios.
    - *Number of Independent Forces/Torques (p):* The script assumes 3 independent forces/torques.
    - *Euler Angles:* The Euler angles for the object orientation are set to [0.0, 0.0, 0.0]. These angles specify the orientation of the circular object in 3D space.
    - *Angle Increment:* When searching for the best finger configurations, the script iterates through possible finger angles with an angle increment of 30 degrees. This increment can be adjusted as needed.

- **Grasp Quality Metrics Calculation & Understanding the Results** :
    - It iterates through all possible finger configurations by varying the angles of the three fingers. For each configuration, it calculates the grasp matrix, which represents the relationship between forces and torques at the fingertips and external perturbations. (Refer the *grasp_matrix_cicrle.py* script)
    - The four grasp quality metrics are then calculated as follows:
        - *Q_MSV(G) (Minimum Singular Value):* Evaluates the minimum singular value of the grasp matrix. It measures the grasp's resistance to external perturbations. A higher Q_MSV(G) value indicates a more stable grasp.
        - *Q_EV(G) (Eigenvalue Volume):* Calculates the product of all singular values of the grasp matrix. It quantifies the grasp's

overall stability. A higher Q_EV(G) value indicates a more stable grasp.

- *Q_GI(G) (Grasp Index):* Compares the minimum and maximum singular values. A value closer to 1 indicates a balanced grasp where the minimum and maximum singular values are similar.
- *Grasp Polygon Regularity Index (Q_GPRI):* Measures the regularity of the grasp polygon formed by the contact points of the fingers. A lower Q_GPRI value implies a more regular grasp polygon.

- **Finding the Best Finger Configurations :**
  - The script maintains a dictionary called *best_configurations* to store the best finger configurations for each of the four grasp quality metrics. The dictionary contains information about the best metric value and the corresponding finger configuration (angles of the three fingers in degrees).
  - For each metric, the script identifies the finger configuration that maximizes/minimizes the metric value. It updates the *best_configurations* dictionary accordingly. For each of the indexes, the best configuration comes out as below:

$$Best\ Configuration\ (in\ degrees) = [\theta_1, \theta_2, \theta_3] = [0\ ,\ 120\ ,\ 240]$$

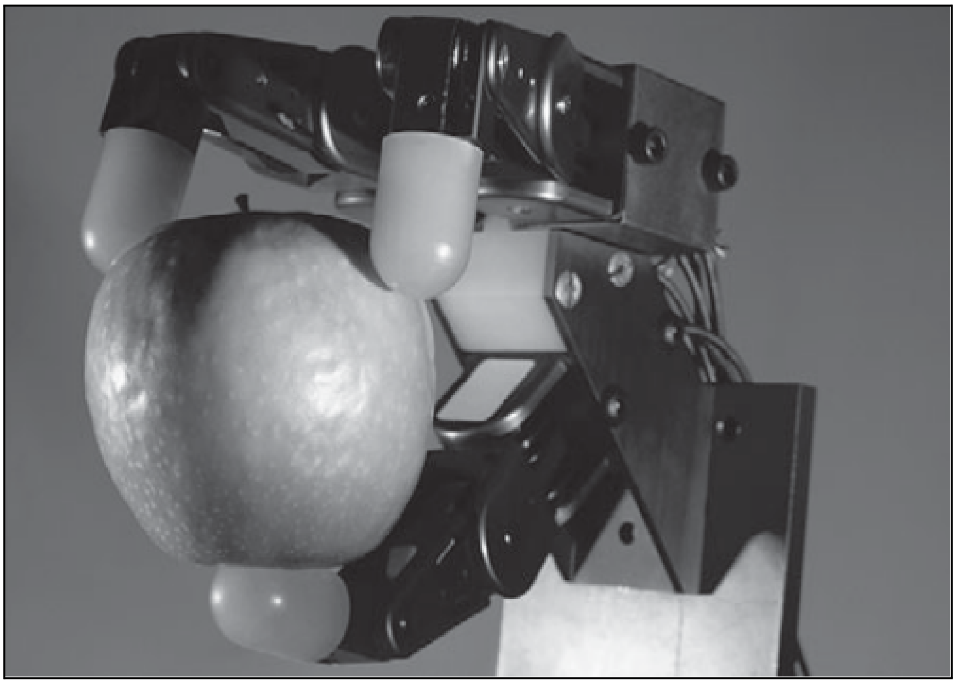

Fig: A three-finger robot hand demonstrating precision grasps

- **Visualization :**
  - The script provides a 3D scatter plot for each metric, visualizing how the metric values change with different finger configurations. Each plot shows the value of the metric as a function of the angles of the 2nd and 3rd fingers. This visualization helps us understand the relationships between the metrics and finger placements.