Indian Institute of Technology, Gandhinagar

ME 692-V

Introduction to Robot Grasping

Semester–I, Academic Year 2023-24

Assignment - 1

By
Kush Patel
[20110131]

Q.1: Write a generalized code for the Grasp matrix. Make appropriate assumptions, but write those clearly.

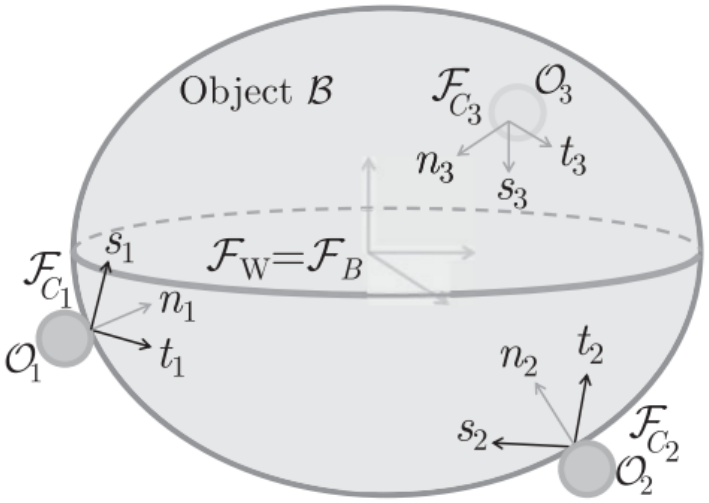

Fig. 1 - Grasped Object

(Note: The provided ***Grasp_Matrix.py*** file contains the python script in order to generate the grasp matrix for a multi-fingered robot hand holding an object with a given surface equation of any random object at specified contact points. Additionally, users can also visualize the object surface and grasp contact points.)

Assumptions:

- *Number of Fingers (k):* The script assumes that we have a multi-fingered robot hand with a specified number of fingers, denoted as 'k'. This value must be provided as an input to the calculator.
- *Degrees of Freedom (m):* The script assumes that the degrees of freedom of the grasping system are known and specified as 'm'. For 2D manipulation, 'm' is set to 3 (3 translational DOFs). For 3D manipulation, 'm' is set to 6 (3 translational and 3 rotational DOFs).
- *Number of Independent Forces/Torques (p):* The script assumes that we specify the number of independent forces and torques ('p') based on the type of grasp or contact. For example, in a point contact, 'p' can be set to 3 (forces in x, y, z directions), while in a line contact, 'p' can be set to 2.
- *Euler Angles (euler_angles):* The script assumes that we provide the Euler angles representing the orientation of the object being grasped. These angles are specified in radians and are used to calculate the rotation matrix that transforms vectors from the body frame to the world frame.
- *Object Surface Equation (object_surface_equation):* The script assumes that we provide an equation that describes the surface of the object being grasped. This equation is in the form of a function that takes three variables (x, y, z) and returns a scalar value. The equation should be expressed with respect to the body frame.

- *Contact Points (contact_points)*: The script assumes that we provide a list of contact points on the object's surface. Each contact point is specified as a 3D coordinate (x, y, z) relative to the body frame of the object. The number of contact points should be at least equal to the number of fingers ('k').
- *Rotation Matrix (rotation_matrix):* The script calculates a rotation matrix based on the provided Euler angles. This rotation matrix is used to convert vectors from the body frame to the world frame.
- *Visualizing Grasp*: The script includes a visualization function that plots the object's surface and the contact points in 3D. It assumes that we may want to visualize the grasp for better understanding.
- *Error Handling:* The script includes basic error handling to check if the number of fingers exceeds the number of provided contact points. If such a mismatch occurs, it prints a warning message.
- *Output Grasp Matrix:* The script calculates the grasp matrix based on the provided inputs and assumptions. The grasp matrix represents the relationship between the forces/torques applied by each finger and the resulting wrench (force and torque) on the object.
- *Output Formatting:* The script formats the calculated grasp matrix as a human-readable string, rounding off the values to three decimal places, and prints it to the console.
- *Simplified Object Shape:* The script assumes a simplified object shape described by an equation. In the example, it uses a sphere equation as a placeholder. we can replace this equation with a more complex one to model different object shapes.
- *Origin of Body Frame:* The script assumes the origin of the body frame of the object to be at coordinates (0, 0, 0) by default. we can modify this origin as needed.

Q.2: Write a generalized code for the Jh matrix(Hand Jacobian). Make appropriate assumptions, but write those clearly.

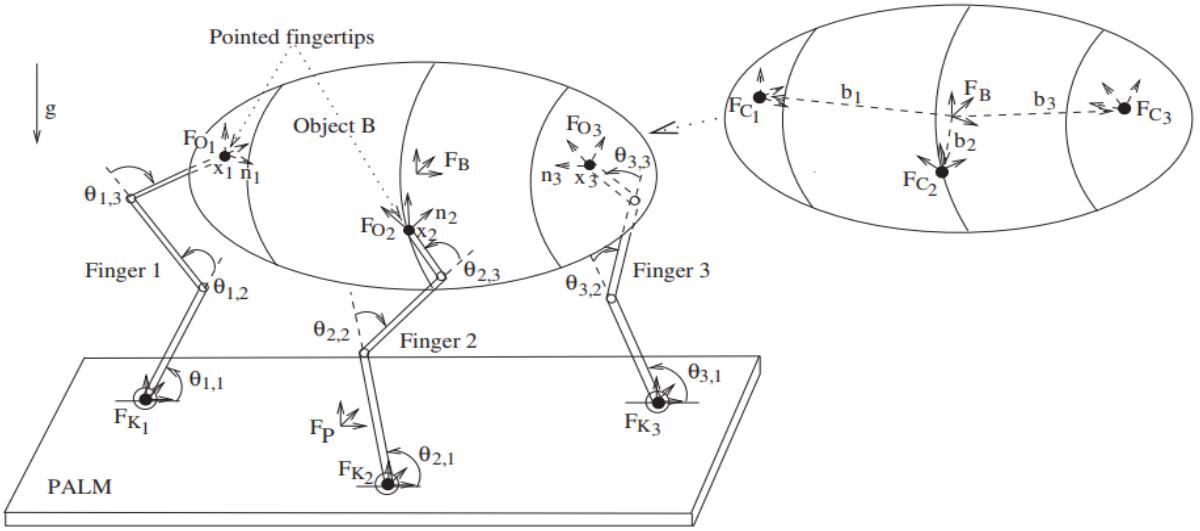

Fig. 2 - Robotic Hand Grasped Object

(Note: The provided ***Hand_Jacobian.py*** file contains the python script in order to generate the hand jacobian matrix for a multi-fingered robot hand holding an object where the robot hand is a serial chain robotic manipulator with specified joint angles and contact positions on the object's surface.)

Assumptions:

- *Number of Fingers (k):* The script assumes a fixed number of fingers, denoted as 'k'. This parameter determines the number of fingers used in the calculation of the hand Jacobian.
- *Number of Joints per Finger (n_values):* The script expects a list 'n_values' that specifies the number of joints for each finger. It is assumed that each finger has a consistent number of joints.
- *Total Number of Joints (N):* The total number of joints 'N' is calculated as the sum of joints across all fingers, which is assumed to be consistent with the provided 'n_values'.
- *Number of Independent Force Components (p):* The variable 'p' represents the number of independent force components, which is typically set to 3 for 3D force and torque calculations. This parameter is assumed to be constant for all fingers.
- *Euler Angles of Object B (euler_angles):* The Euler angles that represent the orientation of the object's body frame 'B' are provided as a list of angles [phi, theta, psi]. These angles are assumed to describe the orientation accurately.
- *Body Origin (body_origin)*: The 'body_origin' parameter specifies the origin of the body frame 'B'. It is assumed that this origin is accurately defined.
- *Object Surface Equation (object_surface_equation)*: The 'object_surface_equation' is assumed to be a function that defines the surface of the object with respect to the body frame 'B'. This equation is expected to be consistent with the object's shape and orientation.
- *Finger Base Orientations (finger_base_orientations):* The orientations of each finger's base frame are provided as a list of Euler angles [phi, theta, psi] for each finger. These angles are assumed to accurately represent the orientation of each finger's base frame.
- *Finger Joint Angles (finger_joint_angles):* The joint angles for each finger are provided as a list of angles for each joint in each finger. It is assumed that these angles accurately represent the joint configurations.
- *Contact Positions (contact_positions*): The 'contact_positions' parameter is a list of contact points where each finger makes contact with the object. It is assumed that these contact positions are correctly specified with respect to the body frame 'B'.
- *Link Lengths (link_lengths):* The 'link_lengths' parameter specifies the lengths of the links in each finger. It is assumed that these lengths accurately represent the physical dimensions of the robotic hand.
- *Coordinate Frames and Transformations:* The script assumes consistent coordinate frame conventions and transformations, such as Denavit-Hartenberg (DH) parameters, to calculate the transformation matrices and Jacobians for each finger.

- *Error Handling:* The script includes error-checking and validation for input parameters. If any inconsistencies or errors are detected, the script provides error messages to guide the user in resolving the issues.
- *Validity of the Calculations:* The script assumes that the mathematical and geometric calculations used to determine the hand Jacobian are valid and accurate within the constraints of the provided input parameters.
- *3D Force and Torque Calculation:* The script is designed to calculate a 3D hand Jacobian, which can handle both forces and torques in three dimensions. It is assumed that the application requires this level of detail.