Rheinische
Friedrich-Wilhelms-
Universität Bonn
Prof. Dr. Maren Bennewitz

Institut für Informatik
Abteilung VI
Humanoid Robots Lab
Adresse:
Friedrich-Hirzebruch-Allee 8
53115 Bonn

Humanoid Robotics

Assignment 5

Due Tuesday, May 22th, before class.

Robotic Manipulation:

1. Forward Kinematics Using DH Parameters

We consider a 2-link planar manipulator with revolute joints. The first link has a length of 1.0, and the second link has a length of 0.5. Both joints rotate in the plane (2D), and the base frame is fixed at the origin (0, 0). Use the Denavit-Hartenberg (DH) convention to describe and compute the robot's forward kinematics.

a. Define the DH parameters for this manipulator and extend the function dh_transform() that returns the corresponding homogeneous transformation matrix. [2 point]

b. Extend the function forward_kinematics_2link() that computes the pose of the end-effector given the two joint angles (in radians). Use the transformation matrix from (a). [2 point]

c. Write code that uses visualize_2link_arm() to plot the arm configuration in 2D. Show the joint positions, links, and the end-effector for at least 5 different joint configurations. [1 point]

2. Inverse Kinematics

We now consider a planar 3-link manipulator, where all joints are revolute and operate in 2D. The link lengths are**: Link 1:** 0.5 m**, Link 2:** 0.4 m, **Link 3:** 0.3 m
The goal of this task is to compute the joint angles that allow the end-effector to reach a desired 2D target position.

a. Extend the function forward_kinematics_3link() that computes the position of the end-effector given three joint angles, using the dh_transform() function from task 1. [1 point]

b. Implement a numerical inverse kinematics solver numerical_ik_3link() using the Jacobian (Moore-Penrose) pseudo-invers method. Start from an initial joint configuration and iteratively update the joint angles to minimize the error between the current and target end-effector positions. Use a fixed step size and a convergence threshold. Explain your stopping criteria and step size selection in a short markdown cell. [3 point]

c. Test your solver with at least three different target positions. Visualize the resulting configurations using matplotlib, showing the links and joints. Query at least one unreachable target and one hard to achieve target (close to joint limits), and discuss how joint limits or unreachable configurations affect the results. [1 point]

UNIVERSITÄT BONN
Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik
Abteilung VI
Humanoid Robots Lab

3. Reachability Map

A reachability map shows the set of all end-effector positions that a robot can achieve, given its joint limits and kinematic structure. In this task, you will compute and visualize the reachability map of the 3-link planar robot arm you implemented in the previous tasks.

a. Explain the concept of a reachability map (RM) in the context of robotic manipulation. How is it generated, and what information does it provide about a robot’s capabilities? [1 point]

b. Describe the purpose of an inverse reachability map (IRM). How is it related to the RM? [1 point]

c. A robot is tasked with picking up objects from a cluttered shelf. Discuss how reachability and manipulability maps can be used to pre-select optimal base poses and joint configurations before executing a grasp. [1 point]

d. Implement a function based on your explanations of a) that generates the reachability map for the 3-link arm used in Task 2. try different link length values and describe the change in the reachability map that you observe from the plot visualization. [2 point]