ASE-9407 Robot Manipulators: Modeling, Control and Programming

Assignment 1: DH Parameters, Forward and Inverse Kinematics

OBJECTIVE

- Learn and use the tools offered by Robotics Toolbox for Matlab to design and analyze an automatic manipulator.
- Link how the theoretical concepts (learnt during the lectures) are encapsulated by the Robotic Toolbox

DESIGN OF A SCARA ROBOT

SCARA robots are widely used in assembly operations. Read about their design and configuration. Your task is to decide the configuration and dimensions for each of the links of your SCARA robot.

Requirements:

The layout of the system where your robot is going to work is shown in Figure 1. Your robot base needs to be placed on top of the plate (orange cylinder). It should pick the green piece, and place it on top of the red piece. The dimensions, positions and orientations of these components are shown in Table 1.

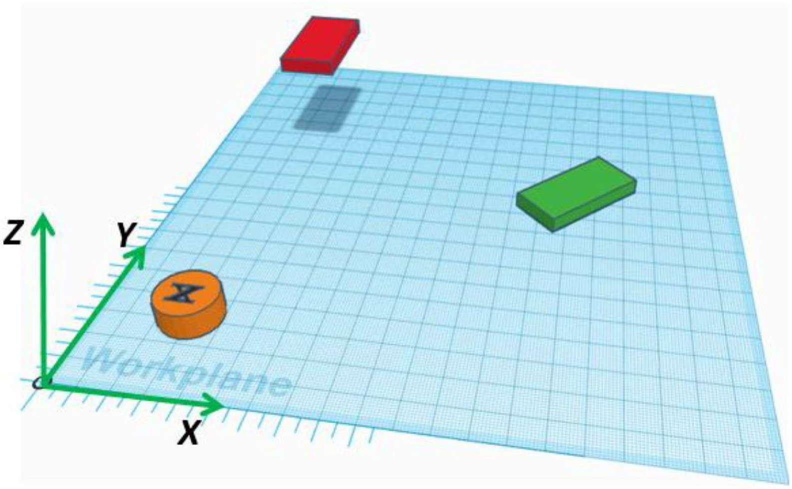

FIGURE 1. SYSTEM LAYOUT: MOUNTING BASE AND WORK PIECES

TABLE 1. DIMENSIONS, POSITIONS AND ORIENTATIONS OF COMPONENTS

Piece	Dimensions(m)	Position(m)	Orientation
Plate (orange cylinder)	Diameter: 0.4 Height: 0.2	x:0.1 y:0.1 z:0	Not rotated
Green piece	Width: 0.05 Length: 0.1 Height: 0.025	x:0.6 y:0.6 z:0	45 degrees (respect to world Z)
Red piece	Width: 0.05 Length: 0.1 Height: 0.025	x:0.3 y:0.8 z:0.1	Not rotated

The positions of the components are given respect to the world frame (shown in green at the left-bottom corner).

The positions given in the previous table refer to the center of the bottom plane of each of the components, as shown in

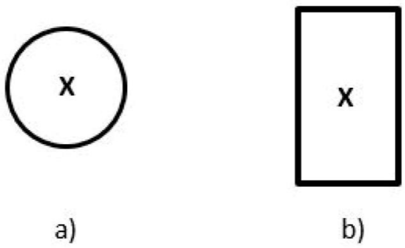

FIGURE 2. A)CYLINDER BOTTOM BASE, B) PRISM BOTTOM BASE

The tool that will grasp the pieces is built with a parallel gripper. In industrial robots, the tools are always attached to mounting plate of the robot, which is located at the end of the robot last link. The dimensions (in meters) of the tool in open configuration are shown in Figure 3. Notice this image shows a tool attached to a non SCARA robot. The gripper stroke (opening-closing travel distance) is 60mm.

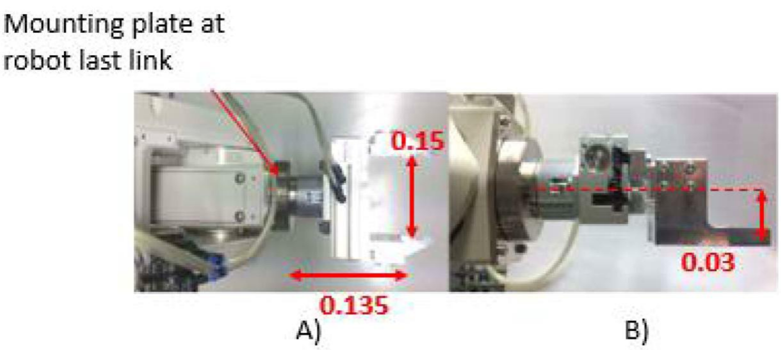

FIGURE 3. ATTACHMENT OF THE TOOL AND LOCATION OF THE TOOL CENTRAL POINT (TCP), UNITS IN METERS. TOOL SHOWN IN OPEN POSITION

APPROACH

1) Assign link frames
2) Determine the modified DH parameters for the SCARA robot of Figure 1. NOTE: parameters like link offset (d) or link length (a) should be assumed intuitively in this first iteration step. Joint twitst(α) can be determined from the figure
3) Model the manipulator with *SerialLink* command and attach the tool to the manipulator, modify also its base
4) Use *teach* command to jog the robot manually and visualize roughly if the working positions/orientations are within the robot workspace
5) Compute the inverse kinematics for the red and green work pieces as shown in Figure 1, if the inverse kinematics method can not find a solution, go to step 2 and modify link offsets (d) and lengths (a) . Be careful with the orientation of the tool. Remember a robot target is not just its position, but also its orientation.

QUESTIONS

Modified DH Parameters

*There is no need to use Matlab for the next 4 points

- Explain clearly any choices you made when placing the link frames. Do not describe here the DH convention, but rather your decisions in some frames where you had freedom to choose their location.
- Draw the robot in two configurations (zero and with offsets). In each of them show clearly the frame assignment.
  - Zero positon (when all joint values are equal to zero)
  - Offset position (with joint values different from zero)
  - Use a tool such as www.tinkercad.com, or any other 3D tool (no need to concentrate on details but rather the links placement and orientations)
  - Draw very clear diagrams
  - Explain if the DH rules fit for both of the configurations
- Explain the different options you had when assigning the link frames
- Provide the modified DH Parameter table of the SCARA manipulator you designed.
- Using the Symbolic Math toolbox, give the transformation matrix between link frames. *(R1: Transformations between links)*
- Give the homogenous transformation matrix for the robot when all the joint angles are set to zero. What is the position in X, Y and Z of the manipulator? *(R2: HT with joints values =0)*

Robotics Toolbox

- Model the robot using MATLAB robotics Toolbox. Give the code you used to create your model. *(R3: Robot model in robotic toolbox)*
- Plot the robot in the zero angle position. *(R4: Robot plot in zero position)*

Forward kinematics

- Determine tool transformation, and apply it to the model you created.
- Determine the base transformation and apply to the model
- Provide the tool transformation. *(R5: Tool transformation)*
- Provide the base transformation *(R6: Base transformation)*
- Plot the robot in different configurations and provide: joint angles, end frame homogenous transformation matrix and position (x,y,z) of the tool tip
  - Zero position *(R7: Robot in zero position)*
  - Offset position *(R8: Robot in offset position)*

Inverse kinematics

- Solve the inverse kinematics for the two work pieces (red and green).
- For each case, provide the code you used to calculate them, the joint angles values and the robot plotted in those positions. *(R9: Inverse kinematics for red and green pieces)*
- How can you verify that the joints angles that you computed, actually will drive the robot tool to the desired position and orientation?

Verifying the Robotic Toolbox forward kinematics

- Compute the forward kinematics with the Robotic Toolbox using fkine, for the green work piece. Provide the HT matrix with the location of the end effector. *(R10: Forward kinematics using fkine)*
- Using the Symbolic toolbox and the transformation matrix between neighbor links, compute the forward kinematics of the manipulator when it is evaluated with the same joint angles, used in the previous point. *(R11: Forward kinematics using symbolic toolbox)*
- Compare the results of both approaches and verify that they are equivalent.
- Document clearly in your script and report the code you used to compute the forward kinematics using the symbolic toolbox.

Assignment submission

Work in groups of three persons. Submit your reports with the answers, diagrams and plots required in the questions section in a pdf file. Submit also the Matlab code you generated properly documented/commented. Both files, should be in a ZIP folder. Submit just one zip file per group.

Points that will be considered for the grading

- Clear report, well structured, clear diagrams. Figure names, references
- Show that you can interpret the concepts
- Consider carefully the orientation of the tool when solving the inverse kinematics
- The pdf report should points in the questions section plus the Matlab requests
- The code will be tested to verify the results and therefore a properly commented code is required.
- When the code is run from Matlab it must show the results according to the Matlab output guideline, see below.

Assignment 1

Matlab Output guideline

- Your Matalab code should display in console the results, variables and plot the figures requested in the assignment.
- The requests are specified in this text as *(Rx: Title)*
- Display the title(Rx:Title) of each request (use command disp). Wait for the user to press key button (waitforbuttonpress) before moving to the next request. You can plot your robot on the same figure, no need to create a new figure every time you need to plot something.
- Provide the correct title to your figures.
- The first item to display is your team number, students name and students number.