University of Maryland- College Park

ENPM662 Introduction to Robot Modeling - Fall 2021

Homework - 4

Due Date : November 17th 2021 11:59 pm

Total - 50

Problem 1: Inverse Kinematics

Consider a KUKA WIIA robot with a pen (L=10 cm) attached as the end effector of the robot along Z direction of the local frame. Using information generated in HW3, derive the Jacobian matrix and then inverse kinematics of this robot (Inverse Jacobian) and solve the inverse kinematic problem so the robot can draw a circle on a wall as shown in Figure 1. **Assume that joint 3 is locked and will not be able to move so the Jacobian matric is square matrix.**

Dimensions are given in Figure 2.

Please assume that the robot is already moved to this configuration

$$q(0) = \begin{bmatrix} q_1 \\ q_2 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} = \begin{bmatrix} 90 \\ 0 \\ -90 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where the pen is in contact with wall at point S and it is perpendicular to the wall. We have updated Figure 1 dimensions to make sure that robot can reach point S with this initial configuration.

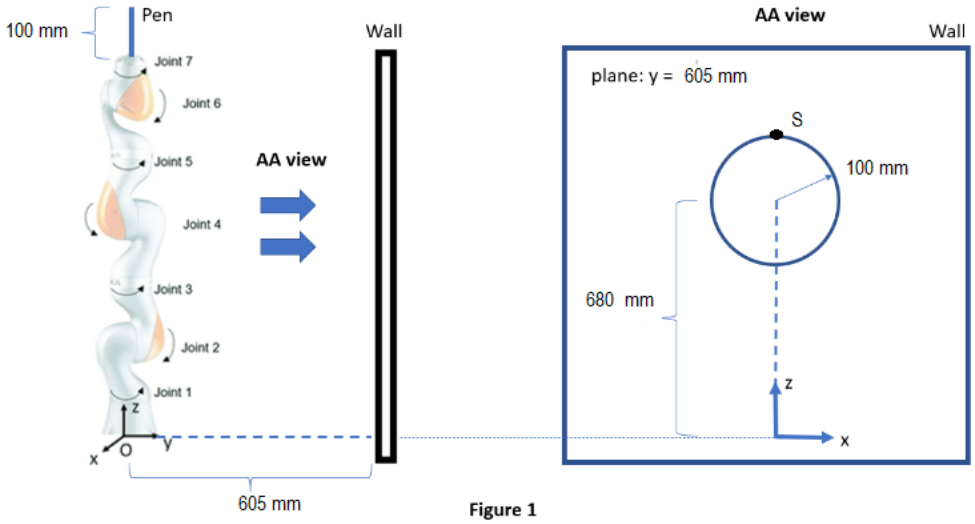

Figure 1

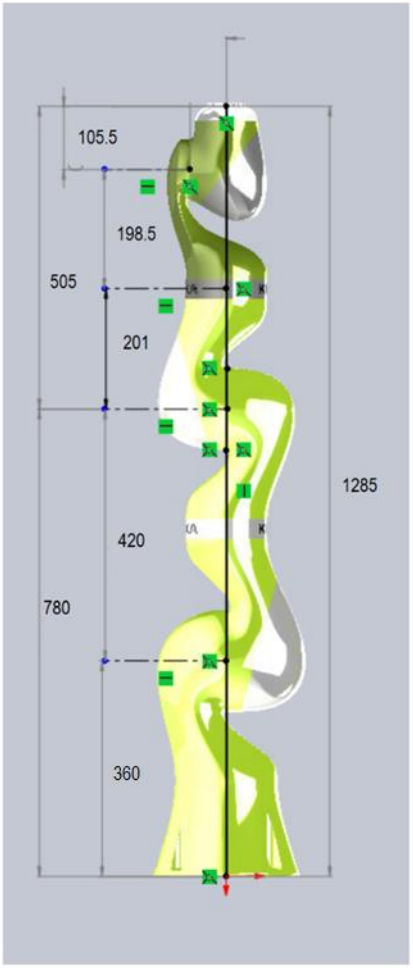

Figure 2

Other notes:

1- You don't need to calculate the inverse kinematics parametrically, just plug in the numerical values to Jacobian matrix first and then find the numerical inverse kinematics matrix in each iteration.
2- Use this equation to calculate the joint angles in each iteration:

   q_current= q_pervious + $\dot{q}$_current . Δt

Deliverables:

Step 1- Python code for forward velocity kinematics equations

Step 2- Python code for inverse velocity kinematics problem that generates joint velocities for a circle defined in Figure 1

Step 3- Plug in the join velocities generated in Step 2 to forward velocity kinematics developed in Step 1 and plot output of the forward kinematics equations that should be a circle as specified in Figure 1

Note: Use Matplotlib to draw the circle as a series of points. See this link:

https://www.kite.com/python/answers/how-to-live-plot-using-matplotlib-in-python