Dynamics and Control of Robot-Environment Interaction

(Fall 2021)

Homework #2 Due date **October** 12^{th}

Problem 1 Closeed-loop inverse kinematics(CLIK) (30)

✓ Consider a 7dof manipulator, which model is given in class. Implement the following controllers using CLIK algorithm.

✓ For each problem choose proper gain.

✓ **Students have to do simulation using position-control mode.**

✓ Desired position, x_d, is expressed in base frame.

$$x_d = [x_p^T \quad x_r^T]^T, \quad x_r = [s_1^T \; s_2^T \; s_3^T]^T$$

- Design trajectory using Jacobian.

$$\dot{q}_d(k) = J^\dagger(q_d(k))\dot{x}_d(k), \quad J^\dagger(q_d(k)) = J^T(JJ^T)^{-1} \tag{1}$$

$$q_d(k+1) = q_d(k) + \dot{q}_d(k)\Delta t \tag{2}$$

  – Move the manipulator from joint initial position to $q = (0, 0, 0, -\pi/2, 0, \pi/2, 0)$ in joint space.

    First, set the start joint positions as *joint initial position.*

    Second, generate cubic spline trajectory for each joint. You can use given example function DyrosMath::cubic in math_type_define.h

    Third, set q_d as command values.

  – And then, move the manipulator to track final end effector pose of $x_p = [0.25, 0.28, 0.65]^T$, $s_1 = [1, 0, 0]^T$, $s_2 = [0, -1, 0]^T$, $s_3 = [0, 0, -1]^T$.

    First, set the start end-effector positions as *initial position.*

    Second, generate cubic spline trajectory($x_d(k)$, $\dot{x}_d(k)$) for end-effector trajectory. You can use given example function DyrosMath::cubicDot in math_type_define.h

    Hint: You do not need to generate cubic trajectory for orientation in this case since desired orientation is the same as initial orientation, but if you want, you can make cubic trajectory of orientation using angle axis theorem.

    Third, calculate q_d.

    Fourth, plot tracking result of end effector and q_d values.

- Design trajectory using CLIK.

$$\dot{q}_d(k) = J^\dagger(q_d(k))\left[\dot{x}_d(k) + K_p(x_d(k) - x(q_d(k)))\right], \quad J^\dagger(q_d(k)) = J^T(JJ^T)^{-1} \tag{3}$$

$$q_d(k+1) = q_d(k) + \dot{q}_d(k)\Delta t \tag{4}$$

where

$$K_p = \begin{bmatrix} k_{p1} & & \\ & \ddots & \\ & & k_{p6} \end{bmatrix} \tag{5}$$

- Move the manipulator from initial position to $q = (0, 0, 0, -\pi/2, 0, \pi/2, 0)$ in joint space.
- And then, move the manipulator to track final end effector pose of $x_p = [0.25, 0.28, 0.65]^T$, $s_1 = [1, 0, 0]^T$, $s_2 = [0, -1, 0]^T$, $s_3 = [0, 0, -1]^T$.

  First, set the start end-effector positions as *initial position.*

  Second, generate cubic spline trajectory($x_d(k)$, $\dot{x}_d(k)$) for end-effector trajectory. You can use given example function DyrosMath::cubic and DyrosMath::cubicDot in math_type_define.h

  Third, calculate q_d. (Use given forward kinematics function to calculate $x(q_d(k))$.)

  Fourth, plot tracking result of end effector and q_d values.
- Compare the results from problem 1.

• Design trajectory using CLIK with weighted pseudo inverse.

$$\dot{q}_d(k) = J^\dagger(q_d(k))\left[\dot{x}_d(k) + K_p(x_d(k) - x(q_d(k)))\right], \quad J^\dagger(q_d(k)) = W^{-1}J^T(JW^{-1}J^T)^{-1} \tag{6}$$

$$q_d(k+1) = q_d(k) + \dot{q}_d(k)\Delta t \tag{7}$$

where

$$W^{-1} = diag(w_1, w_2, w_3, w_4, w_5, w_6, w_7), \quad w_4 = 0.01, w_i = 1(i \neq 4) \tag{8}$$

- Move the manipulator from initial position to $q = (0, 0, 0, -\pi/2, 0, \pi/2, 0)$ in joint space.
- And then, Plot result of trajectory tracking of the end effector to $x_p = [0.25, 0.28, 0.65]^T$, $s_1 = [1, 0, 0]^T$, $s_2 = [0, -1, 0]^T$, $s_3 = [0, 0, -1]^T$. **Here, Do not use step command! Use cubic spline trajectory as above problems.**
- Compare the results from problem 1 and 2. (Compare the output trajectory of 4th joint because the objective of this problem is to see the effect of weight matrix(W))