System HW2

Zhijia Chen

December 2019

1 Q1

The forward kinematics from joint one to end-effector is

$$^0T_e = \begin{bmatrix} r_1 & r_2 & 0 & p_{ex} \\ r_3 & r_4 & 0 & p_{ey} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cos(\theta_{1234}) & -sin(\theta_{1234}) & 0 & l_1cos(\theta_1) + l_2cos(\theta_{12}) + l_3cos(\theta_{123}) + l_4cos(\theta_{1234}) \\ sin(\theta_{1234}) & cos(\theta_{1234}) & 0 & l_1sin(\theta_1) + l_2sin(\theta_{12}) + l_3sin(\theta_{123}) + l_4sin(\theta_{1234}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$cos(\theta_{1234}) = r_1 \tag{1}$$

$$sin(\theta_{1234}) = r_3 \tag{2}$$

$$l_1cos(\theta_1) + l_2cos(\theta_{12}) + l_3cos(\theta_{123}) + l_4cos(\theta_{1234}) = p_{ex} \tag{3}$$

$$l_1sin(\theta_1) + l_2sin(\theta_{12}) + l_3sin(\theta_{123}) + l_4sin(\theta_{1234}) = p_{ey} \tag{4}$$

from equation 1, 2, 3 and 4, we have

$$l_1cos(\theta_1) + l_2cos(\theta_{12}) + l_3cos(\theta_{123}) = p_{ex} - l_4cos(\theta_{1234}) = p_{ex} - l_4r_1 \tag{5}$$

$$l_1sin(\theta_1) + l_2sin(\theta_{12}) + l_3sin(\theta_{123}) = p_{ey} - l_4sin(\theta_{1234}) = p_{ey} - l_4r_3 \tag{6}$$

There are three unknowns, but we only have two equations (5 and 6). Hence, the solutions are infinite.

2 Q2

1.Singularity: When a robot encounters a singularity during movement, the robotic arm would loss one or more degrees of freedom. Also, the the robot end-effector would be blocked in certain directions. Moreover, passing close to a singularity might cause high joint velocities, which should be avoided.
2. Joint limits: Joint limits show the largest workspace that an end-effector can reach.
3.Joint torque: When the joint torque in a robot joint is too large, the joint might break.
4.Path and trajectory planning: Basically, the shorter the path the end-effector travels, the shorter the execution time it will need. Also, the smoother the

end-effector goes, the better the performance might have. Hence we need to consider the path and trajectory planning.
5.Calculation time: If we use iterative inverse kinematic method, small learning rate and small error might lead to high computation time of finding the solution. Hence, we need to consider a proper learning rate and error scope.

3 Q3

When x is positive and y is negative, or x is negative and y is positive, the output of atan2(x,y) is different from atan($\frac{x}{y}$). atan($\frac{x}{y}$) can only give back the arctangent from quadrants 1 or 4, since ($\frac{-x}{y}$) and ($\frac{x}{-y}$) are the same value. In contrast, atan2(y,x) has two input, which can detect whether the numerator and denominator are positive or negative, so it can return arctangent from all the quadrants.

4 Q4

4.1 4.b

$$x = -c_1[a_1 - a_2 s_2 - a_3 s_{23} + a_4 c_{234} + d_5 s_{234}] \tag{7}$$

$$y = -s_1[a_1 - a_2 s_2 - a_3 s_{23} + a_4 c_{234} + d_5 s_{234}] \tag{8}$$

$$z = d_1 + a_2 c_2 + a_3 c_{23} + a_4 s_{234} - d_5 c_{234} \tag{9}$$

$$r_{11} = s_1 s_5 + c_1 c_{234} c_5 \tag{10}$$

$$r_{12} = -s_1 c_5 + c_1 c_{234} s_5 \tag{11}$$

$$r_{13} = c_1 s_{234} \tag{12}$$

$$r_{21} = s_1 c_{234} c_5 - c_1 s_5 \tag{13}$$

$$r_{22} = c_1 c_5 + s_1 c_{234} s_5 \tag{14}$$

$$r_{23} = s_1 s_{234} \tag{15}$$

$$r_{31} = -s_{234} c_5 \tag{16}$$

$$r_{32} = -s_{234} s_5 \tag{17}$$

$$r_{33} = c_{234} \tag{18}$$

from equation 12 and equation 15, we can have:

$$\theta_1 = atan2(r_{23}, r_{13}) \pm \pi \tag{19}$$

from equation 7, equation 12, equation 18, if $cos(\theta_1) \neq 0$, we have

$$\frac{x}{c_1} + a_4 r_{33} + \frac{d_5 r_{13}}{c_1} + a_1 = a_2 s_2 + a_3 s_{23} \tag{20}$$

otherwise, if $sin(\theta_1) \neq 0$,from equation 8, equation 15, equation 18, we have

$$\frac{y}{s_1} + a_4 r_{33} + \frac{d_5 r_{23}}{s_1} + a_1 = a_2 s_2 + a_3 s_{23} \tag{21}$$

let

$$Z_1 = \frac{x}{c_1} + a_4 r_{33} + \frac{d_5 r_{13}}{c_1} + a_1 (\text{or} \frac{y}{s_1} + a_4 r_{33} + \frac{d_5 r_{13}}{s_1} + a_1) \tag{22}$$

from equation 9, equation 12, equation 18, if $cos(\theta_1) \neq 0$, we have

$$z - d_1 - \frac{a_4 r_{13}}{c_1} + d_5 r_{33} = a_2 c_2 + a_3 c_{23} \tag{23}$$

otherwise, if $sin(\theta_1) \neq 0$,from equation 9, equation 15, equation 18, we have

$$z - d_1 - \frac{a_4 r_{23}}{s_1} + d_5 r_{33} = a_2 c_2 + a_3 c_{23} \tag{24}$$

let

$$Z_2 = z - d_1 - \frac{a_4 r_{13}}{c_1} + d_5 r_{33} (or z - d_1 - \frac{a_4 r_{23}}{s_1} + d_5 r_{33}) \tag{25}$$

from equation 20 (or 21), equation 22, equation 23 (or 24), equation 25, we have

$$c_3 = \frac{Z_1^2 + Z_2^2 - a_2^2 - a_3^2}{2 a_2 a_3} \tag{26}$$

$$\theta_3 = \pm arccos(\frac{Z_1^2 + Z_2^2 - a_2^2 - a_3^2}{2 a_2 a_3}) \tag{27}$$

let

$$A = a_3 c_3 + a_2 \tag{28}$$

$$B = a_3 s_3 \tag{29}$$

let

$$c_\gamma = \frac{A}{\sqrt{A^2 + B^2}} \tag{30}$$

$$s_\gamma = \frac{B}{\sqrt{A^2 + B^2}} \tag{31}$$

$$\frac{Z_1}{\sqrt{A^2 + B^2}} = s_2 \frac{A}{\sqrt{A^2 + B^2}} + c_2 \frac{B}{\sqrt{A^2 + B^2}} = sin(\theta_2 + \gamma) \tag{32}$$

$$\frac{Z_2}{\sqrt{A^2 + B^2}} = c_2 \frac{A}{\sqrt{A^2 + B^2}} - s_2 \frac{B}{\sqrt{A^2 + B^2}} = cos(\theta_2 + \gamma) \tag{33}$$

from equation 32, equation 33, we have

$$\gamma = atan2(B, A) \tag{34}$$

let

$$Z_3 = \frac{Z_1}{\sqrt{A^2 + B^2}} \tag{35}$$

$$Z_4 = \frac{Z_2}{\sqrt{A^2 + B^2}} \tag{36}$$

from equation 32, 33, 34, 35, 36,we have

$$\theta_2 = atan2(Z3, Z4) - \gamma \tag{37}$$

if $cos(\theta_1) \neq 0$, from equation 12, 18, 27, 37,

$$\theta_4 = atan2(r_{13}/cos(\theta_1), r_{33}) - \theta_2 - \theta_3 \tag{38}$$

otherwise, if $sin(\theta_1) \neq 0$, from equation 15, 18, 27, 37,

$$\theta_4 = atan2(r_{23}/sin(\theta_1), r_{33}) - \theta_2 - \theta_3 \tag{39}$$

from equation 16, 17

$$\theta_5 = atan2(\frac{r_{32}}{-s_{234}}, \frac{r_{31}}{-s_{234}}) \tag{40}$$

The total number of inverse kinematic solution of YouBot is four.

5 Q5

5.1 5.a

I use cubic polynomial model to calculate the four constraints: a_0, a_1, a_2, a_3. The reason I use cubic polynomial is that I do not know the acceleration required at each checker points, so I cannot use quintic polynomial model.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 10 & 100 & 1000 \\ 0 & 1 & 20 & 300 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_s \\ v_s \\ q_f \\ v_f \end{bmatrix}$$

The q_s at the beginning are set to be qs = [0.00122,0.39156,-0.35444,0.32441,0.01005]. This pose is obtained by using rostopic echo \ jointstates. Although every time the starting position might be different, the disparity is tiny. v_s at the starting point are all set to be zero. After entering the for loop for extracting the position and velocity from data1.bag, the first positions and velocities for the first checker point are set to be q_f and v_f respectively.

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \tag{41}$$

$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 \tag{42}$$

After having q_s, v_s, q_f, and v_f, the cubic polynomial model is used to calculate a_0, a_1, a_2, and a_3. Then a ten times for loop is used to calculate all the q_f and v_f between two checker points, with dt equals to 1. The previous q_f and v_f would be the next q_s and v_s until the for loop ends.

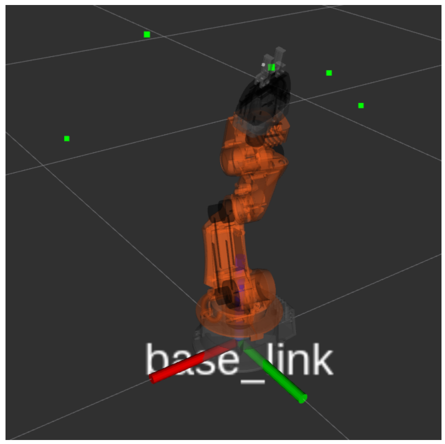

Figure 1: Question5.a

5.2 5.b

I first find all the possible combination of checker point 0,1,2,3,4, which has $5 \times 4 \times 3 \times 2 \times 1 = 120$ cases. Then a for loop is used to store all the positions and velocities in a temporary matrix. After that, the 'forward kinematic offset' function is called to obtain the transformation matrix for the pose at each checker points, so that we can know the position of each checker points. Then a 120 times for loop is used to calculate the total distance for all the possible combinations. The for loop returns 0,1,2,4,3, which is the order that has the shortest distance. Then the corresponding positions and velocities are extracted from the temporary matrix and published. However, the end-effector cannot move in a straight line, so the actual distance might not be short enough.

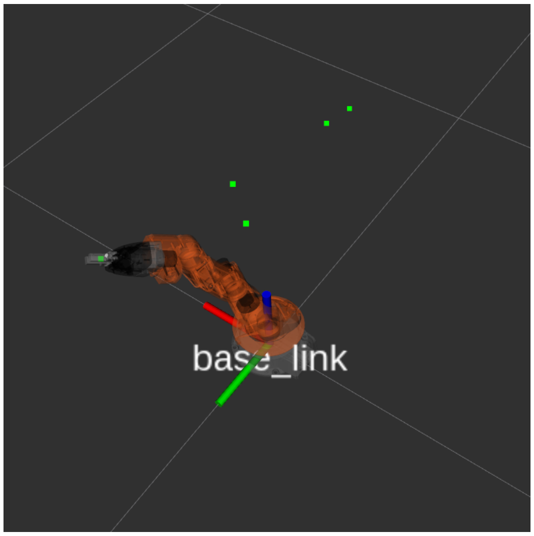

Figure 2: Question5.b