Robotic Systems Engineering

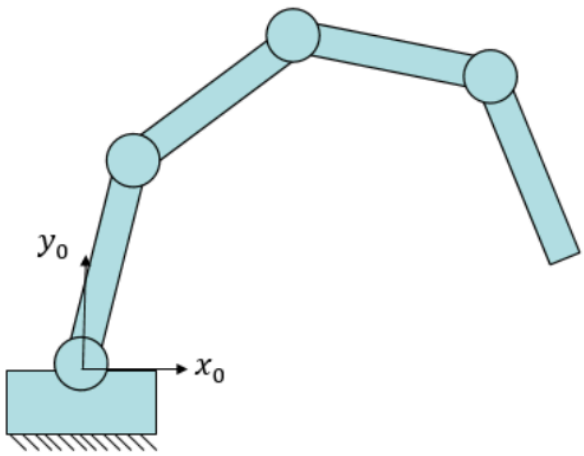
Coursework 2: Jacobian, Inverse Kinematics and Path Planning

WU ZHI

Department of Computer Science

University College London

Questions1



For a 4R-planar manipulator, each moveable joint has two poses (up and down). The base is a fixed point, so the three moveable joints in between have $2^3 = 8$ poses, so it has 8 possible poses to achieve a specific position in workspace. Furthermore, there are three singularity conditions: link1 and 2, link2 and 3, and link3 and 4, and each singularity condition have additional $3 \times 2^2 = 12$ poses. Considering the rotation angle θ for each joint, there are infinity number of solutions in inverse kinematics, because variance in θ can be close to 0.

Questions2

To optimize the inverse kinematics solution, there are many criteria to be considered:

1. minimize number of joints to be moved, minimize the total potential energy of a multilink redundant robot,
2. prevent self-collision and singularity
3. minimize time to desire point

Questions3

(a)

Show in the python code ‘cw2_Questions3.py’ class ‘YoubotTemplate’ function ‘forward_kinematics’ and ‘get_jacobian’.

(b)

$$T_5^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r_{11} = -\sin\theta_1 \sin\theta_5 + \cos\theta_1 \cos\theta_5 \cos(\theta_2 + \theta_3 + \theta_4)$$

$$r_{12} = -\sin\theta_1 \cos\theta_5 - \cos\theta_1 \sin\theta_5 \cos(\theta_2 + \theta_3 + \theta_4)$$

$$r_{13} = -\cos\theta_1 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$r_{21} = \cos\theta_1 \sin\theta_5 + \sin\theta_1 \cos\theta_5 \cos(\theta_2 + \theta_3 + \theta_4)$$

$$r_{22} = \cos\theta_1 \cos\theta_5 - \sin\theta_1 \sin\theta_5 \cos(\theta_2 + \theta_3 + \theta_4)$$

$$r_{23} = -\sin\theta_1 \sin(\theta_2 + \theta_3 + \theta_4)$$

$r_{31} = \cos\theta_5 \sin(\theta_2 + \theta_3 + \theta_4)$

$r_{32} = -\sin\theta_5 \sin(\theta_2 + \theta_3 + \theta_4)$

$r_{33} = \cos(\theta_2 + \theta_3 + \theta_4)$

$x = -\cos\theta_1 (-a_1 + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3) + d_5 \sin(\theta_2 + \theta_3 + \theta_4))$

$y = -\sin\theta_1 (-a_1 + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3) + d_5 \sin(\theta_2 + \theta_3 + \theta_4))$

$z = d_1 + a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3) + d_5 \cos(\theta_2 + \theta_3 + \theta_4)$

As θ_1 and θ_5 have been solved as

$$\theta_1 = \arctan\left(\frac{y}{x}\right)$$

$$\theta_5 = \operatorname{atan2}(r_{21}\cos\theta_1 - r_{11}\sin\theta_1, r_{22}\cos\theta_1 - r_{12}\sin\theta_1)$$

$$\theta_2 + \theta_3 + \theta_4 = \operatorname{atan2}\left(\frac{r_{32}}{-\sin\theta_5}, r_{33}\right)$$

Square the function of x and y, and add them together, we have

$$x^2 + y^2 = (\cos^2\theta_1 + \sin^2\theta_1)(-a_1 + a_2\sin\theta_2 + a_3\sin(\theta_2 + \theta_3) + d_5\sin(\theta_2 + \theta_3 + \theta_4))^2$$

$$\sqrt{x^2 + y^2} = -a_1 + a_2\sin\theta_2 + a_3\sin(\theta_2 + \theta_3) + d_5\sin(\theta_2 + \theta_3 + \theta_4)$$

According to function of r_{32}, $\sin(\theta_2 + \theta_3 + \theta_4) = -\frac{r_{32}}{\sin\theta_5}$, and $\sin(\theta_2 + \theta_3) = \sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_3$

$$\sqrt{x^2 + y^2} = -a_1 + a_2\sin\theta_2 + a_3\sin\theta_2\cos\theta_3 + a_3\cos\theta_2\sin\theta_3 - d_5\frac{r_{32}}{\sin\theta_5}$$

$$(a_2 + a_3\cos\theta_3)\sin\theta_2 + a_3\sin\theta_3\cos\theta_2 = \sqrt{x^2 + y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}$$

Assume $A = a_2 + a_3\cos\theta_3$, and $\mathrm{B} = a_3\sin\theta_3$

$$A\sin\theta_2 + B\cos\theta_2 = \sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}$$

Assume $\cos\alpha = A$ and $\sin\alpha = B$

$$\cos\alpha\sin\theta_2 + \sin\alpha\cos\theta_2 = \sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}$$

$$\sin(\theta_2+\alpha) = \sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}$$

According to the function of r_{33} and trigonometry algorithms, $\cos(\theta_2+\theta_3) = \cos\theta_2\cos\theta_3 - \sin\theta_2\sin\theta_3$, it becomes

$$z = d_1 + a_2\cos\theta_2 + a_3\cos\theta_2\cos\theta_3 - a_3\sin\theta_2\sin\theta_3 + d_5 r_{33}$$

$$(a_2 + a_3\cos\theta_3)\cos\theta_2 - a_3\sin\theta_3\sin\theta_2 = z - d_1 - d_5 r_{33}$$

As we have defined A, B, $\cos\alpha$ and $\sin\alpha$

So,

$$A\cos\theta_2 - B\sin\theta_2 = z - d_1 - d_5 r_{33}$$

$$\cos\alpha\cos\theta_2 - \sin\alpha\sin\theta_2 = z - d_1 - d_5 r_{33}$$

$$\cos(\theta_2+\alpha) = z - d_1 - d_5 r_{33}$$

From $\sin(\theta_2+\alpha) = \sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}$ and $\cos(\theta_2+\alpha) = z - d_1 - d_5 r_{33}$

$$\theta_2 + \alpha = \operatorname{atan2}\left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}, z - d_1 - d_5 r_{33}\right)$$

$$\theta_2 = \operatorname{atan2}\left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}, z - d_1 - d_5 r_{33}\right)$$
$$- \operatorname{atan2}(a_3\sin\theta_3, a_2 + a_3\cos\theta_3)$$

Square $A\sin\theta_2 + B\cos\theta_2 = \sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}$ and $\cos(\theta_2+\alpha) = z - d_1 - d_5 r_{33}$, and sum them together

$$(A\sin\theta_2 + B\cos\theta_2)^2 + (A\cos\theta_2 - B\sin\theta_2)^2$$
$$= \left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}\right)^2 + (z - d_1 - d_5 r_{33})^2$$

Simplify left hand side of Equation above,

$$(A\sin\theta_2 + B\cos\theta_2)^2 + (A\cos\theta_2 - B\sin\theta_2)^2$$

$$= A^2 \sin^2\theta_2 + B^2\cos^2\theta_2 + 2AB\sin\theta_2\cos\theta_2 + A^2\cos^2\theta_2 + B^2\sin^2\theta_2 - 2AB\cos\theta_2\sin\theta_2 = A^2 + B^2$$

Therefore, by substituting $A = a_2 + a_3\cos\theta_3$, and $\mathrm{B} = a_3\sin\theta_3$

$$(a_2 + a_3\cos\theta_3)^2 + (a_3\sin\theta_3)^2$$

$$= \left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}\right)^2 + (z - d_1 - d_5 r_{33})^2$$

$$a_2^2 + a_3^2\cos^2\theta_3 + 2a_2a_3\cos\theta_3 + a_3^2\sin^2\theta_3$$

$$= \left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}\right)^2 + (z - d_1 - d_5 r_{33})^2$$

$$a_2^2 + a_3^2 + 2a_2a_3\cos\theta_3 = \left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}\right)^2 + (z - d_1 - d_5 r_{33})^2$$

$$\cos\theta_3 = \frac{\left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}\right)^2 + (z - d_1 - d_5 r_{33})^2 - a_2^2 - a_3^2}{2a_2a_3}$$

Hence,

$$\theta_3 = \pm\arccos\left(\frac{\left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}\right)^2 + (z - d_1 - d_5 r_{33})^2 - a_2^2 - a_3^2}{2a_2a_3}\right)$$

According to Equation $\theta_2 + \theta_3 + \theta_4 = \operatorname{atan2}\left(\frac{r_{32}}{-\sin\theta_5}, r_{33}\right)$

$$\theta_4 = \operatorname{atan2}\left(\frac{r_{32}}{-\sin\theta_5}, r_{33}\right) - \theta_2 - \theta_3$$

To sum up

$$\theta_1 = \arctan\left(\frac{y}{x}\right)$$

$$\theta_2 = \operatorname{atan2}\left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}, z - d_1 - d_5 r_{33}\right) - \operatorname{atan2}(a_3\sin\theta_3, a_2 + a_3\cos\theta_3)$$

$$\theta_3 = \pm\arccos\left(\frac{\left(\sqrt{x^2+y^2} + a_1 + d_5\frac{r_{32}}{\sin\theta_5}\right)^2 + (z - d_1 - d_5 r_{33})^2 - a_2^2 - a_3^2}{2a_2a_3}\right)$$

$$\theta_4 = \operatorname{atan2}\left(\frac{r_{32}}{-\sin\theta_5}, r_{33}\right) - \theta_2 - \theta_3$$

$$\theta_5 = \operatorname{atan2}(r_{21}\cos\theta_1 - r_{11}\sin\theta_1, r_{22}\cos\theta_1 - r_{12}\sin\theta_1)$$

(c)

Refer to the definition of $\text{atan2}(y,x)$, it is undefined when x equals to 0.

For θ_1 does not exist, it is when the $x = 0$.

For θ_2 does not exist, it is when $sin\theta_5 = 0\ and\ z - d_1 - d_5 r_{33} = 0$

$$and\ \ a_2 + a_3 cos\theta_3 = 0$$

For θ_3 does not exist, it is when

$$\frac{\left(\sqrt{x^2+y^2}+a_1+d_5\frac{r_{32}}{\sin\theta_5}\right)^2+(z-d_1-d_5r_{33})^2-a_2^2-a_3^2}{2a_2a_3} \in (-\infty, 1) \cup (1, +\infty)$$

$$and\ sin\theta_5 = 0\ and\ 2a_2a_3 = 0$$

For θ_4 does not exist, it is when $-sin\theta_5 = 0\ and\ r_{33} = 0$

For θ_5 does not exist, it is when $r_{22}cos\theta_1 - r_{12}sin\theta_1 = 0$

(d)

Show in the python code ‘cw2_Questions3.py’ class ‘YoubotTemplate’ function ‘inverse_kinematics_closed’.

(e)

Show in the python code ‘cw2_Questions3.py’ class ‘YoubotTemplate’ function ‘inverse_kinematics_jac’.

(f)

Singularities is the configuration of a robot when the number of DOF is less than the dimension in which the robot operates, e.g. two revolute joints with identical axes. This causes instability in robot (infinite inverse kinematics solution, infinite joint velocity). It can occur when the manipulator is fully stretched out or two joints axes is fully aligned. By analyzing Jacobian matrix, singularities occur when

$$\det(J) = 0\ or\ \det(JJ^T) = 0$$

Code：

```
# (Q3e) Check for singularity
JT = np.transpose(J)
D = LA.det(J*JT)

if D == 0:
    print 'Singularity'
    noise = 1.003
    theta = [noise * x for x in theta]  # Add a bit of noise to theta to move it out of singula
    k = k + 1
    continue
```

Questions4

(a)

The code of this question named '4a.py'. From ROS bag 'data_q4a.bag', there are 10 time point given, and the positions and velocities of five joints are also can acquire from this bag.

For this situation, cubic model was chosen to perform trajectory planning in joint space, because the information given just contains the time point and the positions and velocities, but the acceleration information cannot be get in this case, so quantic polynomial do not have enough values.

From the cubic polynomial:

$$\begin{bmatrix} 1 & t_i & {t_i}^2 & {t_i}^3 \\ 0 & 1 & 2t_i & 3{t_i}^2 \\ 1 & t_f & {t_f}^2 & {t_f}^3 \\ 0 & 1 & 2t_f & 3{t_f}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_i \\ v_i \\ q_f \\ v_f \end{bmatrix}$$

10 time points divide time series to 9 parts, and in each t_i to t_f, and get values $\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$ of five joints.

In my program, 4 time points are added in each t_i to t_f. Because we have already had the value of a_0, a_1, a_2, a_3 for five joints each time part, so new positions and velocities can be calculated via cubic polynomial.

After publishing the result, the display of the Gazebo simulator is below, and all check points become green.

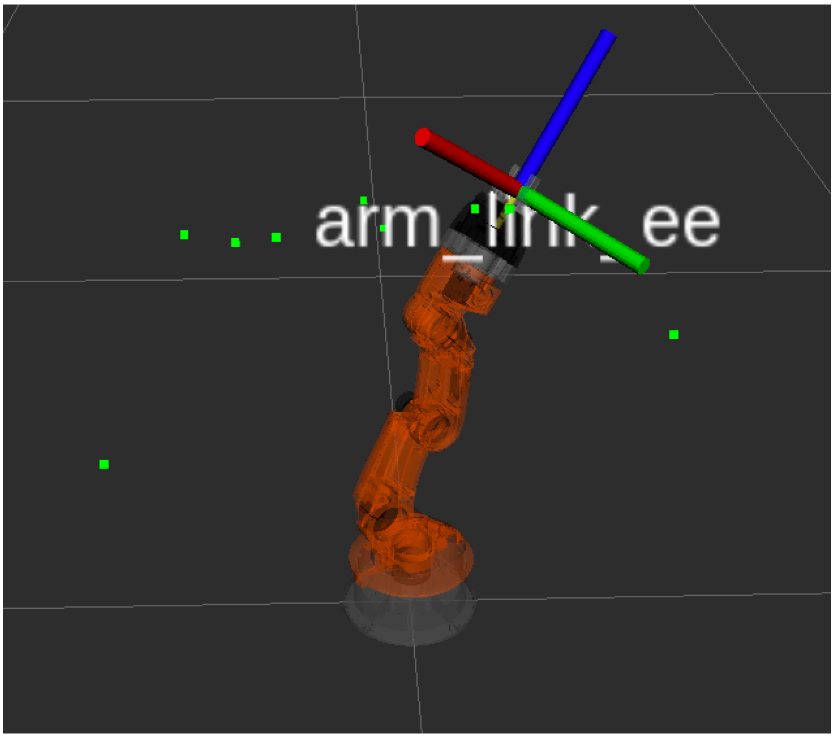


figure 1 The result of 4a in Gazebo

(b)

The code of this question named '4b_class.py'. From ROS bag 'data_q4b.bag', there are 10 time point given, and the quaternion and position x, y, z in Cartesian space are also can acquire from this bag.

Quaternion can be converted to rotation matrix via:

$$Rotaion\ Matrix = \begin{bmatrix} 1 - 2{q_y}^2 - 2{q_z}^2 & 2q_xq_y - 2q_zq_w & 2q_xq_z + 2q_yq_w \\ 2q_xq_y + 2q_zq_w & 1 - 2{q_x}^2 - 2{q_z}^2 & 2q_yq_z - 2q_xq_w \\ 2q_xq_z - 2q_yq_w & 2q_yq_z + 2q_xq_w & 1 - 2{q_x}^2 - 2{q_y}^2 \end{bmatrix}$$

And

$$T = \begin{bmatrix} Rotaion\ Martix & & & x \\ & & & y \\ & & & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, we have 10 T matrix. Screw Path can be used in pose interpolation.

$$T(t) = T_i e^{\log(T_i{}^{-1}T_f)t} \qquad\qquad t \in [0,1]$$

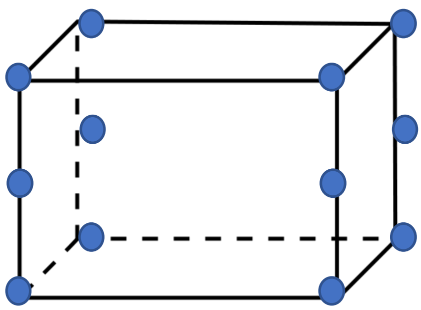
10 time points divide time series to 9 parts, and in each part, we have T_i and T_f. In my program, I add 4 points in each t_i to t_f, so t should be 0.2, 0.4, 0.6, 0.8, and we can get four new T matrix between t_i to t_f.

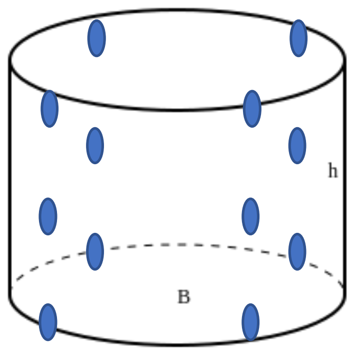
After all, use KDL inverse kinematic to convert T from Cartesian space to Joint space, and positions of 5 joints in Joint space can be get.

(c)

The code of this question named '4c_class.py'. From ROS bag 'data_q4c.bag', there are check points positions x, y, z in Cartesian space. Youbot need to pass the check points and avoid the obstacles. For check points, attractive force will drive the robot move to it, whereas the repulsive force will drive the robot to avoid the obstacles.

There are 5 obstacles in this question: three cuboids and two cylinders. We can get the size of each obstacles from Gazebo GUI, and the information of the center of each obstacle are in topic 'linkstate'. 12 points for each obstacle to calculate repulsive force.





Each obstacle point should have a repulsive force. To simplify the process of repulsive force calculation, the repulsive force is the closest obstacle point for each joint.

Algorithm 1 Gradient Descent Algorithm

procedure GRADIENT DESCENT($\vec{q}_i, \zeta_i, \eta_i, \rho_0$)
  $\vec{q}_0 \leftarrow \vec{q}_i,\ i \leftarrow 0$
  repeat
    $\vec{q}_{i+1} \leftarrow \vec{q}_i + \alpha_i \frac{\tau(\vec{q}_i)}{||\tau(\vec{q}_i)||}$
    $i \leftarrow i + 1$
  until $||\vec{q}_i - \vec{q}_f|| < \epsilon$
  return $< \vec{q}_0, \vec{q}_1, \vec{q}_2, \ldots, \vec{q}_i >$
end procedure

Another point need to mention is the publish for each new point. When we do the calculation process above, the new point will be published once the new position get solved.

All the processes are in callback function. When the new Joint State message was send from topic, the callback function will be executed. The next check point give robot an attractive force and the closet obstacle point give robot repulsive force. All five joints have attractive force and repulsive force. The formulas are below:

$$F_{\mathrm{att},i}(\vec{q}) = \begin{cases} -\zeta_i(o_i(\vec{q}) - o_i(\vec{q}_f)), & ||o_i(\vec{q}) - o_i(\vec{q}_f)|| \leq d \\ d\zeta_i \frac{o_i(\vec{q}) - o_i(\vec{q}_f)}{||o_i(\vec{q}) - o_i(\vec{q}_f)||}, & ||o_i(\vec{q}) - o_i(\vec{q}_f)|| > d \end{cases}$$

$$F_{\mathrm{rep},i}(\vec{q}) = \eta_i \left(\frac{1}{\rho(o_i(\vec{q}))} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(o_i(\vec{q}))} \nabla \rho(o_i(\vec{q}))$$

where $\nabla \rho(o_i(\vec{q})) = \frac{o_i(\vec{q}) - b}{||o_i(\vec{q}) - b||}$, b is the closest obstacle

Then combine this to force in Torques.

$$\tau(\vec{q}) = \sum_{i=1}^{n} J_{o_i}^T(\vec{q}) F_{\text{att},i}(\vec{q}) + \sum_{i=1}^{n} J_{o_i}^T(\vec{q}) F_{\text{rep},i}(\vec{q})$$

The new point can be solved via the Gradient Descent algorithm, and publish the new point.

If

$$||\vec{q_i} - \vec{q_f}|| < \epsilon$$

It means the robot have arrived the current check point, so the next check point will replace current check point. When the robot passes all the check point, the program will finish.

There are servals parameter need to set in this problem, and it is time-consuming to do this, for this problem, I do not know how to set it, but it can be get from multiply experiment.

(d)

In this question, the obstacles are same as the 4(c), but the check points are different, some of the check points are out of the workspaces.

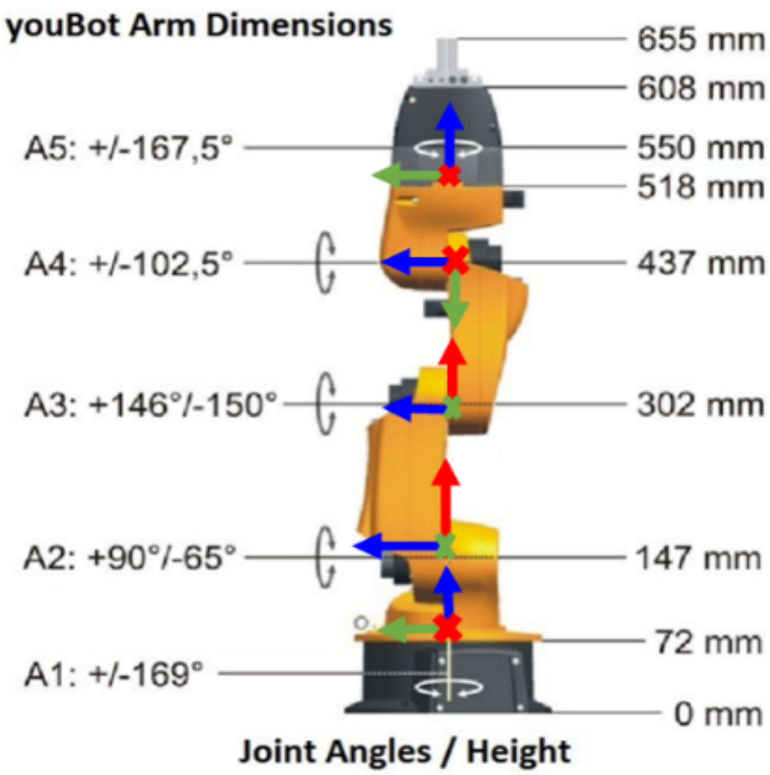
If the current check point is out of the workspace, $||q_i - q_f||$ will keep around a value, but it cannot be less than ϵ, so the next check point cannot be set.

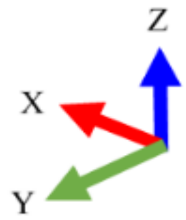
$$||\vec{q_i} - \vec{q_f}|| < \epsilon$$

We can judge the result of $||q_i - q_f||$, and if it keep around a value, and this value is the minimum value of $||q_i - q_f||$ for all steps in Gradient Descent, we can change the attractive point.

Questions5

Derive Jacobian matrix for YouBot manipulator using PoE.





$L_1 = 75$

$L_2 = 302 – 147 = 155$

$L_3 = 437 – 302 = 135$

$L_4 = 518 – 437 = 81$

θ_2 = angle of rotation of link 2

$\theta_2 + \theta_3 =$ angle of rotation of link 3

$\theta_2 + \theta_3 + \theta_4$ = angle of rotation of link 4

And

$\omega_1 = (0, 0, 1)$

$q_1 = (0, 0, 0)$

$v_1 = (0, 0, 0)$

$\omega_2 = (0, 1, 0)$

$q_2 = (0, 0, L_1)$

$v_2 = (-L_1, 0, 0)$

$\omega_3 = (0, 1, 0)$

$q_3 = (L_2\sin\theta_2, 0, L_1 + L_2\cos\theta_2)$

$v_3 = (-L_1 - L_2\cos\theta_2, 0, L_2\sin\theta_2)$

$\omega_4 = (0, 1, 0)$

$q_4 = (L_2\sin\theta_2 + L_3\sin(\theta_2 + \theta_3), 0, L_1 + L_2\cos\theta_2 + L_3\cos(\theta_2 + \theta_3))$

$v_4 = (-L_1 - L_2\cos\theta_2 - L_3\cos(\theta_2 + \theta_3), 0, L_3\sin(\theta_2 + \theta_3) + L_2\sin(\theta_2)$

$\omega_5 = (0, 0, 1)$

$q_5 = (L_2\sin\theta_2 + L_3\sin(\theta_2 + \theta_3) + (L_2\sin\theta_2 + L_3\sin(\theta_2 + \theta_3) + L_2\cos\theta_2 + L_3\cos(\theta_2 + \theta_3) + L_4\cos(\theta_2 + \theta_3 + \theta_4))$

$v_5 = (0, - L_2\sin(\theta_2) - L_3\sin(\theta_2 + \theta_3) - L_4\sin(\theta_2 + \theta_3 + \theta_4), 0)$

Space Jacobian for the Youbot manipulator can be derived using the following formula:

$$J_s(\theta) = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 & \omega_5 \\ 0 & v_2 & v_3 & v_4 & v_5 \end{bmatrix}$$

Questions6