# Optimal Inverse Kinematics of PRRR Planner Manipulator

**Muhammad Ilyas Raza** m.raza@innopolis.university

## 1. Abstract

This assignment present the Calculation of the inverse kinematics of PRRR planar robots for tracking from initial to desired point .The proposed method based on the choice of appropriate initial joint trajectories that satisfy the kinematic constraints to be used as inputs for a optimization algorithm, allows for the optimization for cost functions,such as kinetic energy and Potential, and can provide solutions for both linear and nonlinear trajectories.here i have used proposed the analytical technique and and also quadratics algorithm to find out the solution,and compare the both method and result shown in graphic representation.

## 2. Problem statement

Redundant robot PRRR is shown in figure 1,The cost function in term of the kinetic energy and potential energy is given and constraint for this function is given in form of end effector's velocity in equation below.

$$min_{\dot{q}}\mathcal{J}_c = w_k\frac{1}{2}\dot{\mathbf{q}}^T\mathbf{D}(\mathbf{q})\dot{\mathbf{q}} + w_p\frac{\partial \mathbf{U}(\mathbf{q})}{\partial \mathbf{q}}^T\dot{\mathbf{q}} \tag{1}$$

While respecting the constraints on the desired velocity:

$$\dot{\boldsymbol{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{2}$$

The objective is to minimize the cost function using the constraint to desire points to get the value of the joints velocity $\dot{q}$ , after that we will integrate the these joint velocities to get the value for joints angles

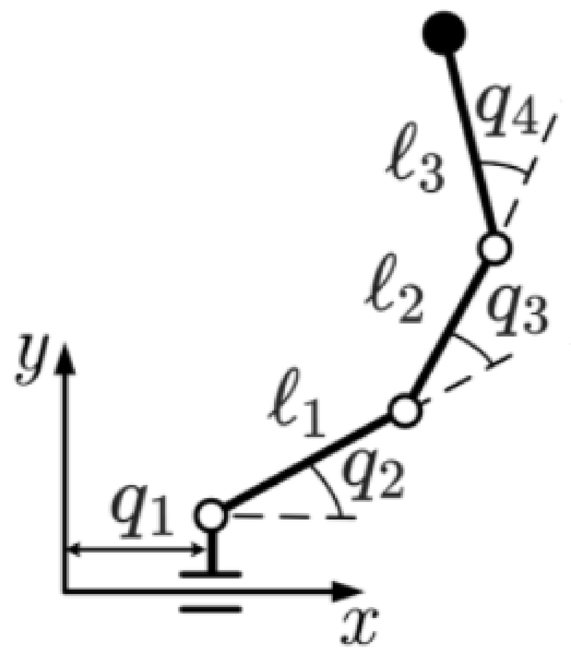



Figure 1: PRRR Robot physical representation

## 3. General Analytical Solution Using Lagrangian Multiplier

There is the fundamental theorem of linear algebra, it can be shown that there exists some $^*R^m$ (called the Lagrange multiplier) that satisfies following relation

$$\nabla J_c + \frac{\delta \dot{x}}{\delta \dot{q}} \lambda = 0 \tag{3}$$

where $j_c$ is given is in equation (1) Taking the gradient of the $J_c$ and derivative of $\dot{x}$ the following results are obtained:

$$\delta J(q) + \frac{\delta x}{\delta \dot{q}} \lambda = 0 \tag{4}$$

$$\omega_k \dot{q} D + J\lambda = -\omega_p \delta u_q \tag{5}$$

$$J(q)\dot{q} = \dot{x} \tag{6}$$

$$\begin{bmatrix} \omega_k D & J \\ J & 0 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} -\omega_p \delta q_u \\ \dot{x} \end{bmatrix} \tag{7}$$

the solutions to the first-order necessary conditions can be obtained, after some manipulation, the following the values for the $\dot{q}$ and $\lambda$ were found as:

$$\dot{q} = G\dot{x} + (I - GJ)(\omega_k D)^{-1} c \tag{8}$$

c is equal to product of gradient of U and $w_p$ , here $w_p$ is weight constant for potential energy and $w_k$ is weight constant of the kinetic energy,below i have defined the values of G and B as below where j is Jacobean and D is dynamics matrix of the PRRR robot.

$$G = (\omega_k D)^{-}1 J^T B \tag{9}$$

$$B = (J(\omega_k D)^{-1} J^T)^{-1} \tag{10}$$

## 4. Case:1 When D=I and U=0

In first case we will consider that our dynamics matrix is identity matrix, and potential energy of the robot is zero then value of the c become zero and our expression in equation (8) will become simpler

$$\dot{q} = G\dot{x} \tag{11}$$

due to identity value of the matrix D and $W_K = 1$ value of the G and B become simpler the final expression for G is given below as

$$G = J^T (JJ^T)^{-1} \tag{12}$$

The final value of the for equation (8) will be given as

$$\dot{q} = J^T (JJ^T)^{-1} \dot{x} \tag{13}$$

**This result is same as pseudo inverse for constraint equation given in (2) , so there is no difference between this method and pseudo inverse method due to case 1 condition**

### 4.1 Implementation and Simulation

For implementation of the system we need to calculate the forward kinematics and Jacobean matrix,The forward kinematics of PRRR robot is given below

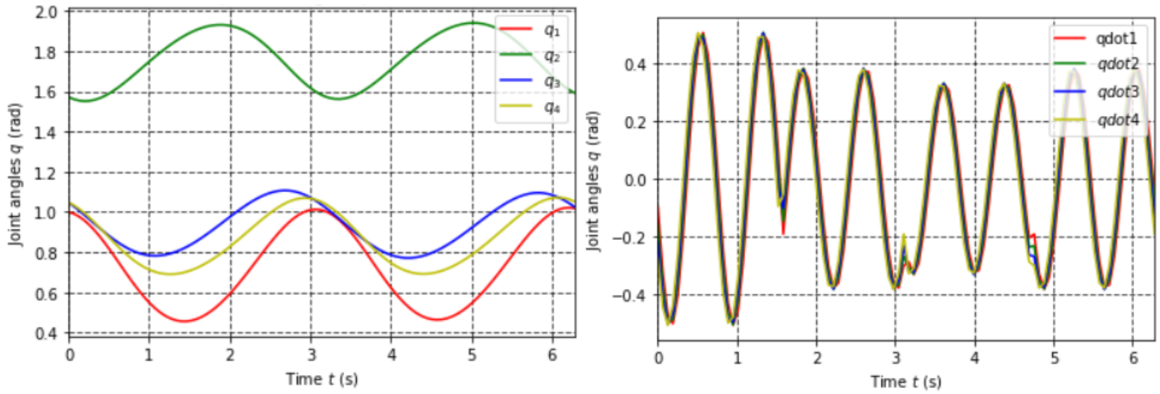
$$F_k = \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} q_0 + l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q + 3) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q + 3) \end{bmatrix} \tag{14}$$

and Jacobin of of the PRRR has been calculated below which is 2*4 matrix

$$\begin{bmatrix} 1 & -l_2 \sin(q_2) - l_3 \sin(q_{23}) - l_4 \sin(q_{234}) & -l_3 \sin(q_{23}) - l_4 \sin(q_{234}) & -l_4 \sin(q_{234}) \\ 0 & l_2 cos(q_2) + l_3 \cos(q_{23}) + l_4 \cos(q_{234}) & +l_3 \cos(q_{23}) + l_4 \cos(q_{234}) & l_4 \cos(q_{234}) \end{bmatrix} \tag{15}$$

$$q_{23} = q_2 + q_3 q_{234} = q_2 + q_3 + q_4 \tag{16}$$

using the value of the Jacobin matrix and i assumed the value of the twisted velocity for end effector $-cos(t), -sin(t)$, which produce the circle trajectory,after solving the equation (8) i got the array for $\dot{q}$ , after integrate the $\dot{q}$ i got the array of the joints angle q , i fed this array into forward kinematic function and draw the angle and also animation for trajectory which follow the circle as can be seen in figure 2




(i) plot for joint angles (ii) Plot for joint velocity

Figure 2: optimization,case:1 when D=I and U=0

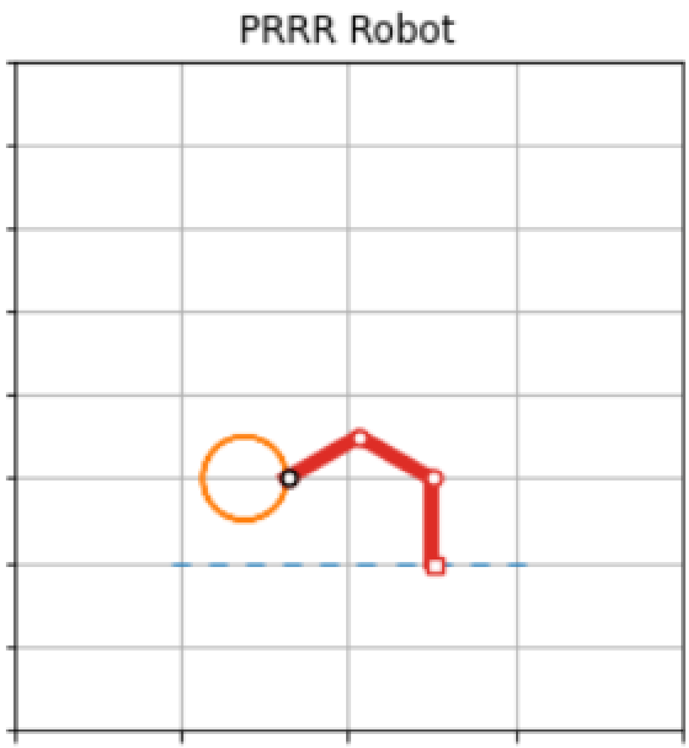



Figure 3: animation pf Circular trajectory following by robot

## 5. Case:2 When include dynamics matrix and Potential energy

In second case i have calculated the dynamics matrix of the PRRR robot which extensive and long matrix has been calculated in colab file, i also have calculated the potential energy for the robot and gradient of the potential energy for each link has calculated which can be seen in here in the equation below

$$U_1 = 0 \tag{17}$$

$u_1$ is the potential energy of the link one which is the prismatic and had no perpendicular distance, so it should be zero

$$U_2 = m_2 g l_{c2} \sin(q_2) \tag{18}$$

where lc2=l2/2

$$U_3 = m_2 g(l_2 sin(q_2) + l_{c3} \sin(q_2 + q_3)) \tag{19}$$

$$U_4 = m_4 g(l_2 sin(q_2) + l_3 \sin(q_2 + q_3) + l_{c4} \sin(q_2 + q_3 + q_4)) \tag{20}$$

the total potential energy of the robot is sum of all joints

$$U = U_1 + U_2 + U_3 + U_4 \tag{21}$$

The gradient of the potential energy is calculated in following matrix

$$\delta u^T = \begin{bmatrix} 0 \\ g l_2 m_2 \cos(q_2)/2 + g m_3(l_2 \cos(q_2) + l3c(q_{23})/2) + g m4(l2 \cos(q_2) + l3 \cos(q_{23}) + l4 \cos(q_{234})/2) \\ g l_3 m_3 \cos(q_{23})/2 + g m_4(l_3 \cos(q_{23}) + l_4 \cos(q_{234}/2) \\ g l_4 m_4 \cos(q_{234})/2 \end{bmatrix} \tag{22}$$

Now I have used the analytical solution for given in equation (8)

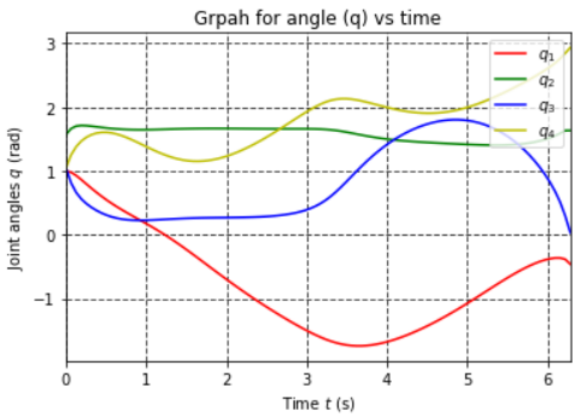
$$\dot{q} = G\dot{x} + (I - GJ)(\omega_k D)^{-1} c \tag{23}$$

after that i have calculated the value of the G and B as given in (9) and (10).

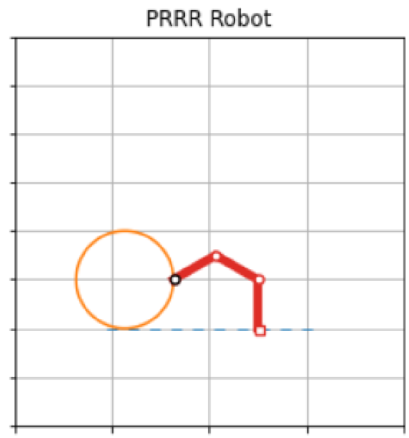
### 5.1 Implementation and Simulation

for this case i also assume that value of the twisted velocity for end effector $-cos(t), -sin(t)$, which produce the circle trajectory,after solving the equation (8) i got the array for $\dot{q}$ , after integrate the $\dot{q}$ i got the array of the joints angle q , i fed this array into forward kinematic function and draw the angle and also animation for trajectory which follow the circle as can be seen in 4

$w_k$ **and** $w_p$ **are the weight coefficient of the kinetic energy and potential energy, when i have put the value**$w_p = 0$**, solution was different and robot motion was abnormal it was colliding with other links, but when i put** $w_p = 0.5$ **robot was acting and following the trajectory in normal behavior**




(i) plot for joint angles




(ii) animation plot for circular trajectory

Figure 4: optimization,case:2 when D=D and U=potential energy

## 6. Bounded Constraints Optimization :

Now the joints velocities are bounded with following constaint

$$\dot{\mathbf{q}}_{min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{max} \tag{24}$$

Now our problem has been not linear , it would be solved by using quadratic technique . In order to solve this problem ,i have used the quadratic solver function. standard form of CVXOPT function is following :

$$min_x \frac{1}{2} x^T P x + q^T x \tag{25}$$

$$Ax = b \tag{26}$$

$$G(x) <= h \tag{27}$$

Our problem with bounded constraint is given below

$$min_x J_c = \omega_k \frac{1}{2} \dot{q}^T D \dot{q} + \omega_p \delta U^T \dot{q} \tag{28}$$

$$\dot{x} = J\dot{q} \tag{29}$$

$$\dot{q} < \dot{q_{max}} \tag{30}$$

$$\dot{q} < -\dot{q_{min}} \tag{31}$$

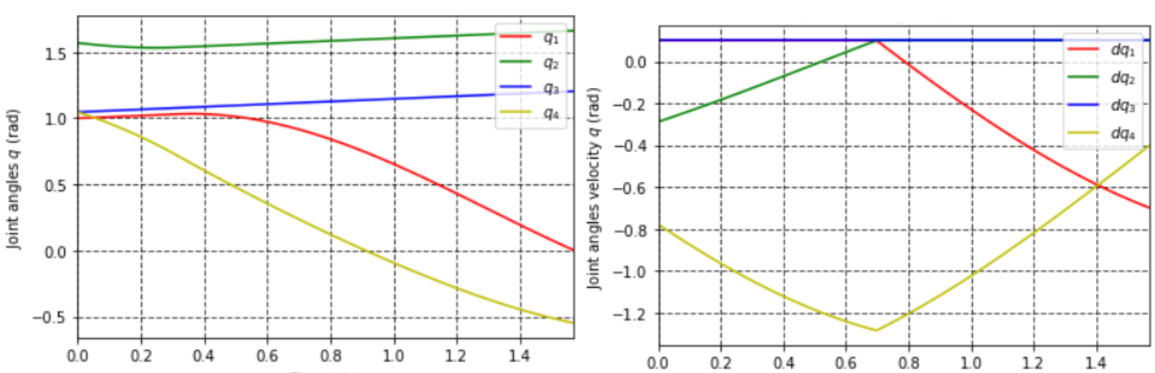
by comparing our system with standard form of the CVXOPT the these constraint P,Q,A,x,b,G,h has following values

$P = D, q^T = \delta U^T, A = J, b = \dot{x}, x = \dot{q}$

G is a 8X4 matrix, which is the stack of two 4x4 identity matrix, where x are equal to joint velocity with dimension 4X1, which makes it equal to 8X1 matrix with four maximum and four minimum values of $\dot{q}$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \\ \dot{q_4} \end{bmatrix} = \begin{bmatrix} \dot{q}_{1max} \\ \dot{q}_{2max} \\ \dot{q}_{3max} \\ \dot{q}_{4max} \\ -\dot{q}_{1min} \\ -\dot{q}_{2min} \\ -\dot{q}_{3min} \\ -\dot{q}_{4min} \end{bmatrix} \tag{32}$$

after finding the value the of the all variable i have used the CVXOPT toolbox to solve the problem and get the solution for joints velocity , then again integrate the joints velocity , desired joint angle has been obtained, graph for joints velocity and angle are below in 5 with trajectory in 6




(i) plot for joint angles

(ii) Plot for joint velocity

Figure 5: quadratic bounded optimization graph

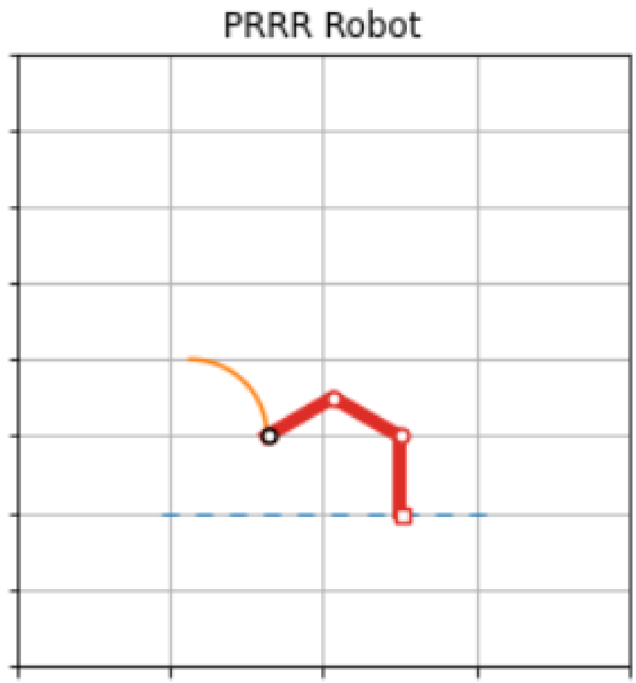



Figure 6: animation of trajectory for bounded constraints of PRRR robot

**there is one problem that when we used the bonded constraint, then robot only follow the limited trajectory, and convergence or drift solution is slow to overcome these problem below i have proposed the better solution**

## 7. Modification to tackle Integration Drift:

The small modification we did helped us in achieve the make the integration more fast and universal we will following constraint in the equation:

$$\begin{aligned} &\min_{\dot{q}} \mathcal{J}_c = w_k \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{D}(\mathbf{q})\dot{\mathbf{q}} + J(q)\dot{q} - \dot{x}_2^2 \\ &S.T: \quad \dot{\mathbf{q}}_{min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{max} \end{aligned} \tag{33}$$

$$\begin{aligned} J(q)\dot{q} - \dot{x}_2^2 &= (J(q)\dot{q} - \dot{x})^T(J(q)\dot{q} - \dot{x}) \\ &= \dot{q}^T J J^T \dot{q} - \dot{x}^T J\dot{q} - \dot{q}^T J^T \dot{x} + \dot{x}^T \dot{x} \\ &= 1/2\dot{q}^T J J^T \dot{q} - 1/2\dot{x}^T J\dot{q} - \dot{q}^T J^T \dot{x} \\ &= 1/2\dot{q}^T J J^T \dot{q} - \dot{x}^T J\dot{q} \end{aligned} \tag{34}$$

to overcome the error between end effector initial position to desired position we will introduce the following error term Let

$$e = x_d - x_e \dot{e} = \dot{x_d} - \dot{x_e} \tag{35}$$

$$J\dot{q} = j(\dot{x_d} - \dot{e}) \tag{36}$$

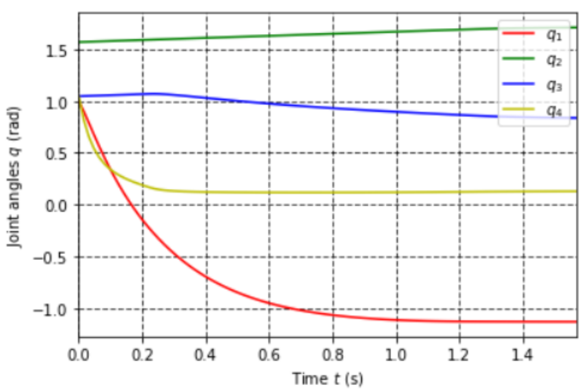
where $\dot{e} + Ke = 0$ K is a positive definite (usually diagonal) matrix, the system (3.71) is asymptotically stable. The error tends to zero along the trajectory with a convergence rate that depends on the eigenvalues of matrix K, the larger the eigenvalues, the faster the convergence, and end effector will follow the trajectory and reached the final position with more accurately. thus $J(q)\dot{q} - \dot{x}_2^2$ could be redefine as:

$$v = -J(q)\dot{q} + \dot{x_d} + ke \tag{37}$$

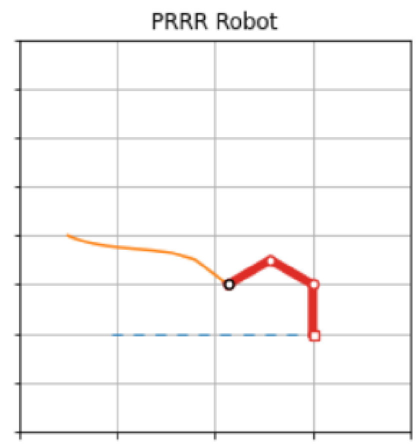
After analytical findings in our inequality constraints we added them to our cost function:

$$J_c = \omega_k \frac{1}{2} \dot{q}^T (D + J^T J)\dot{q} + (\delta U - J^T v^T)\dot{q} \tag{38}$$

Now applying this cost function and using the same bounded constraint the optimization solution would obtain very fast, the obtained graph and trajectory can be seen below in 7




(i) plot for joint angles




(ii) Plot for trajectory

Figure 7: quadratic bounded optimization graph for drift velocity

## 8. Conclusion

This task was about to calculate the inverse kinematics of the PRRR robot,i have analyzed the different technique with different linear and bounded constraints, i have learnt about analytical technique and also used the CVXOPT tool to solve the quadratic problem.

Colab link is Here