Foundations of Robotics: Project-1

Name: Shubham Gupta
Netid: sg7761

Fall 22

1 Objective

Implement the direct and inverse kinematics for the given manipulator: SCARA.

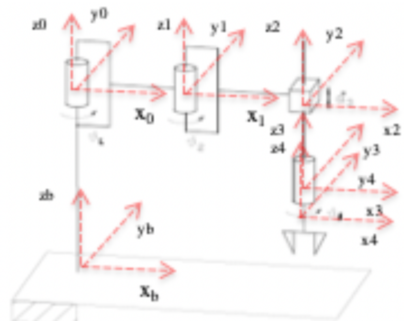


Figure 1: DH diagram for SCARA manipulator.

2 Part-1

Implemented first order kinematic inversion using Matlab and Simulink for the given trajectory using the following methods:

1. Jacobian inverse
2. Jacobian transpose

2.1 Jacobian inverse

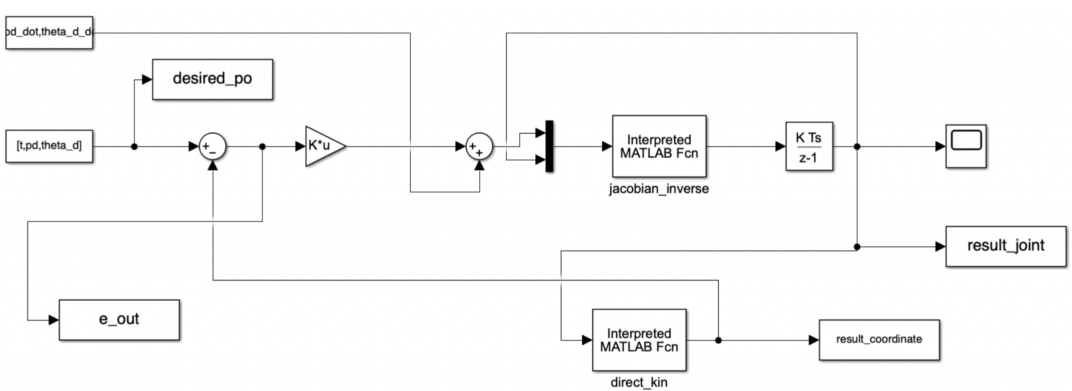


Figure 2: Simulink block diagram for inverse kinematics using Jacobian inverse.

The model uses interpreted matlab blocks to use the existing M files defined in the project structure. The model uses a gain value of 500 in an identity matrix to reach its optimum solution.

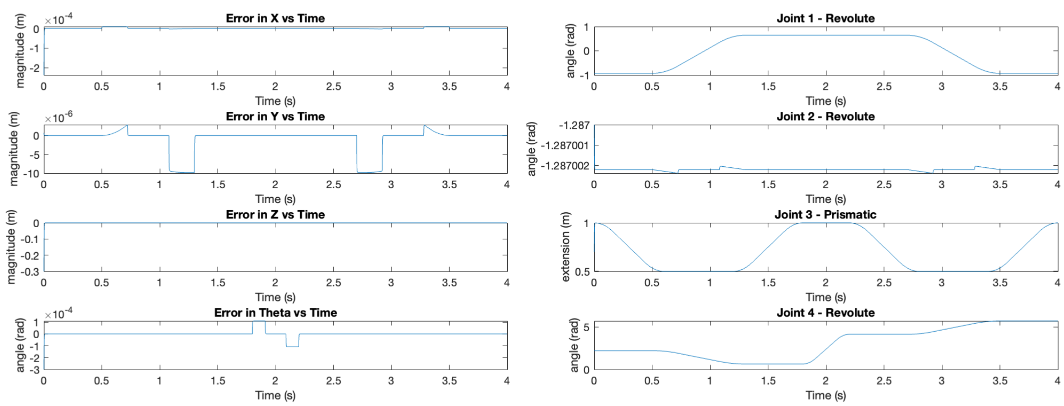
$$HT\_matrix = \begin{bmatrix} \cos(\theta_1+\theta_2+\theta_4) & -\sin(\theta_1+\theta_2+\theta_4) & 0 & a_2\ \cos(\theta_1+\theta_2)+a_1\ \cos(\theta_1) \\ \sin(\theta_1+\theta_2+\theta_4) & \cos(\theta_1+\theta_2+\theta_4) & 0 & a_2\ \sin(\theta_1+\theta_2)+a_1\ \sin(\theta_1) \\ 0 & 0 & 1 & base-d3 \end{bmatrix} \tag{1}$$

$$jacobian\_matrix = \begin{bmatrix} -a_2\ \sin(\theta_1+\theta_2)-a_1\ \sin(\theta_1) & -a_2\ \sin(\theta_1+\theta_2) & 0 & 0 \\ a_2\ \cos(\theta_1+\theta_2)+a_1\ \cos(\theta_1) & a_2\ \cos(\theta_1+\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{2}$$

$$q\_dot = inv(jacobian\_matrix) \cdot Ke \tag{3}$$

In Equation2 we set the third column value as $[0, 0, -1, 0]$ as the direction is opposite to the given trajectory.

Graphs



(a) (b)

Figure 3: Graphs generated for inverse kinematics of SCARA using Jacobian Inverse.**(a)**Error in Operational space of X, Y, Z and Theta across time. **(b)**Value of joint angles / extension across time.

2.2 Jacobian transpose

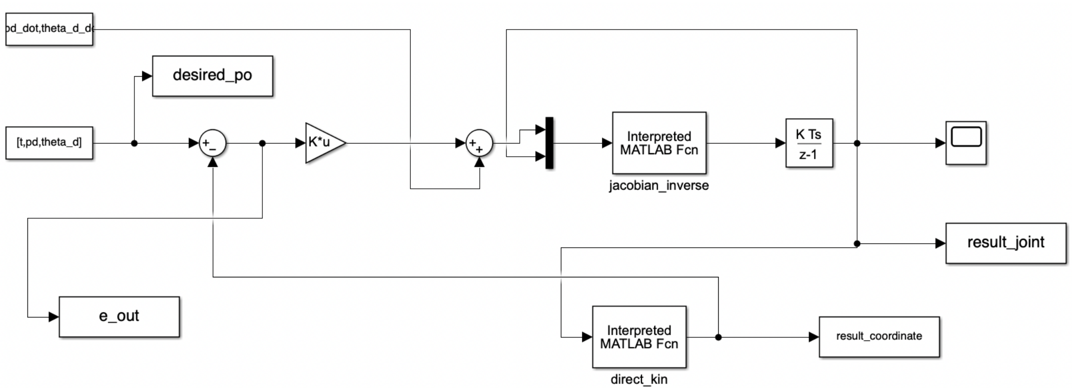
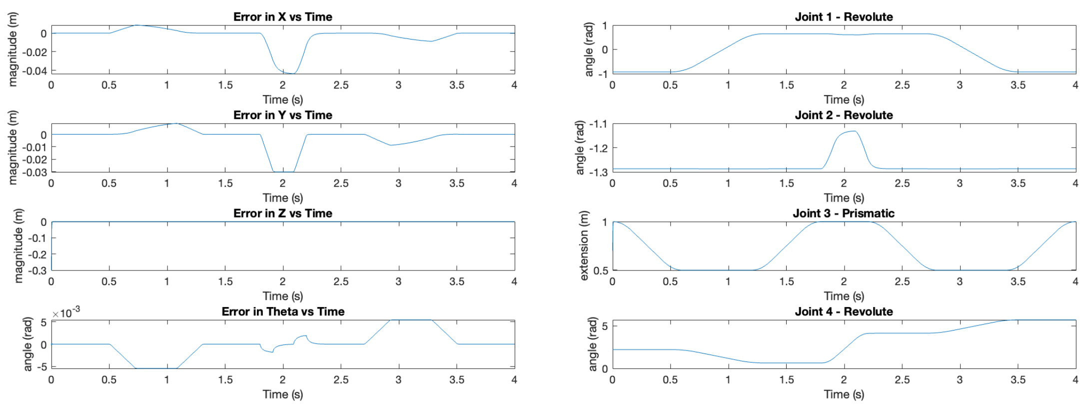


Figure 4: Simulink block diagram for inverse kinematics using Jacobian transpose.

The model uses interpreted matlab blocks to use the existing M files defined in the project structure. The model uses a gain value of 500 in an identity matrix to reach its optimum solution. The homogeneous equation (1) and jacobian matrix (2) from 2.1 will remain the same. Only the final calculation of q_dot will change.

$$q\_dot = transpose(jacobian\_matrix) \cdot Ke \tag{4}$$

Graphs



(a) (b)

Figure 5: Graphs generated for inverse kinematics of SCARA using Jacobian Transpose. **(a)** Error in Operational space of X, Y, Z and Theta across time. **(b)** Value of joint angles / extension across time.

3 Part-2

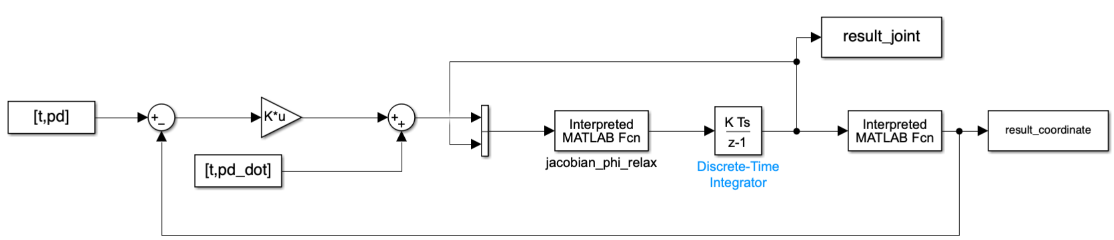


Figure 6: Simulink block diagram for Jacobian pseudo inverse.

Implemented Jacobian pseudo inverse algorithm for maximizing the mechanical joint limits (relax ϕ). ϕ is defined as the sum of all revolute joint angles $(\theta_1 + \theta_2 + \theta_3)$. Since there is no rotation about the Z axis, the sum of angles of the revolute joints is the angular velocity in the Z axis. Therefore we eliminate this row from the jacobian matrix and are left with a non square matrix. This is why pseudo inverse algorithm needs to be applied to relax ϕ.

$$jacobian\_matrix = \begin{bmatrix} -a_2 \sin(\theta_1 + \theta_2) - a_1 \sin(\theta_1) & -a_2 \sin(\theta_1 + \theta_2) & 0 & 0 \\ a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) & a_2 \cos(\theta_1 + \theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{5}$$

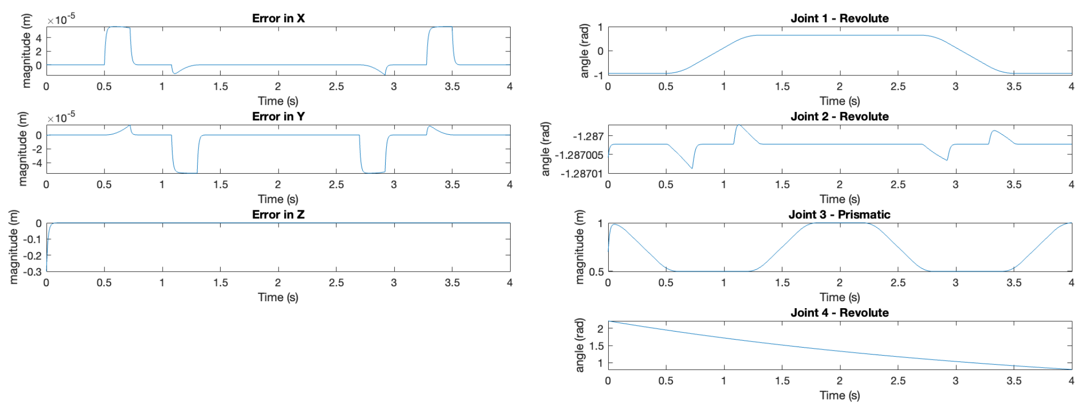
$$pseudo\_inverse\_matrix = jacobian\_matrix^T / jacobian\_matrix \cdot jacobian\_matrix^T \tag{6}$$

$$\dot{q} = pseudo\_inverse\_matrix \cdot Ke + (I_{4\times4} - pseudo\_inverse\_matrix \cdot jacobian\_matrix) \cdot \dot{q}_0 \tag{7}$$

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q} \right)^T \tag{8}$$

$$w(q) = \frac{-1}{2n} \cdot \sum_{i=1}^{n} (\frac{q_1 - \bar{q}_l}{q_1 M - q_i m})^2 \tag{9}$$

Graphs



(a) (b)

Figure 7: Graphs generated for Jacobian pseudo inverse of SCARA.**(a)**Error in Operational space of X, Y, and Z across time. **(b)**Value of joint angles / extension across time.