Robot Kinematics: Forward and Inverse Analysis

Computational Robotics Templates

November 24, 2025

Abstract

This document presents a comprehensive analysis of robot kinematics using the Denavit-Hartenberg (DH) convention. We explore forward kinematics for a 3-DOF planar manipulator and 6-DOF articulated arm, implement inverse kinematics solutions using both geometric and numerical methods, compute the Jacobian matrix for velocity analysis, and visualize the robot workspace. The analysis demonstrates the mathematical foundations essential for robot motion planning and control.

1 Introduction

Robot kinematics is the study of motion without considering the forces that cause it. Forward kinematics determines the end-effector position and orientation given joint angles, while inverse kinematics solves for joint angles given a desired end-effector pose. These fundamental concepts are critical for robot programming, trajectory planning, and real-time control.

2 Mathematical Framework

2.1 Denavit-Hartenberg Convention

The DH convention provides a systematic method for assigning coordinate frames to each link. The transformation from frame $i-1$ to frame i is:

$$
{}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

where θ_i is the joint angle, d_i is the link offset, a_i is the link length, and α_i is the link twist.

2.2 Forward Kinematics

For an n-DOF manipulator, the end-effector pose is:

$$^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot \ldots \cdot {}^{n-1}T_n = \prod_{i=1}^{n} {}^{i-1}T_i \tag{2}$$

2.3 Jacobian Matrix

The Jacobian relates joint velocities to end-effector velocities:

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3}$$

For a revolute joint i:

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} \tag{4}$$

2.4 Inverse Kinematics

The inverse kinematics problem solves for $\mathbf{q}$ given ^0T_n. For numerical solutions, the Newton-Raphson method is:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^{-1}(\mathbf{q}_k) \cdot \Delta\mathbf{x} \tag{5}$$

where $\Delta\mathbf{x}$ is the pose error.

3 Computational Analysis

3.1 3-DOF Planar Manipulator Analysis

3.2 6-DOF Articulated Manipulator

3.3 Trajectory Generation

3.4 Singularity Analysis

3.5 Animation Frames: Robot Motion

4 Results and Discussion

4.1 Forward Kinematics Results

For the test configuration with joint angles $q = [??°, ??°, ??°]$:

Table 1: Forward Kinematics Results

Parameter	**Value**	**Unit**
End-effector X	**??**	m
End-effector Y	**??**	m
End-effector orientation	**??**	degrees
Workspace outer radius	**??**	m
Workspace inner radius	**??**	m

4.2 Inverse Kinematics Results

The inverse kinematics solver successfully found a solution for target position (**??**, **??**) m:

Table 2: Inverse Kinematics Solution

Joint	**Angle**	**Unit**
Joint 1	**??**	degrees
Joint 2	**??**	degrees
Joint 3	**??**	degrees
Position error	**??**	mm

4.3 Trajectory Analysis

The trapezoidal velocity profile trajectory yielded:

- Path length: **??** m
- Maximum end-effector velocity: **??** m/s
- Maximum manipulability index: **??**
- Minimum condition number: **??**

4.4 6-DOF Robot Results

The 6-DOF articulated manipulator end-effector positions:

- Home configuration: (**??**, **??**, **??**) m
- Working configuration: (**??**, **??**, **??**) m

5 Conclusion

This analysis demonstrated the fundamental concepts of robot kinematics including forward and inverse kinematics using the DH convention, Jacobian computation for velocity analysis, workspace visualization, trajectory generation with trapezoidal velocity profiles, and singularity analysis. The numerical methods presented are applicable to manipulators of varying complexity and form the foundation for advanced robot motion planning and control algorithms.