Planar 2-DOF Robot Kinematics

Abstract

This project presents the analytical and computational modeling of a planar two-degree-of-freedom (2-DOF) robotic manipulator. The system was modeled using rigid-body kinematics, with forward kinematics derived to map joint space variables to Cartesian end-effector position. The inverse kinematics problem was solved analytically using geometric principles and the law of cosines, including explicit reachability conditions and multiple solution branches corresponding to different arm configurations.

A Python-based simulation framework was implemented to validate the analytical results through numerical testing and visualization. The Jacobian matrix was derived to relate joint velocities to end-effector velocities, enabling analysis of singular configurations where the manipulator loses instantaneous mobility. These singularities were shown to coincide with configurations where inverse kinematics solutions collapse, providing a consistent physical interpretation.

This project demonstrates how geometry, trigonometry, and linear algebra underpin robotic motion and control in articulated robotic manipulators. By focusing on kinematics and singularity analysis, the work highlights foundational concepts essential to robotics and machine intelligence systems, forming a basis for more advanced topics such as dynamics, control, and motion planning.

Introduction

Robotic manipulators are mechanical systems composed of rigid links connected by joints, designed to position and orient an end-effector within a workspace. Among the simplest and most widely studied classes of manipulators are planar robotic arms, whose motion is constrained to two dimensions. Despite their simplicity, planar manipulators capture many of the fundamental challenges present in real-world robotic systems, including nonlinear kinematics, multiple configurations for a single task, and singularities that limit motion.

Kinematics forms a fundamental mathematical layer of robotics, providing the mapping between joint space and task space that underlies motion planning and control. Forward kinematics determines the position of the end-effector given a set of joint variables, while inverse kinematics addresses the more challenging problem of computing joint variables required to reach a desired Cartesian position. In robotic systems, inverse kinematics is central to motion planning, trajectory generation, and control, and its solutions are often non-unique or nonexistent depending on system configuration and task constraints.

In addition to position analysis, differential kinematics plays a critical role in understanding instantaneous motion. The Jacobian matrix relates joint velocities to end-effector velocities and provides insight into the manipulator's local mobility. Configurations in which the Jacobian loses rank, known as singularities, correspond to physical situations where the robot loses the ability to move in certain directions. Identifying and interpreting such configurations is essential for safe and effective robotic operation.

This project focuses on a planar 2-DOF robotic manipulator as a deliberate design choice. The 2-DOF system is mathematically tractable while still exhibiting key robotic phenomena such as multiple inverse kinematics solutions and kinematic singularities. By analyzing this system in depth, the project aims to demonstrate how geometry, trigonometry, and linear algebra form the core mathematical tools used in robotics and machine intelligence, providing a clear conceptual foundation for more complex robotic systems.

System Definition

- Robot: planar 2-DOF manipulator
- Joints: 2 revolute joints
- Base: fixed at origin
- Coordinate frame: x horizontal, y vertical
- Link lengths: $l1$, $l2$
- Joint limits: $\theta1, \theta2 \in [-\pi, \pi]$ (used for workspace visualization only)

Angles:

- $\theta1$: angle of Link 1 from x-axis (CCW positive)
- $\theta2$: angle of Link 2 relative to Link 1 (CCW positive)

Forward Kinematics

The end-effector position is determined by summing the position vectors of each link. Since the second link rotates relative to the first, its absolute orientation is $\theta1 + \theta2$.

The forward kinematics equations are given by:

$x = l1\cos(\theta1) + l2\cos(\theta1 + \theta2)$

$y = l1\sin(\theta1) + l2\sin(\theta1 + \theta2)$

Visualization and Validation

The forward kinematics model was validated by plotting the robotic arm for known joint configurations. The visualized configurations matched the expected physical behavior, confirming the correctness of the kinematic formulation.

Inverse Kinematics

Inverse kinematics determines the joint angles required for the end-effector to reach a desired Cartesian position. For a planar 2-DOF manipulator, a target point is reachable only if it satisfies the triangle inequality $|l1 - l2| \le r \le l1 + l2$, where r is the distance from the base to the target. When reachable, the manipulator admits two possible configurations corresponding to elbow-up and elbow-down postures.

Analytical Derivation

Using the law of cosines on the triangle formed by the two links and the target position, the second joint angle is obtained as

$\cos(\theta2) = (x^2 + y^2 - l1^2 - l2^2) / (2 \times l1 \times l2)$

which yields two possible solutions corresponding to elbow-up and elbow-down configurations. The first joint angle is then computed by resolving the relative geometry between the target direction and the internal triangle angles.

Inverse Kinematics Implementation

The inverse kinematics was implemented analytically with an explicit reachability check based on the cosine constraint. For reachable targets, both elbow-up and elbow-down solutions were computed and validated by substituting the resulting joint angles back into the forward kinematics equations.

Workspace Plots

A workspace plot is a visual map of every point the robot's end-effector can physically reach given its geometry and joint limits.

The workspace plot was generated numerically using Python. Joint angles were sampled uniformly over the range $\theta 1$, θ2 ∈[−π,π], and the corresponding end-effector positions were computed using the forward kinematics equations. The resulting Cartesian coordinates were plotted to visualise the set of all reachable points in the plane.

Why the workspace is circular or annular

For a planar 2-DOF manipulator with joint limits $\theta 1$, $\theta 2$ ∈[−π,π], the end-effector position is determined solely by the resultant reach of the two links. Because both joints can rotate fully, the arm can sweep all orientations in the plane. The set of reachable points therefore depends only on the achievable radial distances from the base, producing a radially symmetric workspace.

When the minimum reachable radius is zero, the workspace forms a filled circular disk. When the minimum radius is non-zero, the workspace becomes an annular (ring-shaped) region.

Why there is an inner hole (if there is one)

The inner boundary of the workspace corresponds to the minimum reach of the manipulator:

```
r(min) = |l1−l2|
```

If the two link lengths are unequal, the arm cannot fully fold back to the origin, leaving an unreachable central region (inner hole).

If l1 = l2, then r(min) = 0, meaning the end-effector can reach the base point, and no inner hole exists

How link lengths affect the outer boundary

The outer boundary of the workspace is defined by the maximum reach:

```
r(max) = l1 + l2
```

This occurs when the two links are collinear and fully extended. Increasing either link length linearly expands the outer radius of the workspace. The outer boundary is therefore a circle of radius l1 + l2, independent of joint angles, as long as full rotation is allowed.

How joint limits distort the shape

If joint limits are restricted (e.g. $\theta \notin [-\pi,\pi]$), the manipulator can no longer achieve all relative link orientations. This removes portions of the radial sweep, clipping the workspace and breaking its circular symmetry.

Tighter limits on θ_2 increase the inner unreachable region, while limits on θ_1 restrict angular coverage, producing sectors or crescent-shaped workspaces rather than full circles or annuli.

Why points outside this region are IK-impossible

Inverse kinematics solutions exist only for points that lie within the reachable workspace. Any point with radius

```
r > l1 + l2 OR r < |l1 - l2|
```

violates the geometric constraints of the manipulator and therefore has no real joint-angle solution.

Thus, inverse kinematics failure outside the workspace is not a numerical issue, but a physical impossibility imposed by link lengths and joint limits.

Jacobian and Singularities

The Jacobian matrix relates joint velocities to end-effector velocities. For the planar 2-DOF manipulator, the determinant of the Jacobian is proportional to $\sin(\theta_2)$, indicating singular configurations when the arm is fully extended or folded. At these configurations, the manipulator loses one degree of freedom, which is consistent with the collapse of inverse kinematics solutions. At singular configurations, the columns of the Jacobian matrix become linearly dependent, meaning that independent joint motions result in end-effector velocities in the same direction. Consequently, the manipulator loses the ability to move in one Cartesian direction, effectively losing one degree of freedom. This loss of rank implies that the manipulator cannot generate arbitrary end-effector velocities at singular configurations, highlighting the practical importance of singularity avoidance in robot motion planning.

Limitations and Future Work

The present analysis is subject to several deliberate limitations. First, the manipulator is modeled as a planar system, restricting motion to two dimensions. While this assumption simplifies visualization and mathematical analysis, real robotic manipulators operate in three-dimensional space and require additional degrees of freedom to control both position and orientation.

Second, the model considers only kinematics and neglects dynamics. Effects such as link mass, inertia, gravity, friction, and actuator constraints are not included. As a result, the analysis describes geometric feasibility and instantaneous motion but does not address forces, torques, or energy requirements, which are essential for physical implementation and control.

Third, joint limits and mechanical constraints were not explicitly modeled. In practical robotic systems, joint angle bounds and collision avoidance significantly restrict the feasible workspace and must be incorporated into planning algorithms. Additionally, numerical issues arise near singular configurations, where small joint motions can produce large or ill-conditioned end-effector velocity mappings.

Future work could extend this project in several directions. The kinematic framework could be generalized to three-dimensional manipulators with additional joints, enabling independent control of end-effector orientation. Incorporating dynamic modeling would allow the formulation of torque-based control laws and trajectory optimization. Finally, integrating feedback control and motion planning algorithms would bridge the gap between kinematic analysis and autonomous robotic operation, aligning the model more closely with real-world robotics and machine intelligence systems.