Introduction to Robotics: Lecture Notes

Based on Lecture by Professor Oussama Khatib

January 26, 2025

Contents

1 Introduction

Robotics is a multidisciplinary field involving mechanical design, control theory, computer science, and human factors. This course focuses on the mathematical foundations essential for understanding, modeling, and controlling robotic systems. We will explore kinematics, dynamics, control strategies, and advanced topics such as human-robot interaction and haptics.

Robots have evolved significantly from isolated industrial manipulators to systems that interact closely with humans, such as surgical robots, assistive devices, and humanoid robots. This evolution brings new challenges in perception, safety, and control in unstructured environments.

2 Robotic Systems and Sensors

2.1 Determining Robot Configuration

To control a robot, we must know its configuration—the positions and orientations of all its parts. Sensors play a crucial role in providing this information.

2.2 Types of Sensors

- **Encoders**: Measure joint positions (angles or displacements). For a robot with n degrees of freedom, n encoders are needed.
- **GPS (Global Positioning System)**: Provides position data in global coordinates. Limited in precision and does not provide orientation or information about individual joints.
- **Vision Systems**: Cameras and image processing algorithms can infer the robot's configuration by tracking key features or markers.
- **IMUs (Inertial Measurement Units)**: Measure acceleration and angular velocity, useful for estimating motion.

2.3 Challenges in Sensing

- **Redundancy**: Robots with many degrees of freedom require a large number of sensors.
- **Cost and Complexity**: Adding sensors increases the cost and complexity of the system.
- **Accuracy**: Sensor noise and errors affect the precision of the configuration estimation.

3 Spatial Descriptions and Transformations

3.1 Position and Orientation Representation

Understanding the position and orientation of robot parts is essential for kinematic and dynamic modeling.

3.1.1 Coordinate Frames

Robots are modeled using a series of coordinate frames attached to each link. Transformations between these frames describe the robot's configuration.

3.1.2 Rotation Matrices

As before, rotation matrices $R \in SO(3)$ are used to represent orientations.

3.1.3 Homogeneous Transformation Matrices

Homogeneous transformation matrices combine rotations and translations, allowing us to represent the pose of one frame relative to another.

3.2 Forward Kinematics

3.2.1 Denavit-Hartenberg Parameters

Denavit and Hartenberg, two Ph.D. students at Stanford in the early 1970s, developed a systematic method for assigning coordinate frames to robotic links and joints.

The D-H parameters provide a minimal representation of a kinematic chain, with four parameters per link.

- **Link Length** (a_i): Distance from Z_{i-1} to Z_i along X_i.
- **Link Twist** (α_i): Angle from Z_{i-1} to Z_i about X_i.
- **Link Offset** (d_i): Distance from X_{i-1} to X_i along Z_{i-1}.
- **Joint Angle** (θ_i): Angle from X_{i-1} to X_i about Z_{i-1}.

3.2.2 Transformation Matrices

The transformation from frame $i-1$ to frame i:

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

3.2.3 Kinematic Chains

By chaining the transformations, we can compute the pose of the end-effector:

$$T_{0n} = \prod_{i=1}^{n} {}^{i-1}T_i$$

3.3 Inverse Kinematics

Calculating the joint variables $\mathbf{q}$ required to achieve a desired end-effector pose T_d.

3.3.1 Redundancy

For robots with more degrees of freedom than necessary, inverse kinematics has infinite solutions. This redundancy can be challenging but also offers flexibility.

- **Methods to Resolve Redundancy**:
  - Minimizing joint movements.
  - Using secondary objectives (e.g., obstacle avoidance, energy efficiency).

3.3.2 Limitations of Inverse Kinematics

- Computationally intensive for high DOF robots.
- Not always practical for real-time control.
- Humans do not use inverse kinematics consciously.

4 Velocities and the Jacobian

4.1 Linear and Angular Velocities

Understanding how velocities propagate through the kinematic chain is essential for control and dynamic analysis.

4.2 Jacobian Matrix

The Jacobian matrix $J(\mathbf{q})$ relates joint velocities to end-effector velocities:

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

4.2.1 Duality of the Jacobian

The Jacobian also relates forces at the end-effector to torques at the joints:

$$\boldsymbol{\tau} = J^T(\mathbf{q})\mathbf{F}_{\text{ext}}$$

- **Velocity Analysis**: Forward mapping from joint velocities to end-effector velocities.
- **Force Analysis**: Mapping from end-effector forces to joint torques.

4.2.2 Computing the Jacobian

As before, we compute each column of J based on the joint type.

5 Dynamics

5.1 Energy Methods in Dynamics

Instead of using recursive Newton-Euler formulations involving reaction forces, we can derive dynamics using energy methods.

5.1.1 Kinetic Energy

The total kinetic energy K of the robot is the sum of the kinetic energies of all links:

$$K = \sum_{i=1}^{n} \left(\frac{1}{2} m_i \mathbf{v}_i^T \mathbf{v}_i + \frac{1}{2} \boldsymbol{\omega}_i^T I_i \boldsymbol{\omega}_i \right)$$

5.1.2 Mass Matrix via Jacobian

The kinetic energy can also be expressed in terms of joint velocities:

$$K = \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}}$$

Where the mass matrix $M(\mathbf{q})$ is derived from the Jacobian and link inertias:

$$M(\mathbf{q}) = \sum_{i=1}^{n} J_i^T(\mathbf{q}) \mathcal{I}_i J_i(\mathbf{q})$$

- $J_i(\mathbf{q})$ is the Jacobian of link i.
- $\mathcal{I}_i$ includes mass and inertia of link i.

5.2 Dynamics and Control

5.2.1 Dynamic Equations of Motion

The equations of motion are:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

Understanding and modeling these dynamics are crucial for control.

6 Control Strategies

6.1 Joint Space Control

6.1.1 PD and PID Controllers

Control laws that regulate joint positions:

$$\boldsymbol{\tau} = K_P(\mathbf{q}_d - \mathbf{q}) + K_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + K_I \int (\mathbf{q}_d - \mathbf{q})\, dt$$

6.2 Task Space Control

6.2.1 Operational Space Control

Define control laws directly in task space:

$$\mathbf{F}_{\text{ext}} = \Lambda(\mathbf{q})\ddot{\mathbf{x}} + \mu(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{p}(\mathbf{q})$$

- Control objectives are specified in terms of end-effector motion.
- More intuitive and similar to how humans control their limbs.

6.2.2 Attraction to Goal Positions

Implementing controls that attract the end-effector to a desired position, akin to potential fields.

$$\mathbf{F}_{\text{ext}} = -K(\mathbf{x} - \mathbf{x}_d)$$

6.3 Force Control in Contact Tasks

6.3.1 Impedance and Admittance Control

Regulating interaction forces during contact with the environment.

7 Trajectory Planning

7.1 Joint Space vs. Task Space Trajectories

- **Joint Space Planning**: Easier to compute but may result in less intuitive end-effector paths.
- **Task Space Planning**: Direct control over end-effector movement but requires solving inverse kinematics at each point.

7.2 Potential Field Methods

Using artificial potential fields to generate attractive forces towards the goal and repulsive forces away from obstacles.

7.2.1 Attractive Potential

$$U_{\text{att}}(\mathbf{x}) = \frac{1}{2}k_{\text{att}}\|\mathbf{x} - \mathbf{x}_d\|^2$$

7.2.2 Repulsive Potential

$$U_{\text{rep}}(\mathbf{x}) = \frac{1}{2}k_{\text{rep}}\left(\frac{1}{\rho(\mathbf{x})} - \frac{1}{\rho_0}\right)^2, \quad \rho(\mathbf{x}) \leq \rho_0$$

Where:

- $\rho(\mathbf{x})$ is the distance to the obstacle.
- ρ_0 is the influence distance of the obstacle.

7.2.3 Gradient Descent Control Law

$$\mathbf{F}_{\text{total}} = -\nabla U(\mathbf{x}) = \mathbf{F}_{\text{att}} + \mathbf{F}_{\text{rep}}$$

8 Redundancy and Kinematic Control

8.1 Task Prioritization

Using redundancy to perform secondary tasks, such as obstacle avoidance or optimizing joint configurations.

8.1.1 Null Space Projections

$$\dot{\mathbf{q}} = J^\dagger\dot{\mathbf{x}} + N\mathbf{v}$$

Where:

- $N = I - J^\dagger J$ is the null space projector.
- $\mathbf{v}$ is an arbitrary joint velocity vector for secondary tasks.

9 Motion Planning

9.1 Potential Field Methods in Motion Planning

- **Advantages**:
  - Computationally efficient.
  - Real-time obstacle avoidance.
- **Disadvantages**:
  - Possibility of local minima.
  - Not guaranteed to find a path if one exists.

9.2 Dealing with High DOF Robots

Motion planning becomes computationally challenging due to the exponential increase in the configuration space dimensionality.

9.2.1 Approximate Methods

- Sampling-based planners (e.g., RRT, PRM).
- Use of heuristics and simplifications.

10 Advanced Topics

10.1 Human-Robot Interaction

10.1.1 Safety Considerations

Designing robots to be safe around humans:

- **Mechanical Design**: Use of compliant materials, lightweight structures.
- **Control Strategies**: Safe control algorithms that prevent collisions.

10.1.2 Exoskeletons

Wearable robots that augment human capabilities.

- Assist in carrying heavy loads.
- Rehabilitation and assistive technologies.

10.2 Haptics and Teleoperation

10.2.1 Haptic Devices

Provide force feedback to the user, allowing for realistic interactions with virtual or remote environments.

10.2.2 Applications

- Surgical robots.
- Remote manipulation in hazardous environments.
- Virtual prototyping and design.

10.3 Modeling Human Motion

10.3.1 Digital Human Models

Using robotics models to simulate and analyze human motion.

- Kinematic and dynamic models with many DOF.
- Actuation modeled using muscles.
- Applications in animation, ergonomics, and robotics control.

10.3.2 Learning from Human Motion

- Motion capture data used to inform robot controllers.
- Generalizing human movements to robot control tasks.

10.4 Simulation and Realism

- Importance of accurate models for simulation.
- Limitations: Simulations may not always capture real-world complexities.

11 Conclusion

This document provides a comprehensive overview of foundational concepts in robotics theory, focusing on kinematics, dynamics, control, and advanced topics like human-robot interaction and haptics. Understanding these concepts is crucial for advancing robotics research and developing robots capable of complex, dynamic, and safe interactions with the environment and humans.