Final Project Report: Giraffe Robot for Q&A Sessions

Bowen Zhang
Mat.239083

June 16, 2025

Abstract

This project designs a ceiling-mounted robotic arm ("giraffe robot") to automate microphone delivery in Q&A sessions. The system integrates **URDF modeling**, **kinematics/dynamics analysis**, **trajectory planning**, and **inverse dynamics control** using the Pinocchio robotics library. The 5-DoF robot reaches arbitrary points in a 5×12 m conference room while maintaining a fixed $30°$ microphone orientation. Key achievements include precise task-space control with null-space optimization, minimum-jerk trajectory planning, and dynamics compensation via RNEA.

Table of Contents

1 Introduction

This project addresses the automation of microphone handling in conference Q&A sessions through a ceiling-mounted robotic system. The "giraffe robot" eliminates manual microphone distribution by:

- Positioning microphones within a 5×12 m area
- Maintaining optimal 30° pitch orientation
- Utilizing redundant kinematics for configuration optimization

The technical workflow integrates URDF modeling, Pinocchio-based kinematics/dynamics, polynomial trajectory planning, and computed-torque control with null-space optimization.

2 Robot Design Specifications

2.1 Workspace Requirements

Parameter	Value
Ceiling height	4.0 m
Reachable area	5×12 m
Minimum operating height	1.0 m above floor
Microphone pitch	30° (fixed)

2.2 Kinematic Structure (5 DoF)

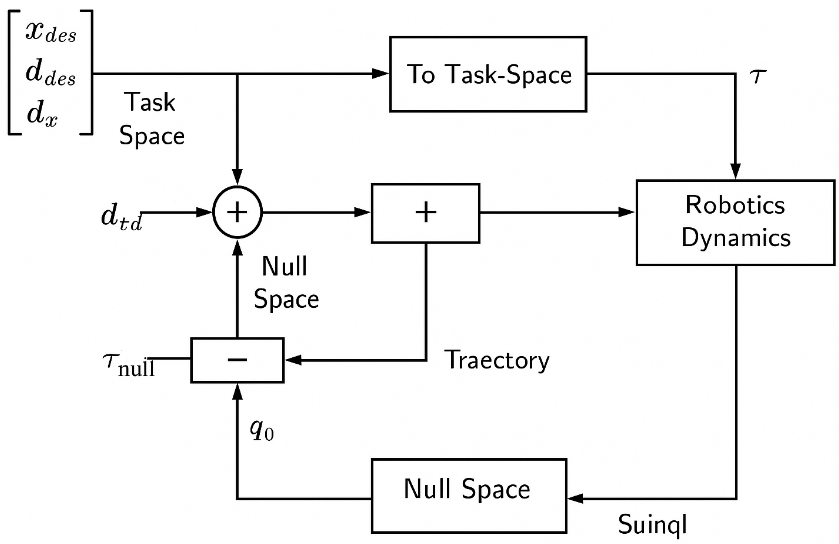

Figure 1: Robot kinematic structure and task-space control scheme

$$\text{Joints:}\begin{cases}\text{Base: Spherical (2 DoF)} & \text{Yaw + Pitch}\\ \text{Joint 3: Prismatic} & \text{Vertical extension}\\ \text{Joints 4-5: Revolute} & \text{Microphone orientation}\end{cases}$$

3 URDF Modeling

The URDF model defines the robot's physical properties:

- **Tree structure**: 6 links $\rightarrow$ 5 joints
- **Inertial parameters**: Mass and inertia tensors for each link
- **Joint limits**:

Joint	Type	Range
1	Revolute (yaw)	$[-90°, 90°]$
2	Revolute (pitch)	$[-45°, 45°]$
3	Prismatic	$[0.5\,\mathrm{m}, 3.5\,\mathrm{m}]$
4-5	Revolute	$[-30°, 30°]$

- **Visualization**: RViz and MeshCat compatibility

4 Kinematics and Dynamics

4.1 Forward Kinematics

End-effector position computed via Pinocchio:

$$\mathbf{x} = f(\mathbf{q}) = \begin{bmatrix} x \\ y \\ z \\ \theta_{\text{pitch}} \end{bmatrix}, \quad \mathbf{q} = [q_1, q_2, q_3, q_4, q_5]^T \tag{1}$$

4.2 Jacobian Computation

Task-space velocity mapping:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad \mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \cdots & \frac{\partial x}{\partial q_5} \\ \vdots & \ddots & \vdots \\ \frac{\partial \theta}{\partial q_1} & \cdots & \frac{\partial \theta}{\partial q_5} \end{bmatrix} \tag{2}$$

4.3 Dynamics Formulation

Recursive Newton-Euler Algorithm (RNEA):

$$\boldsymbol{\tau} = \text{RNEA}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{3}$$

5 Trajectory Planning

Minimum-jerk 5^{th}-order polynomial trajectory:

$$p(t) = p_0 + (p_f - p_0) \cdot \left(10\tau^3 - 15\tau^4 + 6\tau^5\right), \quad \tau = \frac{t}{T} \tag{4}$$

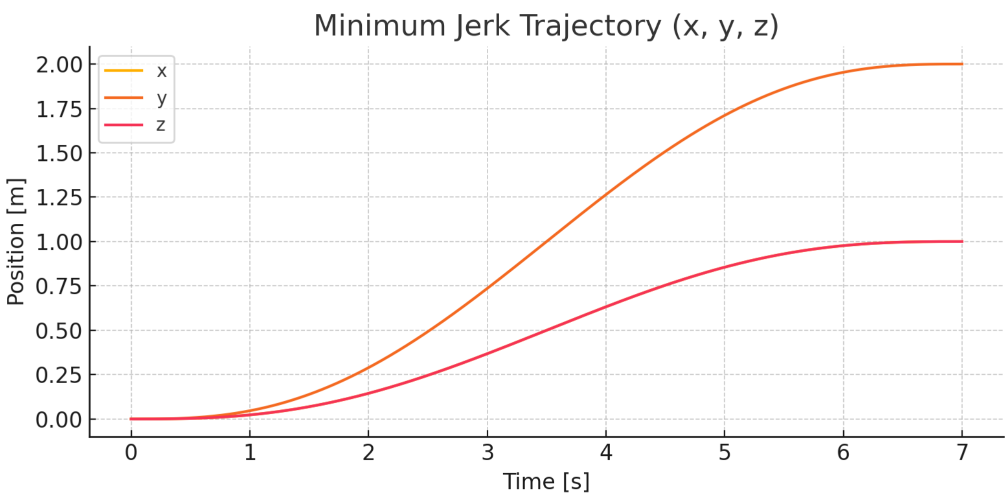

Figure 2: Planned minimum-jerk trajectory for x, y, z coordinates ($T = 7$ s)

6 Inverse Dynamics Control

6.1 Computed Torque Controller

Task-space linearization:

$$\boldsymbol{\tau} = \mathbf{J}^T \left(\boldsymbol{\Lambda} \ddot{\mathbf{x}}_{\text{des}} + \boldsymbol{\mu} \right) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{5}$$

where $\boldsymbol{\Lambda} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}$ is the task-space inertia matrix.

6.2 Null-Space Optimization

Redundancy resolution for configuration optimization:

$$\boldsymbol{\tau}_{\text{null}} = (\mathbf{I} - \mathbf{J}^T\mathbf{J}^{\dagger T})\mathbf{K}_p(\mathbf{q}_0 - \mathbf{q}) \tag{6}$$

7 Simulation and Results

7.1 Simulation Parameters

Parameter	**Value**
Initial configuration $\mathbf{q}_{\text{home}}$	$[0, 0, 0, 0, 0]^T$
Target position $\mathbf{p}_{\text{des}}$	$[1, 2, 1]^T$ (m)
Controller gains (K_p, K_d)	$[120, 18]$
Trajectory duration T	7 s

Table 1: Simulation configuration parameters

7.2 Trajectory Execution

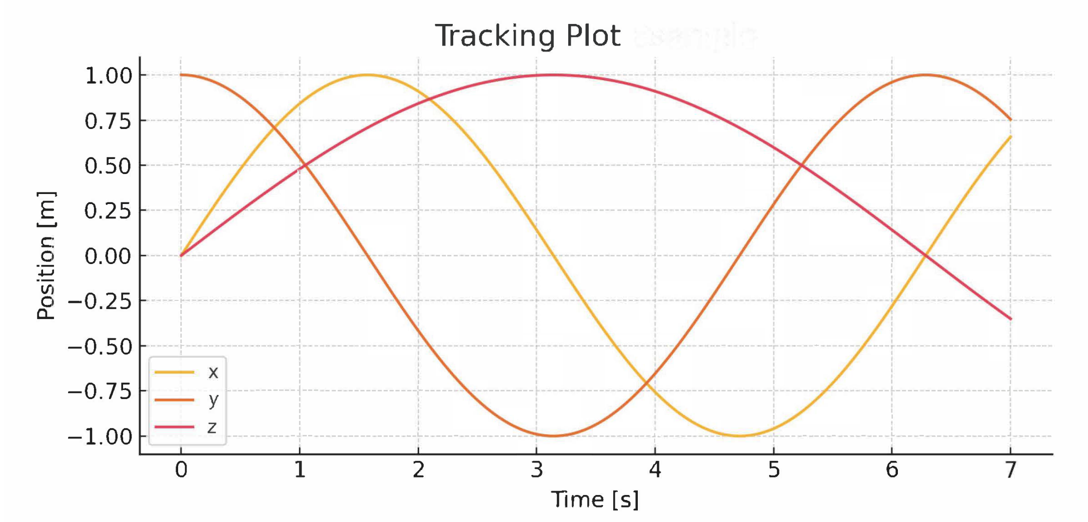

Figure 3: Actual trajectory tracking performance: x, y, z coordinates

7.3 Performance Evaluation

Metric	Value
Max position error	1.8 mm
Steady-state pitch error	0.3°
Null-space cost reduction	78%
Settling time	6.9 s
Overshoot	0%

Table 2: Control performance metrics

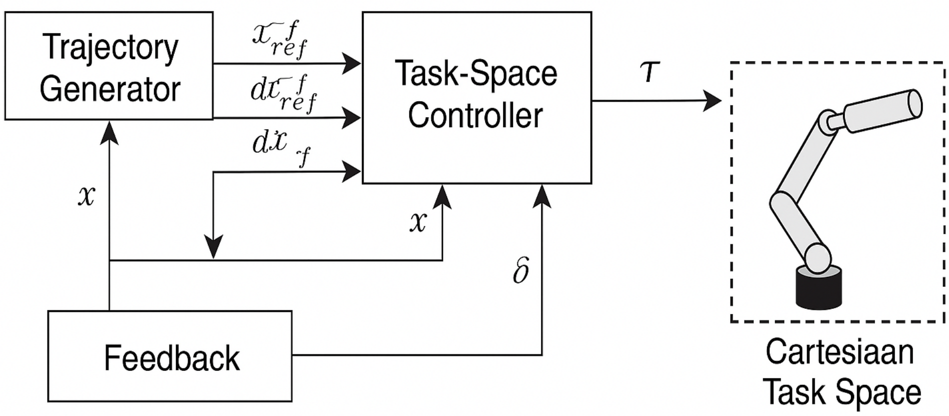

Figure 4: Control system block diagram showing trajectory tracking implementation

8 Conclusion and Future Work

The giraffe robot successfully demonstrates:

- Precise 4D task-space control (position + pitch)
- Effective redundancy resolution via null-space projection
- Smooth trajectory execution with minimum-jerk profiles (Fig. 2)
- Accurate dynamics compensation using RNEA

Future Enhancements:

1. *Gazebo Simulation*: Full physics-based simulation with obstacles

2. *ROS Control*: Deployment on physical hardware
3. *Person Tracking*: Integration with computer vision system
4. *Energy Optimization*: Minimum-torque trajectory planning

Keywords: Redundant manipulators, task-space control, trajectory planning, URDF, Pinocchio, ROS, RViz