Artificial Intelligence and Robotics Course Project

Course Project Specification Document

Project Overview

This project aims to provide hands-on experience in robotics through practical implementation using the CLAIR lab's work environment, originally developed for Yuval Goshen's research under Dr. Sarah Keren's supervision. The project is structured in multiple stages to ensure progressive learning and skill development.

Environment Setup

Required Resources

- Repository: https://github.com/CLAIR-LAB-TECHNION/AIR_robots_sim
- Dependencies: As specified in requirements.txt
- Recommended: Python virtual environment

Technical Prerequisites

- Python environment with all required packages installed
- Access to sim_ur5/mujoco_env/sim_env for environment creation
- Access to sim_ur5/motion_planning/motion_executor for MotionExecutor functionality
- Access to lab_ur5/robot_interface for RobotInterface creation
- Access to lab_ur5/manipulation for ManipulationController functionality
- Understanding of configurations.py for operational boundaries

Project Stages

Stage 1: Simulation Implementation

Task 1: Stack Function Implementation

Objective: Develop a function for creating stable cube stacks at specified locations.

Requirements:

- Function Parameters:
    - Input: List of cube positions
    - Input: Target location for stack placement
- Success Criteria:
    - Accurate positioning of all cubes
    - Stack stability verification
    - Error-free execution

Task 2: Relay Race Implementation

Objective: Create a function managing cube transfer between two robots.

Requirements:

- Function Parameters:
    - Starting position (cube pickup location)
- Process Specifications:
    - Initial pickup by first robot
    - Handover second robot
    - Successful transfer completion

Technical Constraints:

- ur5e_1 robot lacks gripping capability
- Second robot must achieve precise alignment without physical gripping
- Implementation required for **at least one successful configuration**

In this part make sure to record your implementation in the simulation when you submit. Create a python script with your implementation and add it to the zip file.

Stage 2: Lab Implementation

Objective: Transfer simulation implementations to the lab robots.

Here too you will need to implement the two functions you implemented for stage 1. Now we can see the robots in action.

Create a python script with your implementation and add it to the zip file.

Bonus Opportunity:

- Implementation of non-stack structure building
- Focus on achievable, well-defined structures

Stage 3: Independent Project

In this final stage, you will apply the concepts learned throughout the course to design and implement a task planning project. The focus is on developing a complete robotic solution that demonstrates your understanding of task planning, motion planning, and robot control.

Project Requirements:

- Choose an environment:
    1. Simulation environment
    2. Lab robots
- Task Planning Components:
    1. Task Definition
        - Clear description of the chosen task
        - Identification of constraints and challenges
    2. Task Decomposition
        - Break down the complex task into basic operations
        - Identify critical path and if potential bottlenecks
    3. Motion Planning Strategy
        - Design movement sequences
        - Consider obstacle avoidance
    4. Implementation Plan
        - Develop modular code structure
        - Include error handling and recovery
        - Define success criteria

Deliverables:

1. Project Proposal (1-2 pages)
    a. Task description and motivation
    b. Planning approach
    c. Implementation strategy
    d. Expected challenges
2. Technical Documentation
    a. Task planning breakdown
    b. Algorithm descriptions
    c. Code documentation
    d. Test results
3. Implementation
    a. Working code
    b. Video demonstration
    c. Performance analysis

Example Projects:

- Design efficient picking and placing sequences not only for blocks
- Handle different component orientations
- Cooperative Robot Tasks
    - Share workspace effectively
    - Synchronize movements and actions

Note that the important part is your understanding, not the implementation. Therefore, emphasize the project proposal.

Submission Guidelines

Format Requirements:

- Submission Type: ZIP file
- Group Size: Pairs

Required Contents:

1. Modified source files only
2. Documentation:
    a. List of modified functions
    b. Detailed code comments
    c. Implementation explanations
3. Video demonstration of implemented functions

Assessment Criteria

Projects will be evaluated based on:

- Code functionality and reliability
- Documentation quality
- Demonstration clarity

Technical Notes

Important Considerations:

- Refer to configurations.py for operational boundaries
- Consider physical limitations of laboratory equipment
- Document any assumptions or constraints in implementation
- Include error handling and safety considerations

Resource Management

Best Practices:

- Utilize virtual environments for Python
- Install missing packages via pip