Robot Grasping and Mobile Manipulation of Badminton Birdies

Nhi Le, Bhuvan Bolem, Siddharth Garlapati

Problem Statement and Motivation

Objective:

Our project focuses on solving the problem of robotic object grasping, specifically for objects with irregular shapes, such as badminton birdies. We aim to develop an autonomous robotic system capable of detecting, planning, and executing a grasp on a birdie.

Challenges in Grasping Irregular Objects:

- Objects with unconventional shapes, like badminton birdies, present unique challenges
- Irregular geometry: Non-standard shapes make it harder to define stable grasp poses
- This project addresses these challenges by focusing on:
    - Motion planning precision
    - Stability of the grasp during the pick-and-place operation

Technical Overview

1. **Ros Noetic**

   Publishes and subscribes to robot state topics

2. **Gazebo Simulation**

   Simulates the robot and environment in a virtual physics-based world. Custom world creation with a badminton birdie model and placement bin

3. **MoveIt!**

   Handles motion planning and execution for the Fetch robot
   Motion Planning: Ensures collision-free paths for grasping and placement
   Planning Scene: Manages virtual representations of the robot and environment
   Configuration: Set up planning groups, joint limits, and end-effector in the MoveIt Assistant

4. **Python Script**

   Uses ROS interfaces for MoveIt to control the robot. Automates motion planning using MoveGroupCommander. Controls the gripper to pick and place objects.

Workflow and Process

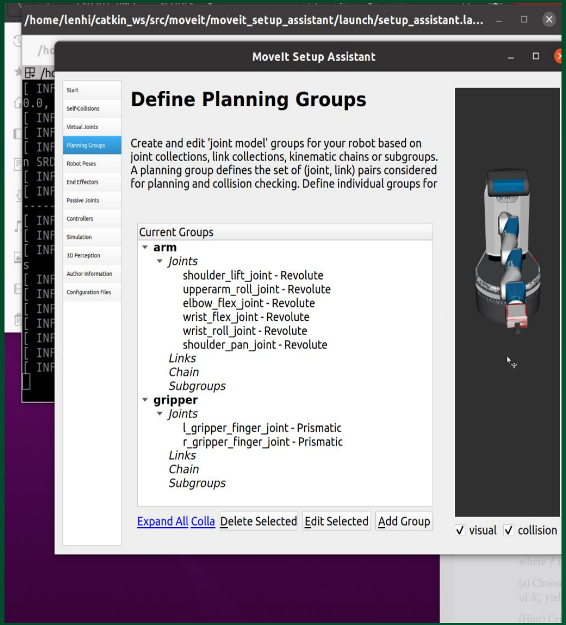
1. Initialization:
   - Launch Gazebo with the Fetch robot and the environment (table, birdie, bin).
   - Use MoveIt! to initialize the robot's arm controller and planning groups.
2. Perception:
   - The robot uses the MoveIt! pipeline to plan and execute actions for picking up the birdie.
3. Motion Planning:
   - The Python script interfaces with MoveIt! to plan the motion of the robot's arm from the table (where the birdie is placed) to the bin.
   - The script executes a grasping operation using the MoveIt! motion planner and the robot's gripper.
4. Grasping:
   - The robot moves the arm to the location of the birdie on the table.
   - It uses MoveIt! to perform inverse kinematics and calculates the best arm configuration to approach and grasp.
5. Placement:
   - After grasping the birdie, the robot moves it to the bin.
   - The arm is re-planned using MoveIt! to execute the drop of the birdie into the bin.
   - The robot places the birdie into the bin and releases it using the gripper.

Implementation Details

Moveit Setup:

Using the MoveIt Assistant, we configured the robot for motion planning:

- **Virtual Joint:**
  We defined a fixed virtual joint between the world and the base_link of the robot. This anchors the robot in the environment and ensures that planning and visualization in MoveIt align accurately with Gazebo.
- **Planning Groups:**
  - **Arm Group:** This includes joints from the **shoulder_pan_joint**, **wrist_roll_joint, etc.**
  - **Gripper Group:** This consists of the **l_gripper_finger_joint** and **r_gripper_finger_joint**, representing the gripper's opening and closing motions.
- **End-Effector:**
  The gripper was set as the end-effector, with the **wrist_roll_link** as its parent link.
- **Planning Pipelines:**
  OMPL was selected as the default motion planning pipeline, which generates collision-free trajectories



Implementation Details

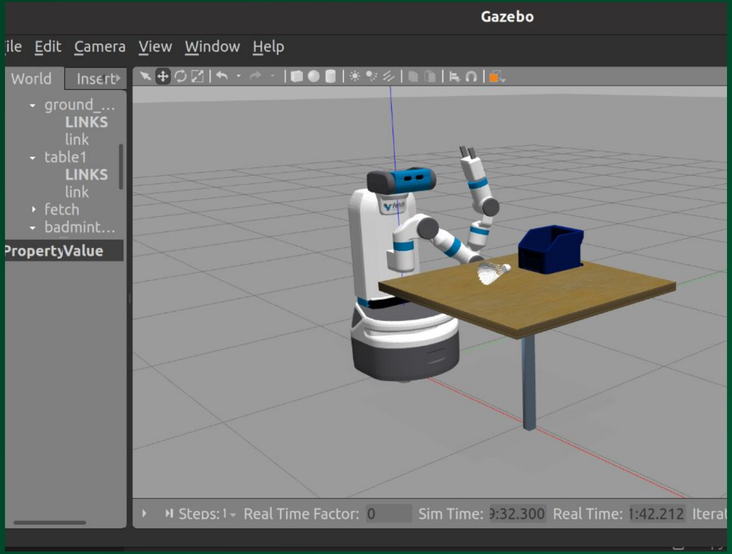
Simulation Setup in Gazebo

Object Model Selection: The target object, a badminton birdie, was represented as a 3D model. We sourced this model from online repositories offering free 3D files in formats like .stl, .obj, or .dae.

Converting the 3D Model: The 3D model wasn't in a compatible Gazebo format, we converted it to .dae. This ensured that the model could be loaded into Gazebo.

Integration into Gazebo: The processed 3D model was added to the Gazebo environment by defining it in a custom SDF file. This involved specifying the model's geometry, material properties, and collision boundaries to make it interactive

Creating the World File: We used Gazebo's .world format to define the entire simulation scene, including: The badminton birdie model, a bin as the placement target, a flat ground plane for stability.



Implementation Details

Python Script

1. Initialization:The script initializes the necessary ROS nodes and MoveIt interfaces. It sets up control groups for the robot's arm and gripper, enabling precise motion planning and execution.
2. Defining the Grasp and Placement Poses: A pick pose is defined above the badminton birdie to ensure the gripper can securely grasp the object. place pose is determined inside the bin, ensuring the birdie is safely deposited at the target location.
3. Motion Planning: The arm's trajectory to the pick and place poses is computed using MoveIt's planning tools. Collision detection and optimization ensure safe and efficient motion.
4. Gripper Control: The gripper opens to approach the birdie, closes to grasp it, and opens again to release it at the placement location. Resetting to Home Pose: After completing the task, the arm returns to a predefined "home" position to prepare for the next cycle.

Future Work

1. **Develop a Demo Video**:
    - Record this after refining the current implementation and testing different object positions.
2. **Expand Task Complexity**:
    - Extend the project to handle multiple birdie
    - Implement sequential pick-and-place operations using MoveIt! Python scripts.

3.. **Real-World Testing**:
- Transition from simulation to a physical robot
- Use an actual Fetch Mobile Manipulator or a similar setup.

Thank you for listening.