CS 6301

Group 8

Dynamic Object Grasping

SAURAV DOSI
ADITYA KULKARNI
FEROZ HATHA

Introduction

- Objective:

The project focuses on developing a control system for a manipulator robot to accurately grasp objects moving along deterministic paths by predicting and adapting to their future poses.

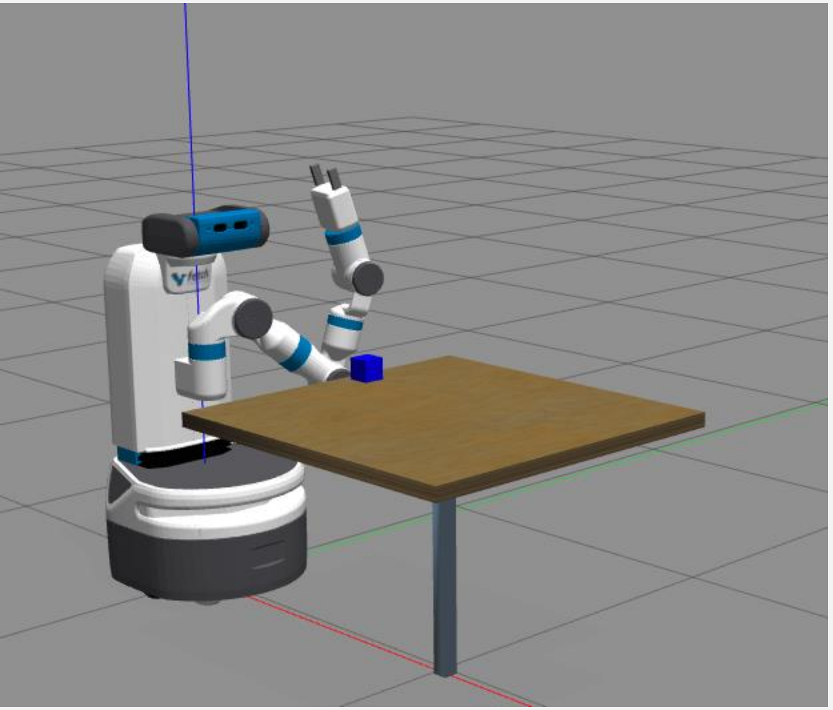
- Methodology:

The approach employs a Recursive Least Squares (RLS) based algorithm for intercept calculation, in conjunction with ROS, MoveIt, and TRAC-IK for inverse kinematics and grasping.

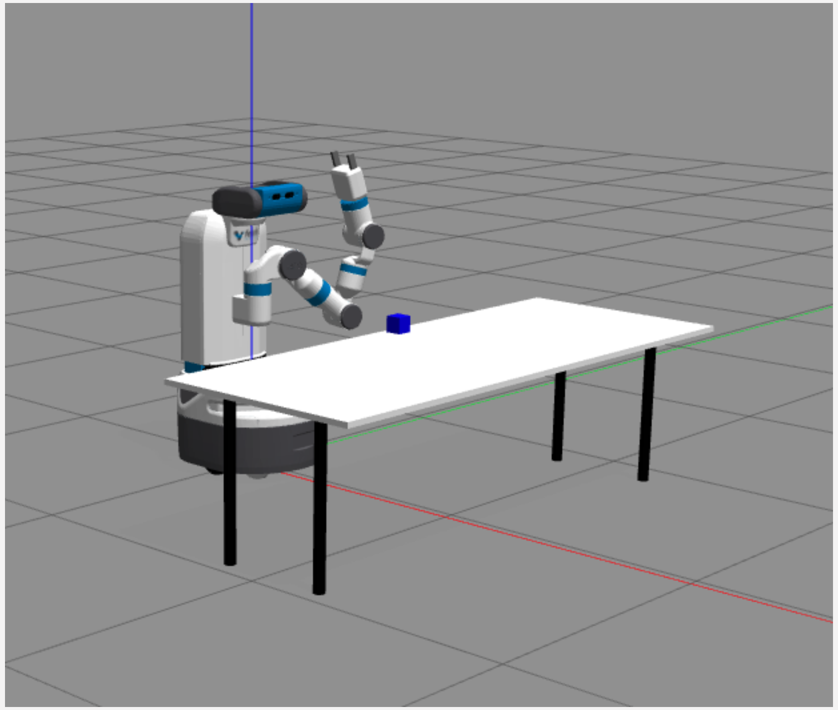
- Applications:

This work has practical implications for industries that rely on automated systems, aiming to enhance efficiency and accuracy in tasks such as conveyor belt sorting and mobile manipulation.

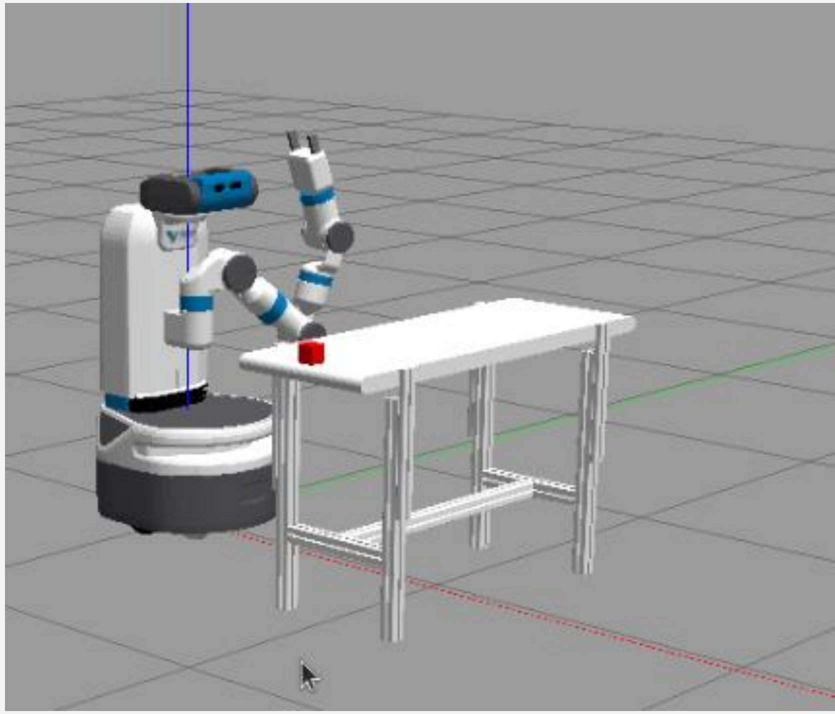
Simulation Environment



GAZEBO CAFE TABLE



LONG TABLE



CONVEYOR BELT

Source

Simulation Environment

- **Components:** Gazebo, Conveyor Belt Plugin

The simulation scene includes a Fetch robot and either a table or a conveyor belt setup. The goal is for the robot to intercept and grasp the moving cube using its arm.

- **Simulation Approaches:**

Two approaches were used to simulate the cube's uniform motion:

  - Use the Gazebo cafe table setting or a custom table setting and run a python script to simulate cube motion.
  - Utlize the Gazebo conveyor belt plugin and incorportate fetch into that setting.

- **Motion Simulation:**
  - Use a Python script to simulate the cube's uniform motion. The script uses ROS nodes to continuously move the cube in Gazebo, updating its position based on a set velocity.
  - The conveyor belt's speed can be controlled via a ROS service, where power levels range from 0 to 100.

- **Challenges:**
  - High frequency vertical oscillations of the cube while simulating uniform motion in the horizontal direction. Common in Gazebo Physics simulations and may be due to default contact parameters between the cube and table surface.
  - With the initial approach of simulating cube motion, the cube continues to have a velocity even after being grasped.

Approach and Methodology

- Trajectory Planning:

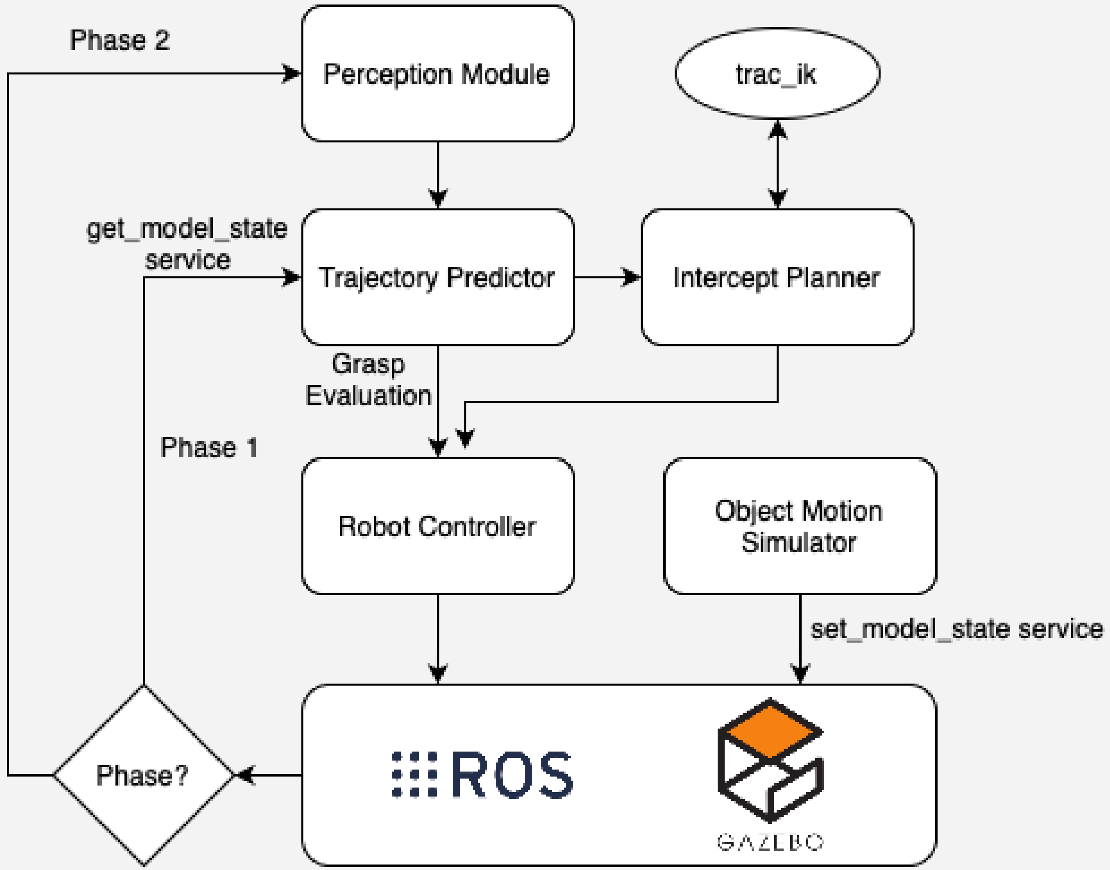
The grasping process involves two main steps: positioning the robot above the intercept point and then performing a vertical descent to grasp the object. The trajectory planning uses MoveIt for collision-free path generation and time estimation.

- Trajectory Prediction:

Recursive Least Squares (RLS) is employed to predict the future positions of the moving object. This method updates predictions based on real-time data from the simulation, allowing the robot to adjust its movements accordingly.

- Interception Algorithm:

An algorithm was developed to calculate interception points by considering time needed for inverse kinematics, gripper alignment, and closure. It uses predicted trajectories to determine optimal interception timing and positioning.



Trajectory Prediction

- Recursive Least Squares for Prediction:

Our system uses Recursive Least Squares (RLS) for predicting the trajectory of objects moving along deterministic paths, allowing the robot to anticipate and align its movements with the object's future positions.

- Error Minimization:

The algorithm defines an error term $e(t) = y(t) - w^T x(t)$, where it continuously refines the parameter vector w to reduce this error over time.

- Recursive Updates:

RLS updates parameters using a gain vector K(t), which adjusts the parameters based on current errors, and updates the covariance matrix to reflect new data, ensuring efficient adaptation to changing inputs.

- Predictive Capability:

By fitting polynomial models to input data, RLS can predict future positions of objects moving along deterministic paths, enhancing the robot's ability to intercept and grasp moving targets accurately.

Intercept Calculation Algorithm

- Initialization and Prediction:

Start by recording the current time and use Recursive Least Squares (RLS) to predict future positions of the moving object until convergence.

- Time Calculation:

Define a range of timesteps and calculate the time needed for the gripper to reach the object, adjusting the timestep range accordingly.

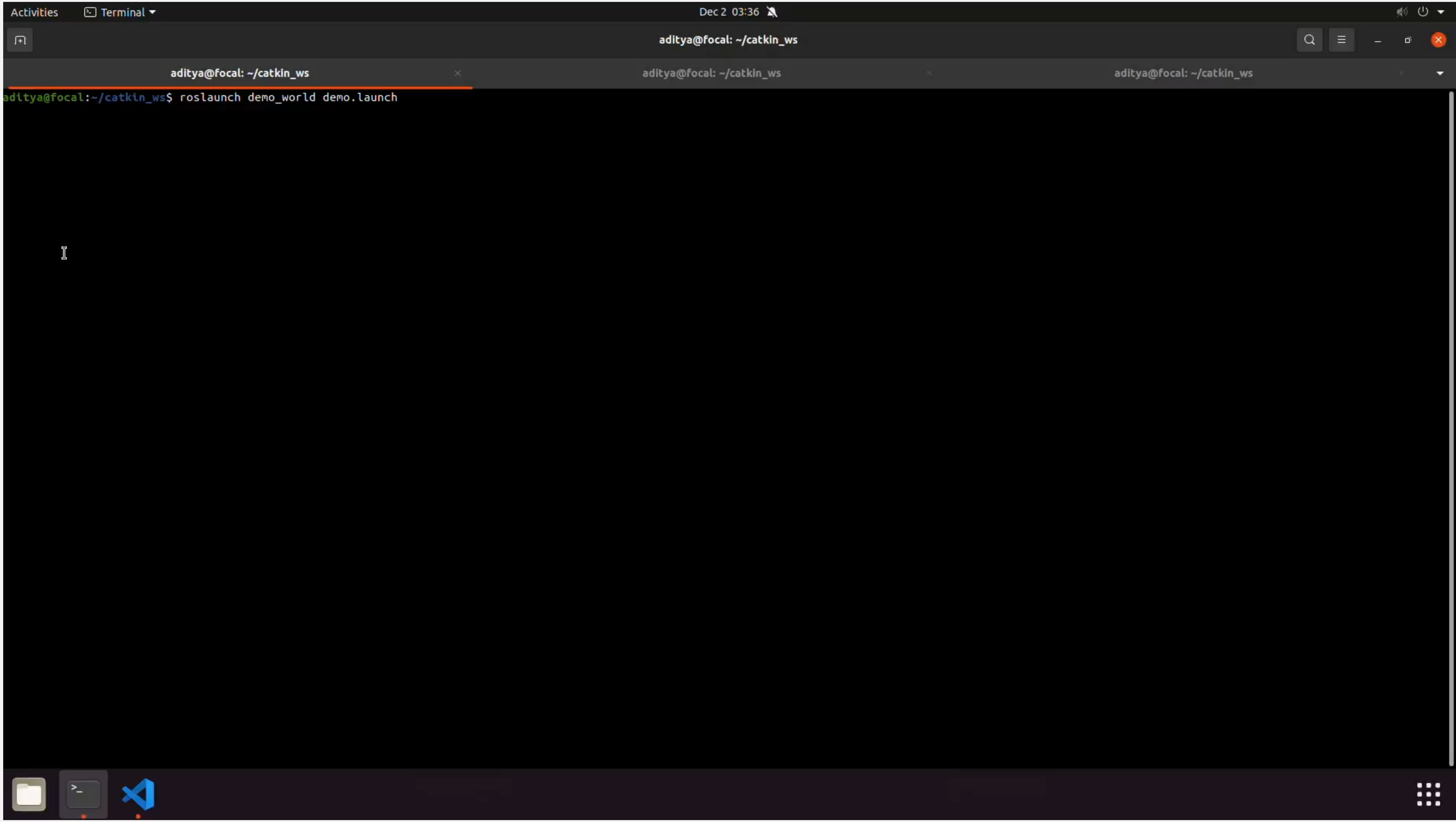
- Trajectory Sampling and IK Solution:

Randomly sample points from the predicted trajectory and use multiple threads to compute Inverse Kinematics (IK) solutions for these points, ensuring efficient computation.

- Execution:

Select the first feasible step with an IK solution, wait for the calculated time, then execute the IK solution and deploy the gripper to grasp the object.

Demo



Activities
Terminal
Dec 2 03:36
aditya@focal: ~/catkin_ws
aditya@focal: ~/catkin_ws
aditya@focal: ~/catkin_ws
aditya@focal: ~/catkin_ws
aditya@focal:~/catkin_ws$ roslaunch demo_world demo.launch

Conclusion

- The project successfully developed a robotic system capable of predicting and intercepting moving objects along deterministic paths, achieving a grasp success rate of 80%.
- The study identified challenges such as handling non-linear paths and varying velocities, which were not fully addressed, indicating areas for further research.
- The findings have practical applications in industries that rely on automated systems, such as conveyor belt sorting, by improving the efficiency and accuracy of robotic grasping tasks.
- Applications: Conveyor Belt Grasp, Efficient Mobile Manipulation Trajectory Planning

Improvements and Future Work

- Enhanced Perception:

Future work will focus on improving the robot's perception capabilities to better handle dynamic environments. This includes refining the system's ability to accurately detect and predict the trajectory of moving objects.

- Non-linear Path Exploration:

The project plans to extend its current linear path approach to more complex, non-linear paths. This will involve developing algorithms capable of handling circular and sinusoidal trajectories, which are more representative of real-world scenarios.

- Variable Velocity Handling:

Another area of improvement is the exploration of non-linear velocities. The current system assumes a constant velocity, but future iterations will incorporate varying speeds to simulate more realistic object movements.

- Reinforcement Learning Integration:

To enhance adaptability in unpredictable environments, the project aims to integrate reinforcement learning techniques. This will allow the robot to learn and adjust its strategies dynamically as it encounters new challenges in grasping tasks.

THANK YOU