Object Reorientation with Help of Environment Contact
Project Proposal

Maede Boroji, Vahid Danesh

Abstract

In this project, we will utilize a Tabletop Franka Panda robot to upright various fallen objects, including cuboids, cylinders, and L-shaped objects. We will assume complete knowledge of the objects' properties, such as dimensions and edges. By identifying the edge of each object that is in contact with the environment, we can determine all the necessary parameters for the motion. This allows us to construct the screw axis which the object will be manipulated about that axis. All simulations will be performed in the Genesis-AI physics-based simulator. This approach enables manipulation of heavy objects by leveraging environmental contact rather than relying solely on grasping. If time permits, we will also explore an extension involving rotational slippage at the grasp point to allow upright manipulation while maximizing manipulability.

1 Problem statement

In general, any rigid body displacement can be expressed as a rotation $\theta \in \mathbb{R}$ about a screw axis $\mathcal{S}$ followed by a translation $d \in \mathbb{R}$ along the axis. By representing the screw axis $\mathcal{S}$ by a unit vector $\boldsymbol{u} \in \mathbb{R}^3$ along the axis and an arbitrary point $\boldsymbol{r} \in \mathbb{R}^3$ on the axis, the screw parameters, using Plücker coordinates, are defined as $(\theta, d, \boldsymbol{u}, \boldsymbol{m})$, where $\boldsymbol{m} = \boldsymbol{r} \times \boldsymbol{u} \in \mathbb{R}^3$. Therefore, the screw motions can be efficiently expressed by the dual quaternions as $D_T = Q_R + \epsilon \frac{1}{2} Q_P Q_R = (\cos\frac{\theta}{2} + \boldsymbol{u}\sin\frac{\theta}{2}) + \epsilon\left(-\frac{d}{2}\sin\frac{\theta}{2} + \sin\frac{\theta}{2}\boldsymbol{m} + \frac{d}{2}\cos\frac{\theta}{2}\boldsymbol{u}\right)$. In our case which is a *constant* screw motion, $\boldsymbol{u}$ and $\boldsymbol{m}$ remains constant and only θ changes. The smooth path in $SE(3)$ provided by the ScLERP is derived by $D(s) = D_1 D_{12}^s$, where $s \in [0, 1]$ is a scalar path parameter and D_{12} is the transformation of $\mathcal{C}_2$ with respect to $\mathcal{C}_1$. D_{12}^s can be computed using $D_{12}^s = \left(\cos\frac{s\theta}{2}, \sin\frac{s\theta}{2}\boldsymbol{u}\right) + \epsilon\left(-\frac{sd}{2}\sin\frac{s\theta}{2}, \frac{sd}{2}\cos\frac{s\theta}{2}\boldsymbol{u} + \sin\frac{s\theta}{2}\boldsymbol{m}\right)$ after extracting the screw parameters $\boldsymbol{u}$, $\boldsymbol{m}$, θ, d from D_{12}.

If $\mathcal{C}_1$ and $\mathcal{C}_2$ have a contact edge in common, the final pose can be achieved by pivoting the object about the common edge, as shown in Fig. 1.

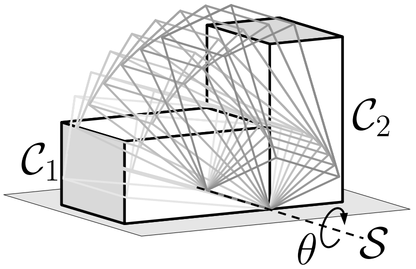


Figure 1: Manipulating cuboid-shape objects by exploiting the environment contact, through pivoting about an edge. Courtesy of image [1]

We will also explore an advanced method in which the manipulator induces object motion through rotational slippage at the grasp point while maintaining a fixed (or nearly fixed) end-effector orientation. This enables more dexterous manipulation while maximizing the robot's manipulability. However, implementation in a physics-based simulator remains uncertain. If time permits, we will attempt it in Genesis-AI; otherwise, results for this specific task will be presented using a non-physics-based simulator, such as Swift [2].

2 Project Overview

Robot Platform and Simulator

We will use the Tabletop Franka Emika Panda robot and simulate it in a physics-based simulator, Genesis-AI [3].

Object Types

The object set will include cuboids, cylinders, and L-shaped rigid bodies. All objects will initially lie in random arbitrary poses on the tabletop and the goal will be to reorient them to an upright configuration.

Simulation Assets

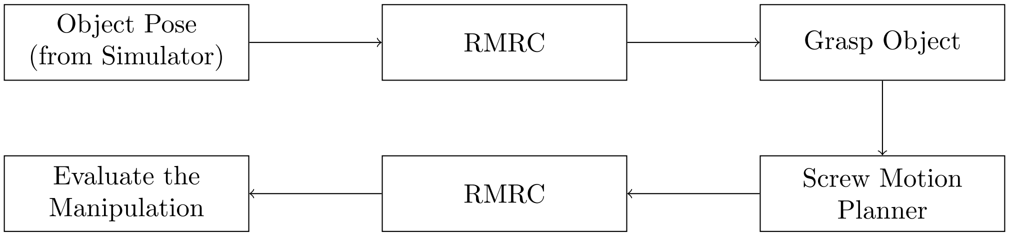
The default Panda robot URDF will be used. For rotational slippage, we will modify the URDF to include an additional (8th) degree of freedom as a revolute joint in between the robot's fingers. This joint will enable rotational slippage between the gripper and the object. The inverse kinematics will be adapted accordingly to solve for this added joint.

Proposed Approach

This will be a model-based control system. The robot begins from a ready configuration. The objects' position are initialized randomly within the robot workspace. The object pose will be obtained directly from the simulator (no perception module).

A resolved motion rate controller (RMRC) will be used to track desired end-effector trajectories along the grasp path to grasp the object and screw motion path to manipulate the object, while avoiding obstacles and respecting joint position and velocity limits. Manipulability will be maximized throughout the trajectory using an optimization-based controller.

Schematic Pipeline



Model Sources

All manipulated objects will be either primitive objects (boxes, cylinders) defined within Genesis-AI or CAD models from open-source repositories.

References

[1] A. Fakhari, A. Patankar, and N. Chakraborty, "Motion and force planning for manipulating heavy objects by pivoting," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9393–9400, 2020.

[2] J. Havlicek and P. Corke, "Swift: A python-based robot simulator." https://github.com/jhavl/swift, 2024. Accessed: 2025-05-01.

[3] G. Authors, "Genesis: A universal and generative physics engine for robotics and beyond." https://github.com/Genesis-Embodied-AI/Genesis, December 2024.