Dexterous Robot Arm to Increase Interactivity for Low Mobility Groups

Anio Zhang

***Abstract*—This proposal outlines the development of a versatile robot arm designed for attachment to wheelchair handlebars, enabling users to interact with various objects such as elevator buttons, card readers, and door handles. The project aims to enhance accessibility and autonomy for wheelchair users by providing a robotic solution capable of intuitive human-object interactions. The arm will be equipped with a wide-angle camera at its extremity, serving as its primary sensory input for environmental perception. The outcomes of this research can potentially extend to other mobile robotic platforms, facilitating enhanced interaction capabilities with human-centric environments.**

I. INTRODUCTION

This project aims to develop advanced pipeline enabling a robot arm, utilizing a wide-angle camera as its primary sensory input alongside actuator encoders, to identify and interact with objects in its vicinity autonomously. Targeting individuals with limited upper-body mobility, the robot arm will facilitate interactions such as pressing elevator buttons, swiping cards, and manipulating door handles. By presenting user-selectable options and executing chosen actions, the system enhances accessibility and independence. Moreover, this technology can be adapted for use on mobile robotic platforms, extending its utility to various environments and applications.

II. METHODOLOGY

To collect the data set we will follow the works done by NYU in this paper. In the initial stages we will simulate the algorithm using PyBullet and other simulation engines. Once the algorithm is in a working state we will implement the above algorithm to work on physical robot arm. We plan on using a smaller arm so that it can be mounted on a wheel chair or another wheeled robot to give it access to interact with objects. The physical robot will be similar to this.

To be more specific:

- **Data collection** We will mount a Intel RealSense camera at the end of a 3d printed emulation of the robots gripper which can be manipulated by a human arm. We will use this setup to collect a few 100 examples of the emulated griper interacting with the objects on the wall. Example grabbing handles, turning on and off switches and interactiving with elevator buttons. In applicable cases we will also transform these datapoints by flipping them as many of them are symmetric vertically (Truning on switches). We will also incorporate and convert the data collected by ”On brining Robots home” paper into our format so that it too can be used in our training process.
- **Experiments with CNN** As our problem relies on the CNN’s capability to recognize the environment the choice of CNN used will be important. To make this decision we will test out various model’s and evaluate their performance for our model.
- **Training the pipe line** With the collected data we will use the URDF of the above mentioned robot arm to simulate the system. We will train the RL model to imitate the actions that we have performed. The expected output of the RL model will be the expected position of the gripper and the gripper’s status.
- **Deployment** Once the model has been trained and performs well in the simulation, we will transfer the model on to a single board computer that can convert the RL model’s output to the joint angles of the robot arm and move it to the required position.

III. Pipeline in Depth

A. Convolutional Neural Network (CNN)

We will first develop the algorithm to take the input from the sensors (cameras) and convert them into a vector that can be further processed by the algorithm. This conversion will be carried out by a CNN. Additionally, we must insure rotational and translational equvariance. CNN by design are equvariant to translation. To ensure rotational equivariance we plan on including some kernels that are perturbed by rotating them. We also plan on experimenting with graph neural network(GNN) with discrete rotational matrices(Not equally distributed along).

Equivariant is important as the robot will not always be on a level surface and the interactable objects will also not always be in a flat surface. Also in many cases the iteractable objects will be viewed from a skewed angles in both pitch and roll. The model must be able to handle these roll and pitch distrubences in the image. Yaw rotation is not much of a consideration as the camera's position on the endeffcor of the robot arm gives us the opportunity to stabilize it via the actuators itself.

B. Deep Reinforcement Learning(RL) Model

The next step would be to convert the above feature vector into the action that the bot needs to take. For this we propose a deep RL model that will convert the above feature vector and the user input(which button to press) and produce a vector that is the location of the end effector of the robot arm. While the task is being performed the above algorithm will continue to run and will continue to update the expected position of the arm. This expected location will be given to the inverse kinematics model that will handle the convertion of these position to the joint angles which are then executed by the robot arm.

IV. Proposed Network Architecture

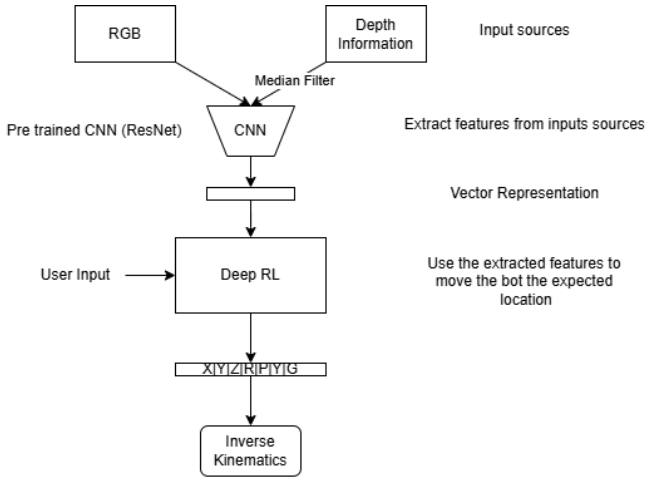

Fig. 1. Flow diagram of the proposed algorithm

V. Timeline

Week	Task	Grading
1-4	Research related to invariant CNN. Implementing and fine tuning the above CNN with the publicly available datasets. Collecting a few additional data points with our own setup.	30%
4-10	Simulating the robot arm in PyBullet and train the deep RL model to perform the tasks. Eg: Grab a door handle, push a few buttons etc. Fine tune the deep RL model and report the results.	30%
10-12	Transfer the above information on to the physical robot arm. Quantify the results in various scenarios.	20%
12-14	Final testing of the simulated and physical bot and writing the report.	20%