Project Milestone: Manipulator control and training based on hand motion detection

Yifei Chen, Deyu Yang, Siying Zhu, Chenhui Ning, Yijun Fang

***Abstract*—This project aims to develop an assistive desktop robotic arm that can recognize and replicate hand motions performed by a user, using a combination of a Leap Motion sensor and computer vision techniques. The proposed gesture-based control system allows users to control the arm using natural hand gestures, and the key challenge is accurately translating the user's intended motions into the robotic arm's movements. This project proposes using velocity control as a simpler alternative to position control, and the system could be used for direct control of the robotic arm or as a means of training the arm to execute specific tasks. The potential benefits include providing a more intuitive and natural means of controlling robotic arms, and facilitating the development of smarter and more capable assistive robotic systems.**

***Index Terms*—Assistive robotics, Imitation learning, gesture-based control, human-robot interaction**

Fig. 1. A example of mimic manipulator control

I. Introduction

In recent years, robotic arms have gained popularity in various industries such as manufacturing, healthcare, and military. However, controlling these machines can be challenging and requires specialized skills. To address this issue, we propose a gesture-based control system for a desktop robotic arm, which enables users to control the arm using natural hand gestures. The goal of this project is to develop an assistive desktop robotic arm that can recognize and replicate hand motions performed by a user, with the key challenge being to accurately translate the user's intended motions into the robotic arm's movements. While previous attempts at using position control have been complicated by the need to solve for both inverse kinematics of the human arm and forward kinematics of the robotic arm, we propose to use velocity control as a simpler alternative. By detecting the user's hand motion through either visual recognition or a wearable sensor, the corresponding velocity and acceleration information can be directly transmitted to the robotic arm. This approach can be used for direct control of the robotic arm or as a means of training the arm to execute specific tasks.

Our proposed system utilizes a combination of a Leap Motion sensor and computer vision techniques to detect and track hand movements, which are then translated into corresponding movements of the robotic arm. While we initially planned to use position control to achieve precise control of the arm, we decided to adopt velocity commands after reviewing the research by Li Fei-Fei's group[1], as it offers greater efficiency and ease of implementation. This system has several advantages, including ease of use, flexibility, and the ability to train the robotic arm through imitation learning, whereby the arm can learn from human demonstrations.

The project aims to develop an intelligent robotic arm that can be trained through imitation learning to perform tasks more efficiently and accurately. Our proposed approach has the potential to provide a more intuitive and natural means of controlling robotic arms, facilitating the development of smarter and more capable assistive robotic systems. In conclusion, the project aims to develop a computer vision-based approach to assistive robotic manipulation that recognizes and responds to hand gestures in real-time, enabling a person to control a robotic arm through simple hand gestures. The proposed system leverages velocity-based control and imitation learning to improve the accuracy and efficiency of the robotic arm over time.

II. Problem Statement

To realize a stable, efficient and intelligent gesture hand control manipulator, we need to answer three key questions. 1. How to track the motion and gestures of hands. 2. What kind of motion control data need to be sent to the manipulator to finish the tasks. 3. What kind of algorithm should be selected to train the learning of the manipulator, so as to achieve the purpose of executing tasks more intelligently.

A. Problem 1

To track the motion and gestures of hands, we propose to use a combination of computer vision and sensor-based approaches. One approach is to use a depth camera, such as the Leap Motion sensor, to detect the position and motion of the user's hands. We will also use Convolutional Neural Networks (CNNs), to analyze image or video data of the user's hands and extract information about their position and motion. The data of hands recognition is enormous. Additionally, we may use machine learning algorithms to improve the accuracy and robustness of the tracking system, since the error of the sensor need to be considered to prevent accident.

B. Problem 2

To control the manipulator and execute tasks, we need to send motion control data to the manipulator. The specific type of data required will depend on the task being performed and the type of manipulator being used. In general, we propose to use velocity-based control, where the velocity and acceleration of the hand are used to directly control the motion of the manipulator. This can be achieved by mapping the hand motion to the joint angles or end-effector position of the manipulator. We can also use algorithms such as Inverse Kinematics to calculate the joint angles required to achieve a desired end-effector position. But we tried to avoid this problem because it would make our controller too complicated.

C. Problem 3

To train the manipulator to perform tasks more intelligently, we propose to use a machine learning algorithm such as Reinforcement Learning (RL) or Imitation Learning. RL can be used to optimize the manipulator's behavior by providing feedback on the success or failure of its actions. Imitation Learning can be used to train the manipulator to mimic the behavior of a human expert, by providing demonstrations of successful task execution. We can also use a combination of both approaches to achieve the best performance. Additionally, we can incorporate techniques such as Transfer Learning or Domain Adaptation to improve the generalization of the learned behavior to new tasks or environments[2].

One of the approach we mainly focus on is the one of the classic algorithm in the deep reinforce learning field. Based on our current information, using the actor-critic CNN in reinforcement deep learning is a suitable approach. The actor-critic method combines policy gradient and value function methods, which can optimize policy and value function simultaneously during the learning process. This method has been successfully applied in many robot control tasks, helping robots learn the optimal policy for specific tasks and adjusting and optimizing it in real-time during task execution.[3] Meanwhile, CNN can automatically extract and classify features from input image data, with good representation and generalization abilities. Therefore, using the actor-critic CNN can effectively solve problems in gesture recognition, motion control, and robot intelligent learning, improving robot performance and efficiency. However, the final decision needs to be based on the actual situation and experimental results.

D. Data will be used

To develop a stable, efficient, and intelligent gesture hand control manipulator, you will need a variety of datasets to train and test the machine learning models.

- **Hand gesture recognition dataset:** This dataset include a wide range of hand gestures that can be used to control the manipulator. It should include images or videos of different people performing the same gestures from different angles and lighting conditions. This problem has been learned for years, so there are a lot of open source data we can use for our project.
- **Motion tracking dataset:** This dataset should include data on the movements and positions of the hand over time, which can be obtained using a sensor such as the Leap Motion device. It should include data for a variety of tasks, such as reaching, grasping, and releasing objects. We are now collecting the data by ourselves.
- **Manipulator control dataset:** This dataset should include data on the movements of the manipulator in response to different control inputs. It can be used to train and test the machine learning model to execute specific tasks, such as picking and placing objects. Since we use a simulation environment, the tracking data can be obtain easily.

E. Evaluation

The evaluation of this project include the following term.

1) **Accuracy of gesture recognition:** Evaluate the model's accuracy in gesture recognition, i.e., how well the model can correctly recognize different gestures.
2) **Precision of motion control:** Evaluate the model's precision in controlling the robot's motion, i.e., how well the model can control the robot to move as expected.
3) **Effectiveness of robot intelligent learning:** Evaluate the model's effectiveness in intelligent learning, i.e., how well the model can continuously optimize the strategy and improve the efficiency and accuracy of executing tasks while executing tasks continuously[4].
4) **Real-time performance and stability:** Evaluate the model's real-time performance and stability when executing tasks, i.e., how well the model can respond to inputs quickly and stably execute tasks in real-time environments.
5) **Generalization ability of the dataset:** Evaluate the dataset's generalization ability for different scenarios and environments, i.e., how well the model can correctly execute tasks in different scenarios and environments.

Currently, we have arranged and designed experiments in the following three areas:

1) **Gesture recognition accuracy evaluation:** A set of gesture datasets can be collected and manually annotated with each gesture's category. The trained model can be used to classify these gestures, and the classification accuracy can be calculated to evaluate the performance of gesture recognition.
2) **Motion control precision evaluation:** Multiple different motion control tasks, such as grasping, carrying, and placing, can be set up. The trained model can be used to control the robotic arm to perform tasks, and the errors during the execution process, such as the error between the target position and actual position, can be recorded to evaluate the precision of motion control.

3) **Intelligent learning effect evaluation:** Reinforcement learning algorithms can be used to train the robot to perform specific tasks, such as completing the task of carrying a specified item within a certain time frame. The time taken and completion rate of the robot can be recorded and compared with benchmark test results to evaluate the effectiveness of intelligent learning.

Of course, the specific evaluation methods need to be selected based on the specific task and experimental scenario.

III. Literature Review

A. Imitation Learning of Manipulation

One of the main work we related with is the ROBOTURK, which proposed by Li's team in 2018. This work introduces a platform called ROBOTURK, which enables remote workers to provide task demonstrations for robotic manipulation tasks through their mobile devices. The authors show that ROBOTURK can collect large amounts of high-quality data for policy learning on multi-step manipulation tasks with sparse rewards. They also show that using larger quantities of demonstrations during policy learning provides benefits in terms of both learning consistency and final performance[1].

B. Hand motion detection in manipulator control

Liang's work presents a new ring-form input device called DualRing, which can sense both hand gesture and movement of hand segments[5]. It can detect the hand pose and movement with two IMU rings and uses a high-frequency AC circuit for on-body contact detection. The interaction space is divided into three sub-spaces based on the sensing information. The system provides comprehensive hand gesture sensing, which outperforms single-ring-based solutions. The user study shows that DualRing is favored for its usability, efficiency, and novelty. Ho-Sun Shin, Asilbek Ganiev, and others designed an interactive method for controlling the motion of a virtual robot arm using different electromyography (EMG) signals[6]. They demonstrated how to use electromyography, gyroscopes, and accelerometers to control the virtual robot arm, and obtained clear and important electromyography data through the MYO muscle signal sensor. The study successfully demonstrated the teaching control of a robotic arm using this data.

Midarto Dwi Wibowo and colleagues used Leap Motion to capture gesture data and applied Naive Bayes algorithm to recognize the 26 letters of the alphabet, except for J and Z, achieving a success rate of 95%[7]. Additionally, Vik and colleagues proposed a method of using Leap Motion for virtual space writing[7].

IV. Technical Approach

To develop a robot that can learn from demonstration, there are three main problems that need to be solved: action data collection, learning from demonstration, and robot simulation. To address the data collection problem, we propose the use of the gesture input device "Leap Motion" for collecting gesture data. The Leap Motion device is capable of tracking hand motion in real-time, and outputting a series of data frames that contain all the relevant information about the hand's motion, including the positions, velocities, directions, and rotations of all the hand's components (fingers, pointables, tools, gestures, etc.). By using the Leap Motion, we can collect data on the current position and speed of the hand, and use this information to identify the meaning represented by the current gesture. This data can then be used to train the robot to perform the desired action. With the help of robot simulation, we can evaluate and optimize the performance of the robot's action execution.

Overall, the use of the Leap Motion device for data collection, along with learning from demonstration and robot simulation, presents a promising approach to developing robots that can learn from human demonstrations.

Fig. 2. The skeleton joints detected by LeapMotion

Once we have collected enough data, the next step is to conduct imitation learning. To achieve this, we can refer to the imitation learning link in DeepClaw Tutorial and use the Gaussian Mixture Model (GMM) in Python to learn the position or speed information of the robot. This will help us to obtain more continuous and reasonable motion trajectories and strategies. After obtaining the optimized motion trajectory and strategy, we can apply it to the robot arm simulation. We conducted a preliminary investigation on four popular robot simulation platforms: PyBullet, ROS, MATLAB, and RoboSuite. PyBullet is ideal for developing and testing robot algorithms, especially reinforcement learning-related ones. It provides efficient physical simulation and motion planning, which is crucial for training reinforcement learning agents. ROS, on the other hand, is more suitable for building robotic applications, rather than providing physical simulation and motion planning. MATLAB is ideal for developing and testing control systems and signal processing algorithms. It is well-suited for projects that require high-level mathematical calculations and analysis. Finally, RoboSuite is designed for developing and testing operational tasks for robot algorithms.

It is suitable for projects that require realistic simulation of robot manipulators and environments, as well as tasks such as grasping, pushing, and lifting. By considering the strengths and limitations of each simulation platform, we can choose the most suitable one for our project.

V. Intermediate Results

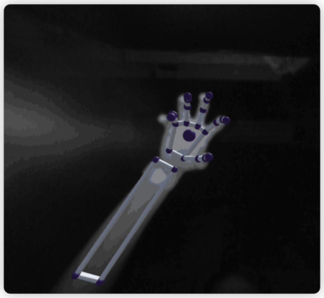

Fig. 3. Hand motion detection

Fig. 4. Gesture detection.

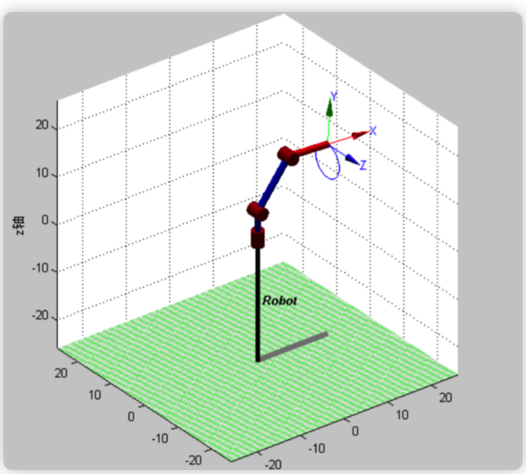

Fig. 5. Robot arm we test in Matlab

We are currently using the Leap Motion gesture input device to collect hand data for my project. When my hand enters the recognition area of the Leap Motion, it automatically tracks and outputs a series of real-time refreshed data frames that contain information about my hand's motion, such as position, velocity, direction, and rotation.

For the simulation environment, we have tested several robot arm that best fit our need. We are now focus on the table manipulator that have less freedom, because we have better performance in our tasks.

References

[1] Mandlekar, A., Zhu, Y., Garg, A., Booher, J., Spero, M., Tung, A., ... , Fei-Fei, L. (2018, October). Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In Conference on Robot Learning (pp. 879-893). PMLR.

[2] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arXiv preprint arXiv:1703.09312, 2017.

[3] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016.

[4] D.Kalashnikov,A.Irpan,P.Pastor,J.Ibarz,A.Herzog,E.Jang, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. arXiv preprint arXiv:1806.10293, 2018.

[5] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. 2021. DualRing: Enabling Subtle and Expressive Hand Interaction with Dual IMU Rings. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 5, 3, Article 115 (Sept 2021), 27 pages. https://doi.org/10.1145/3478114

[6] BOULABIAR M I, COPPIN G, POIRIER F. The Study of the Full Cycle of Gesture Interaction, the Continuum between 2D and 3D[C].16th International Conference, HCI International 2014, 2014

[7] Zeng Hualin, Huang Yuxuan, Chao Fei. A review of research on handwriting robots [J]. Journal of Intelligent Systems, 2016, (1): 15-26.