Zhejiang University-University of Illinois Urbana-Champaign Institute

# Individual Progress Report

# HUMAN-ROBOT INTERACTION FOR OBJECT GRASPING WITH VIRTUAL REALITY AND ROBOTIC ARMS

NAME: Ziming Yan
UIN: zimingy3

Individual Report for Senior Design, Spring 2024

Major: Electrical Engineering

Supervisor: Gaoang Wang & Liangjing Yang

TA: Tielong Cai & Tianci Tang

April 17, 2025

Project No. 26

# Contents

# 1 Introduction

## 1.1 Project Overview

Current robotic systems often struggle with providing an intuitive and seamless human-robot interaction, especially for tasks like object manipulation. Traditional teleoperation methods frequently rely on complex controllers, making natural interaction challenging for users. However, with recent advancements in Virtual Reality (VR) and robotics, there's an opportunity to develop an intuitive interface that allows users to manipulate objects in a virtual environment, with a robotic arm replicating these actions in real-time. This project aims to bridge the gap between human intent and robotic execution by integrating VR with robotic grasping systems, enabling precise and efficient remote object manipulation. This approach opens up new possibilities across various sectors, such as remote work with precise control, and assisting people with conditions like Parkinson' s disease in performing daily tasks.

To be specific, our solution addresses the problem by the following workflow:

**Step 1:** First, we create a digital twin of the target objects alongside the robotic arm in the Unity environment.

**Step 2:** Then, we connect this digital twin with the Meta Quest VR App and render the scene, which is updated in real-time.

**Step 3:** During runtime, the Quest reads in the user's hand trajectory, enabling it to recognize which target object to grasp.

**Step 4:** The Quest continuously returns this information, along with the hand trajectory, to the control module on the PC.

**Step 5:** The control module calculates a set of recalibrated waypoints and sends them to the Robot Operating System (ROS).

**Step 6:** ROS then orders the robotic arm to approach the object in a manner that simulates the user's grasping trajectory.

Robot Experiment Setup

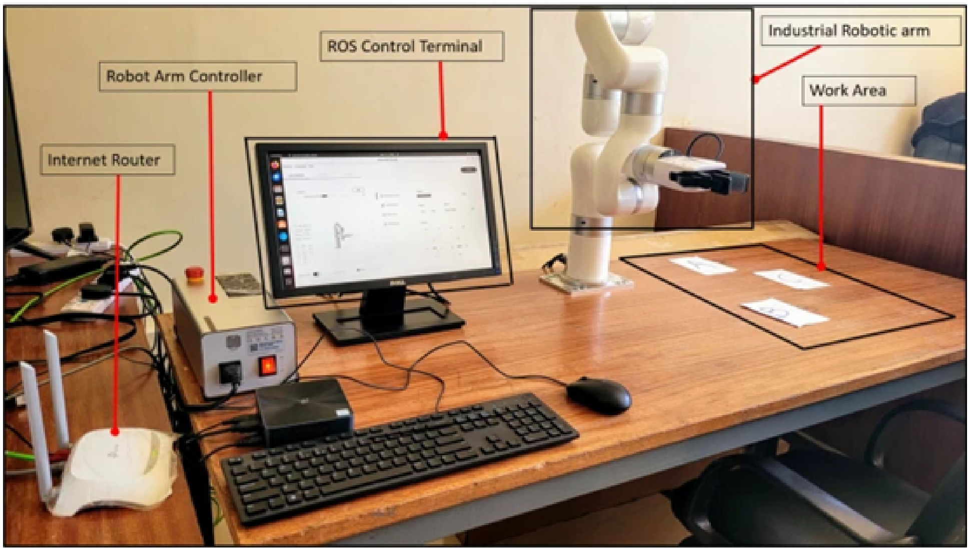


**Figure 1.1** Design Concept

## 1.2 Responsibilities and Role

As the only electrical engineering student in the team, I have a certain knowledge base of embedded chip control development, motor control and robotic arm operation. In addition, I have knowledge of mechanical structure and 3D modeling in my spare time as well as in my research projects. Therefore, I was also responsible for related tasks in the project. Specifically the following aspects:

1. Design the overall task division of the project and draw a Block Diagram
2. Record the progress of the experiment and test data in the lab notebook.
3. Configure the STM32 programming environment and debug the STM32F407 development board.
4. Control the development board to realize the communication with the stepper motor driver board, and then control the stepper motor.
5. Modeling and optimizing the mechanical gripper with mechanical students to control the UR3 arm.

# 2 Individual Design Work

## 2.1 Design and Draw the Block Diagram

The most important task of the project, before starting the division of labor, is the structural distribution, i.e., dividing the functions to be implemented into specific modules and assigning them to each person according to his characteristics, specialties and strengths. I initially divided the project into four subsystems: Mixed Reality (MR), Digital Twins, Control System, and Robot Arm, but after talking with my mentor Yang Liangjing in the group meeting and discussing among the members, I decided to add an auxiliary positioning system to improve the accuracy of the clamping, which is assisted by an external positioning camera.

In addition, due to the limitation of MR devices of campus (It has been used in other projects before the construction of our project), we changed our original design with MR to virtrual reality (VR), and use the hand tracking technology to record the track of hands. With The use of handles would increase the accuracy of route recording, but at the same time increase the difficulty of writing the program. After going through the process of communicating with Jiayu Zhou, who is responsible for writing the program, he agreed to this change. The final Block Diagram is shown below:

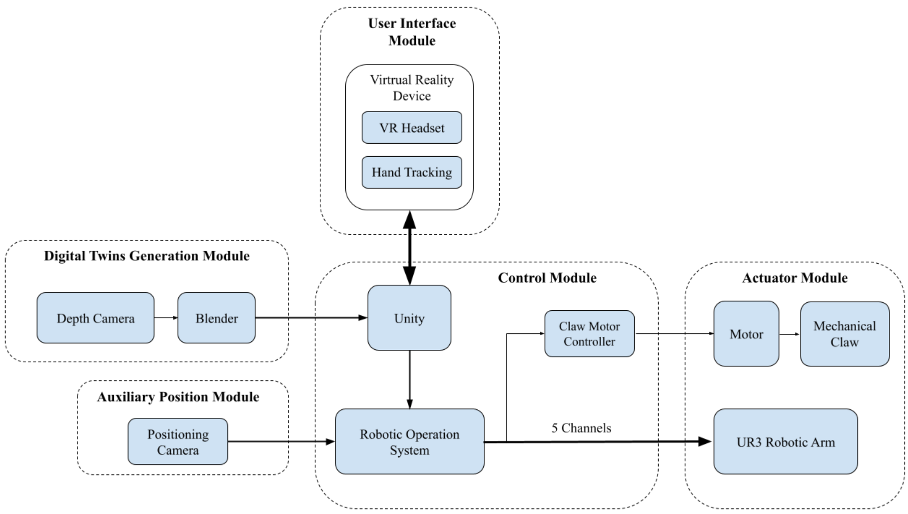


**Figure 2.1** Block Diagram

## 2.2 Control Module

- **Unity Integration:** The Unity will receive the data flow from VR hand tracking and transmit it to the ROS to compute the route of robot arm. In addition, it uses the 3D data from depth camera to reconstruct the environment.
- **ROS (Robotic Operating System) Communication:** According to the position infor-

mation of hands from Unity, ROS could computed the track into parameters of motors to control 5 robot arm joints and pass signal to STM32 motor control board.

- **Motor Controller:** Use STM32 control board to control motor speed, angle and other information with closed loop control.

As for me, I was responsible for implementing and programming motor controller firstly. At first, I wanted to use STM32F104 development board and SG90 rudder to drive the claw and the Jingxing Hu also used rudder parameters to design to 3D model. Through refering to files and websites, I learned how to construct the STM32F103 board compiling environment Keil-MDK [1] and successfully burned some simple programs as tests.

After that, I generate output PWM signal via SWD interface of STM32F103 board to control the angle of SG90 servo.

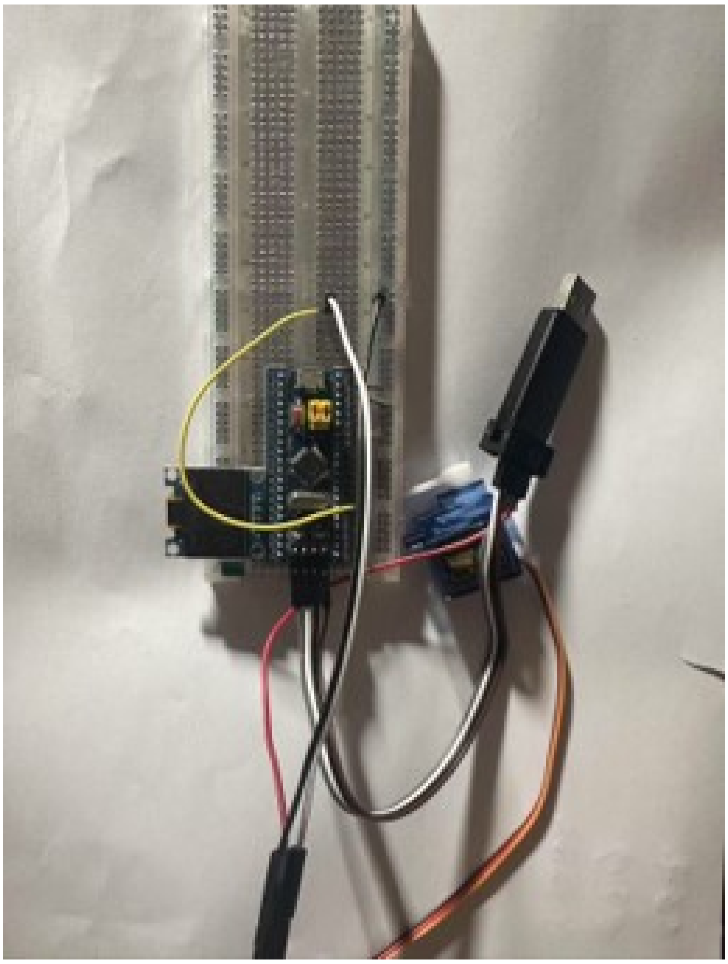

**Figure 2.2** Servo Motor Controlled by STM32F103

Though, the servo was able to drive the claw, it cannot achieve the close loop control. The servo motor was too simple with PWM control, and the torch it applied was also small which could not match our design. Therefore, I wanted to change it to stepper motor, and use STMF407 board to realize the close loop control. After discussion with mentors, they approved my suggestion. The STM32F407 chip has better M4 core and can operate with higher frequency signal compared with F103, which means its control signal would be more accurate. There needs to explain that the 42 stepper motor should be controlled with 4 lines, 2 poles. So, it always needs drive board to control. As for my project, I decided to use Emm42_V 5.0 board to control. The reason is that this board can offer close loop control with same physical demension with motor, I can attached to the back of motor to save space. The following figure shows my design of new control system of stepper motor.

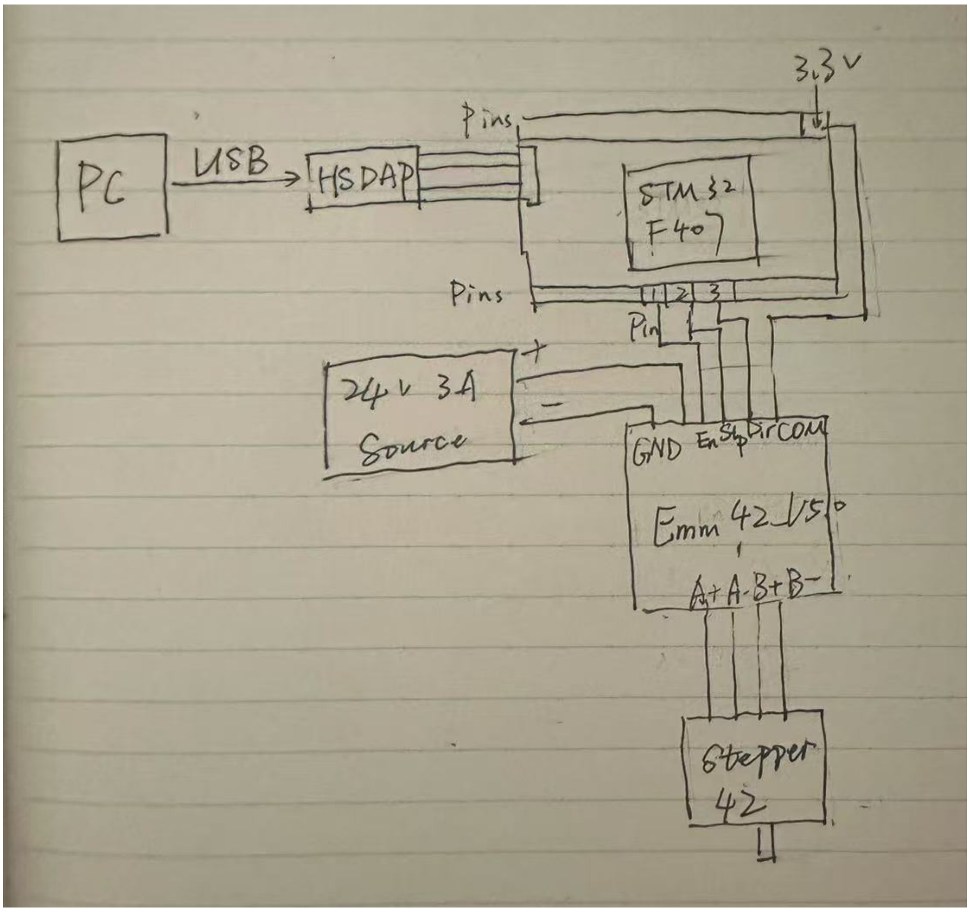


**Figure 2.3** Motor Controller Design Graph

## 2.3 Design Verification Approach

While writing the design document, I wrote the Tolerance Analysis. Through learning essays and thinking verification test, I got the limitation of UR3 robot arm and motor, including the approaches to test verification.

### 2.3.1 UR3 Robot Arm

There are limitations to robotic UR3, ensuring that it can accurately move following the computed route. Motivation speed is one of the most important factors. As mentioned in the instruction document, the general speed tolerance is -150mm/s. This means that if the user configures a 250mm/s speed limit, then the maximum operational speed will be 250-150=100mm/s. Safety tolerances prevent safety violations while allowing for fluctuations in program behavior. For example, when handling a heavy payload, there may be situations where the Robot Arm needs to briefly operate above the normal maximum operational speed to follow a programmed trajectory [2]. An example of such a situation is shown in figure.

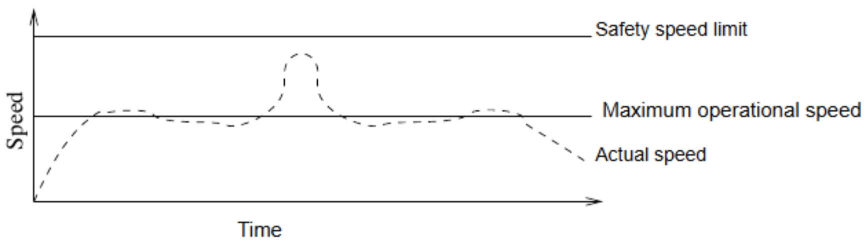


**Figure 2.4** Safety Tolerance Example

### 2.3.2 Torch of Grasping

The precision of grasping, especially whether the claw could successfully hold on to the items, mostly relies on the force of the claw applied. In other words, it depends on the torch range that the motor could supply. To verify the feasibility of our design, we want to use FEA ( Finite Element Analysis) software, like ANSYS, COMSOL, or Abaqus.

For example, the object material is plastic and can obtain 60 MPa maximum stress before damage. The friction coefficient ($\mu$) is 0.3. After simulation, we find we need to apply 15N force on the object, the contact area is $3mm^2$, according to the formula:

$$Pressure(P) = \frac{Force(F)}{Area(S)} = \frac{15}{3} \cdot 10^{-6} = 5MPa < 60MPa$$

We could ensure claw will not break the object if grasped successfully. Therefore, we could compute the corresponding Torch(T) with formula:

$$Torch(T) = Length(L) \times Force(F)$$

## 2.4 Digital Twins Generation Module:

I learned related knowledge to help the digital twin development of Yuchen Yang. The environment is scanned by a depth camera to get a 3D point cloud file. The depth data can be reconstructed into a 3D mesh by the SDK or plug-ins in Unity3D [3], and reconstructed into a virtual environment consistent with reality through Unity3D rendering [4].

## 2.5 Record Research Notebook

As I mentioned in the introduction, I was responsible for the recording of progress in the research notebook. For example, when Jingxing needs parameters of UR3 robot arm and servo motor at the beginning of 3D modle development, I recorded all the parameters needed in the notebook. In addition, I recorded the discussion and ideas for the future research.

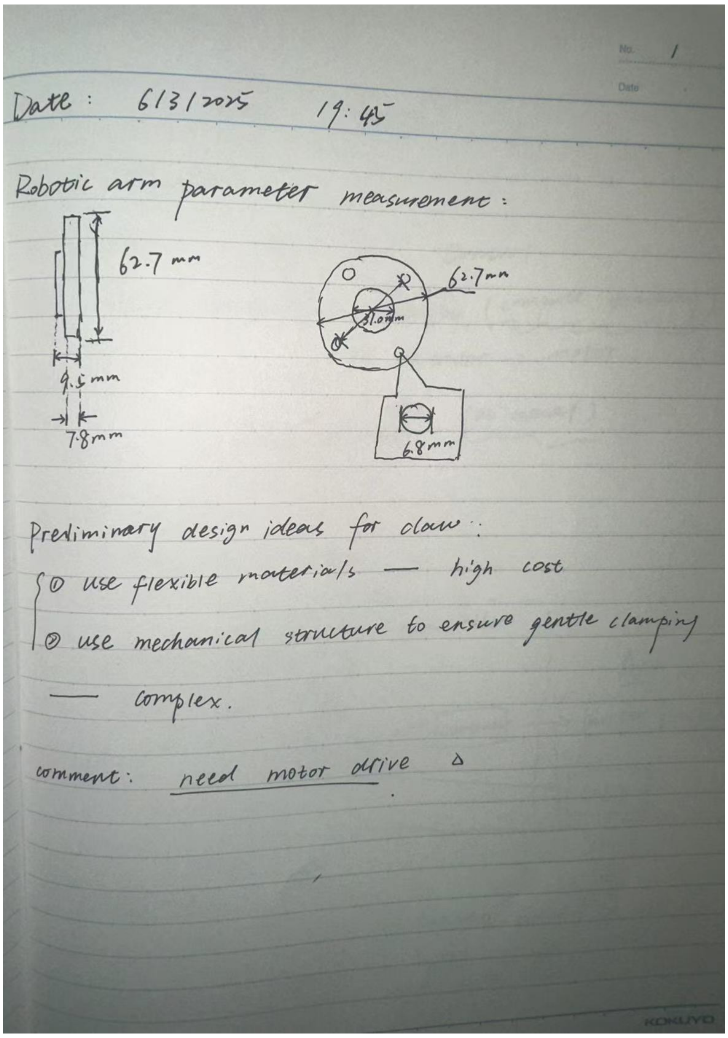


**Figure 2.5** Notebook

# 3 Conclusion

## 3.1 Self-assessment

Up till now, I have finished the STM32F407 board initialization successfully. Actually, my assignment progress is slightly behind of my schedule. I think the reason for the formation of this situation is that the preliminary route does not match the expected results, such as the preliminary F103 development board and servo combination can not meet the project requirements after a series of understanding before deciding to change to 42 stepper motor and F407 this trial and error time cost is higher. In addition, in the STM32F407 chip debugging to modify the loopholes and knowledge of stepper motor control knowledge also spent more time.

Although the progress is not the fastest, I think I played many irreplaceable roles in the team, such as the contribution of thinking during the discussion process, the presenter-sharer in the group meeting discussion, slides production, and report writing, which cannot be shown in the form of words and pictures. Overall, I think we will be able to complete our Senior Design successfully.

## 3.2 Plans for remaining work

| **Duration** | **Milestone** |
|---|---|
| 1 Week | Compile and burn STM32 code and debug by linking to stepper motor |
| 1 Week | Connect the claw control signal from ROS to STM32 |
| 1 Week | Realize the closed-loop control of the motor |
| 1 Week | Test the grasping and finish the Final Report |

## References

[1] DaniaoXP. “Stm32f103 入门学习.” Accessed: 2025-03-16. [Online]. Available: https://blog.csdn.net/daniaoxp/article/details/100101149.

[2] U. Robots. “Ur3/cb3 original instructions manual.” 2025-04-10. [Online]. Available: https://www.manualslib.com/manual/2019611/Universal-Robots-Ur3-Cb3.html?page=104.

[3] Unity Technologies. “Camera component reference.” 2025-04-01. [Online]. Available: https://docs.unity3d.com/cn/Packages/com.unity.render-pipelines.universal@12.1/manual/camera-component-reference.html.

[4] Unity Technologies. “Mesh renderer.” Accessed: 2025-04-01. [Online]. Available: https://docs.unity3d.com/6000.0/Documentation/ScriptReference/MeshRenderer.html.