ECE 445

SENIOR DESIGN LABORATORY

PROJECT PROPOSAL

MassageMate: Smart Robot Masseur for ECE 445

Team #14

HAO BAI (haob2)
WENTAO YAO (wentaoy4)
KE XU (kex5)
XIUYUAN ZHOU (xiuyuan5)

TA: Yutao Zhuang

March 5, 2023

Contents

1 Introduction

1.1 Problem

High-intensity work tends to cause fatigue in people's neck and waist, so they need massage to relax their shoulders and neck, but frequent visits to massage parlors cost quite time and are expensive. Massage benefits can include reducing stress and increasing relaxation, reducing pain and muscle soreness and tension, improving circulation, energy and alertness, lowering heart rate and blood pressure, and improving immune function.

However, human massage also has some drawbacks and risks. Human massage may directly cause new injuries, mostly bruises and nerve lesions, aggravate existing injuries and chronic pain problems, distract patients from more appropriate care, mildly stress the nervous system, or even cause blood clot, nerve injury or bone fracture in rare cases. Some chronic pain patients may be disastrously traumatized by intense massage . Moreover, some customers may not want to be touched by an unfamiliar person, or they have some body privacy that hope to be hidden from others, then a Robotic Masseur can help.

1.2 Solution

The project involves a phone app that allows users to control a robotic masseur with voice or touch. The app communicates with a cloud processing unit that uses ASR and large language model to understand the user's massage preferences and generate massage instructions. The instructions are sent to a local processing unit on Raspberry Pi that controls the robotic arm and massage head. The local processing unit also receives feedback from the robotic system for further adjustment. We also designed a massage head on the end of robotic arm.

The Robotic arm we use is OpenMANIPULATOR-P, controlled through DYNAMIXEL SDK through U2D2 connected to Raspberry Pi. We also designed the massage head, the head is basically a half-ball shape, fixed on a rack and pinion structure controlled by a high frequency motor. Another rack and pinion structure on the back of it will control the movement of massage head in vertical direction with the extension arm. The local control unit could adjust the massage force, frequency and height of this massage head, and receive sensors' outputs from it for processing.

1.3 Visual Aid

As shown in Figure 1, the proposed solution involves the use of a robotic arm for massaging a human. The human will lie on a bed, and the robotic arm, consisting of an arm and a dropper attached to the end, will control the movement and height of the massaging tool.

The robotic arm will be connected to a control panel that facilitates data transmission

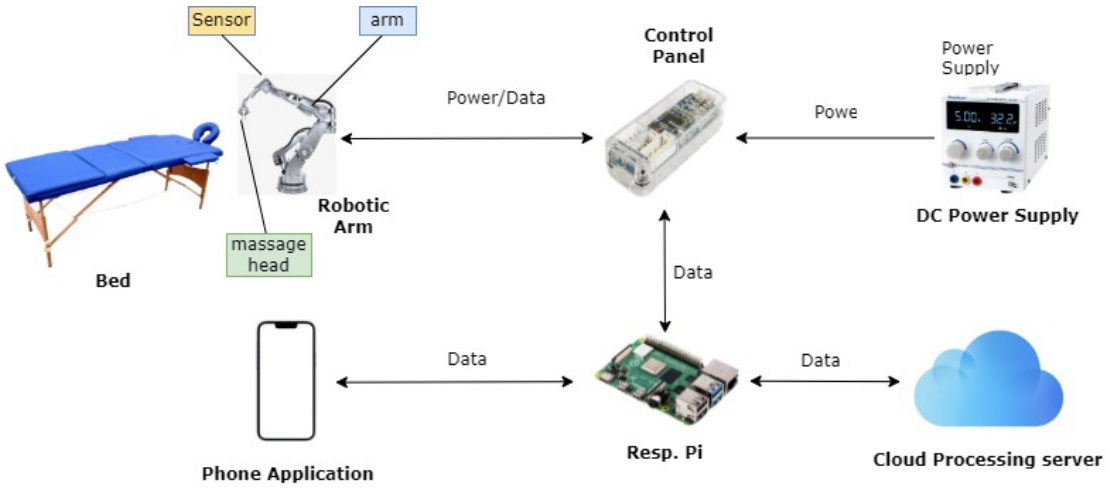

Figure 1: Pictorial Illustration

between the arm and the device's power supply. Additionally, the control panel will be connected to a Resp. Pi board responsible for processing data related to the arm's movement. The Resp. Pi board will communicate with a phone application that allows users to visualize and control the arm's movements.

To facilitate communication between the human and the device, the phone application will feature a program that can translate spoken commands into machine code that can change the state of the robotic arm. This approach aims to provide a safe and effective means of massaging the human while minimizing the potential for harm or discomfort.

1.4 High-level Requirement List

- The robotic arm and massage head can move according to the user's instructions and feedback, with appropriate distance, velocity, frequency and strength for massage.
- The phone app can record the user's voice and touch inputs and communicate with the cloud processing unit in real time.
- The cloud processing unit can use ASR and finetuened Natural Language Models to understand the user's intention for massage, and generate massage instructions for the local processing unit on Raspberry Pi.
- The local processing unit (Raspberry Pi) Could process massage instructions from cloud server in real-time, controlling control robotic arm and robotic heads simultaneously.

2 Design

2.1 Block Diagram

Figure 2 shows the block diagram of our design, which consists of two systems: Robotic System and Control System. The Robotic System includes the robotic arm and the massage-head that we designed to perform massage to the users. The Control System allows the user to control the robot through a phone app that we developed. The app sends messages to our cloud processing unit, which analyzes user input and generates massage instructions. These instructions are then sent to our local processing unit, which controls our robotic masseur.

2.2 Subsystem Overview

2.2.1 Robotic Arm Subsystem

The robotic arm we use in the project is OpenMANIPULATOR-P from ROBOTIS Co., Ltd. [1] This robotic arm is based on ROS and OpenSource, and it is composed of DYNAMIXEL-P modules. The DYNAMIXEL modules are the basic components of the robotic arm. They have a robotic modular form, and they can be combined together to form the arm with a chain method. To control the OpenMANIPULATOR-P from a PC, we use a U2D2 USB communication converter. This communication converter is fixed on a U2D2 Power Hub and connected with a TTL line. The Hub connects to the robotic arm also with a TTL line, supplying power and sending control instructions.[2]

After the voice detection module detects the user's voice and converts it to structured instructions, it sends them to the Robotic Instruction Converter Program that we designed in the PC. This program converts these high-level instructions to the instructions in DYNAMIXEL SDK, which is the software development toolkit for controlling OpenMANIPULATOR-P through the communication converter.

2.2.2 Massage-Head Subsystem

The control system will compare the actual data with the set value, the force sensor can get real-time data of massage pressure and send it to control system, the Rack and pinion(L) will control the vertical movement of the extra arm, the Rack and pinion(S) will control the massage pressure, and the High-frequency motor will control the vibration of massage-head with crankshaft structure.

We use mechanical arm to control x/y position, then large pack and pinion(red part) works to control massage-head goes down. Force sensor sends back real-time pressure and when there is a certain amount of pressure, which means massage-head touches user's back, the large pinion(red part) stop work. As for the crankshaft structure, when axis rotates, the pole (yellow part which will keep vertical) goes up and down according to sine function, and push the massage-head(semicircular part) to go up and down. When the high-frequency motor drive axis rotates, the massage-head (semicircular part) mas-

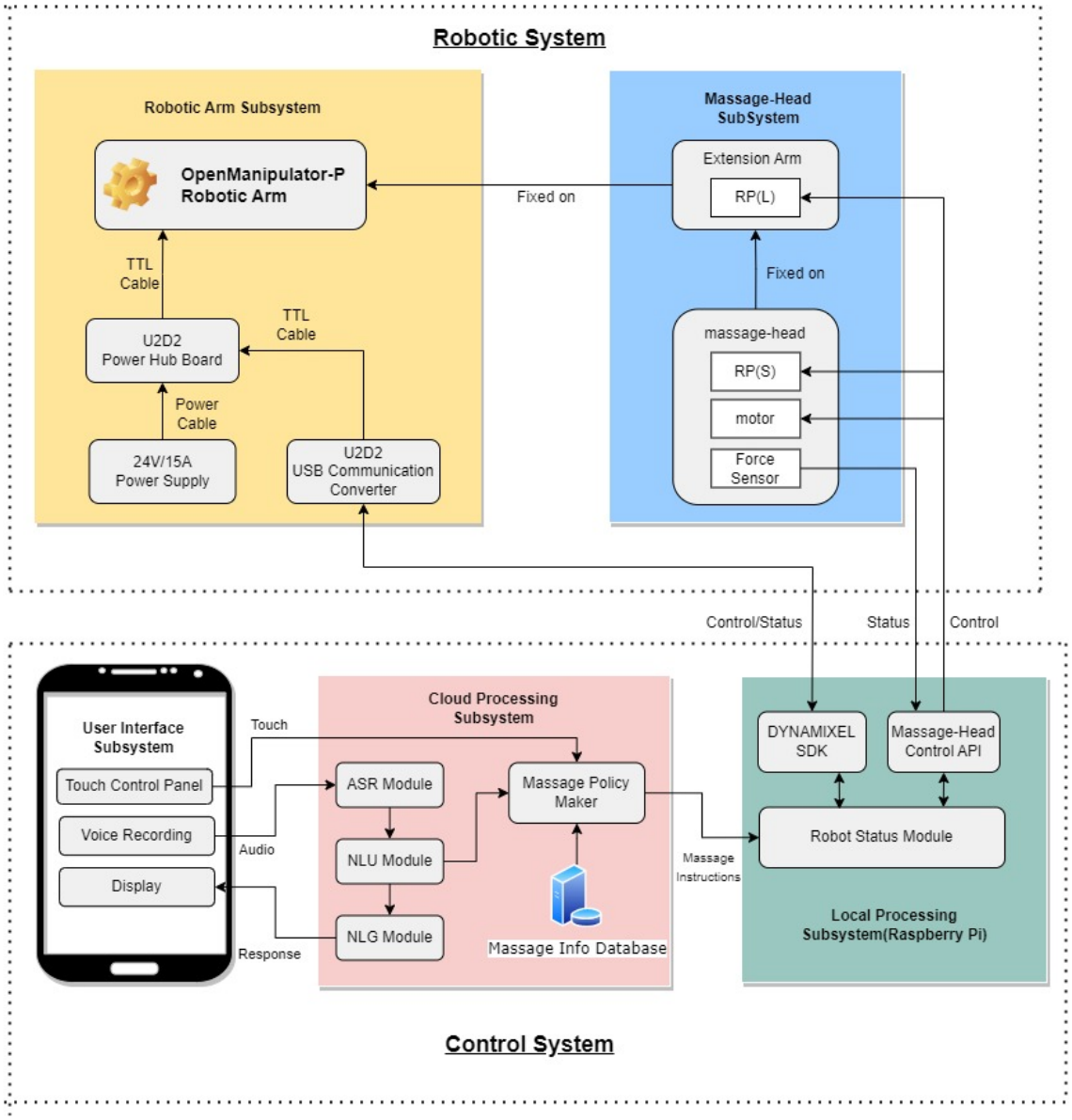

Figure 2: Block Diagram

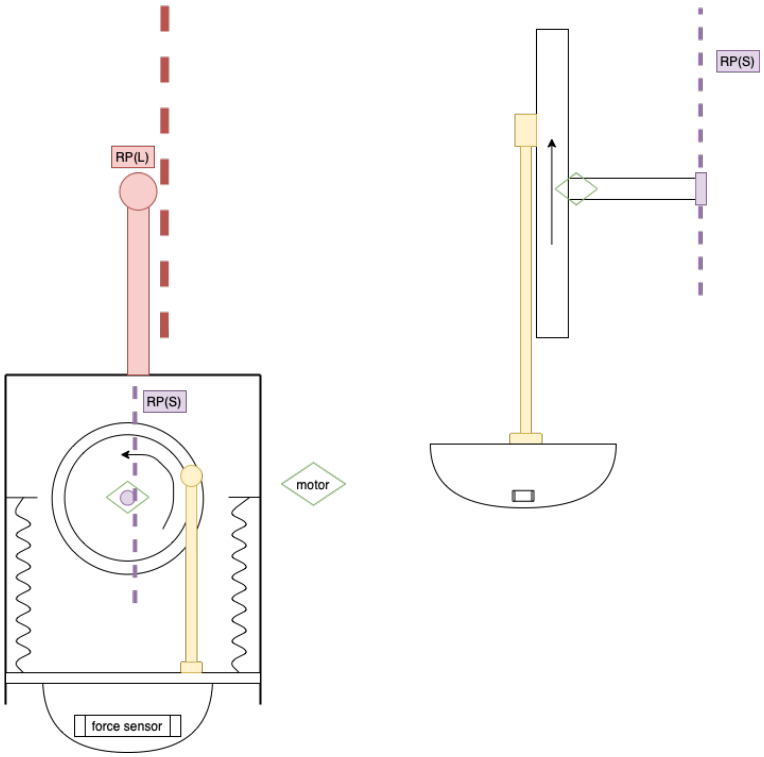

Figure 3: Massage-Head design

sage quickly. When pinion(purple) moves down, the massage-head(semicircular part) goes down and improve massage pressure.

2.2.3 User Interface Subsystem

As shown in Figure 4, the button in the upper-right corner is a switch, the center box shows the position of the masseur robot arm relative to the person, and the buttons in the lower-left corner are designed to move forward, downward, leftward and rightward, the "rotate" button can let a robotic arm knead, the swivel knob can reset the force, and the emergence key can pause the MassageMate.

2.2.4 Cloud Processing Subsystem

The control system is essentially a task-oriented dialog system. It construs of 4 main modules: NLU, DST, PM, and NLG.

- NLU (Natural Language Understanding): The NLU module is responsible for understanding the user's natural language input, such as speech or text.**texbook** It performs several tasks such as tokenization, named entity recognition, intent classification, and slot filling. Tokenization breaks the user input into individual words, while named entity recognition identifies important entities such as names, dates, and locations. Intent classification predicts the user's intention based on their in-

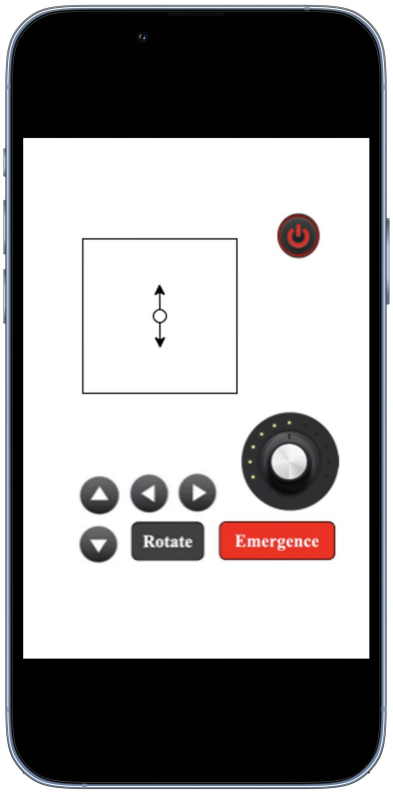

Figure 4: Phone App for Users

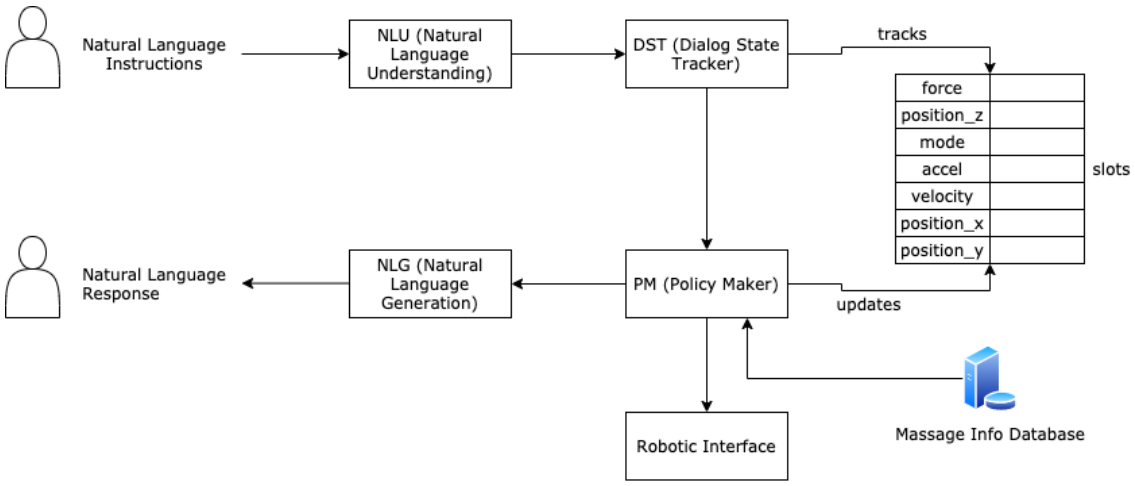

Figure 5: Slot Diagram

put, while slot filling identifies specific pieces of information that the user has provided.[3]

- DST (Dialogue State Tracking): The DST module keeps track of the current state of the conversation. It uses the output from the NLU module to update the dialogue state, which represents the current context of the conversation. The dialogue state typically includes information such as the user's goal, the user's preferences, and the current task being performed.[4]
- PM (Policy Manager): The PM module is responsible for determining the appropriate response to the user's input based on the current dialogue state. It uses a combination of rules and machine learning algorithms to generate a response that meets the user's goal and takes into account the current context of the conversation.
- NLG (Natural Language Generation): The NLG module is responsible for generating the final response to the user's input in natural language. It takes the output from the PM module and generates a response that is grammatically correct, semantically meaningful, and appropriate for the context of the conversation. [5]

The interaction between these modules typically involves a feedback loop where the output of one module is used as input for the next. For example, the NLU module outputs the intent and slots extracted from the user's input, which are used by the DST module to update the dialogue state. The PM module then uses the updated dialogue state to generate an appropriate response, which is passed to the NLG module for natural language generation. The resulting response is then presented to the user, and the process starts again with the user's next input. [6]

In our task, there are several slots for filling: '[force, position_z, mode, acceleration, velocity, position_x, position_y]'.

- 'force': This slot refers to the amount of force to be applied during the massage. The value for this slot could be a numerical value that specifies the force in Newtons or some other unit of force.
- 'position_z': This slot refers to the vertical position of the robotic arm during the massage. The value for this slot could be a numerical value that specifies the position in meters or some other unit of length.
- 'mode': This slot refers to the mode or type of massage that the user wants. The value for this slot could be a string or categorical value that specifies the type of massage, such as deep tissue massage, Swedish massage, or shiatsu massage.
- 'acceleration': This slot refers to the rate at which the robotic arm should accelerate during the massage. The value for this slot could be a numerical value that specifies the acceleration in meters per second squared or some other unit of acceleration.
- 'velocity': This slot refers to the velocity or speed at which the robotic arm should move during the massage. The value for this slot could be a numerical value that specifies the velocity in meters per second or some other unit of velocity.

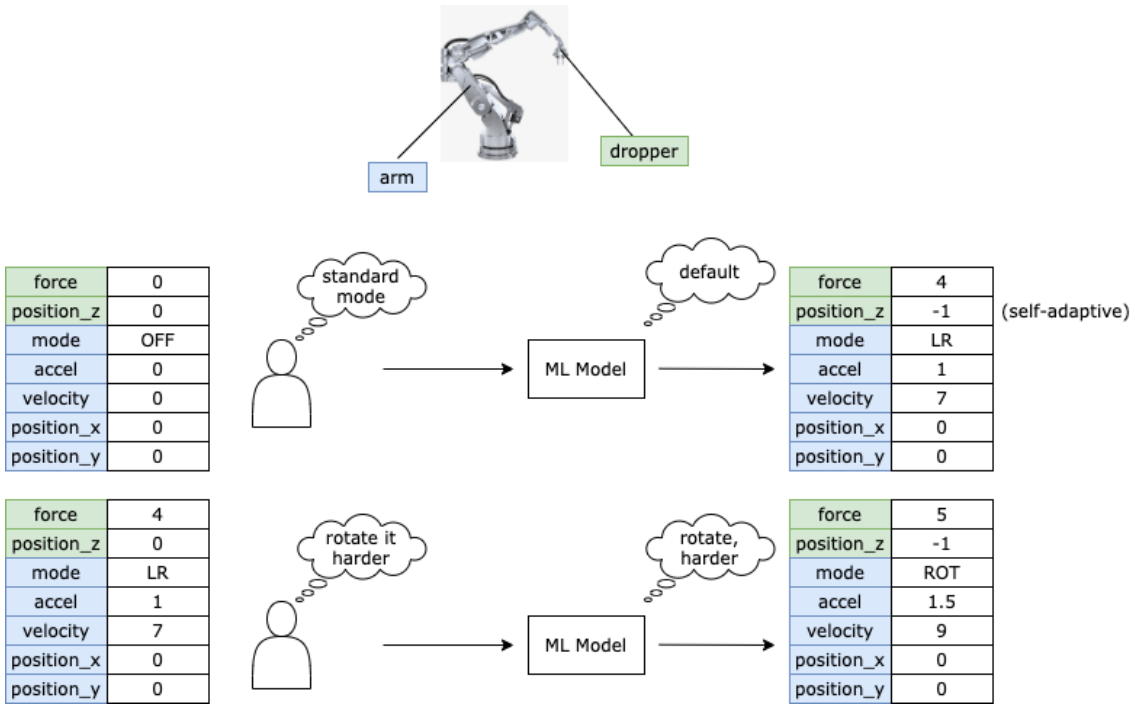

Figure 6: Slot Pictorial

- 'position_x': This slot refers to the horizontal position of the robotic arm during the massage. The value for this slot could be a numerical value that specifies the position in meters or some other unit of length.
- 'position_y': This slot refers to the lateral position of the robotic arm during the massage. The value for this slot could be a numerical value that specifies the position in meters or some other unit of length.

Slot filling & updating example:

After the user utterance, the ASR module parse the voice to text.

Upper: from 'OFF' mode to 'LR' (left-right move) mode. It's default massaging mode.

Lower: from 'LR' mode to 'ROT' (rotation move).

2.2.5 Local Processing Subsystem

We use Raspberry Pi to implement the local processing unit for our robotic system. We design a Robot status Module program in the Ras.Pi to process the massage instructions from Cloud Processing subsystem. Whenever our robot status module get the instruction message from our cloud server, it will firstly analyze the instruction, convert it to specific motion parameters of both robotic arm and massage head. Then, it will send control signals to robotic arm and massage head through DYNAMIXEL SDK and Massage-Head Control API separately. At the same time, both of the control interfaces will also receive status from robotic system.

2.3 Subsystem Requirements

2.3.1 Robotic Arm Subsystem

- The robotic arm must operate in appropriate velocity and position accurately as input by the instructions.
- The communication delay between robotic arm and control system should be small to support real-time application.
- The motion of robotic arm must not hit people or other components surrounding it.

2.3.2 Massage-Head Subsystem

- Force sensor must fast enough to send back real-time data.
- Racks and pinions must have enough mechanical strength so that when the massage head vibrates, it will not be broken. Those also need appropriate room won't occupy too much room but can be well adjusted.
- Motor must have high enough frequency and low noise.

2.3.3 User Interface Subsystem

- The user interface should be simple and easy to understand considering easy access and operation.
- Overall response time should be rapid and the delay of information transmission to the server should be small.
- Security must be a high priority to prevent users from receiving fraud and abuse that endangers hardware, data, and even human security.

2.3.4 Cloud Processing Subsystem

- The task-oriented dialogue system requires a robust and reliable infrastructure to support the processing of user input and generation of appropriate responses.
- The system should be able to accurately classify the user's intent with a high degree of accuracy, extract named entities from the user's input with a high degree of accuracy, and handle input in one language at a minimum.
- It should be able to accurately update the dialogue state based on the user's input, generate an appropriate response that meets the user's goal and takes into account the current context of the conversation, and handle basic conversations with a limited number of turns.

2.3.5 Local Processing Subsystem

- The communication delay between local processing system and cloud server must be small to support real-time application
- The control program in this subsystem should operate both robotic arm and massage head simultaneously. And get motion status information from them as well.

2.4 Tolerance Analysis

- The motion accuracy of robotic arm might be hard to reach for dedicated massages on human body.
- When recording audio commands, the sound may be overridden by ambient noise.
- When a move command is issued through the mobile app, the exact amount of movement needs to be determined through trial and error
- The force sensor's electrical signals need to be converted into data and transmitted in real time to the control system, which calculates the average pressure and compares it with the target value. It needs to rule out the changes caused by the body's breathing or other normal causes, and finally determine which pinion is moved and if the pinion is moved up or down.
- Mechanical construction requires accurate data to integrate each part, which is difficult for teams without ME students.
- If the task-oriented dialogue system cannot meet all requirements, the minimum requirements for each subsystem should be prioritized.
- The NLU subsystem should be able to accurately classify the user's intent and extract named entities from the user's input with a high degree of accuracy and handle input in one language at a minimum.
- The DST subsystem should be able to accurately update the dialogue state based on the user's input and handle basic conversations with a limited number of turns.
- The PM subsystem should be able to generate an appropriate response that meets the user's goal and takes into account the current context of the conversation and handle basic conversations with a limited number of turns.
- The NLG subsystem should be able to generate a response that is grammatically correct and semantically meaningful, appropriate for the context of the conversation, and handle one language at a minimum.

3 Ethics and Safety

Developing a project that involves the use of robotic arms for massaging humans presents both ethical and safety considerations. The ethical issues that need to be considered in-

clude issues of privacy, informed consent, and the potential for harm to the client.

The IEEE Code [7] of Ethics emphasizes the importance of the safety and welfare of the public. Therefore, the project should prioritize the safety and comfort of the human clients. The ACM Code of Ethics [8] emphasizes the importance of avoiding harm, and the project should strive to avoid any potential harm to the clients.

To avoid ethical breaches, the project team should obtain informed consent from the clients before beginning the massage. The clients should be informed of the benefits and risks of the massage, as well as any potential discomfort or pain that may arise. The project team should also respect the privacy of the clients by ensuring that any personal information obtained during the massage is kept confidential.

In terms of safety, the project team should adhere to relevant safety and regulatory standards. For instance, the American National Standards Institute/Robotic Industries Association (ANSI/RIA) safety standards outline safety requirements for industrial robots. The team should also review state and federal regulations, industry standards, and campus policy to ensure that the project complies with relevant safety standards.

Potential safety concerns that need to be addressed include ensuring that the robotic arm is programmed correctly to avoid injuring the clients, and that the robot's movements are gentle enough to avoid causing pain or discomfort. The project team should also consider potential risks such as electrical hazards, fire hazards, and the potential for the robot to malfunction.

In conclusion, developing a project that involves using robotic arms for massaging humans requires careful consideration of ethical and safety concerns. The project team should prioritize the safety and welfare of the clients, obtain informed consent, and adhere to relevant safety standards to avoid potential harm.

References

[1] OpenManipulator. "OpenManipulator Pro." (2023), [Online]. Available: https://emanual.robotis.com/docs/en/platform/openmanipulator_p/overview/ (visited on 03/08/2023).

[2] OpenManipulator. "U2D2 e-manual." (2023), [Online]. Available: https://emanual.robotis.com/docs/en/parts/interface/u2d2/ (visited on 03/08/2023).

[3] J. Allen, *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc., 1995.

[4] A. R. Williams Jason D., "The dialog state tracking challenge series: A review," *Dialogue & Discourse*, vol. 7, no. 3, pp. 4–33, 2016.

[5] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[6] H. Chen, "A survey on dialogue systems: Recent advances and new frontiers," *Acm Sigkdd Explorations Newsletter*, vol. 19, no. 2, pp. 25–35, 2017.

[7] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/08/2023).

[8] ACM. "ACM Code of Ethics." (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/08/2023).