ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT DRAFT

MassageMate: Smart Robot Masseur for ECE 445

Team #38

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

TA: Yutao Zhuang

May 9, 2023

Abstract

Our proposed solution involves the development of an intelligent robotic masseur capable of performing automated massage.The project utilizes a robotic arm equipped with customized massage head, integrated with various sensors to provide an intelligent and personalized massage experience. Users can control the robotic arm using voice instructions and a camera interface. The Natural Language Processing (NLP) module analyzes user instructions to determine the optimal force, frequency, and massage position for the massage head. Additionally, a visual detection model identifies suitable massage points and guides the arm to those specific locations. The vital signal sensor continuously monitors the user's physiological responses, allowing the robotic arm to automatically adjust the massage force according to the user's body condition.

Keywords : Robotic Masseur, Therapy, Healthcare, OpenMANIPULATOR-P

Contents

1 Introduction

1.1 Problem and Solution Overview

1.1.1 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. As an important component of Traditional Chinese Medicine (TCM), science of acupoint therapy 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, the number of acupuncturists who truly have qualified acupuncture skills is limited.

Besides, human massage also has some drawbacks and risks. Recognizing, positioning acupoints, and massaging is heavily depends on the skills of practitioners. 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 [1] can help. Therefore, a machine or a robot that can perform the acupoint therapy automatically will reduce the workload of practitioners and benefit to people's daily health care.

1.1.2 Solution (System Overview)

The project utilizes Robotic arm and several sensors to massage users intelligently. We use a customized massage head, working together with the robotic arm to do massage. The user can control the robotic arm with voice instructions as well as a camera. The NLP module (using voice instructions) will analyze user's in order to control the force, frequency as well as the position for the massage head. Moreover, the visual detection model will detect the massage points that are suitable for massaging, and lead the arm to move to that point. In addition, the vital signal sensor will also provide feedback to the body condition information so that the robotic arm could adjust the massage force automatically.

The Robotic arm we use is OpenMANIPULATOR-P, controlled through DYNAMIXEL SDK through U2D2 connected to the Arduino UNO. 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.

We also have a vital signs monitoring subsystem to detect whether the human feels comfortable or not. When the human feels uncomfortable, his blood pressure will go up and heart rate will also exceed the threshold. When this happens, we want the massage head to slow down its movement, thus making the person more comfortable.

For the vision part, we use a camera to locate where the person is, and use a linear regression model to map the positions to the program. We have several pre-defined points on the human body stored in the program, and will let the massage head to certain locations.

1.2 Visual Aid

[!h] The figure 1 is the pictorial representation of our project.

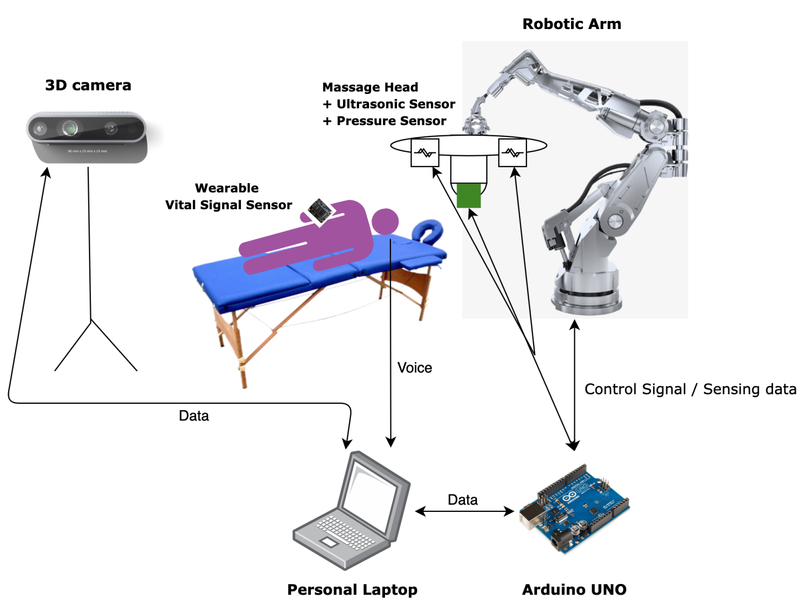

Figure 1: Visual Aid

Figure 2 shows the final product that we made.

Figure 2: Photograph of the final product.

The figure 3 is block diagram for our project. The robotic arm and robotic control subsystem is to control the motion of robotic arm with massage head. In the massage head subsystem, we designed a simple massage head with several sensors on it to perform massage on human's back. The acupoint subsystem is to detect acupoints using computer vision models. And the NLP subsystem is to process user's input instructions. For now, the user interface subsystem is simplified as just receiving user's voice instructions. And Vital signal subsystem is going to integrated together for now due to the large noise from the sensors.

2 Design

2.1 Design Procedure

This work proposes a design made up of 3 subsystems: the computer vision subsystem, the natural language processing system, and the vital signs detection system. The three systems are designed to be separate to reduce complexity.

We can also go a merged way. For example, we can merge the natural language processing and computer vision part, so that the user can talk when in a visual detection mode. However, this merge will introudce too much complex for the design because we need two threads. The voice system should always be listening to the microphone, which will be a separate thread.

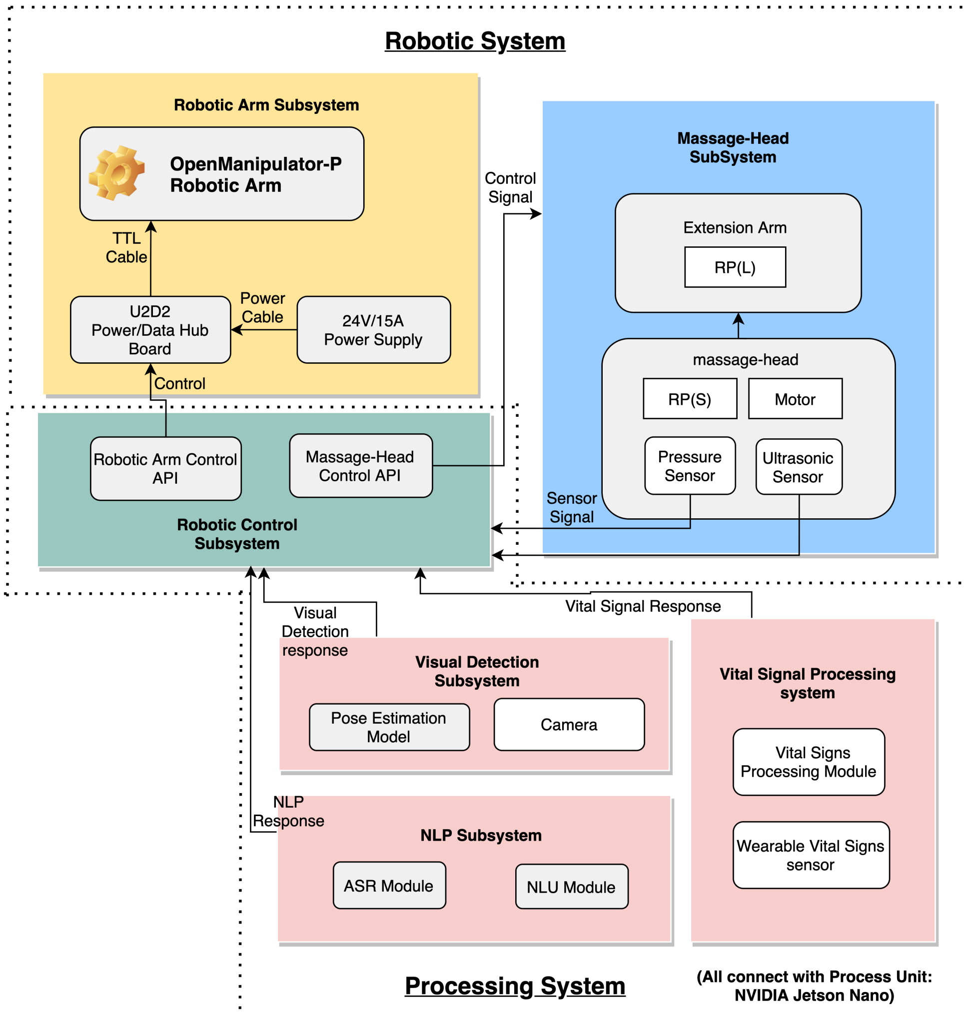

Figure 3: Block Diagram

We utilized lots of tools in our design. For softwares, we used the Arduino software, Google voice translation API, the Intel camera software, etc. For hardwares, we used the Intel camera, the OpenMANIPULATOR-P robotic arm, the built-in microphone, the force sensor, the vital signs sensor etc..

2.2 High-level Requirements (Points Summary Table)

- 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 ultrasonic sensors can adjust the position of the massage head when it detects a movement of the person being massaged, and a camera is applied to detect the

points of the human.

- The force sensor can detect the force on the human and whether the robotic arm should move up (too much force) or down (too little force).
- The Processing unit can use ASR and finetuened Natural Language Models to understand the user's intention for massage, and generate massage instructions for controlling the robotic arm and massage head.

2.3 Design Details

2.3.1 Processing Unit

The processing unit in our system is the hardware that supports collecting and processing data from sensors and robotic arm, running deep learning model inference, and control the robotic system. Therefore, the processing computer needs to contain GPU as well as having USB interfaces to receive signals from Arduino.

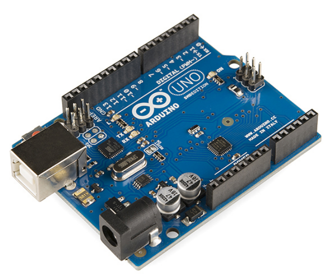

Figure 4: Arduino UNO ONE

We choose Arduino UNO (see Figure 4) as our connection between sensors, robotic arm and the computer. It is powered by 5V/3A power supply with Type-A power connecter. The Arduino UNO board has many different types of hardware interfaces. The USB Type A will be used to connect the robotic arm, the 40-pin Exp Header is used to connect sensors in our system. 3D camera is connected to the Camera Conn Interface.

2.3.2 Robotic Arm Subsystem

The robotic arm we use in the project is OpenMANIPULATOR-P from ROBOTIS Co., Ltd.[2] 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. And the U2D2 will be connected to USB-A interface in Jetson board.The specification of robotic arm is shown in 5. It needs 24V/15A DC power supply, stationed at a movable base. This robotic arm has six-degree of freedom. It can generally reach nearly 700mm in length, with maximum payload 3kg.

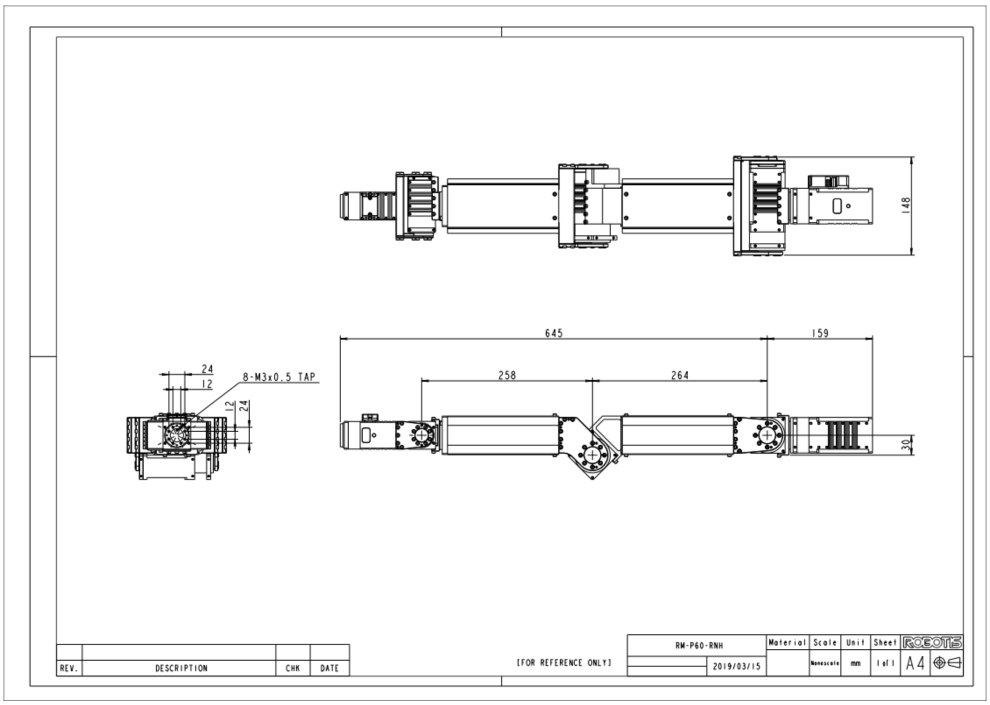

Figure 5: OpenMANIPULATOR-P Specification

2.3.3 Robotic Control Subsystem

The functionality for the control system is to gather and process all sensing data, and then generate massage policy to control the robotic arm and massage Head. It will connect directly to NLP subsystem, Visual detection subsystem, vital signal subsystem and sensors on massage head to receive input. Then, through the processing from the control program, it will send control instruction in robotic arm control API and massage head control API for controlling them separately.

Robotic Control The robotic arm is connected with a communication converter called U2D2. It support ROS to control its motion in task space. It is connected as the figure 6 shows. To control the motion of robotic arm, we use the following ROS service:

- **/open_manipulator/goal_task_space_path** : Set the goal pose in task space at $pos = (x, y, z), Quat = (w, x, y, z)$
- **/open_manipulator/goal_joint_space_path_from_present**: Set the goal pose offset in task space at $\Delta pos = (\Delta x, \Delta y, \Delta z), \Delta Quat = (\Delta w, \Delta x, \Delta y, \Delta z)$

We utilize those API for our control program of the robotic arm. Whenever the robotic arm receive the messages from perception subsystems and user inputs, it will then calculate the target location in both $pos = (x, y, z)$ and $Quat = (w, x, y, z)$ as well as the time for motion t to control the motion speed. And the control program will also get robotic arm pose information in real time, which could help the control program to adjust arm's motion.

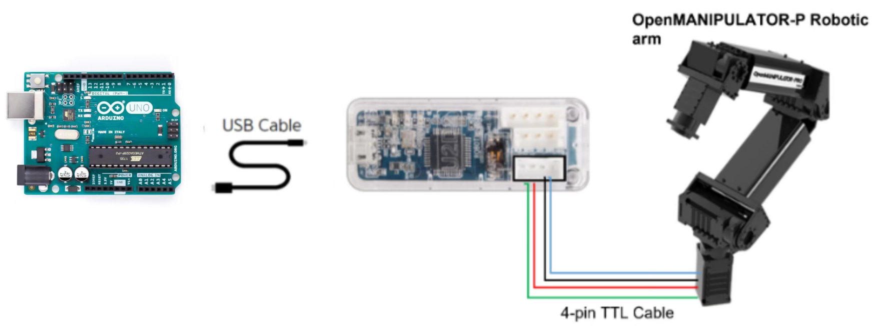

Figure 6: Robotic Arm connection

Massage Head Control The massage head is controlled by the Arduino UNO chip.

Through a specialized chip**??**, the signal is translated to force value. If the pressure value is too large/small, the robotic arm will move up or down the massage head.

And single chip also controls the vibration of massage-head by control the high-frequency motor12.

Around the massage head is a ultrasonic ranging module. When the measured distance value increase suddenly or two values has too much difference, they think message head leave body area, and force massage head stop moving.

For our final design, we only use one ultrasonic distance sensor to detect the distance to human's body. The reason for that is rely on the consideration of ups and downs on human's back, which may cause large error when we average the values from the two sensors. By using one sensor, we first move the sensor upon the proposed position, detecting the distance of it and then move our massage head down based on sensors' value.

Control Program The following Flowchart figure7 shows how to control the robotic arm to do massage based on a specific massage point and force given. Input to the control

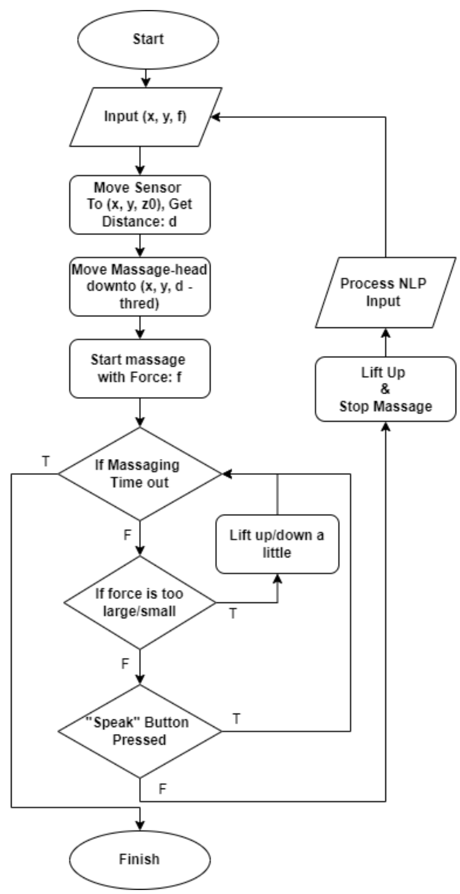

Figure 7: Control Program

program is the massage location and force. Firstly, it will move the ultrasonic distance sensor upon the massage point (x,y), get the distance (d) to human's back on that point. Then, the massage head will move down to the human's back at that point with a safety threshold (thred). And then turn on the motor on massage head, beginning massage phase. During this massage phase, when the pressure sensor, which is fixed on massage head, detect that the pressure is too large/ too small, it will move up/down with a small

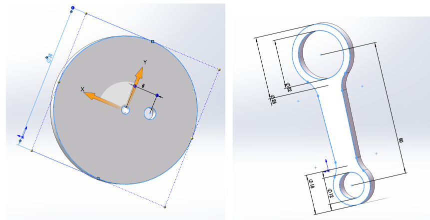

Figure 8: Eccentric Gear and Rotate-rod

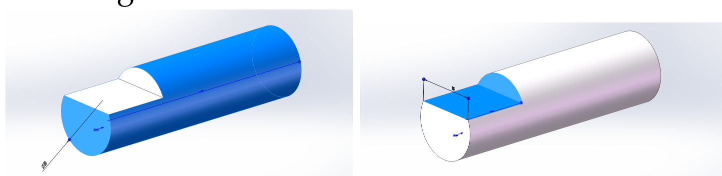

Figure 9: Rectilinear Moving Rod

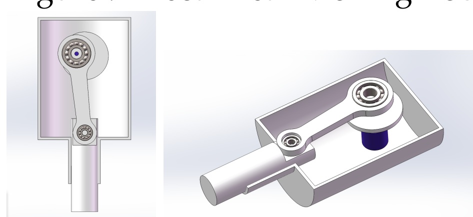

Figure 10: Massage Head

value (dz) accordingly. Additionally, in the massage phase, when user press the ”speak” button, the robotic arm will lift up and stop the massage head, then user can speak the voice instructions.(This serve both as a user instruction input button and the emergency stop button).The control program will follow this instruction for next massage. In the end, when the time for massage a single location is reached, the program will finish.

2.3.4 Robotic Massage Head Subsystem

The massage head consists of crankshaft construction and a rubber ball. Crankshaft includes eccentric gear2.3.4, rotate-rod8 and rectilinear moving rod9.

Offset 4mm of eccentric gear’s center, rotate-rod connects with gear through a bearing (GB 608, OD 22mm and ID 8mm). the other end of rotate-rod connect with rectilinear moving rod through another bearing (GB 604, OD 12mm and ID 4mm). Rubber ball is connect to rectilinear moving rod with pressure sensor stuck to it. 10

We use R370 motor12 to drive eccentric gear. When eccentric gear rotates, rotate-rod the goes up and down according to sine function, and push the rectilinear moving rod and rubber ball.

Figure 11: Pressure Sensor

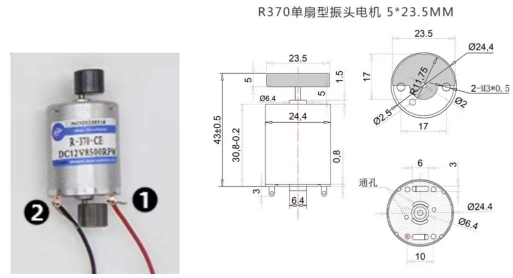

Figure 12: Motor

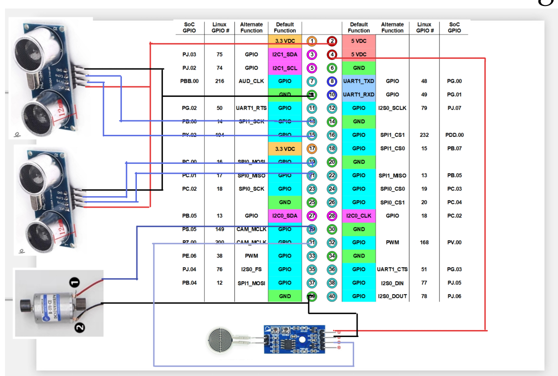

Figure 13: Massage Head Sensor Layout

Finally, constrained by 3D printing technology and strength needs, I decided to directly modify a molded massager,throw away the original control module and add the sensors and wiring we needed. I have found a factory to help us make this part. Its principle of converting circular motion into straight motion is the same as the previous idea.

2.3.5 Visual Processing Subsystem

The Vision Processing Subsystem is built on the MediaPipe framework. MediaPipe is chosen for its real-time optimization and compatibility with various platforms. The Pose Landmarks module utilizes pretrained models to detect human body landmarks and outputs their 3D coordinates. The identified landmarks, including the left and right shoulders and hips, define the relevant massage region.

We have another comparison that is shown in Figure 15.

The subsystem focuses on 2D vision information rather than depth data due to accuracy issues and computational limitations. Force and ultrasonic sensors are proposed for more effective pressure detection during massage sessions.

In this task, we will use Intel RealSense D435i Depth Camera shown as 16, which could offer quality depth for a variety of applications. With a range up to 10m, this small form

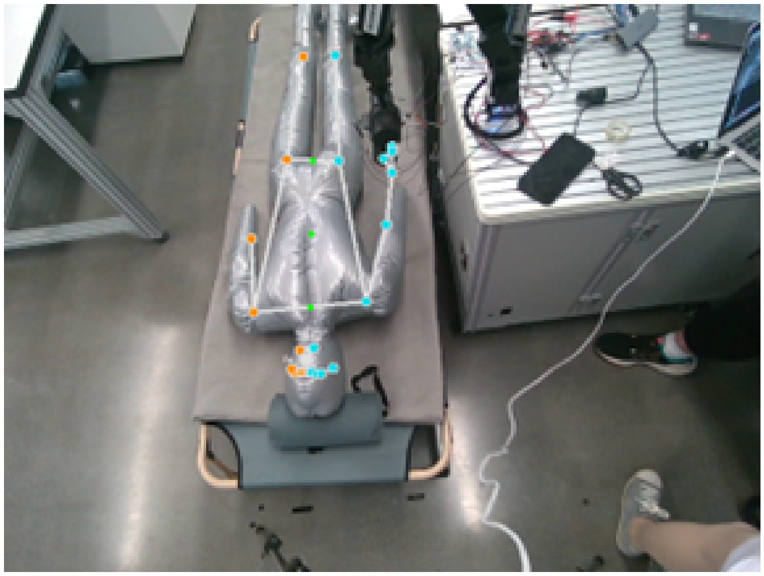

Figure 14: Pose estimation using camera.

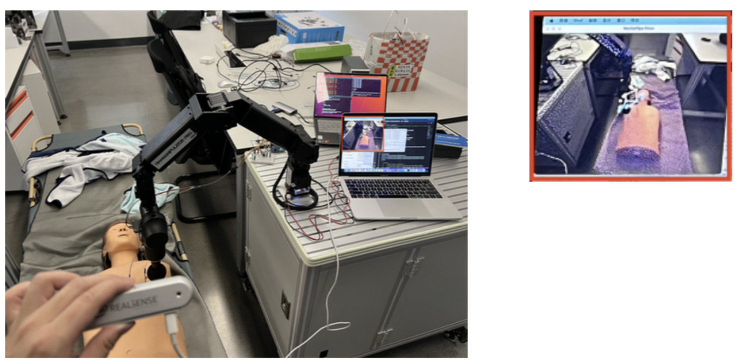

Figure 15: Pose estimation using camera (comparison).

factor camera can be integrated into any solution with ease, and comes complete with our Intel RealSense SDK 2.0 and cross-platform support. And it will be connected to Arduino UNO with USB-C* 3.1 Gen 1.

The Position Calculation and Visualization module captures webcam frames, calculates average positions of landmarks, and overlays target locations using OpenCV for visual representation. Evaluation of pose detection involves testing with real human users, acknowledging influences such as clothing color and lighting conditions. The Coordinates Mapping module maps camera space coordinates to robotic arm task space using a linear regression model, allowing accurate prediction of coordinates and performance evaluation. Overall, the Vision Processing Subsystem aims to enhance the MassageMate project with effective and accurate locating capabilities.

Figure 16: Intel RealSense D435i Depth Camera

2.3.6 NLP Subsystem

The NLP subsystem aims to use voice to control the massage head.

Modules

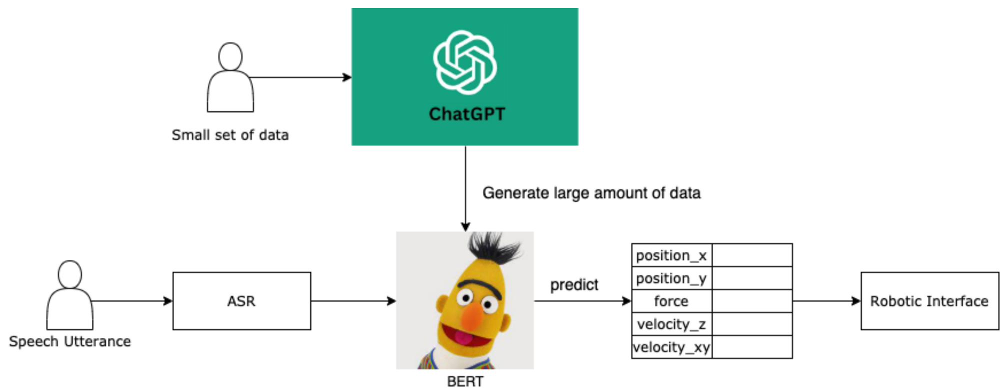

Figure 17: The overall architecture of the components of the NLP subsystem.

ASR (Automatic Speech Recognition)

This block receives signal input from the microphone from a phone or computer and converts the speech to text using the Google speech translation API.

NLU (Natural Language Understanding)

Block Description: The NLU module is responsible for understanding the user's natural language input, such as speech or text. It performs several tasks such as tokenization, named entity recognition, intent classification, and slot filling [3].

Contribution to Overall Design: The NLU module is critical to the overall design of the task-oriented dialogue system as it is the initial step in processing the user's input. It extracts important information such as the user's intent and any relevant entities, which is then used by the subsequent modules to generate an appropriate response.

Interfaces with Other Blocks: The NLU module interfaces with the DST module to provide information about the user's intent and any relevant entities. It also interfaces with the PM module to provide information about the user's input.

Train

`NLU` and `PM` will construct one model. We currently plan to use `ChatGPT` to distill a `Flan-T5` model on the given slots. The training input should be `(user command, current state)` and the output should be `(update of current state)`. The training data will be generated by `ChatGPT` and we will manually finetune the `Flan-T5` model on the synthetic data. Note that we will control the change of `DST` to ensure that the change is acceptable and safe for the person being massaged.

We do NOT use ChatGPT directly for this task because of the RISKY zero-shot performance. We want to make sure the human being massaged is safe.

Inference

An inference example is shown below. The arm controls the blue slots and the dropper controls the green slots.

`position_x`: this parameter gets the horizontal position (left or right on the human back).

`position_y`: this parameter gets the lateral position (upper or lower on the human back).

`force`: this parameter gets the force.

`velocity_xy`: this parameter helps to determine the force.

`velocity_z`: this parameter helps to determine the force.

When any one of `force, velocity_xy, velocity_z` is less than zero, the system will reduce the force (lift the massage head).

Design Alternatives

Except for the BERT implementation, we can use the ChatGPT implementation, that calls the OpenAI API to do question-answering. However, ChatGPT has these drawbacks:

- Unsecured connection. The network in the real world may not always be functioning. We need a local method.

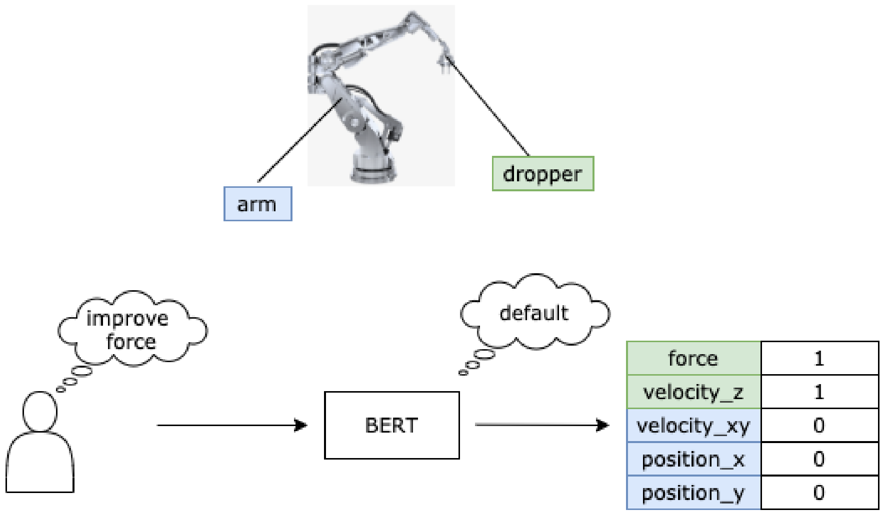

Figure 18: The illustration of the overall architecture of the working pipeline.

- Bad response format. It's hard to control the format of the answer provided by ChatGPT.

2.3.7 Vital Signal Processing system

The vital signal processing system uses a watch wearing on the human's wrist. The watch is composed of one chip (Vital Signs Monitoring Chip) MQ-R-12. The chip is shown below.

The chip layout is shown below.

This chip gathers the heart rate, which we use to judge whether the user is comfortable or not. When the heart rate exceeds the threshold, we shut the system down. The chip also includes data like blood pressure and SpO2, but we don't use them because they do not seem to be effective.

Design Alternatives As an alternative to the chip attached to a finger, we can also use a chip that's fixed to the human's palm or wrist. Here we use the chip attached to the finger to best control the cost.

Figure 19: The Vital Sighs Monitoring Chip MQ-R-12 pictorial graph.

序号	信号名称	功能	说明
1	VCC_IO	数字 IO 供电	2.7V ~ 3.6V
2	RST	复位	低电平有效
3	URX	模块 UART 接收	电平与 VDD_IO 一致
4	UTX	模块 UART 发送	电平与 VDD_IO 一致
5	STA	保留	悬空
6	GND	GND	

Figure 20: The Vital Sighs Monitoring Layout.

3 Verification

3.1 Requirements and Verification

3.1.1 Processing Unit

- Requirements
  1. The Arduino UNO has enough Pins and interfaces to connect sensors. The connection on board should be stable under the motion of robotic arm and massage head.
  2. The communication latency between PC, Arduino and robotic arm should be low enough to make sure acquiring data and sending instructions works in real time. Generally the time for Arduino to get sensor data and the communication delay between Arduino and Sensors should below 50ms.
  3. The data got from Ultrasonic sensor should be accurate enough to support the distance detection.
- Verification
  1. The Arduino, bread board and motor control module is fixed on the base of robotic arm. When the robotic arm is working, those control modules and robotic arm will not influence with each other.
  2. Connect all sensors onto the Arduino board, we test the time for getting data. The Arduino delay for acquiring sensor data is about 10ms, and for each data acquire program, we have 10 loops. The average data acquire time is 133.24ms. So there are 33ms additional delay caused from Python program and Serial Communication between Arduino and PC.
  3. Running the Control program and machine learning models at the same time, and doing some functionality tests to check if the Processing Unit could work fine. The vision model could easily reach at least 50FPS with all subsystem running, while the voice detection model will have delay for 3 to 5 second due to network connection of ASR model.
  4. We did Ultrasonic sensor test by putting it at 5cm, 10cm, 20cm and 40cm to the wall. The error rate for the testing results are shown in the table 1. In general, the error rate for the test cases are below 10%, around 5%, which imply that the sensor is usable for feedback control of robotic arm. It has the error ratio about

Expected value	Real Value	Error Rate
5cm	5.38cm	7.6%
10cm	10.12cm	1.19%
20cm	19.91cm	0.45%
40cm	37.35cm	6.625%

Table 1: Ultrasonic Sensor Test Result

1.19%.

3.1.2 Robotic Control & Robotic Arm Subsystem

- Requirements

  1. The Control subsystem could incorporate all sensors to massage at specific point in task space with appropriate force on human's back. It should make sure that massage head would neither push too hard on human's back nor not reach on human with the feedback loop of pressure sensor.

  2. An immediate stop mechanism is set to make sure that when in the emergency situation that user want to stop immediately, the robotic arm could lift up and stop massage program.

- Verification

  1. We test the massage control program by massage on a soft mannequin. The robotic arm is to move at random position in the soft mannequin fully charged by gas. And we perform 50 test cases on that, counting the number of test case that are 1. appropriately massage. 2. Too hard (The vibration of massage head will be stuck on the fully charged mannequin). 3. Not reached (The massage head doesn't reach the appropriate height). For all 50 test cases, 21 cases are succesfully massaged, 12 cases with force too hard and 17 cases no reached.

  2. The immediate stop button is the same as the "speak" button for NLP subsystem. During massage phase at the soft mannequin, we randomly press the button to check if the robotic arm could respond in time. Among all 20 tests, in 14 cases the robotic arm could respond in time, 6 cases it doesn't respond due to program logic. However, when we quickly press the button repeatedly, the robotic arm could stop immediately.

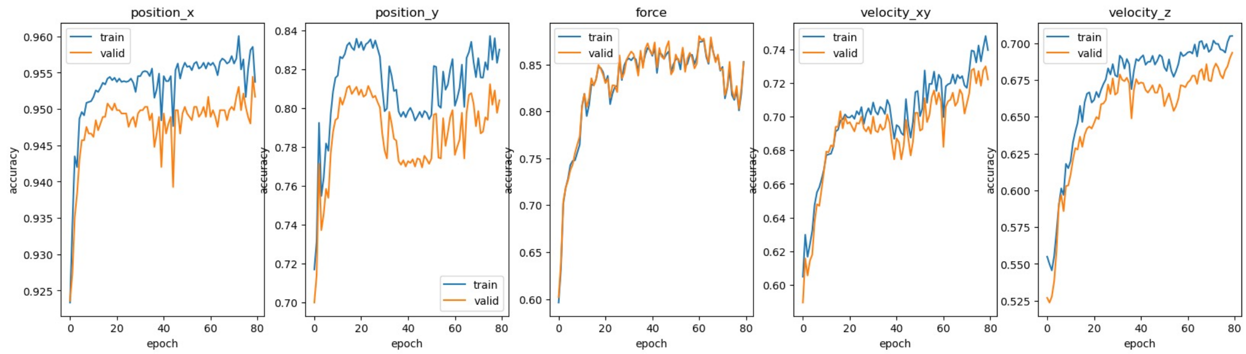

Figure 21: Accuracy results of each slot of the NLP modules.

3.1.3 Voice Controller Subsystem

- Requirements

  1. We aim to achieve a safe and accurate control using voice.
  2. We want to provide useful functionalities on voice control.

- Verification

  1. Results are shown in Figure 21.

3.1.4 Vital Signs Monitoring

- Requirements

  1. The final system should be able to adjust the robotic arm based on the statistics it collect from the vital chip.
  2. When the heart rate reaches 90, we want the system to stop immediately.

- Verification

  1. When the heart rate improves and reaches 90, the machine stops.
  2. When the human runs, the chip successfully detects the increased heart rate and makes the machine stop. We tested on 10 cases and 8 were successful, so the accuracy is 80%.

3.1.5 Massage Head Subsystems

- Requirements

1. Elements must have enough mechanical strength so that when the massage head vibrates, it will not be broken.
2. Motor must have high enough frequency and multiple modes.

- Verification

1. It can work at maximum frequency for a period of time without the structure falling apart.

3.1.6 Visual Processing Subsystem

- Requirements

1. Though the distribution of human features in the human body is highly dependent on the key points of the human skeleton, our model can perform well in learning local and spatial features.
2. A relatively large dataset of human body images is vital, thus we adopt the pre-trained Mediapipe tool open sourced by Google.

- Verification

1. Our code can be executed without bugs so that model can be trained and predictions can be made.
2. The model can perform well so that massaging points can be located and mapped accurately.

3.2 Tolerance Analysis

In order to achieve the functionality of controlling robotic masseur, the system have following data for the analysis:

volume of metal part:
eccentric gear:$7035mm^3$
rotate-rod:$5485mm^3$
rectilinear moving rod:$20468mm^3$
total mass of metal part is about $0.00785\dot{(}7035 + 5485 + 20468) = 259g$
the remaining part is some small sensor.
The head of the arm can mostly lift up to 30 kilograms, and mass of massage head with

响应时间	< 1ms
恢复时间	< 15ms

Figure 22: sensor

two ultrasonic ranging modules is no more than 1 kilogram, so the robotic arm can list massage head easily.

response time of sensor is less than 1 ms
response time of robotic arm's control module is about 16 ms
total response time is less than 20 ms, is short enough to adjust position and force to comfort user.

For the Acupoint detection system, since the camera is setup 1 meter above human, the model need to be able to detect keypoint with the rescale of image. The model accuracy need to reach above at least 60%, and the frame rate should be above at least 45FPS.

The Vital signal module should filter the noise from detection, there should be at least 5 clean features for model to inference. And the average slot filling accuracy could reach at least 90 percent in the evaluation set for those 5 features.

4 Cost and Schedule

4.1 Cost Analysis

Our fixed development costs are estimated to be ¥40/hour, 10 hours/week, in total 12 weeks for four people. The total human cost is ¥19200.

The main parts are concentrated in robotic arm. Our costs are estimated as follow:

Name	Quantity	Unit Price	Total Cost
Arduino UNO	1	¥199	¥199
OpenMANIPULATOR-P Robotic Arm	1	¥96766	¥96766
Intel RealSense D435i 3D Camera	1	¥2829	¥2829
Ultrasonic ranging module	2	¥4.8	¥9.6
RP-C18.3-LT Pressure sensor	1	¥29.8	¥29.8
Pressure sensor data translation module	1	¥8	¥8
R370 motor	1	¥8.5	¥8.5
Metal part in message head	1	about ¥400	about ¥400

As for Arduino UNO, robotic arm and 3D camera, we can use components that are readily available in the school laboratory.

Ultrasonic ranging module, pressure sensor and motor are purchased from companies, and may not reach the requirements for our design. Each part listed may be broken when they are transported or used. So we need flexible budget to choose and purchase new elements.

The total cost is about ¥119649.9.

4.2 Schedule

The schedule is shown as in the table 23

5 Conclusions

5.1 Accomplishment

In conclusion, this work presents MassageMate: a robotic massager that uses visual detection and voice recognition to lead the robotic arm to do massaging. This system utilizes sensors like the pressure sensor, ultrasonic, microphone, camera.

In the end, we achieved three demos:

Week Number	Wentao	Ke	Xiuyuan	Jack
3.19-3.26	Inital set up of robotic arm	Inital set up of robotic arm	study how to transmit singles to control systerm	Survey slot filling and dialog system.
3.27-4.2	Write Python API for the control of robotic arm	Study acupoint detection method	Make metal part of massage and connect it to robotic arm	Survey vital signs monitoring subsystem.
4.3-4.9	Connect and test all sensors	Connect Camera and try different detection methods	Make metal part of massage and connect it to robotic arm. Able to control its motor	Implement slot filling and dialog system.
4.10-4.16	Finish the feedback control for sensors on massage head	Evaluate different methods and choose the best one to optimize	Study relationship between strength and position of massage and user's confort level	Implement vital signs monitoring subsystem.
4.17-4.23	Integrate robotic control with acupoint detection, doing camera calibration	code user-interface for control through phone App or handle	Improve pressure control	Integrate the slot filling and dialog system.
4.24-4.30	Integrate the Vital signs subsystem with robotic control	Do real-human safety and functionality test	Test mechanical reliability test	Integrate vital signs monitoring subsystem.
5.1-5.7	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo
5.8-5.22	Improve the functionality and write final report	Improve the functionality and write final report	Improve the functionality and write final report	Improve the functionality and write final report

Figure 23: Schedule

Visual detection demo We build our visual detection model based on human pose detection models to detect and calculate the massage location on human's back including shoulder, side waist and near spine acupoints. The control system could convert the massage point from image space to task space for massage automatically.

Voice recognition demo The voice recognition demo shows that our system can detect and recognize the intention of the user on how he wants to change the force, velocity and position of the massaeur.

Vital signs detection demo The vital signs detection demo shows the capability of our system to process vital signs data and use a threshold on the heartrate so that the system should stop working.

5.2 Weakness & Uncertainty

However, our system still has some parts need to be improved. Firstly, due to the accuracy of ultrasonic sensor and pressure sensor, the instant distance and force will have error within around 10%. The inappropriate force exert on pressure sensor may not be detected during massage. So to guarantee safety, we set a very conservative massage force threshold on that, which may lead to massage head not reach to human. Secondly, the voice detection subsystem has quiet large delay, so that the implementation of voice instruction may not respond quickly. Thirdly, the detection of human massage pose is rely on the cloth that user wear. For person wearing too dark clothes may not be detected properly by the vision model.

5.3 Future Work

In the future, we want the massaeur to be smarter. For example, the vital signs monitoring system should be a machine learning model that automatically detects whether the user is comfortable or not. We also want the NLP system to be able to handle commands that are more complex. Most essentially, the camera can detects and adjust the human pose on the fly.

5.4 Ethics

Before the user actually uses MassageMate, he needs to sign a waiver form (disclaimer) that all danger caused by MassageMate should be obligated to himself. Our teams's massage robot is not a substitute for medical advice or treatment. The robot has risks including but not limited to bruising, soreness, muscle strain, and injury.

We also want to refer to the IEEE code [4] and ACM code [5] to show that our work agrees to these codes and welcomes them.

References

[1] T. W. Huanbin Gao Shouyin Lu, “Research and development of chinese medical massage robot,” *Robot*, vol. 05, pp. 43–52, 2011.

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

[3] Q. Chen, Z. Zhuo, and W. Wang, “Bert for joint intent classification and slot filling,” *arXiv preprint arXiv:1902.10909*, 2019.

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

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