Robotic Operation of ICU Equipment in Contagious Environments

Computer Integrated Surgery II Project Report

2021 Spring

Tianyu Wang
Department of Biomedical Engineering
Email: twang98@jh.edu

Jiayin Qu
Department of Applied Math and Statistics
Email: jqu5@jh.edu

Mentor: Dr. Axel Krieger, Dr. Balazs Vagvolgyi
Instructor: Dr. Russell H. Taylor

Table of Content

Introduction

1. Abstract

Ventilators and infusion pumps are critical equipment in treating COVID-19. Routine setting changes require staff to enter ICU which requires consuming a full set of PPE and exposes staff to risk of infection by COVID-19. Unfortunately, due to security concerns, equipment can not access networks[1]. In this report, we discuss parts of a novel teleoperated robotic system that aims to recognize key ICU equipment, operate them, and project key information from such equipment straight back to the operator to reduce material cost and risk of exposure for ICU staff during routine checkup. More specifically, we report on our progress and result of training an object recognition model to recognize key ICU equipment and implementing an novel end-effector design with corresponding UI to control the robot.

2. Clinical Motivation

The COVID-19 pandemic has caused a tremendous global surge of ICU devices and staff demand. Since the virus is highly contagious, and directly impacts the respiratory system, it is believed that about ⅓ of the patients who have COVID-19 will require ICU admission at some point[2]. This poses great challenges towards the medical system as hospitals are overloading to take on COVID patients in ICU. As of February, the average national ICU occupation is still at 73 %[3], this is an almost 10 % raise compared to 2010 [4]. Most Significantly, given COVID's mode of transmittance, medical workers are at risk every time they enter an ICU room to do routine adjustments and check-ups for patients. Furthermore, entering the ICU requires consuming a full set of PPE. Not to mention the amount of time it would take to properly dress in PPE. Therefore, ICU teams need a novel way to remotely monitor and adjust settings on key ICU equipment in order to efficiently monitor and provide consistent healthcare service to all patients as we cope with ICU need surge due to COVID-19.

Figure 1. Time estimate of putting on and doping off a PPE suit[1]

3. Problem Statement

The problem identified above indicates that ICU teams need a novel way to remotely monitor and adjust settings on key ICU equipment in order to efficiently monitor and provide consistent healthcare service to all patients as we cope with ICU need surge due to COVID-19. To achieve this, we break down the problem into four functional segments to mimic the workflow of an ICU staff entering into the room: Identify Device, Move to Device, Operate Device, Information Feedback.

4. Goal

To achieve the tasks mentioned above, we envision a 6 DOF robotic arm mounted on a mobile platform that is capable of recognizing the target object, finding its relative position and aligning itself with the control panel to perform various interactions based on pre-programmed configuration with its end-effector using remote command.

However, due to time and physical limitations, we simplify the problem from 6 DOF to 3 DOF by assuming that the robot has already aligned itself in an ideal position parallel to the device interface. Furthermore, we model the buttons, screen and knobs using an oscilloscope shown below, as it contains multiple interaction modality and is a good representation of medical devices in ICU.

Therefore, our goal is to recognize the device, load its pre-configured interface layout, and make a 2D cartesian robot capable of manipulating the oscilloscope using its end-effector. This robot will be controlled with a working user interface that allows the user to remotely control the robot.

5. Prior Work

Prior work by Vagvolgyi et al[1]. proposed an teleoperated robotics system utilizing a 2D cartesian robot mounted on a ventilator. Using a proxy computer and a camera mounted on a fixed location, they were able to achieve remote control of the ventilator using a stylus on a cam and information feedback using the camera. However, their solution is limited to one device per robot and only concerns interactive modality such as button and touchscreen, thus has limited generalization ability.

Figure 2. The 2D Cartesian robot designed for controlling touchscreen [1]

There are also many prior designs of end effector that take on different forms and purposes when it comes to fitting in the last piece of puzzle between driving an robotic arm and completing physical operations. Here we focus on impactive end effectors such as grippers. Researchers have conducted prior research comparing the different ways of actuating a gripper as shown below. Furthermore, companies like On-Robot and Zimmer Group have also designed multiple industrial level tried-and-tested layouts of gribber that we can adapt when designing our own. Thus, based on these prior work, we can hope to integrate closely our design needs and the derived experience from academic and industry to arrive at the optimal design for our own purpose.

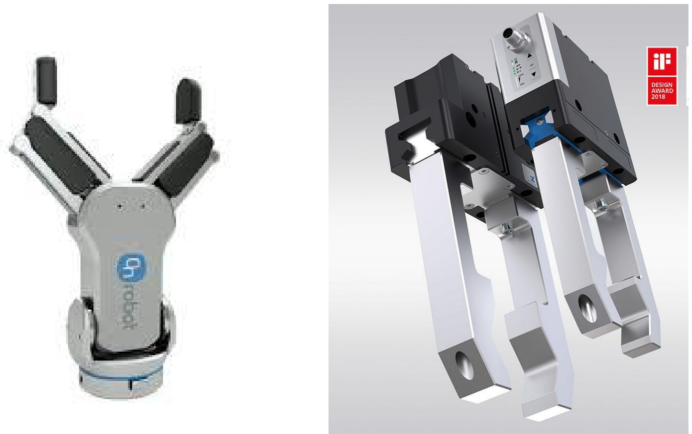


Figure 3. On Robot Sample[5], Zimmer Group Sample [6]

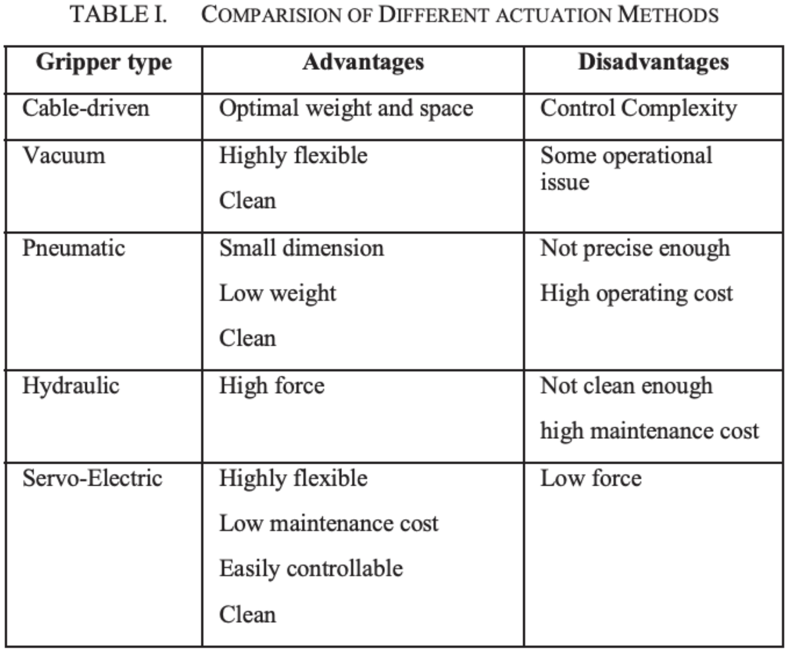


TABLE I. COMPARISION OF DIFFERENT ACTUATION METHODS

Gripper type	Advantages	Disadvantages
Cable-driven	Optimal weight and space	Control Complexity
Vacuum	Highly flexible Clean	Some operational issue
Pneumatic	Small dimension Low weight Clean	Not precise enough High operating cost
Hydraulic	High force	Not clean enough high maintenance cost
Servo-Electric	Highly flexible Low maintenance cost Easily controllable Clean	Low force

Table 1: Comparison of Different Actuation Methods[7]

6. Proposed Solution

Based on our needs and problem statement, we propose the following workflow of solution.

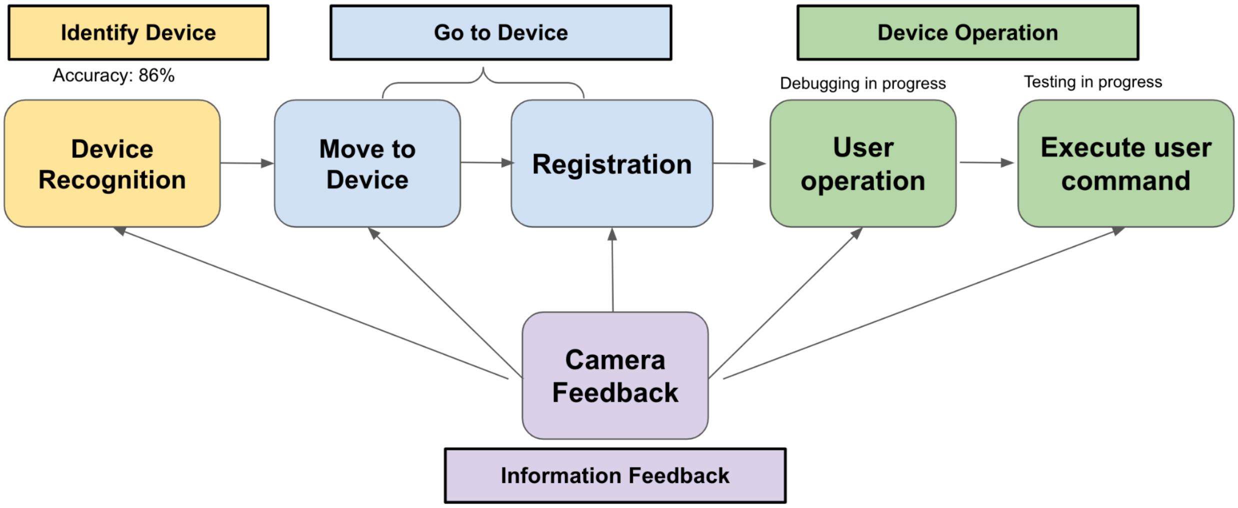


Figure 4. Proposed Solution Workflow with Corresponding Functional Segments

As mentioned above, we will be only focusing on identifying the device and device operation portion of the solution. We modeled target devices with oscilloscopes. Simplified the problem from 6 to 3 DOF by assuming the robot has already aligned with the device at an ideal position.

For identifying the device, we trained an object recognition model to recognize the device from distance, and defined json structure to load corresponding interface configuration for device based on user input. Currently it has an classification accuracy of 86 %.

For device operation, we built a user interface based on python-ROS system that allows the user to remotely control the robot; In addition, we designed an novel end-effector to interact effectively and accurately with the target device and built a 2D cartesian for testing end-effector.

Lastly, we incorporated live camera feedback to project important information to user as well as monitor the end-effector interaction with devices. However, this portion is still under construction and not optimized for output yet.

Detailed progress and result can be seen below in the corresponding technical section.

Technical Summary

A. Identify Device

Object Recognition

The recognition of the oscilloscope adapts the YOLO (you only look once) algorithm. The benefits of YOLO is that it can achieve relatively fast recognition speed and the learning accuracy of the general representations is also high compared with other object detection methods including DPM and R-CNN[8].

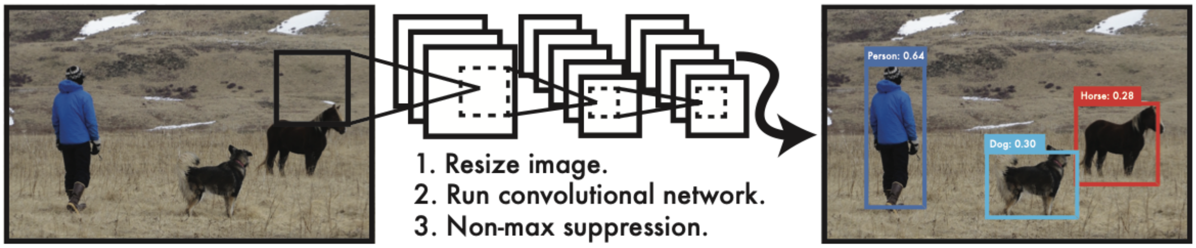


Figure 5. General workflow of YOLO[8]

We use the LabelImg tool kit to manually annotate the position of the oscilloscope in the image. The whole training data set includes 200 images of the oscilloscope from various viewpoints and distance, whereas the testing data set has 100 images.

Figure 6. Labelling the image in the training data set

The training process is fulfilled by taking advantage of the two core libraries: OpenCV and ImageAI. The model is trained with a batch size of 10 and takes 23 epochs of training time. The eventual model we choose achieves 86% accuracy on the test data set.

B. Move to Device

Feature Recognition

The feature recognition is achieved through Harris's edge and corner detector[9]. The detector reads the image and calculates the gradient to find the edge and corner of the object. This would help us vaguely find the position and the orientation of the object.

Feature Matching

After the robot moves to a position that faces the control panel of the designated device, feature matching is required for finer registration. The 2D image from camera input will be compared with the standard image in which the pose is desired. A homography matrix would be calculated that contains the information of rotation and linear distance[10].

C. Operate Device

User Interface Design

The graphic user interface (GUI) is constructed based on QtCreator and QtDesigner that have ROS workspace. ROS is a versatile environment which also allows for the communication among various different terminals, making it a good platform for remote communication between control devices. QtCreator allows us to build a GUI that uses ROS command so that the users can interact with the robot through the GUI-ROS-Arduino system.

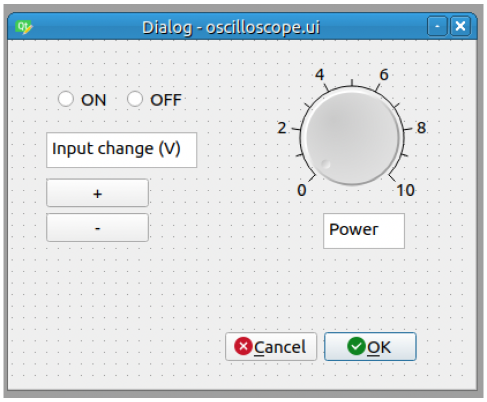
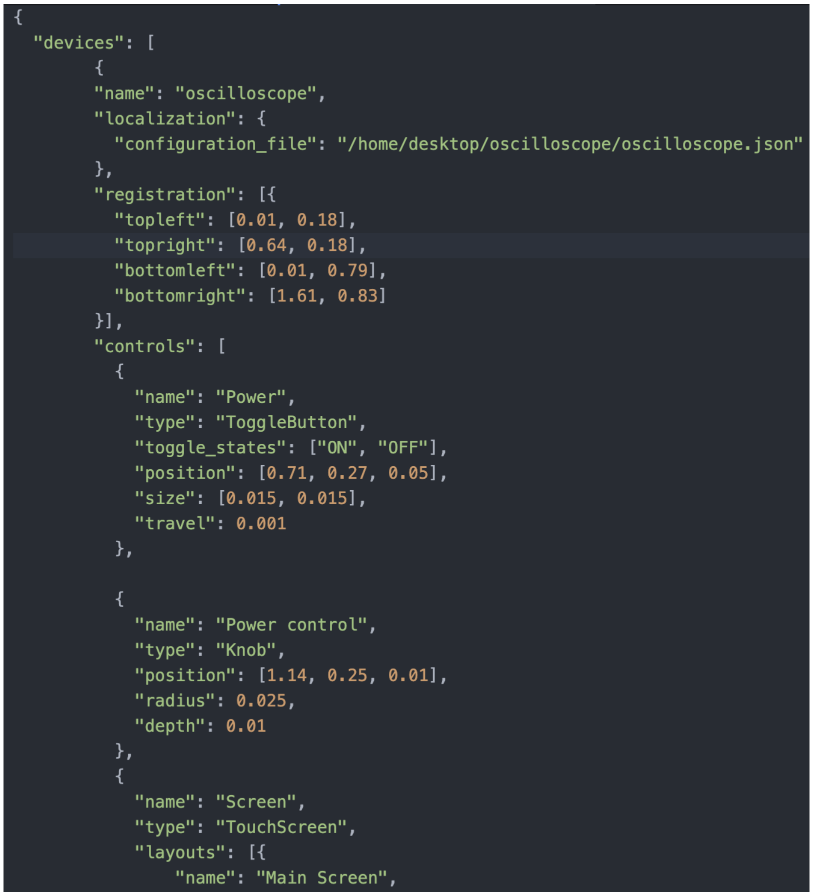


Figure 7. Sample user interface (camera view separated from the GUI)

Configuration File

The configuration file of the device in the format of json is required. In the configuration file, the position, type, shape as well as the way the users can interact with buttons are recorded. While the user is interacting with the GUI, the robot will follow the predefined instructions to interact with the device.



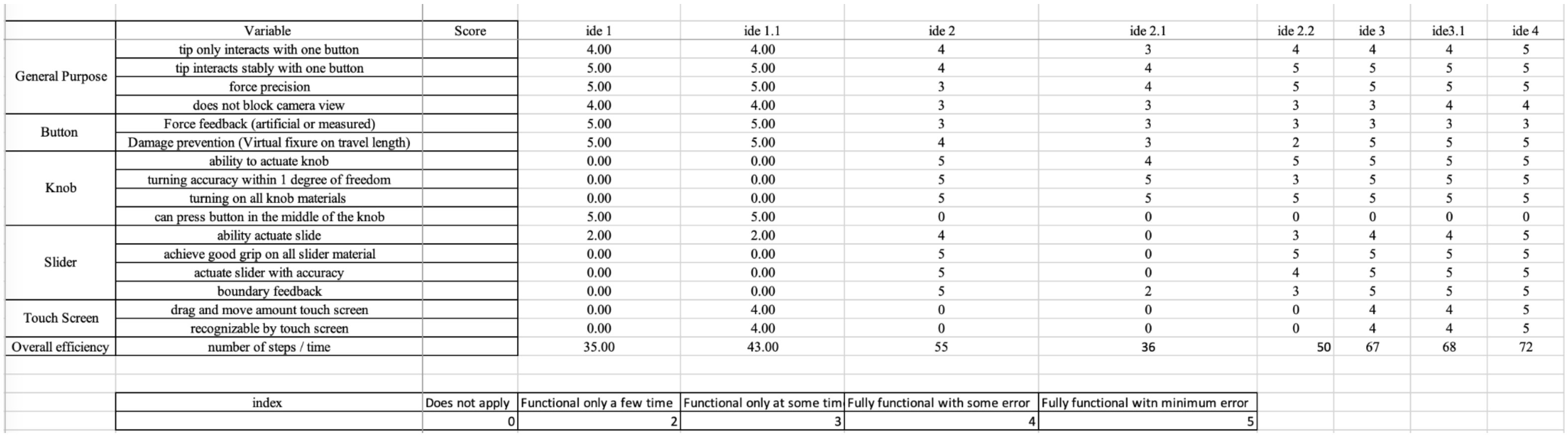
```
{
  "devices": [
        {
        "name": "oscilloscope",
        "localization": {
          "configuration_file": "/home/desktop/oscilloscope/oscilloscope.json"
        },
        "registration": [{
          "topleft": [0.01, 0.18],
          "topright": [0.64, 0.18],
          "bottomleft": [0.01, 0.79],
          "bottomright": [1.61, 0.83]
        }],
        "controls": [
          {
            "name": "Power",
            "type": "ToggleButton",
            "toggle_states": ["ON", "OFF"],
            "position": [0.71, 0.27, 0.05],
            "size": [0.015, 0.015],
            "travel": 0.001
          },

          {
            "name": "Power control",
            "type": "Knob",
            "position": [1.14, 0.25, 0.01],
            "radius": 0.025,
            "depth": 0.01
          },
          {
            "name": "Screen",
            "type": "TouchScreen",
            "layouts": [{
                "name": "Main Screen",
```

Figure 8. The sample configuration file of the oscilloscope in json format

End Effector Design

Our design of the end-effector is generated from a combination of need criteria and design evaluation we did in the beginning of the project, detailed criterias are shown below:



	Variable	Score	ide 1	ide 1.1	ide 2	ide 2.1	ide 2.2	ide 3	ide3.1	ide 4
General Purpose	tip only interacts with one button		4.00	4.00	4	3	4	4	4	5
	tip interacts stably with one button		5.00	5.00	4	4	5	5	5	5
	force precision		5.00	5.00	3	4	5	5	5	5
	does not block camera view		4.00	4.00	3	3	3	3	4	4
Button	Force feedback (artificial or measured)		5.00	5.00	3	3	3	3	3	3
	Damage prevention (Virtual fixure on travel length)		5.00	5.00	4	3	2	5	5	5
Knob	ability to actuate knob		0.00	0.00	5	4	5	5	5	5
	turning accuracy within 1 degree of freedom		0.00	0.00	5	5	3	5	5	5
	turning on all knob materials		0.00	0.00	5	5	5	5	5	5
	can press button in the middle of the knob		5.00	5.00	0	0	0	0	0	0
Slider	ability actuate slide		2.00	2.00	4	0	3	4	4	5
	achieve good grip on all slider material		0.00	0.00	5	0	5	5	5	5
	actuate slider with accuracy		0.00	0.00	5	0	4	5	5	5
	boundary feedback		0.00	0.00	5	2	3	5	5	5
Touch Screen	drag and move amount touch screen		0.00	4.00	0	0	0	4	4	5
	recognizable by touch screen		0.00	4.00	0	0	0	4	4	5
Overall efficiency	number of steps / time		35.00	43.00	55	36	50	67	68	72

index	Does not apply	Functional only a few time	Functional only at some tim	Fully functional with some error	Fully functional witn minimum error
	0	2	3	4	5

Figure 9. Design Criteria and Evaluation for different ideation

The implemented design is based on ideation 2.2 where it features 2 DOF two-finger claw design. It contains a squeezing joint and a rotating wrist joint for gripping and turning knobs respectively. When separated, the design can utilize a single finger to press and interact with buttons and touchscreen as well. At each tip of the finger, is a piece of rubber that can be recognized by touchscreen and to increase friction for interaction with buttons and knobs.

Note that although we have a final ideation design that excels in more area and results in a higher evaluation score, after discussion with our project mentor, we decided to go with ideation 2.2 instead of our final design due to design 2.2's wide availability on the market, making it easier to purchase and continue with prototype rather than printing from CAD design which may encounter more in further issue of compatibility and print accuracy. Thus we adapted off-the shelf products based on our own design criteria instead of 3D printing CAD for prototyping purposes. Rubber end was attached after.

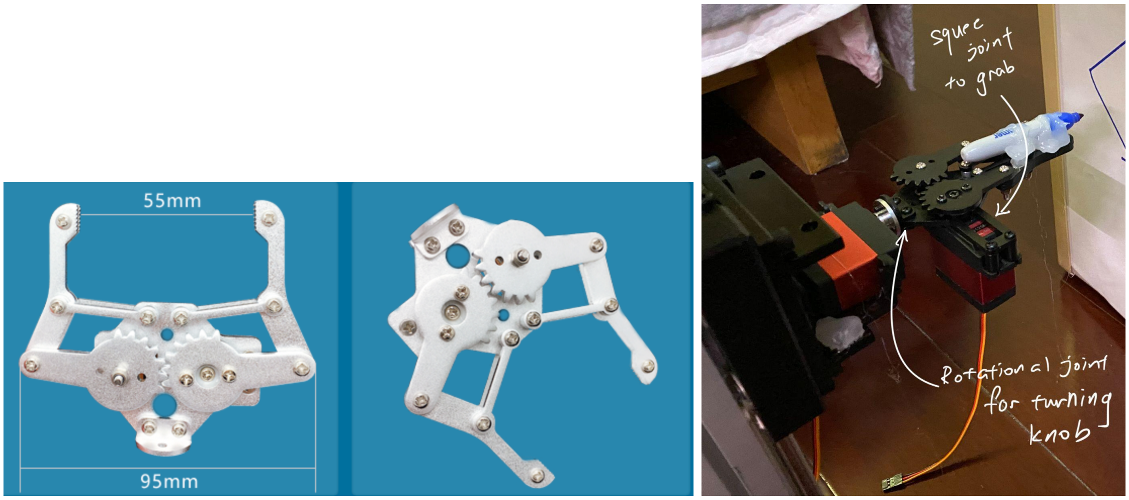
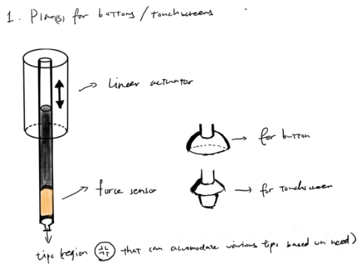


Figure 10. Illustration of final implemented end-effector design and assembled end-effector with rotating and squeezing joint.

Ideation 1.1 Single finger prong with replacable tip



Abstract:

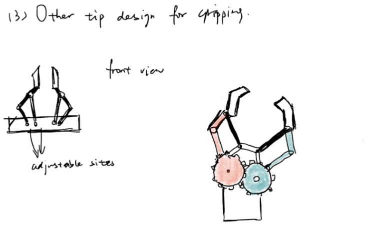
Design is similar to that of ideation 1 except now our tip are replacable with different shape and material to accomate touch screen operation possibility.

Design Feature:

Replacable Tip for specific button pressing and specific touchscreen operation.

Score: 43/80

Ideation 2.2 Forceful Gripping two finger



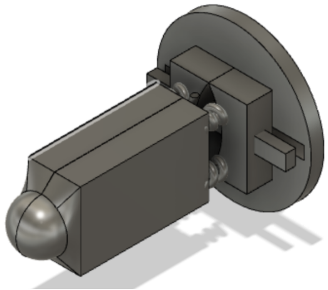
Abstract：
This design is similar to that of ideation 2, but we switched the two slab to more mechanical complex geometry design to amplify the force exerted by servo on knob to ensure better grip.

Design feature:

1. Geometric finger design for better force exertion
2. Reduced number of servo, two finger driven by one servo using gears.

Score: 50/80

Final End Effector Design



Abstract: Stylus integrated 2 Finger design for end effector

Feature:
1. integrated stylus at the tip
2. integrated spring to absorb impact
3. Potentially replace Spring with actuator to achieve faster movement

Score: 72/80

Figure 11. Illustration of different design history

Test Bed Construction

Here we utilizes a 2D cartesian robot to simplify prototyping, assuming 6 DOF robotic arm moves only in 2D plane parallel to device when close up interaction. Based on our testing needs, we envisioned an assembly shown by CAD in figure 8. The 2D cartesian robot is mounted on a Z-direction slide for location adjustment. It utilizes 3 stepper motors for localization and two servos for end-effector manipulation. All the motors are powered by an independent power supply and driven through arduino. All the servos are driven by arduino as well.

Similar to the end-effector, after discussion with our mentor, we then purchased off the shelf linear belt drive and assembled them to avoid compatibility issues and reinventing the wheel.

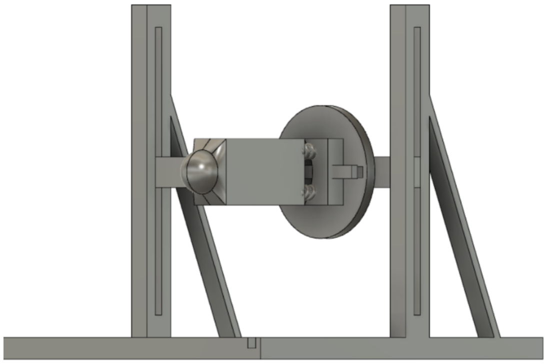


Figure 12. Illustration of envisioned test bed assembly with end-effector mounted on a 2D cartesian robot.

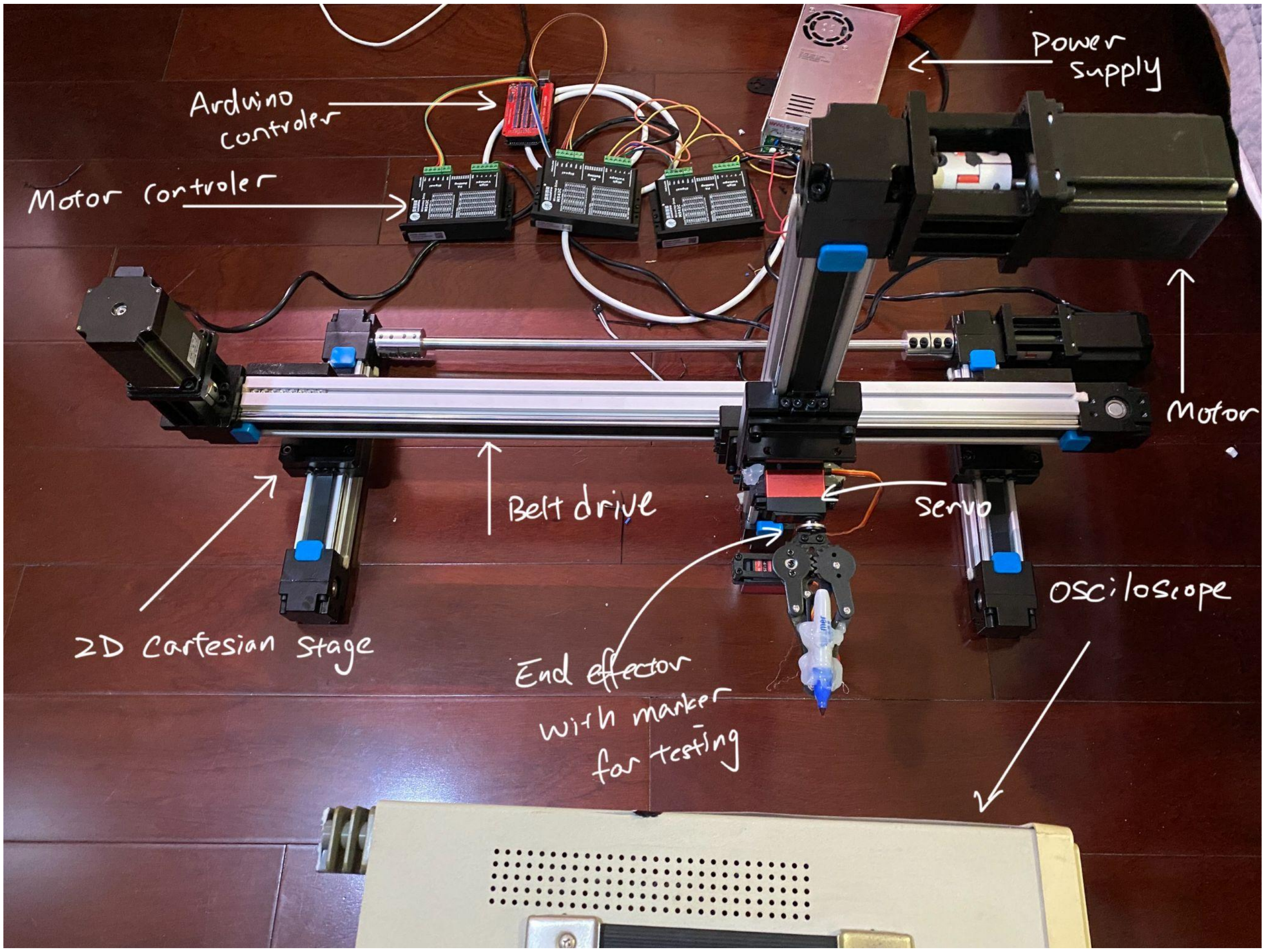


Figure 13. Illustration of fully assembled robot with label during testing.

Testing and result

Testing of the end-effector is conducted using the above testing set-up, where a paper is attached to the oscilloscope and utilized a marker to represent the trajectory of the end-effector movement in space. As shown in the close-up image in figure 9. We asked the robot to step horizontally for some distance, comeback to center, then step vertically for some distance and return to center. Then starting from the center we ask it to draw a square rotated 45 degree. The figure is shown below in figure 10 with the robot drawing in blue and expected trajectory in red dash lines.

From the result, we observed a relatively good path following the ability of our robot end-effector. However, it is obvious that the robot is struggling at corners or sharp turn arounds as shown above. Future optimization should be able to solve this issue.

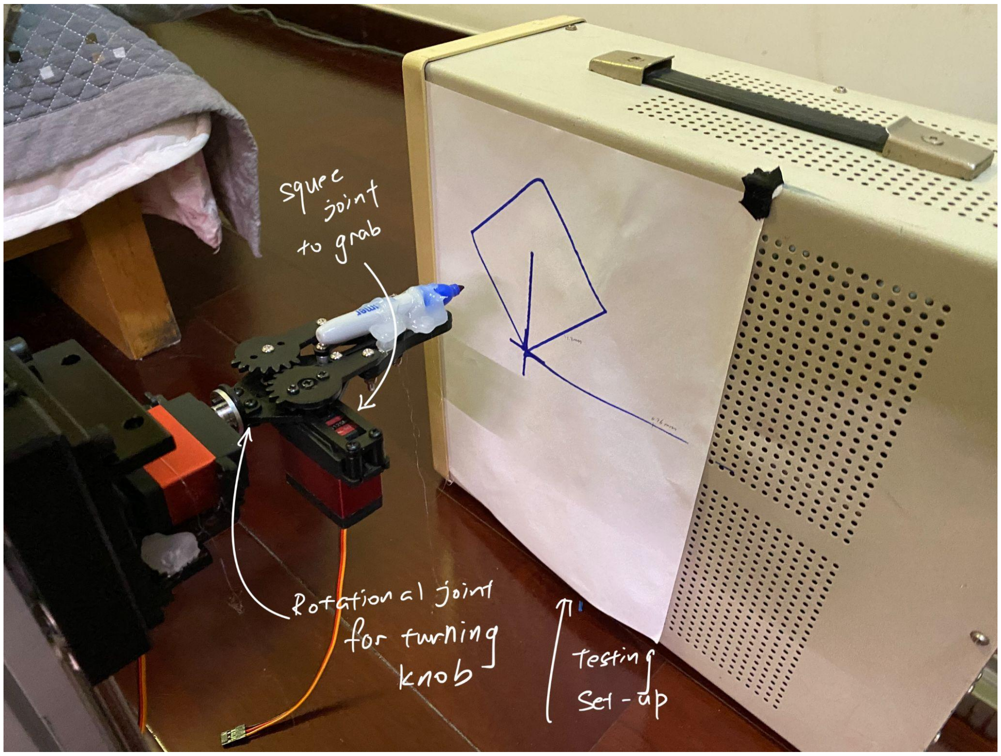


Figure 14. Close up image of testing set-up in process.

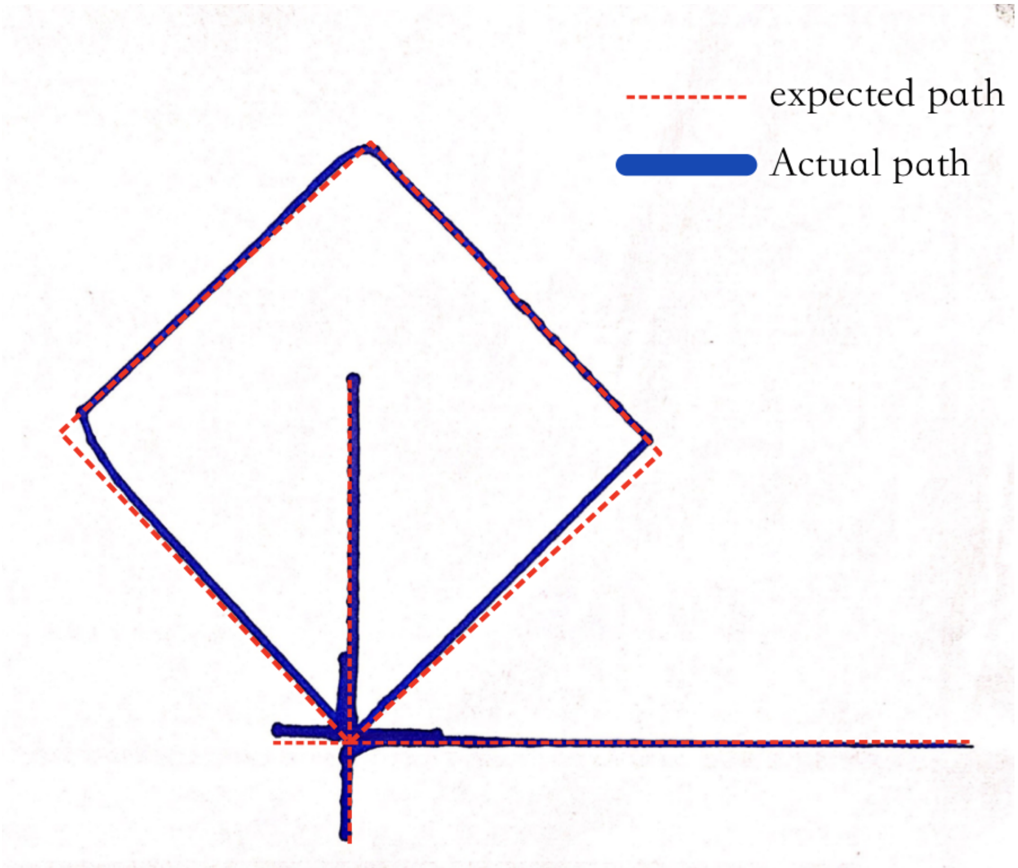


Figure 15. Robot end-effector trajectory (blue solid line) overlaid with expected trajectory (red-dash line).

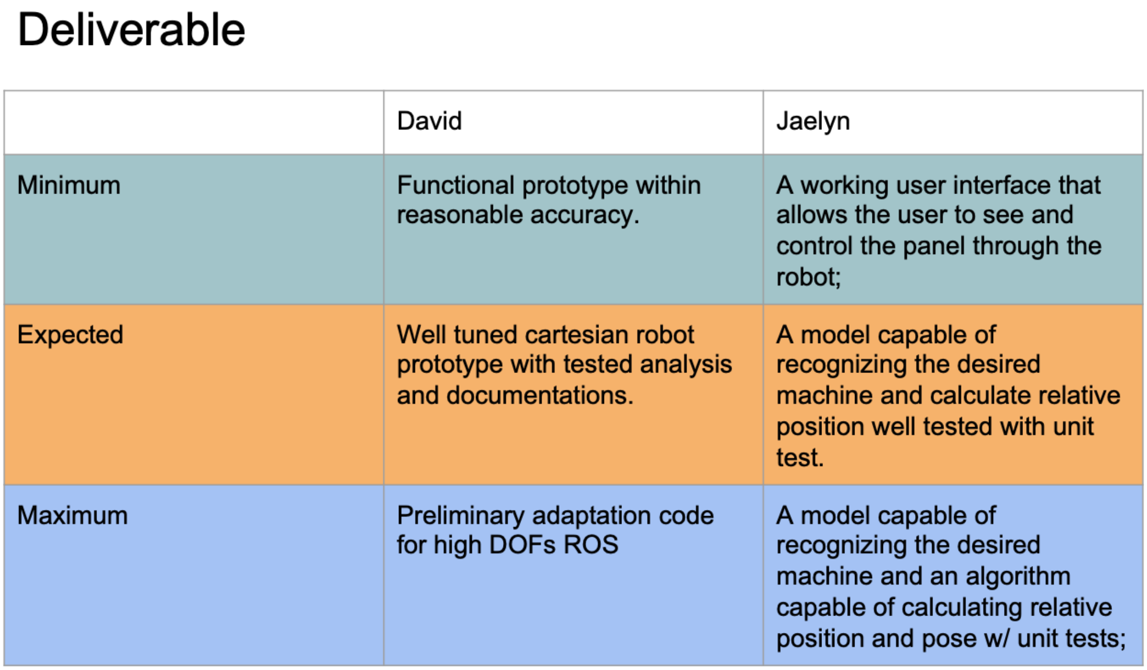
Management Summary

1. Who did what?

Tianyu Wang: end effector design, building the 2D cartesian robot, direct control of the robot via Arduino, accuracy test of the robot,

Jiayin Qu: object recognition algorithm design, feature recognition algorithm design, configuration file in json format design, ROS UI framework design

2. Deliverables and Status



Deliverable

	David	Jaelyn
Minimum	Functional prototype within reasonable accuracy.	A working user interface that allows the user to see and control the panel through the robot;
Expected	Well tuned cartesian robot prototype with tested analysis and documentations.	A model capable of recognizing the desired machine and calculate relative position well tested with unit test.
Maximum	Preliminary adaptation code for high DOFs ROS	A model capable of recognizing the desired machine and an algorithm capable of calculating relative position and pose w/ unit tests;

Minimum:
    David: Completed
    Jaelyn: Completed
Expected:
    David: Still ironing out some detail regarding end-effector accuracy and updating some documentation. 70% Completed
    Jaelyn: Still exploring options for distant pose recognition;
Maximum:
    David: Future work
    Jaelyn: Same as expected

3. Future Work

Due to various limitations, our current work is completely developed based on a 2D Cartesian robot. Our eventual goal, as stated in previous parts, is to develop a mobile robot with 6DOF that can move in ICU and operate with various medical devices as needed. Therefore, the work that would be needed to achieve the goal would include but not limited to the following:

Pose recognition

In a 2D cartesian robot, the robot would be fixed to a position that best fits its operation of the control panel on the oscilloscope. In the case of a 6DOF robot, it might not be directly facing the control panel. The algorithm that recognizes the relative pose between the robot and the device would be required.

Movement path design

After receiving the information regarding the relative pose and distance between the robot and the device, the robot is expected to move to the designated position with the help of movement path design. The path would be automatically generated based on live camera input and the path itself must be short and clear of occlusion. We are also considering letting the user take control so that the movement of the robot can be controlled through a remote controller.

Calibration

Due to the complexity of the movement of a 6 DOF robot, calibration would be of a more important issue so ensure the precision of the interaction between the robot and the device.

End effector design

In the current design of the 2D Cartesian robot, the modalities its end effector can interact with are rather limited. Certain functions are not incorporated in this design, especially the intrusion finger that is supposed to be interacting with the touchscreen on some types of ventilators.

4. Lessons Learned

During the whole designing, much is learned in the process. For the mechanical design part, we learned things including operating motors/servos through Arduino, assembling electrical circuits, designing of design evaluation charts, etc. For the computer vision and software designing part, we learned things including doing object recognition, generating configuration files in json format, constructing virtual machines, building a user interface via QtCreator/QtDesigner in a ROS workspace.

5. Acknowledgements

We would like to thank Dr. Krieger and Dr. Vagvolgyi for their patience and support. Dr. Krieger provides us with sufficient guidance in building the cartesian robot and Dr. Vagvolgyi offers us great help in exploring computer vision and UI construction related topics. We would also want to thank Dr. Taylor for helping improve our presentation skill.

References

[1] B. P. Vagvolgyi *et al.*, "Telerobotic Operation of Intensive Care Unit Ventilators," *ArXiv201005247 Cs*, Oct. 2020, Accessed: Apr. 12, 2021. [Online]. Available: http://arxiv.org/abs/2010.05247.

[2] S. M. Abate, S. A. Ali, B. Mantfardo, and B. Basu, "Rate of Intensive Care Unit admission and outcomes among patients with coronavirus: A systematic review and Meta-analysis," *PLOS ONE*, vol. 15, no. 7, p. e0235653, Jul. 2020, doi: 10.1371/journal.pone.0235653.

[3] M. Conlen, J. Keefe, L. Leatherby, and C. Smart, "How Full Are Hospital I.C.U.s Near You?," *The New York Times*, Dec. 16, 2020.

[4] H. Na and P. Sm, "Critical Care Medicine Beds, Use, Occupancy, and Costs in the United States: A Methodological Review.," *Crit. Care Med.*, vol. 43, no. 11, pp. 2452–2459, Nov. 2015, doi: 10.1097/ccm.0000000000001227.

[5] "Robot arm grippers and EOAT components | OnRobot." https://onrobot.com/en/products (accessed Feb. 25, 2021).

[6] "Home - Zimmer Group." https://www.zimmer-group.com/en/ (accessed Feb. 25, 2021).

[7] N. Kumar *et al.*, "Design, development and experimental assessment of a robotic end-effector for non-standard concrete applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1707–1713, doi: 10.1109/ICRA.2017.7989201.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *ArXiv150602640 Cs*, May 2016, Accessed: May 06, 2021. [Online]. Available: http://arxiv.org/abs/1506.02640.

[9] "OpenCV: Harris Corner Detection." https://docs.opencv.org/3.4/dc/d0d/tutorial_py_features_harris.html (accessed May 06, 2021).

[10] "Image alignment and registration with OpenCV - PyImageSearch." https://www.pyimagesearch.com/2020/08/31/image-alignment-and-registration-with-opencv/ (accessed May 06, 2021).