ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

An Intelligent Assistant Using Sign Language

Team #27

QIANZHONG CHEN (qc19@illinois.edu)
HANWEN LIU (hanwenl4@illinois.edu)
HAINA LOU (hainal2@illinois.edu)
YIKE ZHOU (yikez3@illinois.edu)

TA: Xiaoyue Li

March 24, 2023

Contents

1 Introduction

1.1 Problem

An Intelligent Assistant (IA) is software that can provide services and interact with the user, typically by performing automated tasks and assisting with daily activities. With the advent of computer vision and natural language processing technologies and the emergence of smart home accessories, intelligent assistants have revolutionized how people interact with technology.

Most intelligent assistants use Voice User Interface (VUI) as a primary means of communication. Some of the most prevalent examples include Siri from Apple, Cortana from Microsoft, and Alexa from Amazon. VUI provides several advantages, such as hands-free operation, faster input, and greater convenience. However, VUI is not always suitable for people with hearing or speech problems, hindering their ability to use these intelligent assistants.

People with hearing or speech impairments often face significant challenges in accessing information, participating in social interactions, and performing daily activities. Therefore, developing technologies that meet their unique needs and facilitate communication and engagement can significantly improve their quality of life.

1.2 Solution

We propose to develop an intelligent assistant that uses sign language as its primary communication standard. Sign language will enable people with hearing or speech impairments to interact with intelligent assistants effectively. By leveraging the latest advancements in computer vision and natural language processing, our intelligent assistant will recognize sign language and respond in real-time, making it a powerful and accessible tool for a broader range of users.

In recent years, intelligent assistants are becoming more personalized. Some assistants use data analysis to learn about individual users and their preferences and provide more accurate, relevant, and reasonable services and recommendations. Our intelligent assistant using sign language consists of four subsystems: Input and output subsystem, Gesture recognition subsystem, System and control subsystem, and Bionic hand subsystem.

The input and output subsystem includes a camera that receives input from the user through sign language and a displayer that reveals the interaction between the user and the intelligent assistant to those who do not understand sign language.

The gesture recognition subsystem receives the visual signal from the input and output subsystems. Gesture recognition subsystem applies *You Only Look Once* (YOLO)[1], an object detection algorithm, to detect the 3-dimensional position of the user's hand. Then it utilizes *Mediapipe*[2], developed by Google, to collect the relevant position for the wrist in real-time.

The system and control subsystem receives the decision of the gesture, which the bionic hand needs to do from the gesture recognition subsystem. It translates it to Pulse-Width Modulation (PWM) signals to control the movement of servo motors. Our control system uses Microcontroller Unit (MCU) to output signals and an advanced computing unit to deploy the machine learning model.

The bionic hand subsystem is the most complex part under the input and output subsystem. It is responsible for communication with the user.

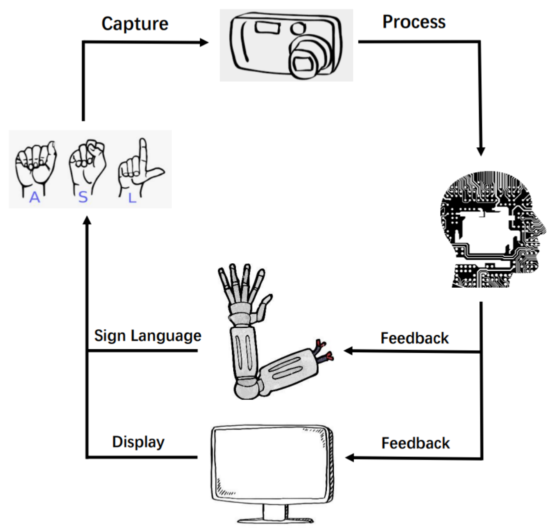

Figure 1: Visual Aid

1.3 High-level Requirements

Build an end-to-end model using the Mediapipe framework combined with different machine learning models, including Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and Gate Recurrent Unit (GRU).

To implement a good interaction experience, the time used by the user doing sign language to display the dialogue and the response of the bionic hand should be at most 30 seconds.

The bionic hand can move free and fluently as designed, all of the 12 degrees of freedom fulfilled; the movement of a single joint of the finger does not interrupt or be interrupted by other movements; the bionic hand could work for one hour in a roll and two years in total.

2 Design

2.1 Block Diagram

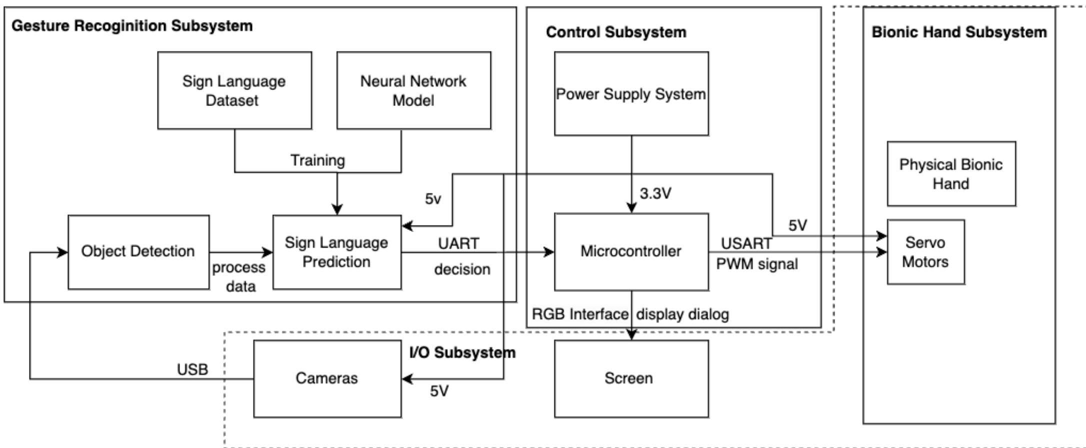

Figure 2: High Level System Overview

2.2 Physical Design

In this part, we demonstrate the system overview in Fig. 3 and Fig. 4. Our system includes two bionic hands, one camera for gesture recognition, one STM32 micro-controller and development board, and one Liquid Crystal Display (LCD) screen to visualize the information.

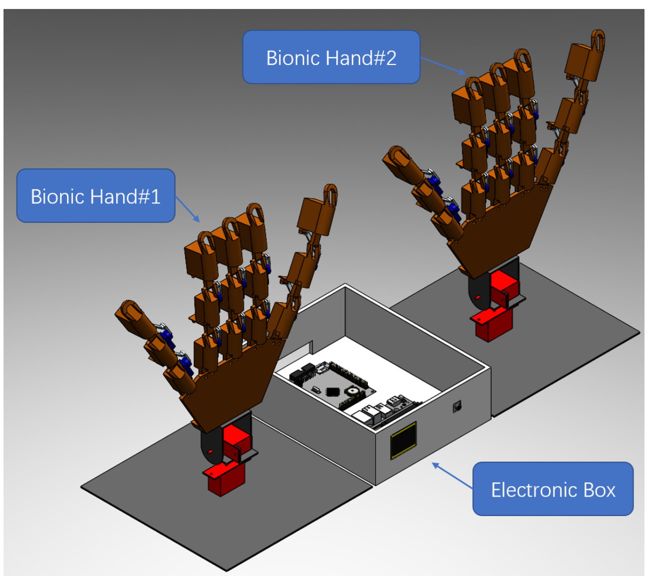

Figure 3: Overview of the System Physical Design

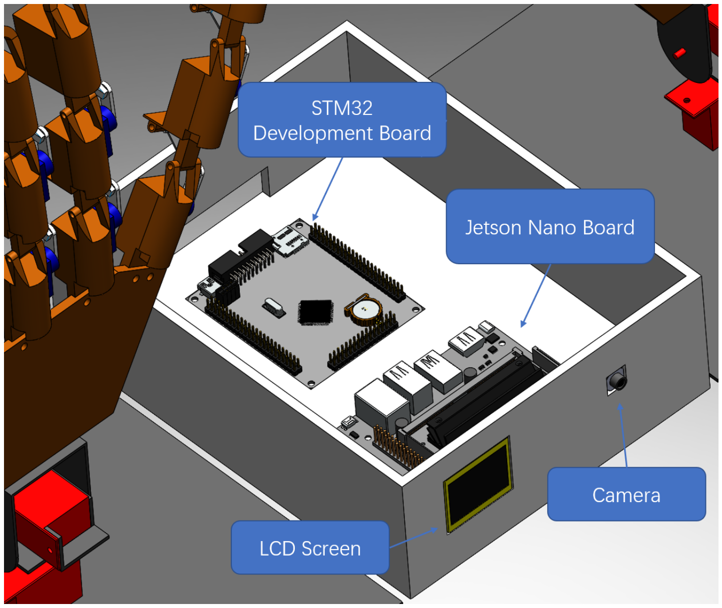

Figure 4: Detailed Layout of Electronic Box

2.3 Bionic Hand Subsystem

2.3.1 Description

The bionic hand subsystem consists of two identical bionic hands, forming a system with 24 degrees of freedom (DOF). The bionic hand subsystem is responsible for delivering the motion planned by the control subsystem and interacting with the user directly. From bottom to top, each hand has a moveable platform, 10 SG-90 servo motors, and a plastic hand. The moveable platform, driven by two RDS3115 digital servo motors, holds the plastic bionic hand and provides two extra DOF. The SG-90 servo motor is directly fixed inside the fingers, and its output shaft will connect the finger's moveable part with a linkage and drive the finger to rotate around the joint. Therefore, the finger part, motor, and linkage will form a basic 4-bars link system and move smoothly. The combination of fingers' movements will form different gestures. The bionic plastic hand comprises 26 parts, as shown in Fig. 5. We plan to manufacture them with 3D printing using PLA material.

2.3.2 Requirements and Verification

Considering the system functionality and durability, we have listed requirements and verification in Table. 1.

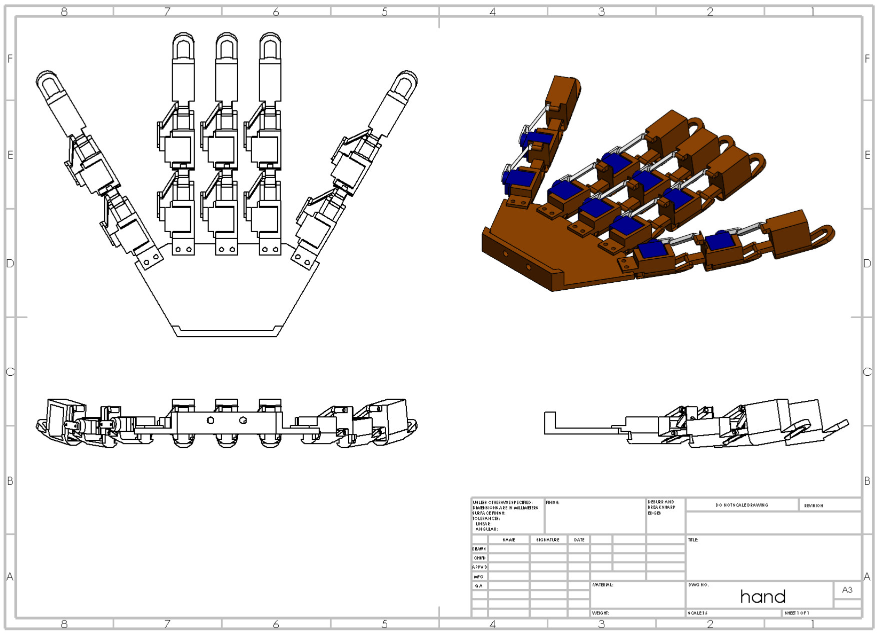

Figure 5: Engineering Drawing of the Bionic Hand

Table 1: Requirements and Verification for Bionic Hand

Requirements	**Verification**
1. Every finger has 2 DOF	A. Using CAD to assembly and validate the design. B. Print the finger and check the smoothness of motion.
2. Finger can bend for more than 45 degree	A. Using ”motion study” function on CAD software to have a throughout dynamic simulation.
3. Motion of finger is finished within 500 ms	A. Connect the servo motor, linkage, and finger to test and record the motion time.
4. The palm can steadily fix five fingers and avoid shaking	A. Using CAE software to have statics simulation and analysis on palm part under various situations. B. Test the strength of palm part before assembly.
5. Motion of each finger would not cause intervention with others	A. Conduct dynamics simulation to validate the design.

2.4 System & Control Subsystem

2.4.1 Description

The system & control subsystem is aimed at translating from sign language predictions to PWM signals and delivering the signals to servo motors, which makes it possible for communication and control of the whole system. It contains one development board with a microcontroller and a computing unit that supports real-time gesture recognition. The schematic diagram of control system components is shown in Fig. 6.

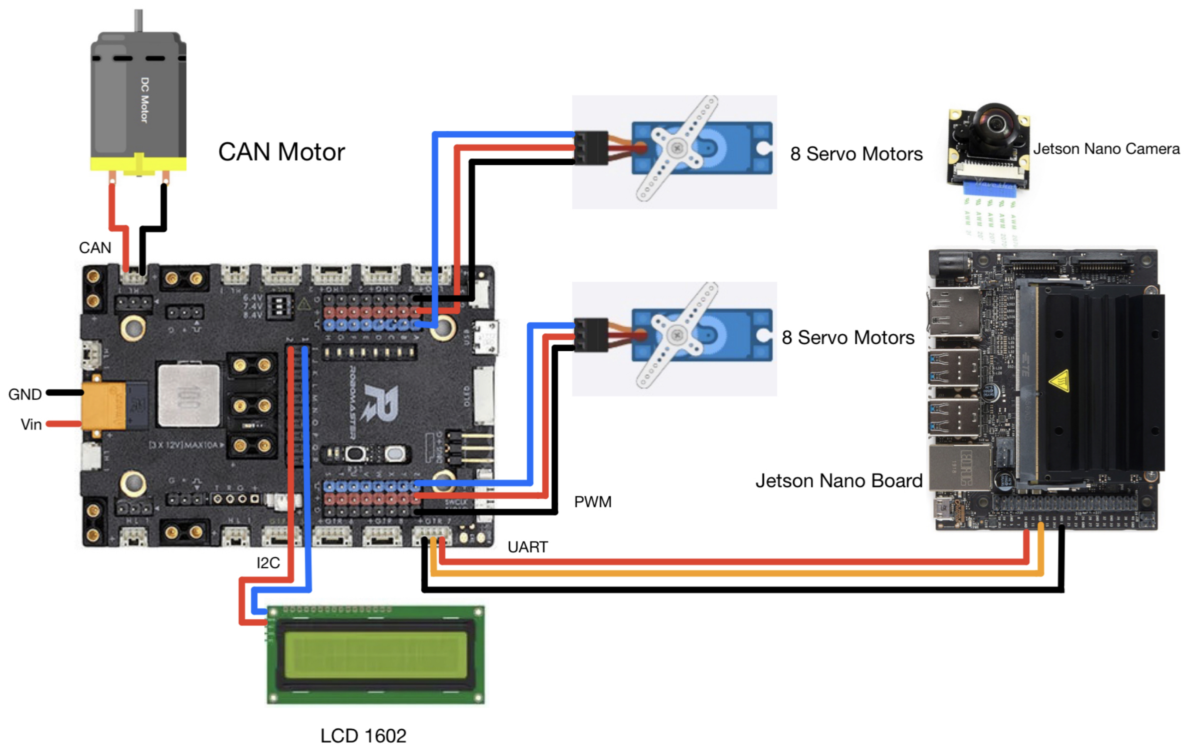

Figure 6: Schematic Diagram of Control System Components

2.4.2 Connection with Other Subsystems

- Connection to the power source: Our MCU is driven by 3.3V.
- Connection to the gesture recognition subsystem: Receives predictions from machine learning model.
- Connection to the bionic hand subsystem: Output PWM signals to each servo motor to control the motion of the bionic hand.
- Connection to the input and output Subsystem: The camera will be connected to our computing unit to capture the gesture, and LCD will be connected to MCU via I2C.

2.4.3 Microcontroller & Development board

We choose STM32F427IIH6 as our microcontroller. It controls the motion of 24 servo motors on the bionic hand by sending PWM signals. The specific motion signal should be generated based on the decision from the computing unit Jetson Nano through Universal Asynchronous Receiver/Transmitter (UART). We use Robomaster Development board A as our main control board. It operates on ChibiOS which provides a hardware abstraction layer.

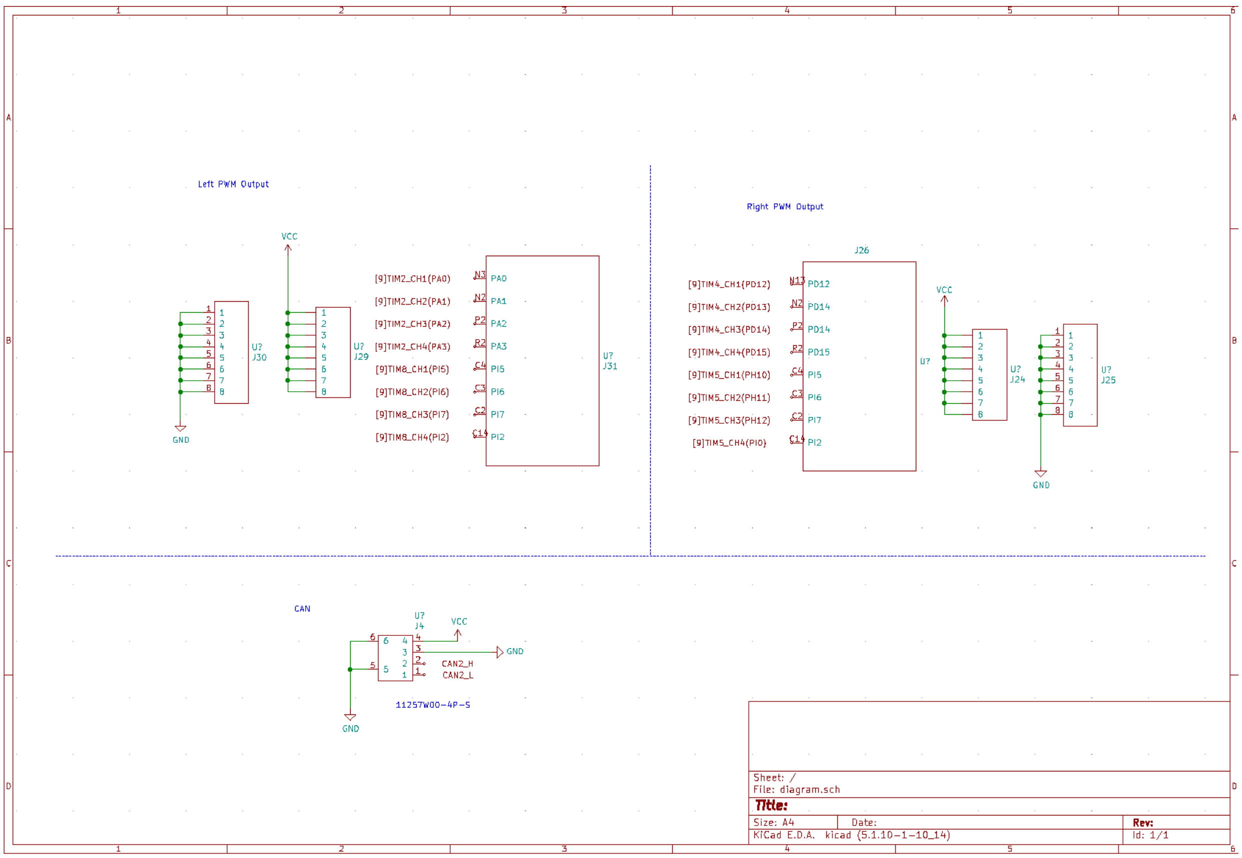

Figure 7: Schematic Diagram of PWM and Controller Area Network (CAN) Output Pin

Table 2: Requirements and Verification for the Microcontroller

Requirement	Verification
1. The microcontroller needs to output 20 stable PWM signals with 50 Hz frequency. Stable means the error of motor deflection angle should not exceed 15 degrees.	A. To test 20 PWM signals, we need to connect it to Oscilloscope to see the waveform. B. We use Oscilloscope to sample PWM waveform after the motor starts. To stabilize, we will use pulses between 1ms and 2ms as drive signal, in order not to go beyond the range of having linear relationship between pulse width and rotation angle. C. Use embedded operating system ChibiOS function to get PWM driver frequency configuration.
2. The delay from microcontroller receives decision message from Jetson Nano board to output corresponding PWM signal should be less than 5 seconds.	A. We will use shell to debug and test the program on MCU, start timer and output the time when MCU receives the decision message and end timer when it outputs PWM signal to motors.
3. The degree of servo motors should not exceed the maximum degree. Otherwise, the mechanical parts will be damaged.	A. Check if the PWM configuration program has the 2.5ms high level pulse restriction.

2.4.4 Computing Unit

We will use Jetson Nano as our computing unit in this project. Jetson Nano is a small, powerful processing unit that can run multiple neural networks in parallel for applications like image classification, object detection, and speech processing [3]. We use this platform to deploy our sign language recognition model. It will intake data from the camera unit and perform computing onboard after getting the result from the gesture recognition subsystem. The computing unit, powered by a development board, will send all decisions to MCU via UART.

Table 3: Requirements and Verification for the Computing Unit

Requirement	Verification
1. Can generate gesture prediction results within 10 second.	A. Start Jetson nano camera and do sign language, 3D points on hand captured as input data B. (Start timer) process input data, verificate in our trained machine learning model and make predictions. C. (Stop timer) ensure the predictions can be made within 10 seconds after user has done the gesture
2. Can generate gesture prediction results with 90% prediction accuracy	A. Run the whole program and do the same hand gesture based on our dataset for 20 times. B. Record the feedback from our bionic hand. C. Use the ground truth feedback to calculate accuracy.

2.5 Input & Output Subsystem

2.5.1 Description

The input and output subsystem includes one camera and one LCD. The camera module captures the user's hand gesture as input data. The LCD displays sign language dialogs to other people as text to help people who do not know sign language learn the conversation between the intelligent assistant and the user.

2.5.2 Connection with other subsystems

- Connection to the power source: The camera and LCD are connected to a 5V power source. Jetson Nano board and MCU supply the camera and LCD power, respectively.

- Connection to the control subsystem: Jetson Nano board and MCU supply power to the camera and LCD, respectively. The camera inputs hand gesture data through a USB wire, and the LCD displays sign language dialogues.

2.5.3 Camera

A webcam is used for capturing real-time hand gestures of the user. The input resolution should be 3280*2464 with 79.3 field of view. It will be connected to Jetson Nano board by Flexible Flat Cable (FFC).

Table 4: Requirements and Verification for the Camera

Requirement	Verification
1. Camera should have enough resolution to capture the characteristics of hand.	A. Connect the camera to Jetson Nano Board. Run gesture recognition, if the 3D-coordinate of critical points on hand are captured, the camera works.
2. User standing 0.5m - 1.5 m in front of our camera can be captured by Jetson nano camera easily	A. We will start camera and gesture recognition system, testing from standing 0.5 m in front of camera to standing 1.5 m in front of camera. If too close or too far to meet the requirement, we will change the camera.

2.5.4 LCD

When a user uses sign language to communicate with our assistant, not only our bionic hands will give users feedback, but also the dialog between users and the assistant will be displayed on LCD as text. We use 12864LCD in this project. It can be connected to development board via Inter-Integrated Circuit(I2C).

2.6 Gesture Recognition Subsystem

2.6.1 Description

The gesture recognition subsystem, which consists of object detection and sign language prediction, is mainly used to extract features from the images passed by the camera and then feed them into a pre-trained model for further prediction.

Table 5: Requirements and verification for LCD

Requirement	Verification
1. LCD driven voltage needs to be 3.0V ∼5.0V	A. Use multimeter to test the STM32 output voltage
2. Character shown on the LCD screen should display at a fast speed. Whole dialog should be shown on the screen in 5 second.	A.We can use timer to record the display process. B. We can also use shell to debug the display program and check the duration time.

2.6.2 Connection with other subsystems

- Connection to the input and output subsystem: Receives the image delivered by the camera.
- Connection to the system and control subsystem: Passes the prediction results for further control used.

2.6.3 Feature Extraction and Prediction

Object Detection

Considering the cost-effectiveness and convenience, we proposed adopting computer-vision-based techniques to detect objects rather than sensor-based ones. However, most computer vision-based methods consisting of gesture segmentation and hand shape estimation have high demands on high computing power, which indicates that a platform with robust processors is needed. Given that we do not have the equipment to support those methods, we select an open-sourced framework, MediaPipe, developed by Google, to detect the users' body movements.[2] By using the machine learning pipeline under MediaPipe's hand tracking solution and pose detecting solution, we can accurately get high-fidelity 3D-coordinates (i.e., x, y, z-axis) key points quickly, which is lightweight enough to run in real-time devices. Figure 8[4] shows the hands' landmarks returned by the framework.

Sign Language Prediction

We treated recognizing static signs image as a baseline, and the final goal is completing a dynamic recognition subsystem. As shown in Figure 9, we need a pre-trained model to predict the meaning of American Sign Language (ASL) input images. Before introducing the machine learning algorithms and models we selected, we must have enough high-quality datasets to train and test our models. We plan to take 400 photos of each sign language that must be classified for the static recognition dataset and combine open-

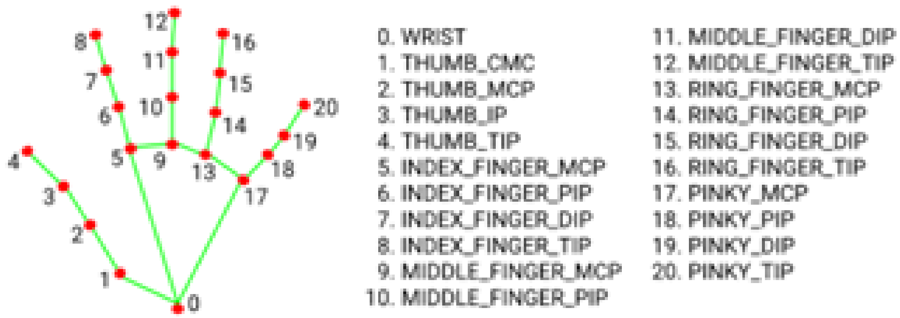

Figure 8: Hand's Landmarks

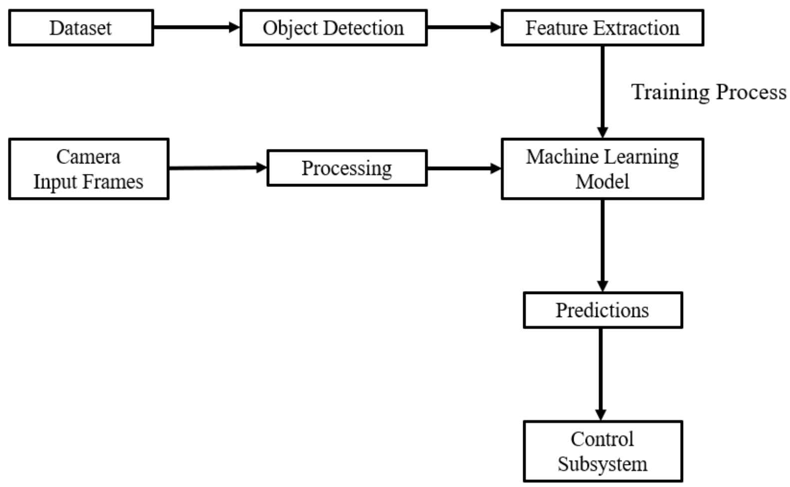

Figure 9: Gesture Recognition Subsystem Workflow

sourced dataset[5] Kaggle if more samples are needed. Each sign movement sample will be recorded in a video of 30 frames for dynamic recognition. Thirty frames is an appropriate length for us to conduct a complete sign language. Since video recording is time-consuming, we plan to record forty videos for each type of sign movement.

After obtaining sufficient datasets, we could utilize MediaPipe to extract the critical landmarks on hands and poses. Figure 10 shows one image sample's key points representation in the 3D coordinate system. We learned that the returned landmarks change as the relative position of the hand in the image changes. To solve the problem, we will select one fixed point as a reference and then adjust all other landmarks accordingly relative to the selected point. In this case, we could successfully eliminate the influence of location effects on the outcome. After pre-processing, we could use the updated coordinates information as the data features and feed them to models for training.

We will not deploy complicated neural networks in static recognition tasks because traditional machine learning algorithms can achieve well enough results. Considering the number of features is much lower than the number of samples when dealing with static

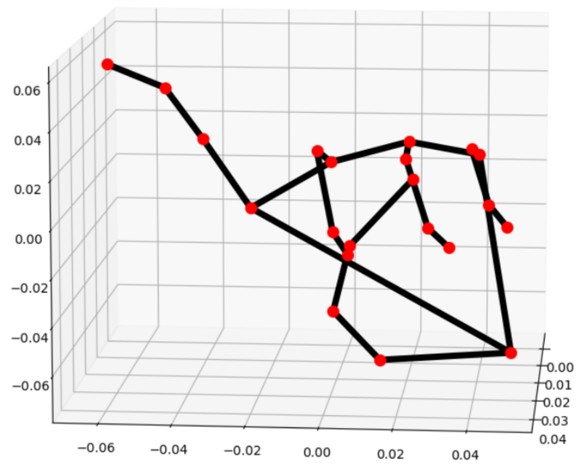

Figure 10: Hands' Landmarks in 3D Coordinates

classification tasks, we adopted Support Vector Machine (SVM), which performs better in high-dimensional space than other traditional models[6]. We will test the model in the different kernels to find the one that best fits our dataset. In the case of dynamic recognition, we proposed adopting recurrent neural networks (RNNs), which contain memory storing information from previous states' computations. Thus, they can deal with time series and sequential data. While traditional RNN may occur problems of gradient vanishing, we planned to adopt LSTM, a variant of RNN, to prevent the potential risk. Due to the possible lack of data, we may encounter the problem that our data set is insufficient to train LSTM with many parameters well. Hence, Gate Recurrent Unit (GRU) is our alternative solution for its less parameter so that we could perform better with limited data and more efficiently[7]. In this case, we proposed to adopt GRU and LSTM as our machine learning.

2.6.4 Requirement & Verification

Table 6: Requirements and Verification for Gesture Recognition Subsystem

Requirement	Verification
1. The accuracy of the recognition should be above 90%	A. Divide the dataset into the training dataset (80%) and testing dataset (20%). B. Taking the strategy of cross-validation to train and evaluate the model. C. Input real-time video captured by the camera to check the performance of the trained model.
2. The model's whole response time should shorter than 500ms	A. Group members perform different sign movement and record the time from the end of the action to the display result formore than 20 times. Calculate to get the mean valueas the final response time.

2.7 Tolerance Analysis

Given that our machine learning models' performance depends to a large extent on the dataset's quality. Especially in dynamic sign language recognition, our dataset should be established by recording as clips of videos, and it is inevitable for us to generate a lot of low-quality data due to various reasons, including the camera's resolution, lighting conditions, and the correctness of the recorders' sign language movement.

We will evaluate the results using performance matrices, including accuracy, precision, recall, and F1 score, to evaluate our outcome quantitatively. Accuracy represents correctly predicted labels from the whole dataset, as shown in Equation 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision measures the number of actual positives in all the positives predicted by the model, which is a good measurement when the cost of False Positive is high. Equation 2 gives the mathematical formulation. Recall calculates the number of actual positives predicted correctly by our model, which is a good measurement when the cost of a False Negative is high. Equation 3 gives the mathematical formulation. The F1 score combines Precision and Recall and represents both properties. Equation 4 gives the mathematical formulation.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Recall = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

In the above equations, TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively. Confusion Matrices are also proposed to understand the models' performance better.

Concerns related to the bionic hand subsystem mainly circled with fingers' motion smoothness and material strength. The core system can be abstracted to a four-bar linkage for motion smoothness. Therefore, detailed kinematics and dynamics simulation is necessary. Initially, we referenced the simulation tool used in UIUC TAM 212 course[8]. Due to the space limitation, we had the presupposed ground link length (g) and input link length (a). By changing the output link length (b) and floating link length (f), we have combinations of four-bar linkages that can deliver satisfied trajectories on the output node, which links to the finger part driven by the motor. We conduct dynamics simulation for those candidate four-bar linkages using the MATLAB program developed in UIUC ME 370 lab and project. By calculating the position, velocity, acceleration, and torque on the output node for the whole cycle, we chose the best four-bar linkage that suffers little impact and vibration. The kinematic and dynamic simulation results are shown below in Fig.11 and Fig. 12. The formulas we used are listed below:

$$\text{Grashof index:} \quad \boldsymbol{G} = \boldsymbol{s} + \boldsymbol{l} - \boldsymbol{p} - \boldsymbol{q} \geq 0 \tag{5}$$

$$\text{Validity index:} \quad \boldsymbol{V} = \boldsymbol{l} - \boldsymbol{s} - \boldsymbol{p} - \boldsymbol{q} \geq 0 \tag{6}$$

where $\boldsymbol{l}$ is the longest link, $\boldsymbol{s}$ is the shortest link, and $\boldsymbol{p}$, $\boldsymbol{q}$ are the rest two links.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -R_{12y} & R_{12x} & -R_{32y} & R_{32x} & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & R_{23y} & -R_{23x} & -R_{43y} & R_{43x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_{12x} \\ F_{12y} \\ F_{32x} \\ F_{32y} \\ F_{43x} \\ F_{43y} \\ F_{14y} \\ T_{12} \end{bmatrix} = \begin{bmatrix} m_2 A_{CG2x} \\ m_2 A_{CG2y} \\ I_{CG2}\alpha_2 \\ m_3 A_{CG3x} - F_{Px} \\ m_2 A_{CG2y} - F_{Py} \\ I_{CG3}\alpha_3 - R_{Px}F_{Py} + R_{Py}F_{Px} \\ m_4 A_{CG4x} \\ 0 \end{bmatrix} \tag{7}$$

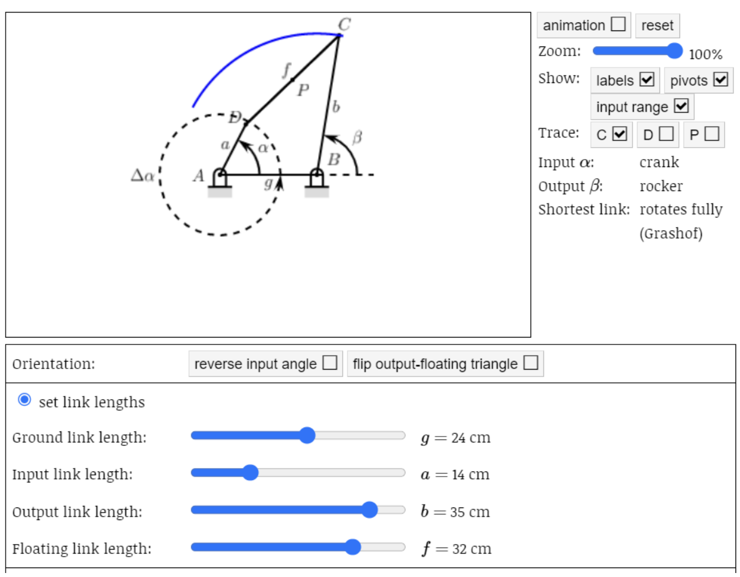

Figure 11: Kinematics Simulation Results of Proposed Four-bar Linkages

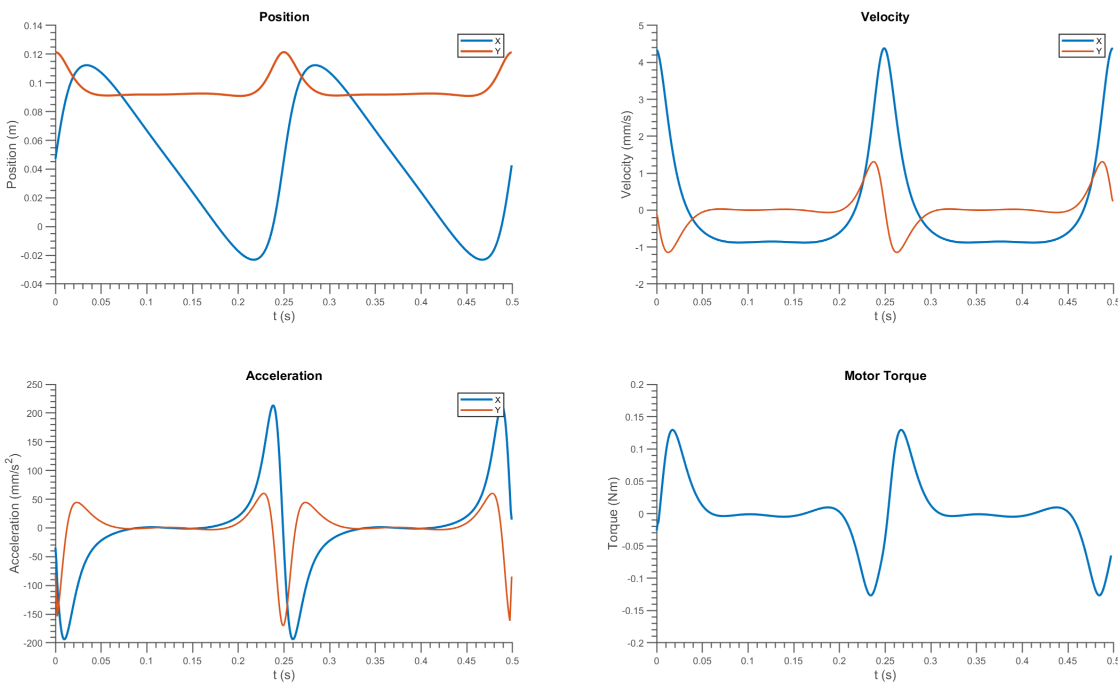

Figure 12: Dynamics Simulation Results of Proposed Four-bar Linkages

For critical parts, we conduct statics simulation and failure analysis with computer aids engineering (CAE) software to verify and improve our design. For instance, as demonstrated in Fig .13, the palm we initially designed have right-angle sides, which may cause stress concentration and even yielding under impact. Afterwards we strengthen the part and fillet the right-angle sides. The optimized design performs much better under the same load conditions as shown in Fig. 14.

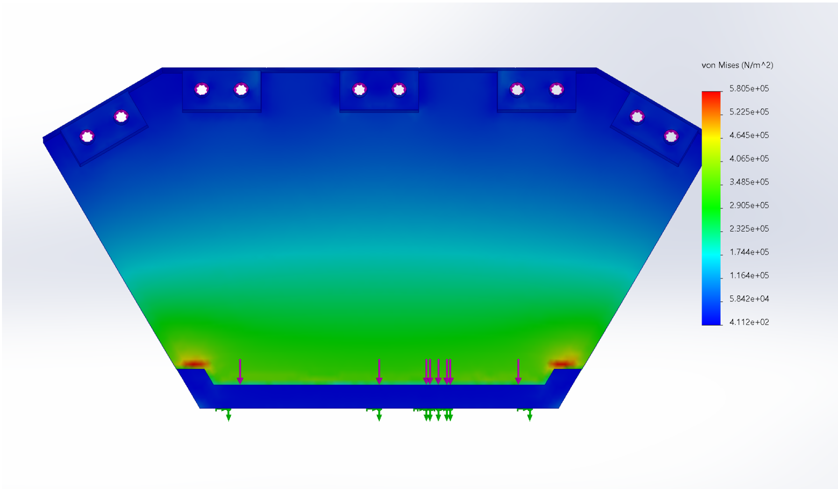

Figure 13: Statics Simulation for Initial Palm

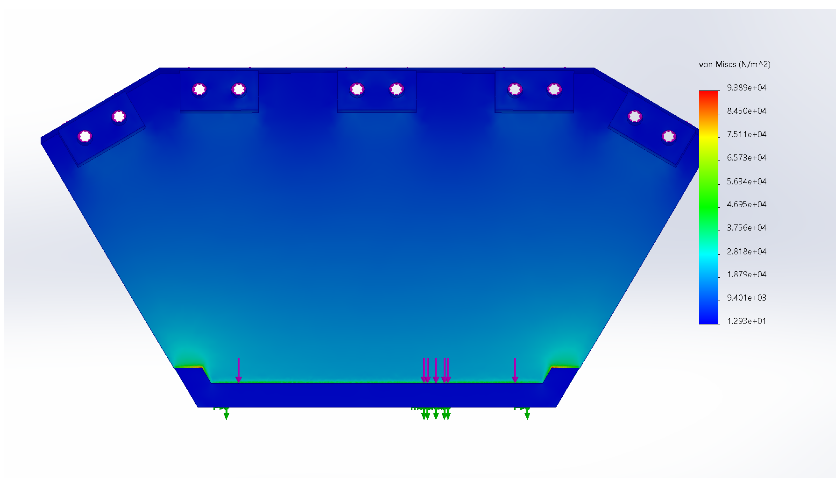

Figure 14: Statics Simulation for Optimized Palm

The tolerance analysis lies on control subsystem concentrates on the stability of PWM signals. In order to make sure the signals would not interfere with each other, and servo motors would be move correctly, some other control modules might also be added to our development board.

3 Cost & Schedule

3.1 Cost

As engineers, all teammates' labor is valued and should be evaluated according to the total time to complete the project. Our group decides to work at least 15 hours per week per person on designing, testing, and validating our project and at least 3 hours per week to document our work. The graduate Research Assistant at the University of Illinois will get paid 40$ per working hour. Therefore the total labor cost of our team will be:

$$4 \cdot \frac{\$40}{hr} \cdot \frac{18hr}{week} \cdot 10week \cdot 2.5 = \$72,000$$

At the same time, the cost of the prototype is estimated at $235.3 in total:

Table 7: Prototype Cost Estimation

Part	Cost per piece	Piece	Total
Electrical Platform (2 DOF)	$20	2	$40
Bread Board	$2.5	1	$2.5
Servo Motor (SG90, 9 gram)	$1	20	$20
Steal Rod (ϕ 2.8mm * 1m)	$5	1	$5
M3/M4/M5 Bots & Nuts	$2.5	2	$5
3-mm Bearings	$0.4	2	$0.8
Jatson Nano Development Board (B1)	$150	1	$150
Camera	$12	1	$12
Total		30	$235.3

Therefore, our total cost will be estimated at 72,235.3$.

3.2 Schedule

week	Hanwen Liu	Yike Zhou	Haina Lou	Qianzhong Chen
3/20	Set group rules, distribute work, and read embeded system development manual	Extract Features of an open-sourced dataset by MediaPipe. Train and evaluate several machine learning models.	Buy hardware materials and learn embedded RTOS development process	Iterate and refine the mechanical design. Manufacture the Version#0 finger and combine with servo motor for simple demonstration
3/27	Construct our static sign language dataset and find effective features	Complete the static recognition model and test it on a real-time scenario	Implement a small demo: programming to output PWM signals and drive one servo motor	Confirm the mechanical design, pace up to manufacture the bionic hand
4/3	Deploy pre-trained model on the device to test the performance	Build and train LSTM based on open-sourced dataset	Output different signals to schedule all 24 motors	Continue to manufacture the bionic hand and purchase the electrical platform holding the hand
4/10	construct our dynamic sign language dataset	Build and train GRU based on open-sourced dataset	Test the motors which are installed in bionic hand to realize different hand gestures	Work with Haina to tune the motor and control code, turning the simple motion of fingers to gesture

week	Hanwen Liu	Yike Zhou	Haina Lou	Qianzhong Chen
4/17	Work on embedded programming to make dialog display on LCD	Adjust the model's structure and parameters on our dataset	Connect LCD to STM32 development board and continue work on other different hand gestures	Continue to work on gestures, manufacture the electronic components box, get ready for system test
4/24	Work with Haina to connect Jetson Nano board to STM32, and perform different hand gesture based on prediction	Explore more models that may yield good outcomes	Help Yike deploy model on Jetson Nano board. Work with Howie to connect Jetson Nano board to STM32	Tune and refine the details of system's mechanical parts, start preparing final demo
5/1	Combine and test all the subsystems. All together			
5/8	Prepare mock demo and the final report draft. All together			
5/15	Prepare functionality demonstration video and the final report. All together			

4 Ethics & Safety

When designing a product, safety is our top priority. To ensure "the safety, health, and welfare of the public"[9], it is vital to notify users of potential hazards and minimize the possibility of systematic danger caused by misuse of our work. This means guaranteeing that users are aware of any potential risks associated with using our intelligent assistant and are given clear instructions, warnings, and possible solutions when danger happens. In addition, it is essential to implement safety features and safeguards that can help prevent accidents and minimize the consequences and risks of any incidents that could happen.

As engineers, we are responsible for addressing unforeseen risks to ensure the safety of users. Regarding the movement of the fingers controlled by pulling on strings, we must implement safety features to prevent any harm caused by misuse. This could include providing clear instructions on how to use the product and warnings and possible solutions when danger happens. Additionally, we could design the strings to minimize the possibility of jerking or twisting, such as using more vital strings or providing guides for proper string movement. We could include safety covers or guards around the strings to avoid tangling or wrapping around the user's neck or other body parts.

For the sharp or protruding parts of the product, we must mitigate the risks by adding protective covers, smoothing the edges, or even redesigning the product. We could also provide clear instructions on handling the product to prevent harm.

In case of a system malfunction or short circuit, we could incorporate safety features to minimize the consequences and risks of any incidents that could happen. We should use materials less likely to cause harm from accidents, such as fire-resistant materials. Additionally, we could design the product with an emergency shut-off feature that could quickly disconnect the power source in case of a malfunction.

The Occupational Safety and Health Administration (OSHA) [10] provides guidelines for ensuring workplace safety, while the Federal Communications Commission (FCC) [11] sets standards for electronic devices' safety.

The soul of design is to help people's life easier. As a society, striving for respect, inclusivity, fairness, and equilibrium is vital, ensuring everyone has access to the tools and resources they need to live fulfilling lives without "discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression"[9]. Our core is to help people with hearing and speech problems could also interact with intelligent assistants.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection." (Jun. 2016).

[2] C. Lugaresi, J. Tang, H. Nash, *et al.* "Mediapipe: A framework for perceiving and processing reality." (2019), [Online]. Available: https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf.

[3] N. Developer. "Jetson Nano Developer Kit." (2021), [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (visited on 03/08/2023).

[4] Google. "Hands." (2021), [Online]. Available: https://google.github.io/mediapipe/solutions/hands.html (visited on 03/08/2023).

[5] AKASH. "ASL Alphabet." (2018), [Online]. Available: https://www.kaggle.com/datasets/grassknoted/asl-alphabet (visited on 03/24/2023).

[6] A. Halder and A. Tayade, "Real-time vernacular sign language recognition using mediapipe and machine learning," *Journal homepage: www. ijrpr. com ISSN*, vol. 2582, p. 7421, 2021.

[7] G. H. Samaan, A. R. Wadie, A. K. Attia, *et al.*, "Mediapipe's landmarks with rnn for dynamic sign language recognition," *Electronics*, vol. 11, no. 19, p. 3228, 2022.

[8] Matthew-West. "UIUC TAM 212 Course Website." (2021), [Online]. Available: http://dynref.engr.illinois.edu/aml.html (visited on 03/23/2023).

[9] IEEE. "IEEE Code of Ethics." (2020), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/08/2023).

[10] "Law and regulations." (2023), [Online]. Available: https://www.osha.gov/laws-regs (visited on 03/24/2023).

[11] "Federal communications commission." (2023), [Online]. Available: https://www.fcc.gov (visited on 03/24/2023).