ECE 110/120 Honors Lab

Fall 2022

HandCV

Robotic hand controlled with computer vision

Ryan Li (rl31) - ECE 110 & 120
Harbin Li (hdli2) - ECE 110 & 120

1 Introduction

We will be using computer vision to capture the movement of a human hand and build a robotic hand that replicates the movement of the human hand. If successful, this prototype can serve to perform delicate, but dangerous operations without putting the operator in risk. Our demonstration of using computer vision to simulate human movements can also be used in developing humanoids with smoother and lifelike motions.

1.1 Background Research

Prosthetic limbs and humanoid robotics have been seeing more development as they are a gateway to solving problems with disabilities, as well as becoming the staple for automation. It reduces risks involved with tasks that require high precision or high strength, which robots have a higher potential of achieving. A group last semester did a similar project. They used computer vision to control a robotic arm with two joints. We wanted to strive to include the precision of the fingers in our hands more so than the arm motion. Therefore, we are extending this previous project and focusing on the hand aspect.

Our project is split up into mechanical, electrical, and software components that will be worked on individually and then integrated together.

For the computer vision part, we will be using OpenCV and mediapipe to identify the landmarks on our hand. OpenCV and mediapipe are both python libraries that facilitate computer vision projects. OpenCV will provide most of the commands that we will use in our code, and mediapipe, which is a machine learning module, will help us identify the hand landmarks from the image input of our camera. Then, using the coordinates of the hand landmarks, we can write a program using Python that identifies hand movement through the relative position of the landmarks.

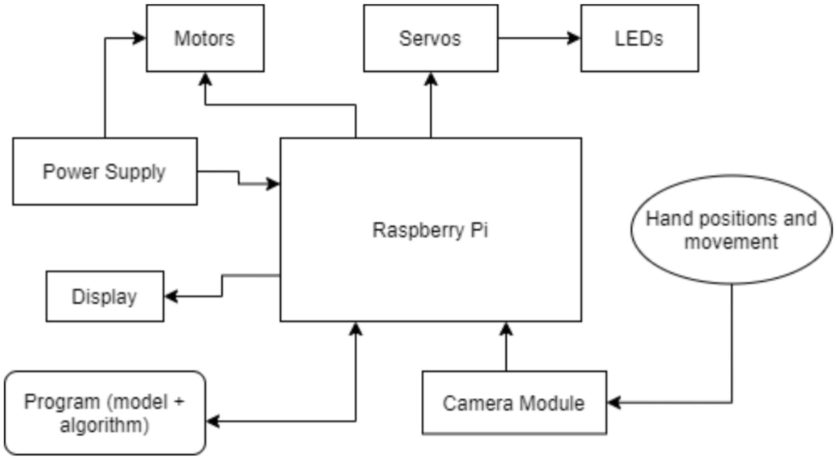
For the mechanical robotic hand, we will be designing the parts in CAD and going through iterations of design to make a system where each finger can be controlled by a single servo. There will be a line of string for extension and another line for retraction. The motion of the arm will be controlled by two DC motors. Both the servos and the DC motors will be encased in 3D printed structures.

Lots of the electrical components we need will already have premade boards with the functionality that we want. We'll have a Raspberry Pi computer due to its ability to smoothly run Python code, as well as breakout boards for motor controllers. We'll also use LEDs to indicate the positions of the fingers as a way to debug and for initial prototyping.

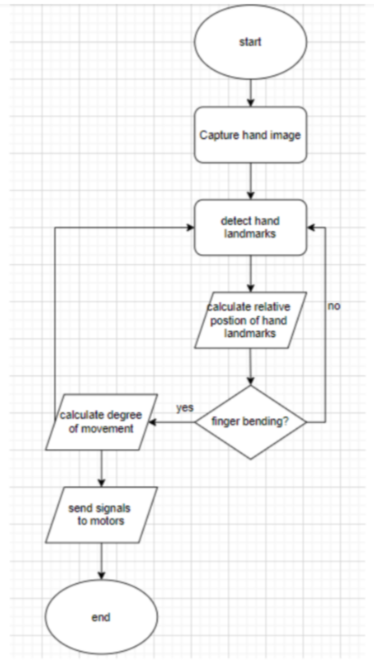
All these subsystems will be slowly integrated together so that bugs and design flaws can be addressed at a smaller level before integration and make for a more efficient design process.

1.2 Flowcharts and Diagrams

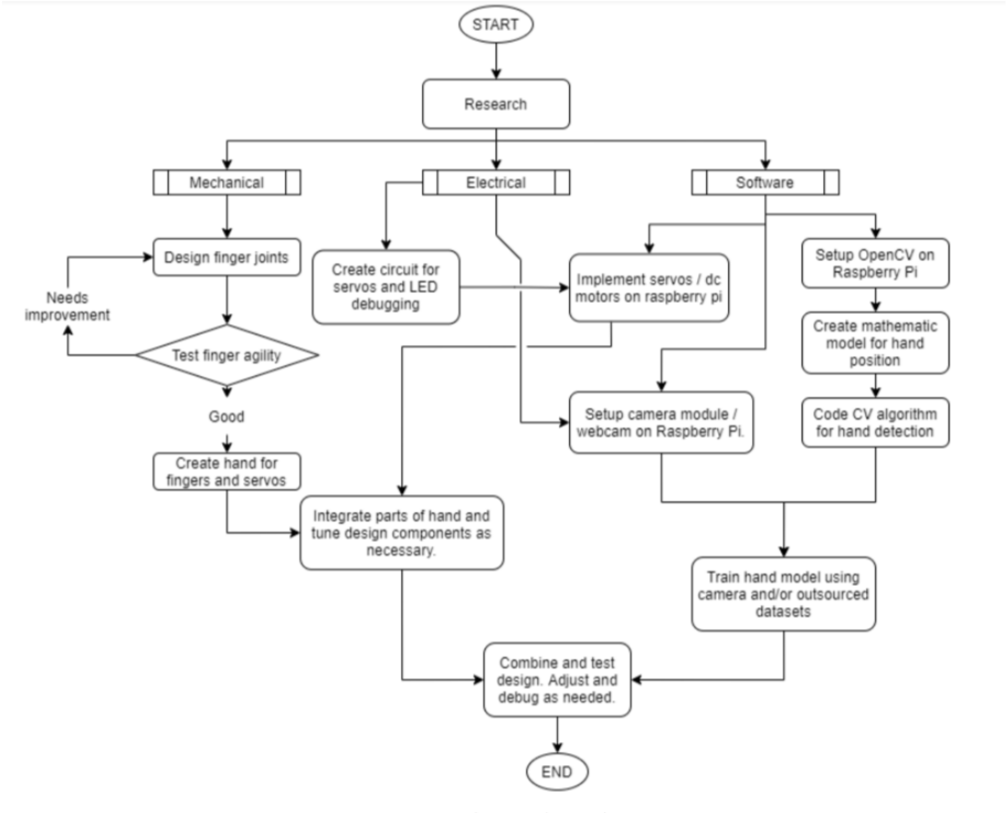
Below, we have diagrams of all the different components and how they interact with one another. There is also a brief flowchart that describes the control process during the execution of the final product.



Block Diagram of All Components



Control Process


START
Research
Mechanical
Electrical
Software
Design finger joints
Needs improvement
Test finger agility
Good
Create hand for fingers and servos
Create circuit for servos and LED debugging
Implement servos / dc motors on raspberry pi
Setup OpenCV on Raspberry Pi
Create mathematic model for hand position
Setup camera module / webcam on Raspberry Pi.
Code CV algorithm for hand detection
Integrate parts of hand and tune design components as necessary.
Train hand model using camera and/or outsourced datasets
Combine and test design. Adjust and debug as needed.
END

Project Outline Flowchart

1.3 Timeline

Week 5	Tasks
5	Finish project proposal and continue research
6	Complete first full prototype of fingers. Setup OpenCV.
7	Make edits to fingers and design hand.
8	Final draft of fingers. Finish designing mathe model for OpenCV and write initial code.
9	Complete LED debugging circuit. Test with RPI.
10	Test fingers with servos and adjust servo mounts on hand.
11	Finish coding algorithm for improving hand model. Start training with the camera.
12	Integrate hand and arm with motors into the RPI with breakout boards. Train model.
13	Initial test.
14	Debugging.
15	Project wrapup.

1.4 Parts List

Part Name	Vendor	Purpose	Quantity	Price/unit
Raspberry Pi		Main computer for running CV	1	*
Raspberry Pi Cam Module		Hand detection camera	1	*
9g Servos	Adafruit	Controls strings for fingers	5	$5.95*
Stepper Motor	Sparkfun	Controls wrist and elbow	2	$26.95
DC Motor Controller	Adafruit	Control motor direction and power	2	$4.95
Fishing Line		For stringing	1	$3.50*
			Total:	$63.80

*Already obtained

1.5 Potential Challenges

- How to convert the data collected from our camera to movement instructions for the robotic hand will be a challenge.
- Using limited budget on cheaper servos that must be tuned to have better precision.
- Allocating enough time to integrate all parts of the hand.
- Using motors with enough torque to move arm components

References

[1] S. Shakhadri, ”Hand Tracking System using OpenCV — Build a Hand Tracking System”, *Analytics Vidhya*, 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/07/building-a-hand-tracking-system-using-opencv/. [Accessed: 26- Sep- 2022]

[2] V. Kumar, Q. Wang, W. Minghua, S. Rizwan, S. Shaikh and X. Liu, ”Computer vision based object grasping 6DoF robotic arm using picamera”, *Ieeexplore.ieee.org*, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/8384653. [Accessed: 26- Sep- 2022]

[3] L. Tian, N. Magnenat Thalmann, D. Thalmann, and J. Zheng, "The making of a 3D-printed, cable-driven, single-model, lightweight humanoid robotic hand," *Frontiers*, 01-Jan-1AD. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2017.00065/full. [Accessed: 26-Sep-2022].