Developing a Soft Robot Inspired by Octopuses for Efficient Ladder Climbing

Abstract

In the current study of robotics, most research focuses on rigid robots using rigid structures and materials, for example, room-cleaning robots, cars, and mechanical dogs. These robots use stiff materials, shafts, and motors and are generally non-flexible. This raises a problem while trying to operate on uneven surfaces. Even though these robots are capable of dealing with certain uneven surfaces, but still lack general adaptability. In the future, we may need robots that can climb up cliffs, between crevices, or even foreign planets, which need robots to have extreme versatility. In this case, we need a new type of robot capable of such tasks, but how? When I saw octopuses, I came up with the idea of creating a flexible, octopus-like robot to accomplish such tasks. Some applications of flexible robots include rescuing people buried underground after earthquakes, climbing vertical cliffs with claws, and substituting humans for potentially dangerous jobs.

From observing and studying octopuses' motions, I can develop a soft robot using soft materials, an Arduino mainboard, motors, and strings to recreate specific movements. This project combines hardware and software skills to create. With strings and motors, the flexible arm can grab onto bars and pull itself upwards. With two such components, it can repeat the holding and releasing process to climb upwards periodically. Ultimately, it is capable of repeatedly climbing upwards on a ladder. This project uses these different motors and explores how they impact the accuracy and efficiency of the robot as well as how to control soft robot arms.

Keywords: Hard Robots/ Soft Robots, Flexibility, Adaptability

Table of Contents

1 Introduction

1.1 Background of the study and its purposes

Robotics has always captivated me, regardless of whether it has flexible arms or stiff structures. I used and saw a variety of devices growing up. However, I never learn how to build robots with soft components like rubber. I have heard that soft robotics is evolving and has a lot of potential benefits. In order to create a soft robot that can perform various tasks, including climbing ladders, I launched this project. This research has the potential to grow into more robust robots in the future. As robotics advance, this feature can turn into grabbing objects and becoming multifunctional arms!

Figure 1.1 Soft Robotics Arm

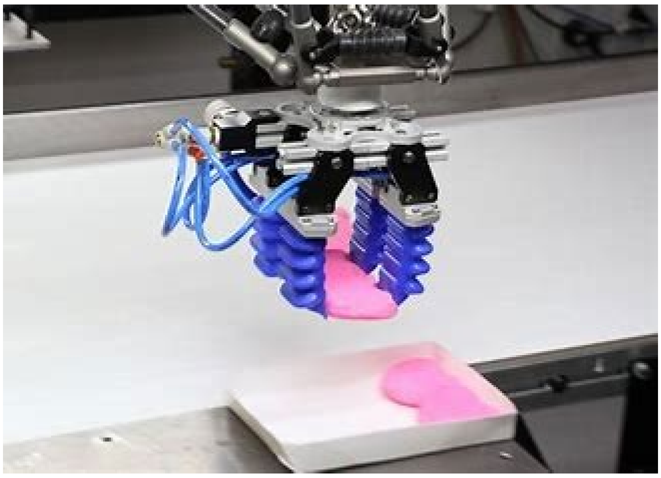

Figure 1.2 Soft Robotics Hand

When compared to hard robotics arms, soft robotics arms are more favored due to their incredibly high versatility and adaptability. In factories, arms are made out of hard metal, and

Compared to hard robotics, soft robotics arms are more favored due to their incredibly high versatility and adaptability. In factories, arms made out of hard metal and joints can only turn and rotate at certain angles, while a soft arm with malleable materials can bend at any point and perform a lot more functions than stiff robot arms. This difference is even more critical when looking at areas destroyed by tornadoes or earthquakes. Many people are waiting to be saved under cracks or in collapsed buildings in these areas. In this scenario, soft robotics can shine because of its high versatility. Furthermore, soft robots can navigate under the ground, inside crevices, and tiny gaps because of their malleability. These are all of its advantages over stiff robot arms.

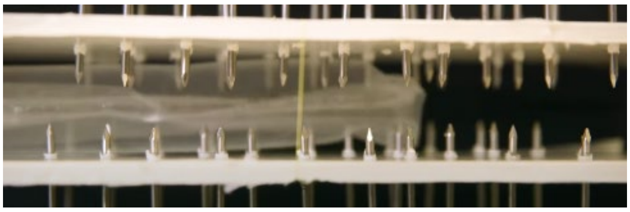

Figure 1.3 Soft Robot Transverse Through Spikes and Narrow Opening

Octopuses are most commonly studied when developing soft robots because of their strong and versatile tentacles. Their tentacles are extremely important for the study of soft robotics because these tentacles are the perfect model for soft robotics. Their tentacles can bend in any direction and into any form while maintaining a powerful force is an ideal model for scientists to study and replicate.

1.2 Current state of the study

From two decades ago to now, technology in soft robotics has increased drastically. The increase includes improving materials, software, control scheme, and robot designs. It quickly catches up to hard robotics and can complete complex tasks that traditional hard robotics could not.

Currently, there are many applications of soft robotics already in the world. Soft robotics arms design concepts based on octopus by M. Cianchetti, A. Arienti, M. Follador, B. Mazzolai, P. Dario, and C. Laschi are present. In their research, they used sophisticated experiments to recreate octopus arm movements. They undiscovered parts of the octopus' movement and rearranged muscles and components in the arm to simplify controls.

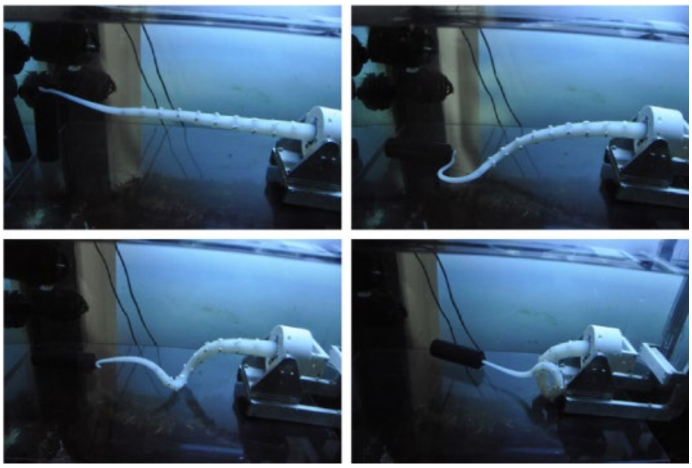

Figure 1.4 Soft Octopus Arm

Besides, design concepts and exact measurements of octopus arms are done in a research paper called “Soft robotic arm inspired by the octopus: I. From biological functions to artificial requirements” by L Margheri, C Laschi, and B Mazzolai. They discussed potential uses of soft robotics and design concepts for futuristics, novel robots that could derive from octopuses.

Another research aims at climbing various columnar objects. This is done in a paper called “The soft multi-legged robot inspired by octopus: climbing various columnar objects” by Kazuyuki Ito, Yoshihiro Homma, and Jonathan Rossiter. They designed a soft robot, Taoyaka-S II, with unique structures, plastic strings, and ribbons. They can control the robot with motors and is capable of climbing various columnar objects.

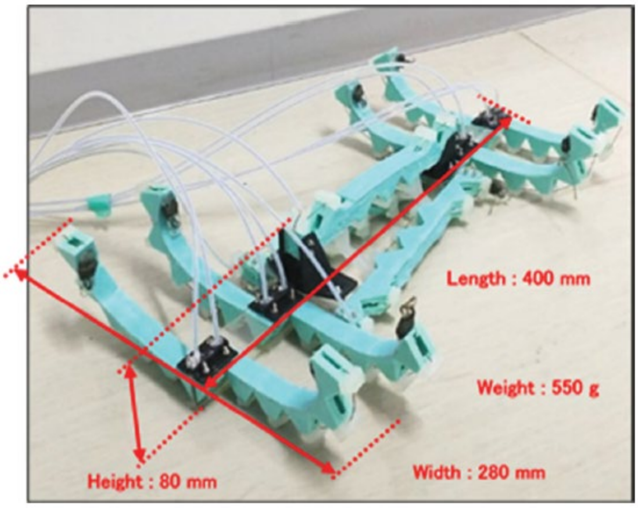

Figure 1.5 Taoyaka-S II

The robot comprises several parts, including the legs and the main body. When the top leg grabs tight and the bottom leg releases, the body art will pull the bottom of the robot upwards. Then the bottom part will tighten, and the top leg will be released, which allows the robot to be pushed upwards.

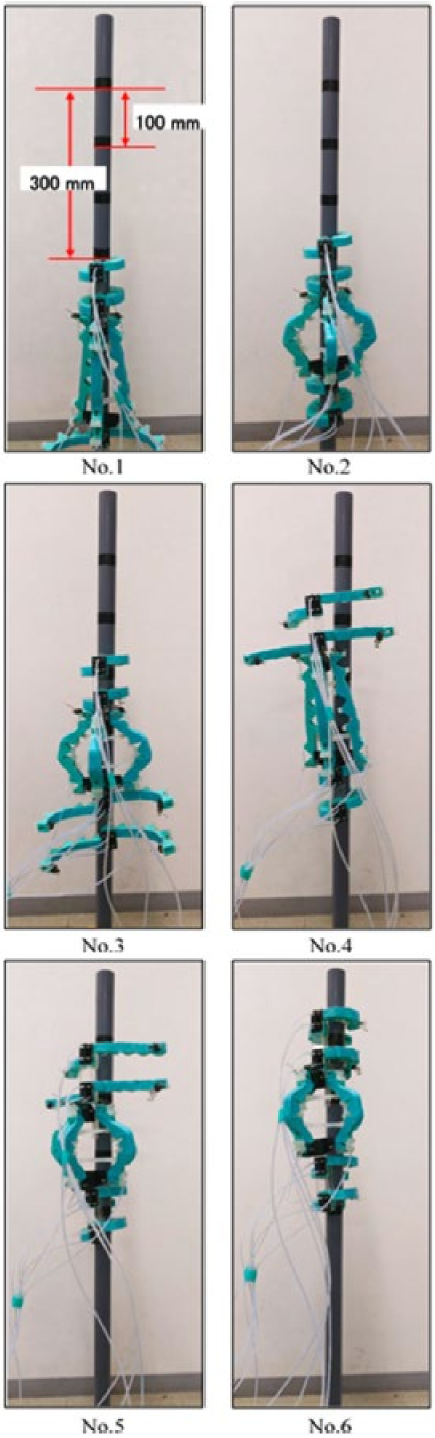

Figure 1.6 Taoyaka-S II Climbing

1.3 Plan of the study

1.3.1 Overall Project Plan

Overall, I expect to build a robot capable of self-correcting and climbing consistently for the current project. This can be achieved through a variety of sensors and motors. Then, with a good program that can drive the robot automatically or controlled by humans, it can climb ladders consistently and correct itself during problems.

Besides, this project is expected to undergo different iterations to improve based on previous designs. For example, a new iteration might include an improved chassis version, a strong-arm control system, or a change in overall robot structure design.

1.3.2 Hardware

The hardware we will use includes an Arduino mainboard, an angle sensor, a Bluetooth or inferred remote control unit, a servo, two motors, a motor control unit, and a chassis that is either 3D printed PLA plastic, or laser cut plywood. The Arduino mainboard will be responsible for the complete control of the robot, while the angle sensor will provide the tilt angle for the servo to correct at the bottom of the robot. The motors will be responsible for pulling the robot upwards and climbing. The robot can also be controlled with either Bluetooth or infrared because of the respective sensors.

1.3.3 Arduino Program

With the proper programming language provided by Arduino, I can program the robot to receive wireless information and act based on this information in real-time. Besides, the robot will also be able to make adjustments based on itself and do specific programmed automatic actions. This could be done with various libraries that can control different parts of the robot.

1.3.4 Expected Performance

For this project, we are expected to be able to climb at a consistent rate 1 interval every 7 seconds as the arm is contracting and pulling the robot upwards. The material that this robot's arm uses must have a stiffness of over 40 to guarantee that it can support itself while it is released from the ladder. Besides, the robot arm should require less than 100 newtons of force to retract completely. Otherwise, the strength of the motor may not be enough, and it may not be able to hand on the ladder properly.

1.3.5 Cost and Budget

This project is expected to cost 500 dollars to make. The Arduino starter kit is around 120 dollars. The new mainboards that I might use should cost approximately 50 dollars. Wires should cost approximately 10 dollars. A new set of motors and motor drivers should cost around 50 dollars. Material costs such as silicone, rubber, plywood, and PLA plastic should cost at most about 100 dollars. This plan uses around 330 dollars, leaving 170 dollars to purchase anything else necessary for the current budget.

1.3.6 Future Plans for the Current Project

However, the study's current state is limited to climbing ladders; however, it has a bright future and high potential. Since the arms are made of rubber, it offers great versatility when needed to bend in irregular shapes. Therefore, this arm can become multifunctional and become equipped with more tools, and be used for other purposes. In the future, I am planning to make these two arms able to move by themselves and can also grab items after attaching claws to their ends. Furthermore, with more powerful motors and intricate designs, it has vast potential to expand.

In the near foreseeable future, this project could be developed into a multi-use robotic arm. One such use scenario is in outer space when astronauts need to exit the spaceship; such a soft robot arm can guarantee that they will be secured. Another use case is for disabled people. They may use this arm as an entire arm. In the future, with the development of neuroscience and technology, people will be able to control such an arm with their minds.

2 Mechanical Structure

2.1 Version 1

2.1.1 Overall Structure Design – V1

The first soft robot design contains two rubber arms and multiple laser-printed plates connected via screws. This holds the rubber arm together well and provides enough room for motors and other Arduino modules.

The diagram above demonstrates the overall structure of the robot. It is rather simplistic to reduce weight, and the arms can be controlled efficiently using Arduino motors and a mainboard. The circuits and connections in the system keep the motor powered and controlled.

2.1.2 Arm Design – V1

The arm performs a critical function in this robot. It is responsible for the primary climbing function. Therefore, it needs to be durable and as light as possible. Multiple materials have acceptable weights and stiffness. In version 1, the material choice is rubber, with a stiffness of 60. Besides material choice, the design is also extremely important. Since the arm needs to be capable of bending and grabbing on the bars, the tip is curved and indented with grooves. Therefore, when fitting a string through, it will be able to bend to the opposing side and grab the bar. When further pulling the string, the bigger indents near the stem of the arm bend and then pull themselves upwards.

Finally, it is also important to not on how to produce an arm like this. There are three steps in producing a compatible rubber arm. First, the model of the arm is created in CAD software, and then the mold. Next, the mold is 3D printed without the top plate and the top plate with screws. Finally, pour the liquid rubber into the mold and then leave it all day and night to solidify. Some trimming may be required as there may be redundant rubber connected to the arm.

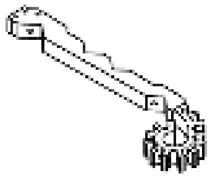

Figure 2.1 Arm V1

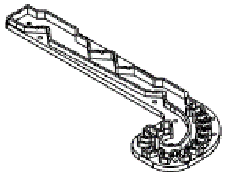

Figure 2.2 Arm Mold

2.1.3 Chassis Design – V1

In the first version, chassis pieces are laser printed and then connected with screws. It is essentially a big box with screw holes, an opening on one end to fix the arm, and a large empty area for any Arduino components. The screw holes are carefully designed and placed such that each piece will be stably connected. Every sharp edge is rounded such that it will not accidentally injure anyone. While at rest, the arm will still have support from the backplate to stay roughly in the same spot. Besides, the chassis contains room for strings to store after being coiled up by the motor, as well as providing additional mounting room for additional equipment like MPU-6050.

The following parts are responsible for the overall chassis integrity. There are multiple of the same pieces, and all of them are connected with joints and screws. The following figures will include the CAD of the chassis.

The top plate is the same as the bottom plate; they are the two biggest plates in the chassis; they hold the two arms at the back and the motors and act as a central connection for every other piece.

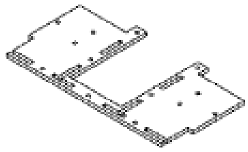

Figure 2.3 Top and Bottom Plate

The side plates connect the top and the bottom plate from the very outer side; they contribute greatly to the structural integrity of the chassis.

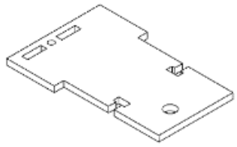

Figure 2.4 Side Plate

The elbow plate is the part connecting the top and bottom at the center. Besides adding additional connections to the over chassis, they also serve the function of securing the rubber arm at the two sides.

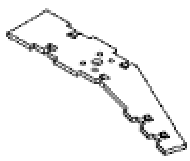

Figure 2.5 Elbow Plate

Finally, the back plates add the last piece needed for an extremely secure chassis. Besides, it also contains holds and other parts for wires to come through.

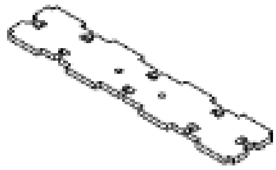

Figure 2.6 Back Plate

2.2 Version 2

2.2.1 Overall Structure Design – V2

The second version of the robot uses a 3D-printed outer shell instead of laser-printed plates. A 3D-printed outer shell offers two compartments where motors can be secured in place with screws instead of being held in place by tape. This is a much more stable solution and solves a lot of difficulties encountered in version 1, such as unstable components and heavy chassis. However, the general idea of the design remains similar.

2.2.2 Arm Design – V2

The arm design remains primarily unchanged for Version 2 of the robot as the Version 1 arm proves to be durable and is suitable for this task. However, it is created using a stiffer version of rubber, and the arm is now attached to the chassis a little bit differently.

Since every piece is Custom 3D printed, it needs screw holes and a support plate. Therefore, a piece is specially created for this purpose, which will be further discussed in the chassis design-v2 section.

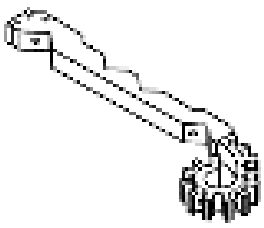

Figure 2.7 Arm Design - V2

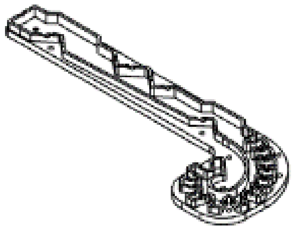

Figure 2.8 Arm Mold Design – V2

2.2.3 Chassis Design – V2

In designing the chassis for version 2, many changes must be adapted. First, in order to provide a sturdier structure, all parts are 3D printed, which offers much higher structural support and integrity. Secondly, the arrangement in version 1 of the robot is not very space efficient. In version 2, this problem is addressed by reorganizing the space and moving the motor from the inside to the outside (now they are placed within two additional compartments). Then the entire inside is occupied by other Arduino sensors and wires.

This version of the chassis includes two parts, motor holder piece and arm holder piece.

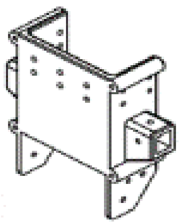

Figure 2.9 Motor Holder Piece

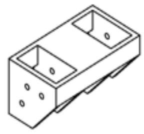

Figure 2.10 Arm Holder Piece

The Motor Holder Piece and Arm Holder Piece are connected with six screws. The following figure demonstrates the final assembly of the chassis.

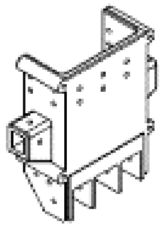

Figure 2.11 Final Chassis Assembly

After all, the chassis is strong for holding Arduino pieces and motors. It offered a lot of space for pieces and wires.

In conclusion, this design makes sure the robot has both strong structural integrity, as well as lighter weight.

2.3 Version 3

2.3.1 Overall Structural Design – V3

In the final version of the robot, it uses laser-printed pieces again as the laser-printed chassis, which is lighter than the 3D-printed chassis. In addition, five laser-cut pieces are screwed in tightly to each other to provide room for arm mounting, motors, and other components. This design saves much weight from the previous two versions and has excellent structural integrity.

The motherboard and some accessories are mounted outside of the chassis with strong screws and soldered cables. This guarantees that everything is well-connected and still offers excellent stability and integrity.

2.3.2 Arm Design – V3

This arm version uses a stiffer material to further improve upon the previous version. It is a 3D printable material instead of molded rubber, saving much weight and offering higher stiffness. However, this change also requires more motors to bend it properly. Besides, the designs of the joints on the arm are further refined so that it bends in order. Bending in order is extremely significant as the arm must grab the ladder before lifting itself upwards.

Another improvement in this design is the string that goes through the center of the arm. In addition, it now includes a soft rubber tube in all the opening of the arm, which help prevent the string from snapping.

During testing, it is shown that different joint designs could make a big difference as the arm design should allow the arm to bend in proper order. There are a few different arm designs; each bends in a slightly different order.

Figure 2.12 Final Arm

2.3.3 String Mechanism – V3

Since the motors generate torque, the string must be connected to the motor to be pulled back. In this project, a custom piece with a base cylinder and a small screw protruding out of it is used. Both ends of the cylinder are connected to two motors as they are responsible for one arm, and one end of the string is connected to the small screw protru ding out of the cylinder. This structure can pull string effectively and is a significant upgrade compared to the first two versions.

2.3.4 Servo and Weight Design – V3

In this version of the robot, the original weight is replaced with a battery, and the servo is mounted inside the chassis with an open hole on the surface. The battery has a holder and two wires connecting and powering the robot. The robot can be further improved with this design and save much weight again.

2.3.5 Chassis Design – V3

This new chassis is 3D printed 0.125-inch wood and assembled with screws. It is essential to adjust the laser printer and make sure that the holes designed on two different connecting pieces match each other's position. Besides, this chassis must accommodate the new servo and weight design, which requires a hole in the chassis. Therefore, each panel of the chassis component has many different connecting screw holes. The chassis comprises one top plate, two side plates, two middle plates, one back plate, one bottom plate, and two small plates.

The top plate has a big square opening for the servo to spin and connect the servo. This is a very important part of the robot's functionality. Besides, the top plate connects to both side plates and mounts the motherboard alongside several other components. In designing this plate, I used many joints and screws to ensure the side and top plates stayed well connected.

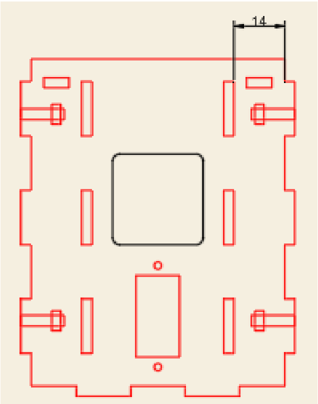

Figure 2.13 Top plate CAD

The side plates are also extremely important for the robot as motors are connected to it with plastic brackets and screws, and arms are connected to it with three screws. They make sure that the motors are well stabilized as well as the structural integrity of the entire robot.

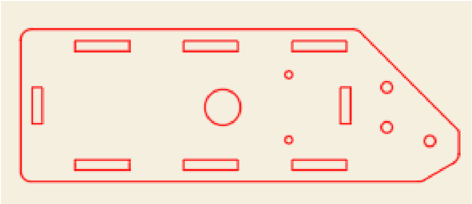

Figure 2.14 Side Plate CAD

The middle plates are connected to the top plate and bottom plate. It is used to mount the second motor for each arm. This piece needs to be connected tightly to the top plate, and bottom plate as the motor spinning can generate much torque, and the arm has screw holes connecting to it.

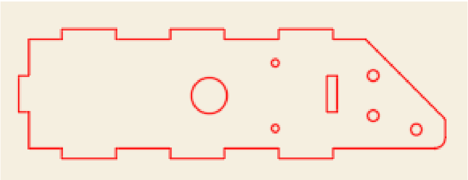

Figure 2.15 Middle Plate CAD

The Bottom plate is simply a piece connecting the side plates and middle plates. It is alongside the top plate; make sure the plywood won't twist internally and cause damage. It's a critical piece for structural integrity.

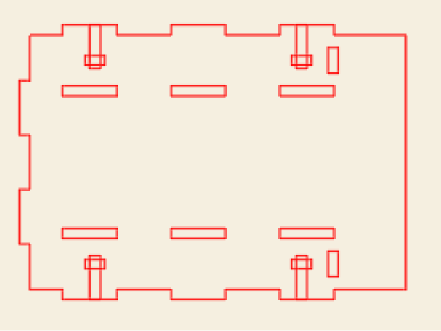

Figure 2.16 Bottom Plate

3 Software and Electronics

3.1 Programming Software: Arduino

Arduino is a company that designs and sells single-board microcontrollers and microcontroller kits for creating digital devices. It is also open source so that everyone can find design information online. Regarding hardware, every single Arduino board contains a different amount of flash drives, features, pinholes, and processing units. This ensures that there is an option for nearly every single project. Regarding software, Arduino has their own language based on C++ and its extended libraries. Also the Arduino community also takes advantage of the open-source infrastructure that Arduino uses to write their own libraries for extended hardware plugins and extensions.

The Arduino system allows simple control over motors, servos, and sensor reading values. These all facilitated the development of this project and improved its functionality. It is the overall best programming language and control for this project.

3.2 Version 1

3.2.1 The Overall Design – V1

This robot needs different components in order to function; besides, several key components must be present in the robot for crucial functionality. First, there is a mainboard that controls the entire robot. Besides, there are two motors inside the chassis connected to two strings that can bend the arms to climb. Finally, there is a weight connected to the servo at the very bottom of the robot, which can balance it when only one arm is grabbing onto the ladder. This is the first version of the robot, which only includes the barebone requirements of this robot.

3.2.2 The Arduino Mainboard – Arduino Uno R3 – V1

The Mainboard is the fundamental part of this robot. It compiles code and then runs it. It also accesses different parts of the robot and controls all of the parts of the robot. The first version of the robot uses Arduino Uno R3 as the main board. It is one of the best mainboards with excellent functionality, speed, and port availability. One drawback to this board is that the Arduino Uno R3 mainboard cannot run two things at the same time. However, it isn't very important since this robot doesn't need to run two things at the same time.

Figure 3.1 Arduino Uno R3 Mainboard

3.2.3 Motors and Arm Control – V1

This part of the robot utilizes motors that are relatively small, as well as having a big gear ratio that sacrifices speed. This is because they must be capable of pulling the entire robot upwards. Each of the motors is connected to the motor driver board, which is then connected to a 9V battery and a pin port on the Arduino mega pro mainboard. These motors complete their function well and are easily controllable with the library and motor driver board.

In controlling the arm, it utilizes the tension of a string that goes through the middle of the rubber arm and attaches it to the very end of the arm. Then, as the motors pull and tighten the string, the arm would bend and grab a bar on the ladder. After a while, when the string is released, the arm would recover to its natural position. However, due to the chassis design of this version of the robot, the string often snaps due to additional twists and turns.

These motors are carefully chosen to work well. It's equipped with a gear ratio aimed towards having higher torque, so it can pull the robot upwards with a consistent force.

The motors can't operate just by themselves. The motors must be connected to L298N first, then connect L298N to an external battery and a PWM port on the Arduino mainboard. The L298N acts as a bridge between the motor and the mainboard; it will transfer the command from the mainboard to the motors and power them using an external battery.

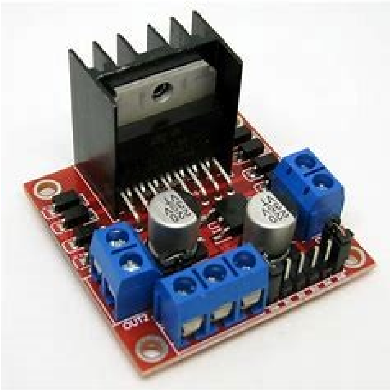

Figure 3.2 L298N

3.2.4 Angle Sensor – MPU6050 and Servo – V1

The previous section discussed to use of servos to correct the robot's tilt, but what if the robot cannot detect tilt? The robot will be completely oblivious to the tilt without an angle sensor. This is where MPU 6050 is used in this robot. It is attached close to the center of the chassis. When a tilt is detected, it will control the servo to rotate accordingly to correct the tilt. However, due to the restriction of Arduino mainboards, this feature cannot be turned on while climbing upwards. Thus, it is only used between moves during climbing, other steps are estimated, and the values are pre-defined rather than detected.

Figure 3.3 Angle Sensor – MPU6050

Then the angle feedback from MPU6050 will be mapped accordingly to the servo, which will control the weight to balance the robot. It can guarantee that the robot will not tilt significantly such that the arm cannot grab on as well as resist external influences.

Figure 3.4 Arduino Servo – SG90

3.2.5 Infrared Remote Control & Receiver – V1

The control of this system is a system of inferred remote control and its receiver. The inferred remote control is powered by a battery, and the receiver is connected to the Arduino mainboard. In order to use this inferred system, a library called “IRemote” must be installed on the Arduino IDE. It is the driving library for the inferred receiver and controller. It allows users to create inferred receivers and receive values from the controller. Based on the different returned values from different key presses, it is able to do different actions, such as climbing or moving the weight. This is exceptionally useful during testing and experimentation.

Figure 3.5 Arduino Infrared Module

Figure 3.6 Arduino Infrared Remote

The advantage of an infrared remote is the ability to remotely control the robot. For example, specific commands and actions for debugging and testing can be set to each of the buttons of the remote. However, infrared remotes suffer from poor connection and weak signals.

3.2.6 Arduino Program Logic– V1

The first version of the program is straightforward, and it simply involves climbing up the ladder and some adjustments with weights. Therefore, the program simply consists of a while loop and repeatable commands.

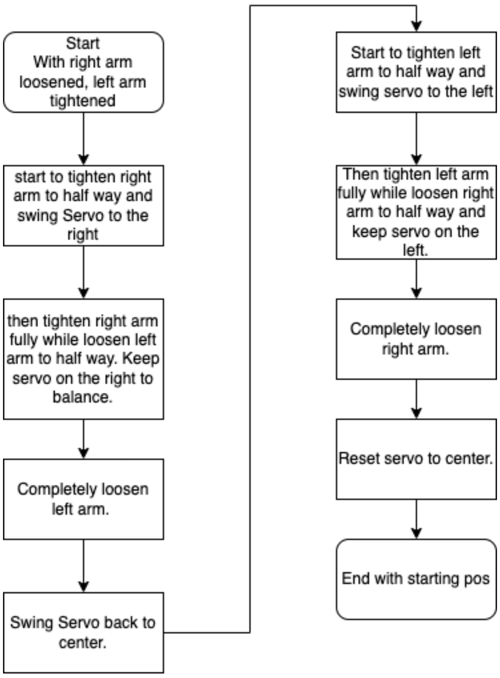

Figure 3.7 Arduino Program Logic

This program only climbs up once, and the program needs to be run multiple times for it to go up constantly.

3.3 Version 2

3.3.1 The Overall Design – V2

This design remained primarily unchanged from version 1, with the only few changes being the chassis, motor position, and mainboard change. Firstly, the chassis became 3D printed, which allows for exact positioning from every component as well as reducing weight from the previous generation. Secondly, the motor position has shifted to allow the string to be pulled easier and slip less often. Third, the mainboard has been changed to a smaller version of Arduino Uno R3 while still offering enough ports to connect all of the wires. Finally, the infrared module is discarded in favor of a Bluetooth module instead. This gives more control with the additions of the slider, toggle button, and much more.

3.3.2 The Arduino Mainboard - V2

In the previous version, it is shown that the mainboard is bigger than what is required heavy. This version of the robot uses a smaller mainboard which offers enough ports while also shrinking the size and weight to allow the robot to climb better.

3.3.3 Motors and Arm Control - V2

This version uses the same motor as the previous generation because of its proven functionality. Besides, the new chassis design adds more space for the motor to spin and a specific room on the side of the chassis to prevent the string from snapping.

The motors are still controlled via L298N since it offers enough power for the motors to run and the capability to control two at once.

3.3.4 Angle Sensor and Servo– V2

This version still uses the same MPU6050 sensor with the same filtering algorithm. It is proven to be accurate enough to control the bottom weight for balancing. Its position had been changed from the top of the robot to the center since the top of the robot was now fully occupied by the arm and its new attachment chassis. The Weight design and servo control remain unchanged and are still attached at the bottom of the robot.

3.3.5 Bluetooth Module – V2

This robot version uses a Bluetooth module instead of the infrared module. The Bluetooth module offers greater functionality, better control, and better connectivity. With an app called blinker (available on IOS and Android), the Bluetooth module will be connected to the device and then can be set up with different buttons, sliders, etc.

This feature is great for debugging and more specific robot controls as it offers an infinite option of buttons and controls.

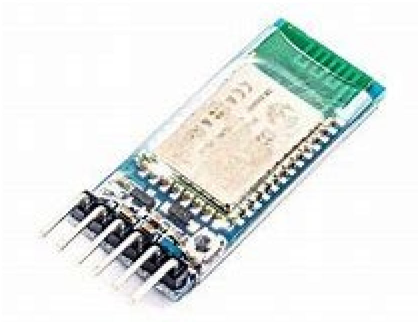

Figure 3.8 Arduino Bluetooth Module BT-18

Figure 3.9 Blinker App Page

3.3.6 Arduino Program Logic – V2

Since this robot is designed for climbing up ladders, and thus the programming is designed for it to be able to do it consistently. However, due to the drawbacks of Arduino, only one task is able to execute at the same time, which eliminates the possibility of actively adjusting the weight at the bottom with feedback from MPU6050. However, it is still functional as it can be used to adjust the robot's tilt when grabbing the second arm on the higher bar of the ladder. This is still a great use scenario for the weight.

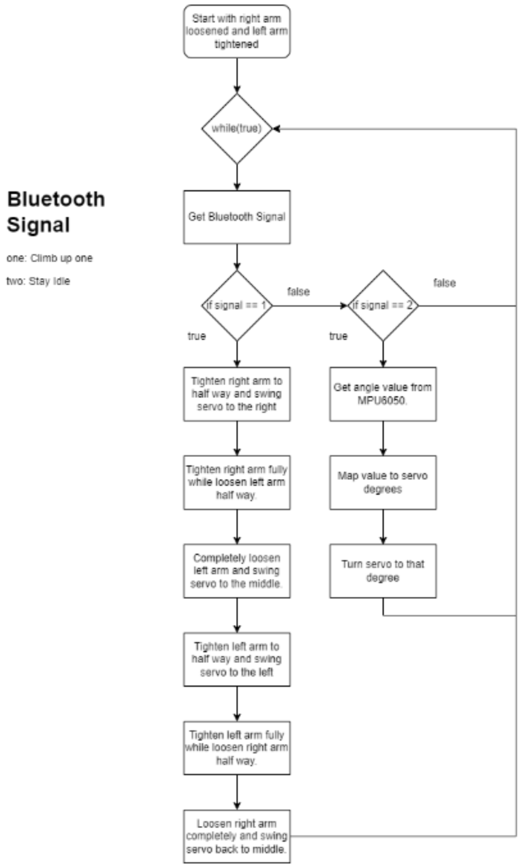

Figure 3.10 Overall Program Logic – V2

3.3.7 Final Product – V2

The following figure is the second version of the robot with the Arduino mainboard taken off; it still includes the arm, chassis assembly, motors, and other sensors. It still resembles many similarities to the first version, while the chassis and layout of the robot are different from the last generation.

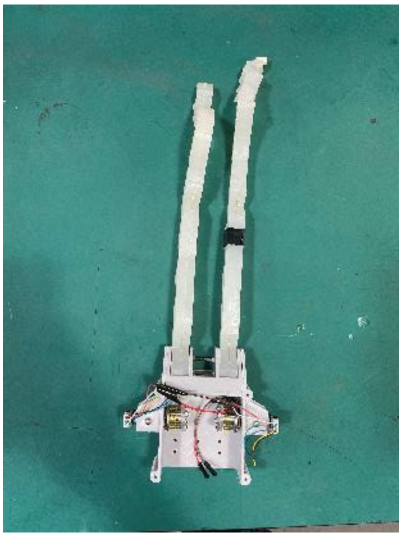

Figure 3.11 Overall Assembly Without Mainboard – V2

It's most effective at climbing ladders with some minor tweaks and changes mentioned in the sections above for this iteration of the robot.

3.4 Version 3

3.4.1 The Overall Design – V3

The final version of the robot returned to using laser cut wood to save weight and the main structural components are mounted outside of the case with screws and wires are soldered together. Due to some issues regarding arms in the previous two version, this now have a stronger and sturdier 3D-printed arm and is capable of bending. However, this arm now requires four motors to bend it properly. The motors are now mounted with a bracket that's connected to the case with screws. Besides, the original weight is replaced by a battery instead. The arm is also further changed to make sure that it bends in the proper order and with the right strength. There are also a lot of other minor changes in part choices and mount order, which will be further discussed in the following sections.

3.4.2 New Mainboard – esp32 – V3

This mainboard has all the required features while further reducing size and weight. The wires are now directly soldered to the mainboard and breadboard, which makes it extremely stable and reliable compared to previous versions.

Figure 3.12 ESP32 Mainboard

3.4.3 New Motor and Motor Control – V3

This version uses four motors instead of two due to the increased stiffness of the arm. Besides, the motors are now controlled with A4950 instead of L298N because of their decreased size and weight.

Figure 3.13 A4950 Motor Controller

3.4.4 Angle Sensor and Servo – V3

This version of the robot uses the same MPU6050 and servo as Version two.

3.4.5 Bluetooth Module – V3

This version of the robot uses the same Bluetooth module as Version two.

3.4.6 Arduino Program Logic – V3

This version of the robot has Similar functionality to the previous version. However, it now includes the ability for users to control robots. So the motors will spin when the users can control the three motors. The two-arm motors will be controlled with buttons. When the forward button is pressed, the respective arm motor will spin and retract the arm, making the robot ascend the ladder. When the stop button is pressed, the respective motor will stop. When the downward button is pressed, the motor will start to release the arm. Besides the arm motors, the servo that controls the weight is controlled via a slider, where the slider is initially at the center and can be adjusted by sliding left or right.

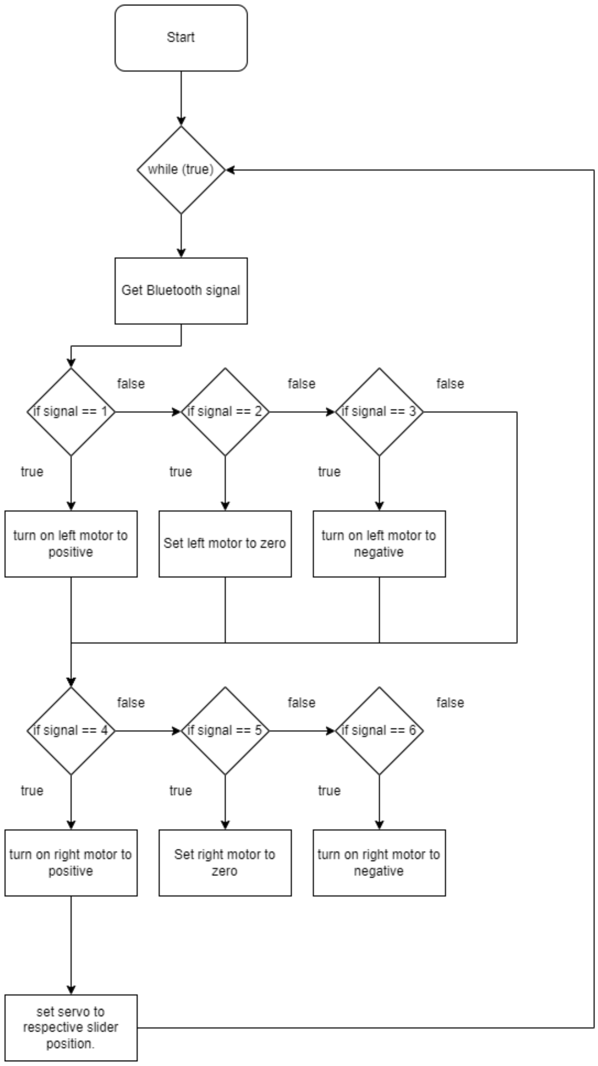

Figure 3.14 Overall Program Logic – V3

3.4.7 Final Product – V3

The following figure is the final robot with all the chassis, motors, and sensors.

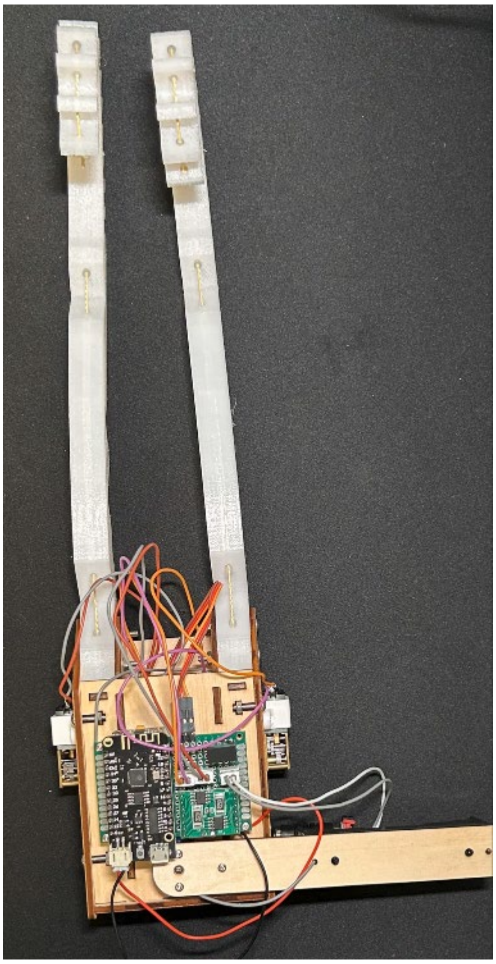

Figure 3.15 Overall Assembly– Version 3

As mentioned, it uses laser-cut plywood as the chassis and ESP 32 as the main board. It also used a new arm system that works incredibly better than previous versions. It is capable of climbing fast and pulling much weight. The appropriate sensors, motors, and programs allow the robot to function well as a unit and be extremely effective.

3.5 Features of the Final Robot

3.5.1 Soft Arm Shape Shift

This feature is crucial for this robot. The robot needs to be able to climb a ladder up consistently, which requires the arm to be able to bend predictably as well as in order so it can grab the ladder before bending upwards.

The arm is tested multiple times for each version to guarantee it will function as intended. Throughout the three versions, the arm is the part that received the most attention and upgrades as it is one of the most critical components of the entire robot.

The following diagram shows all the different arms considered for version three.

Figure 3.16 Different Arms Tested for Version 3

(Tip of the arm is joint zero and joint one to four from if from top to bottom)

From left to right, the arm becomes better and more suited for this robot. The arm has to be mostly redesigned due to the material change from the previous versions.

The first version of the arm that's tested for robot version three is very short. The shorter size saves a lot of weight from the robot. However, it's both hard to pull on, and it persists many issues that a long version of the arm doesn't.

From the first one's failure, the second design is created. This time, it uses the same longer design as previous generations. However, it encountered a problem where joint two start bending before joint zero was fully bent. This is a problem as the arm must first fully grab the ladder before other joints start bending and pull the robot upwards.

Version three of the arm is an attempt to solve the problem; it adds more material to each joint to prevent it from bending earlier. However, the same issue still persists on other joints as well.

In the fourth version of the arm, the design of joint zero is changed and to facilitate it to bend with a smaller force. With this change in design, some materials of the joint are removed. This version successfully removed any problem with the robot, and the joints were all bent in the correct order. However, with this new design, the arm requires 75 newtons of force to be able to pull it; it's higher when compared to the arms of the previous generations.

In the first stage, the arm is completely relaxed and in its natural shape.

Figure 3.17 Arm Four Bend Stage One

In the second stage, only joint one is bent, and other three joints are still mostly in their natural state.

Figure 3.18 Arm Four Bend Stage Two

In the third stage, joint two is fully bent while the other two are starting to bend.

Figure 3.19 Arm Four Bend Stage Three

In the fourth stage, the arm is fully bent, and it will lift the robot upwards.

Figure 3.20 Arm Four Bend Stage Four

Here is a table to describe the forces required to bend the abovementioned arm.

Table 3.1 Force Chart for Arm Four

Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Total Force
5 (N)	7 (N)	7.39 (N)	8.5 (N)	11 (N)	62 (N)

3.5.2 Automatic Balancing

During experimentation, it is shown that robots tend to tilt sideways when only one arm is connected to the ladder. This significantly impedes the next arm from being able to grab onto the ladder successfully. Therefore, the solution is to add an additional weight at the bottom and

Turn it with a servo with respect to the tilt angle. This method can correct the robot's tilt after experimentation.

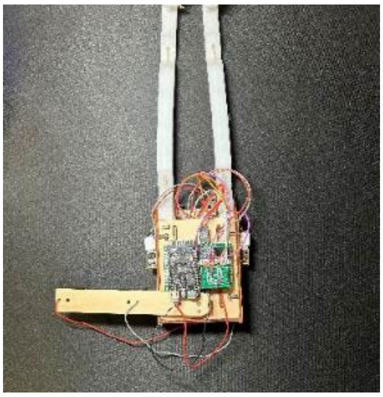

Figure 3.21 Weight Left Balance

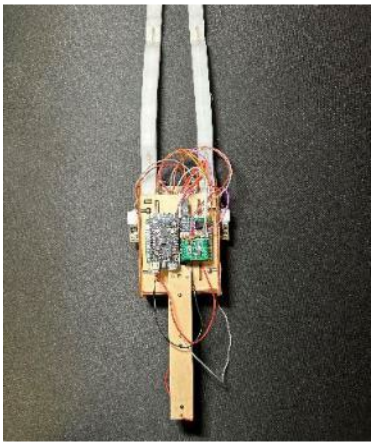

Figure 3.22 Weight Middle Balance

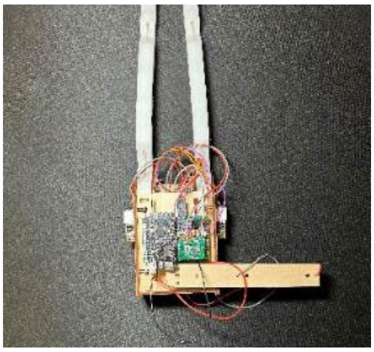

Figure 3.23 Weight Right Balance

3.5.3 Bluetooth Connectivity

This robot also features a Bluetooth module since version two, as version one includes an infrared remote and sensor. However, the infrared sensor is not very sensitive as well as the controller, which causes problems from time to time. Therefore, the robot shifted from using an infrared sensor and controller to using a Bluetooth module and an app on a device. (As for the app page, please refer to previous chapters) This Bluetooth connection is extremely stable throughout testing. This feature enables everyone to be able to control the robot when connected to it via Bluetooth in the future.

On the blinker app, there are many buttons and a slider that can control the robot's movement. It's accurate, has a relatively lower latency when compared to infrared controls, and has a more stable connection than infrared, as infrared connections can be blocked easily.

3.5.4 Programmable Actions

This robot is programmed with Arduino, which allows the user to make custom actions through the programming app. The robot currently supports custom weight control and arms control. This feature is essential when automatic climbing fails, as the users will be able to interrupt and use the custom controls to control the robot. Besides, the robot allows users to create automatic actions through Arduino programming software. For example, users will be able to program the robot to go upwards and downwards automatically, swing its weight and do much more.

4 Experimentations

4.1 Robot Arm Bend Force Testing

4.1.1 Bending In Correct Order

The arm in this robot must bend in order from joint zero to joint four. Otherwise, from observation, the arm may not grab to the next bar, which would result in total failure. If it bends in that specific order, then joint zero would first completely bend and grab onto the bar. Then, the arm will start to pull the robot upwards to the next level, and the same repeats with the second arm. If other joints start to bend before joint zero fully bends, the arm will not be able to grab onto the next bar.

The following experiments on arms are all prototypes for the third version of the robot.

4.1.2 Force Testing – 1

Table 4.1 Force Testing – Version 1

Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Total Force
5	7	7.5	8.5	11	75 (N)

From this graph, it's easy to tell that the joints will bend out of order as the force required to bend joint two is less than joint one. This would cause joint zero to disconnect from the arm.

Figure 4.1 Stage 1 – V1

Figure 4.2 Stage 2 – V1

Figure 4.3 Stage 3 – V1

From these three stages for the first version of the arm, it's clear to see that the bend sequence isn't great, and the arm will have chances where joint zero can fail to grab on to the bar. Besides, the joints at the bottom of the arm are bending simultaneously.

4.1.3 Force Testing – 2

Table 4.2 Force Testing – Version 2

Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Total Force
5	9	7.5	8.5	9	67 (N)

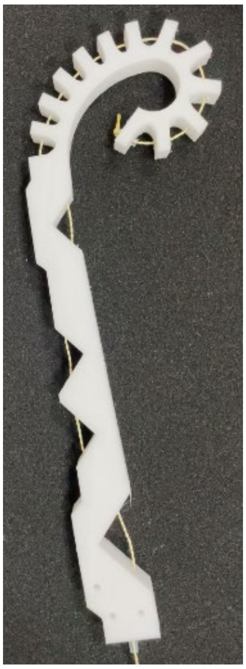

Figure 4.4 Stage 1 – V2

Figure 4.5 Stage 2 – V2

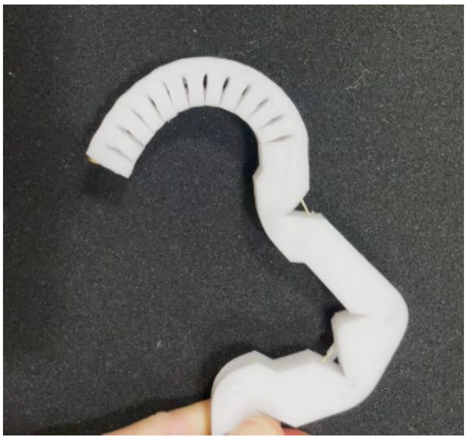

Figure 4.6 Stage 3 – V2

This Revised version of the arm performs much better and bends in roughly the correct order. However, the force required to fully bend this arm is too high at 67N. The motors may have issues bending these arms, and they can decrease in terms of the force needed to bend. Besides, the joints at the bottom of the arm still tend to bend before joint zero fully bend. These two issues can be solved simultaneously with a slight change in the joint designs.

4.1.4 Force Testing – 3

Table 4.3 Force Testing – Version 3

Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Total Force
5	7	7.39	8.5	11	62 (N)

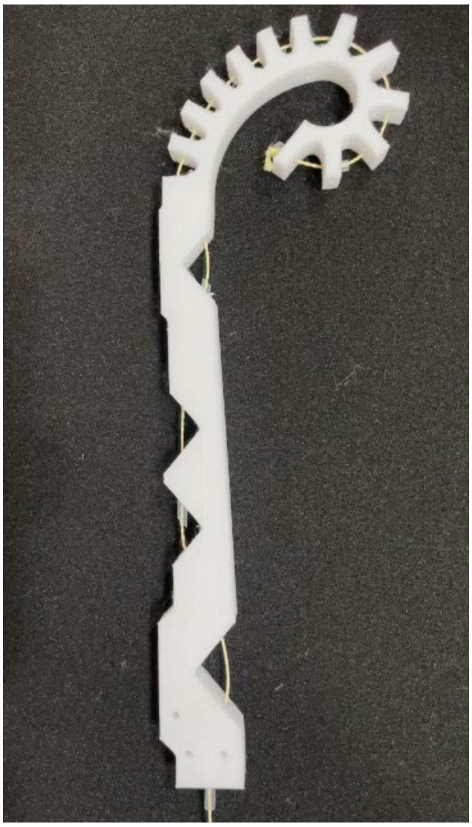
Figure 4.7 Stage 1 – V3


Figure 4.8 Stage 2 – V3

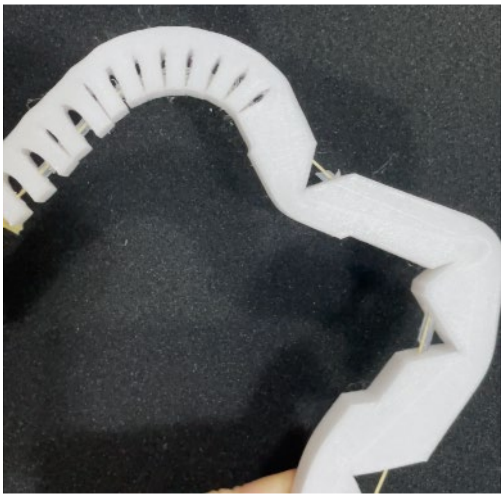

Figure 4.9 Stage 3 – V3

This is the third version of the arm. From the stages, it's easy to conclude that the arm bends in a better sequence, and the overall force required to bend the arm is further lowered to 62 newtons. However, joint zero doesn't fully close when other joints start to bend, so the force needed to fully bend joint zero need to be further lowered. The final version of the arm that's used in the robot is the arm showed in the section 3.5.1.

4.1.5 Results and Analysis

From previous results and improvements. It's certainly sure that the different ways that the joints are modified and the forces that are required to bend all five joints. Several joints are thickened, and some other parts of the weakened. When there are less material on a joint, making it thinner, the joint is easier to bend, and the bending sequence is just determined by the forces needed to bend these joints. Therefore, it important to organize all of the forces needed to bend these joints one by one. Thus, the force needed to bend joint zero should be the lowest and gradually increase from joint one to four, with an exception that joint two and three are the same.

5 Conclusion

5.1 Advantages

5.1.1 Advantage Against Traditional Metallic Robots

This robot uses certain malleable materials in the arms such that it can complete this task much more efficiently. For example, a traditional metallic robot needs many joints or specifically designed hooks to complete such tasks, while this robot only requires a few motors and a generic malleable arm. This is an obvious advantage against traditional robotics, and in the future, it could be further improved and utilized in many other areas.

5.2 Problems and Disadvantages

5.2.1 Limitations of Arduino

This limitation of the Arduino system caused trouble for me as I was using the delay and while loop to control it to climb up the ladder. However, in the meantime, I also used the angle sensor and a servo to maintain the weight on the bottom for it to be balanced. Arduino's lack of multi-threading means that if I were running the code that blinds up, then the balancing code wouldn't work.

Figure 5.1 Typical Arduino Mainboard

General Arduino mainboard lacks the capability to multithreading, meaning that at one time, only one process can run. This causes some significant issues as all of the programs have to be written to run within one process, which can cause delays and imperfection.

However, I'd found a solution to this issue. I've just set the servo position to a fixed angle while in the climbing code, and anywhere outside of that, it's going to be just like before. Besides, when the robot is controlled with Bluetooth, it will check all conditions in one process and have the same effect as multithreading.

5.2.2 Limitations of Hardware

The mainboard and many pieces are held in place by many screws outside of the robot, which is less ideal. The mainboard and these instruments should be contained within the robot rather than in the open. Although almost every wire is soldered and pieces are held together closely, it still lacks structural integrity, and motors still have a chance to fall off.

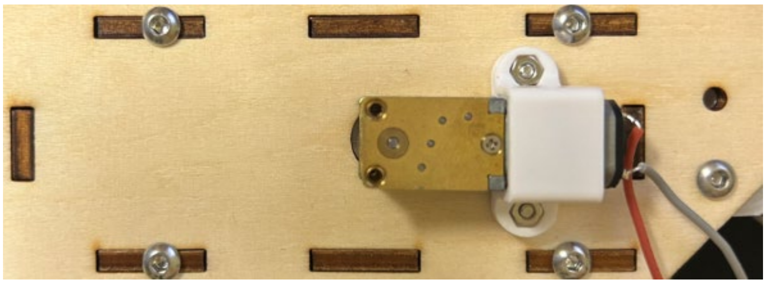

Figure 5.2 Side of the Final Robot

Besides, due to the size of the robot, stronger motors cannot be used, which limits the option to more motors and slower speeds for higher torque. The optimal solution is to increase the housing size and allow a bigger motor to fit within the robot. However, due to availability limitations, it wasn't an option to use stronger and faster motors.

5.2.3 Disadvantages

Despite the advantages of soft robotics, disadvantages are still very common in soft robotics. Firstly, at the level of this project, the arm can only perform about one function. For example, the current arm can only climb ladders well, and to achieve other functions, the arm needs to be redesigned for that specific purpose. Besides, soft robots aren't extremely accurate at the level of this project. Since the entire arm is controlled by a string, it's not extremely accurate. It has minor differences, while hard robots can be more accurate with specific motor controls at joints. However, as soft robotics advances, it can be fixed.

So far, the observed difference, about ±1 cm, isn't great enough to cause any issue; however, when it is applied to more specific domains that require higher accuracy, it might fail.

Figure 5.3 Arm and String

5.3 Final Completion Versus Original Plan

The original plan was to produce an arm to climb ladders and grab particular objects. The arm is then specially designed such that the arm is capable of such tasks. Joint zero is specifically designed to curl around an object to grab it. However, it is poor at grabbing due to the lack of friction.

On the other hand, joint zero is extremely capable of climbing ladders consistently. The orientation of this project is thus turned to climbing ladders.

5.4 Future Outlook

This project has vast potential in the future to develop. The malleable arm can be bent in different ways and can be further upgraded with advanced technology. This project can be turned into a tentacle to help astronauts in their future space mission and an extra arm for disabled people.

One such future implementation of this project is in outer space. One such soft arm can benefit astronauts in outer space while not in a spaceship. With some redesign of the first part, it can grab many things so that the astronauts can focus on their work. Besides, since the arm is malleable, it could be used to grab the astronauts so that they do not flow away too far. Also, it can help astronauts to go to places where they want to go at outside of the spaceship. Such advantages are achievable with soft robotics, while traditional robotics is more complicated.

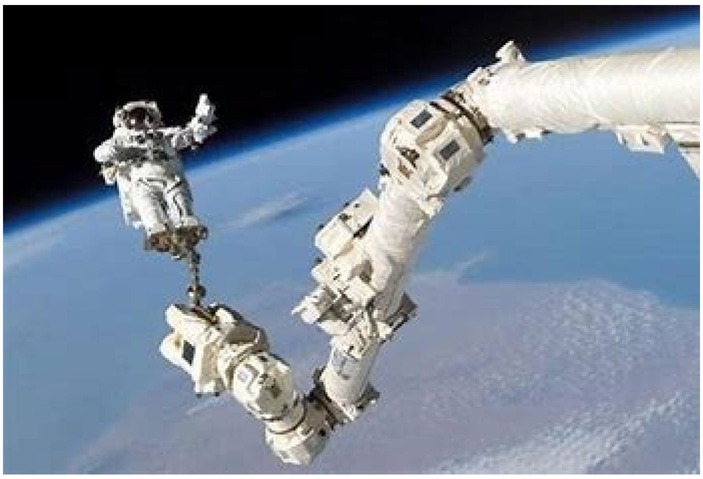

Figure 5.3 Current Stiff Space Arm with Joints

Soft robot arms, if implemented correctly in space, can be used for greater versatility and it capable of more effective movement and efficiency.

Another implementation is to help disabled people. For example, if such a project is further improved and advanced with more modern technology. It is possible to become an assistant arm for disabled people. For example, with a thought in mind, they can control this arm to grab onto different objects, and if a pair of arms, climb up a ladder with ease.

Figure 5.4 Current Artificial Arm with Joints

A new arm with soft material would provide more advantages than a traditional hard robotics arm since it is greater versatility and more controlling options. Hard robots need joints to be able to move around freely, while soft robots can move anywhere at any angle because of the nature of materials and design concepts.

Reference

[1] Margheri, L., Laschi, C., & Mazzolai, B. (2012). Soft robotic arm inspired by the octopus: I. From biological functions to artificial requirements. Bioinspiration & Biomimetics, 7(2), 025004. https://doi.org/10.1088/1748-3182/7/2/025004

[2] Mazzolai, B., Margheri, L., Cianchetti, M., Dario, P., & Laschi, C. (2012). Soft-robotic arm inspired by the octopus: II. From artificial requirements to innovative technological solutions. Bioinspiration & Biomimetics, 7(2), 025005. https://doi.org/10.1088/1748-3182/7/2/025005

[3] Iida, F., & Laschi, C. (2011). Soft Robotics: Challenges and Perspectives. Procedia Computer Science, 7, 99–102. https://doi.org/10.1016/j.procs.2011.12.030

[4] Cianchetti, M., Arienti, A., Follador, M., Mazzolai, B., Dario, P., & Laschi, C. (2011). Design concept and validation of a robotic arm inspired by the octopus. Materials Science and Engineering: C, 31(6), 1230–1239. https://doi.org/10.1016/j.msec.2010.12.004

[5] Calisti, M., Corucci, F., Arienti, A., & Laschi, C. (2015). Dynamics of underwater legged locomotion: modeling and experiments on an octopus-inspired robot. Bioinspiration & Biomimetics, 10(4), 046012. https://doi.org/10.1088/1748-3190/10/4/046012