# Robotic Octopus Arm

## **Table of Contents**

Project Overview	2
Mechanical Design & Geometry	
The Spiral Unit	
Actuation System	
Servo and Electronics Housing	
Final Assembly	6
Fabrication	7
3D Printing	7
Assembly	7
Electronics & Power	8
Control Software	9
Arduino Code (C++)	9
Challenges & Iterations	10
The Lifting Issue	10
Winch Slip & Torque Transfer	
Conclusion & Future Work	10

## **Project Overview**

This project involves the design, fabrication, and control of a cable-driven robotic arm biologically inspired by octopus tentacles. Unlike traditional rigid robots that use discrete joints, this arm utilizes a continuous backbone to achieve a smooth curling motion.

The goal was to create a low-cost, modular platform capable of high-articulation grasping using FDM 3D printing and hobbyist electronics. The system relies on a logarithmic spiral geometry, allowing the arm to curl tightly onto itself or wrap around objects of varying sizes.

This project helped me gain a deeper understanding of mechanical design and fabrication — 3D printing in this case — as well as mechanical and electrical integration to make a final product. The intended application of this arm is to grant the ability to grab various objects of complex and awkward geometries, which more conventional robotic arms might have trouble grasping.

## Mechanical Design & Geometry

### The Spiral Unit

The arm consists of a series of discrete tapered units. The geometry is derived from a logarithmic spiral equation, scaled down by a factor ( $\beta = 0.94$ ) for each subsequent unit. This taper ensures that when the cable is pulled, the arm curls from the tip (lowest stiffness) to the base (highest stiffness).

The core of the design is its ability to perform versatile grasping through curling. To achieve this, the arm is not a simple taper; it is a logarithmic spiral, defined by the polar equation:

$$\rho = ae^{\beta\theta}$$

Where  $\rho$  is the radius,  $\theta$  is the angle, and  $\beta$  controls the growth rate of the spiral.

By designing the discrete units to follow this curvature, the arm maintains a constant grasping angle regardless of the object's size. This allows the robot to wrap around a thin cable just as effectively as it wraps around a large cylinder, mimicking the biological versatility of an octopus tentacle.

To prevent unwanted twisting (torsion) of the different units during actuation, I designed the units with 2 notches and 2 groves (scaled appropriately) for each subsequent unit to fit into in order to counter the unwanted torsional force and keep all of the units facing the desired direction. This can be seen in the images below.

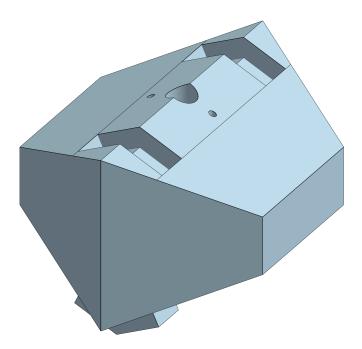


Figure 1: Isometric view of the base hexagonal unit. All subsequent units are exactly the same unit, but each scaled by a factor of 0.94 from the previous unit.

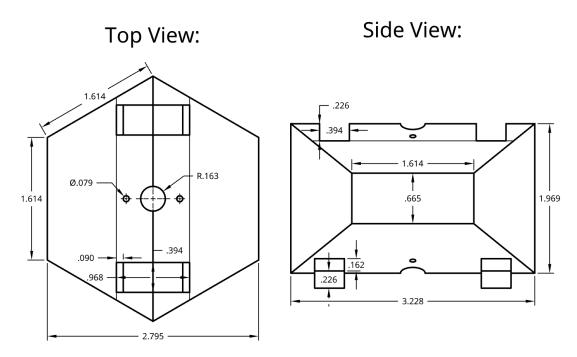


Figure 2: Dimensioned drawings of the base hexagonal unit from Figure 1. Note that all dimensions are in inches. Also note that the angle of the between the horizontal and the angled surface is 25°

The angle of 25° was chosen because of a bending angle of 50° total (the 25° of each adjacent unit adding up), this allows the arm to curl into a very tight spiral across 20 units (more than a full 360° loop), allowing it to wrap around small objects. If the angle were smaller the arm would

be stiffer but wouldn't curl tightly, and if the angle were larger, the units might become unstable or the gap would be too large for the cable to bridge smoothly.

#### **Actuation System**

The actuation is provided by a high-torque (55kg-cm), GX3255BLS continuous rotation servo. The servo drives a custom-designed winch spool that pulls a 0.8mm diameter UHMWPE braided line running through the perimeter of the spiral units. The cable is threaded through both sides of each unit so that each direction of servo rotation curls the arm in the direction of the rotation. A central elastic cable (5/16" OD) provides the restoring force, returning the arm to a straight position when the cable is released, as well as keeps all of the units aligned.

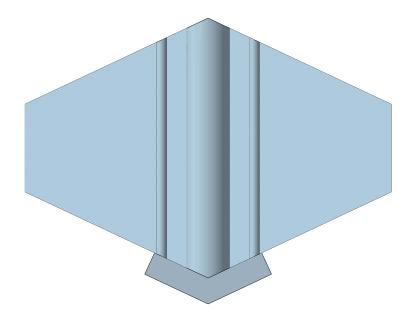
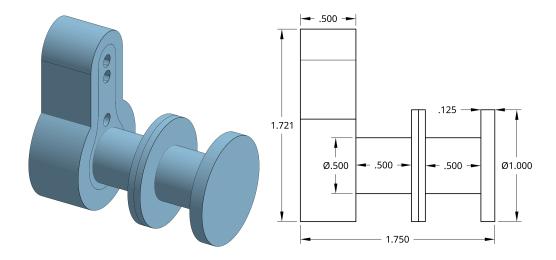


Figure 3: Cross section of base hexagonal unit from Figure 1 to show the actuation cable and inner tube holes.



*Figure 4:* Isometric view (left) and dimensioned drawing (right) of aforementioned winch. Note that all dimensions are in inches. Also note that the back of the winch is designed to pressfit into the servo horn but its dimensions are not relevant for this documentation, and are therefore omitted for clarity.

#### Servo and Electronics Housing

The wiring, servo, and arm were all organized and secured using a custom-designed housing. This housing features cutouts of hexagons on the sides, which serve two purposes: they allow for zip ties to secure the wiring externally, and they minimize filament usage during printing. The housing's dimensions and images are provided below:

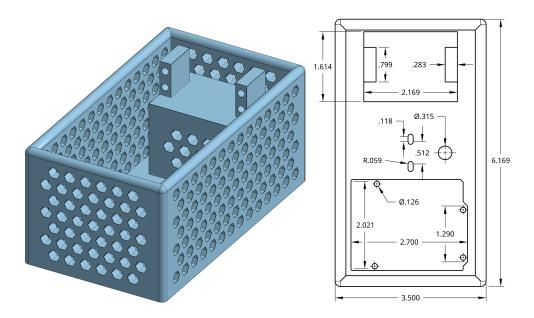


Figure 5: Isometric view (left) and dimensioned drawing (right) of aforementioned housing. Note that all dimensions are in inches.

## Final Assembly

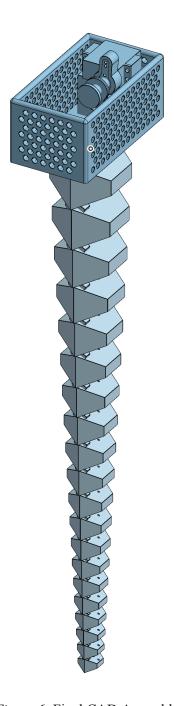


Figure 6: Final CAD Assembly

Link to Full Onshape Document

### **Fabrication**

#### 3D Printing

The spiral units and base housing were printed using PLA with 30% infill, which provided adequate strength for each of the units and the housing. The winch was printed using 50% infill PETG to provide the necessary extra strength and rigidity to withstand the forces of curling the arm and firmly holding objects. Images of the housing and the winch attached to the complete assembly are included in the Conclusion section.

The arm units and winch were printed on a Prusa MK 4S, and the housing was printed on a Prusa Mini.

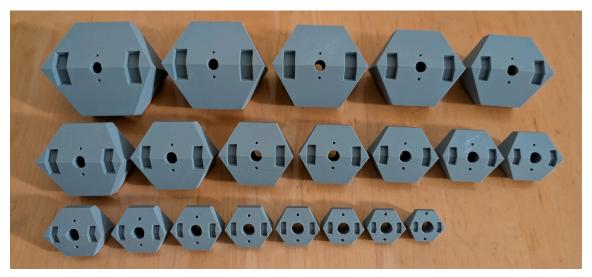


Figure 7: All 20 units all 3D printed in PLA

### Assembly

The assembly relies on a central elastic tube threaded through all units, capped at the tip and ziptied at the base to provide compressive force. This compression keeps the slots and extrusions of each unit interlocked with the next.



Figure 8: All 3D printed units assembled onto the center elastic tube

#### **Electronics & Power**

The control system is built around an Arduino R4 WiFi. Because the high-torque servo draws significant current (up to 5-7A under load), the system isolates the logic power from the motor power. The complete wiring and organization around the housing is shown in the <u>Conclusion</u> section.

- Microcontroller: Arduino R4 WiFi.
- Actuator: 55kg-cm Brushless Continuous Rotation Servo.
- Power: External 5V 15A power supply connected via a modified barrel jack adapter.
- Wiring: Common ground established between the external supply and the Arduino to ensure signal integrity.

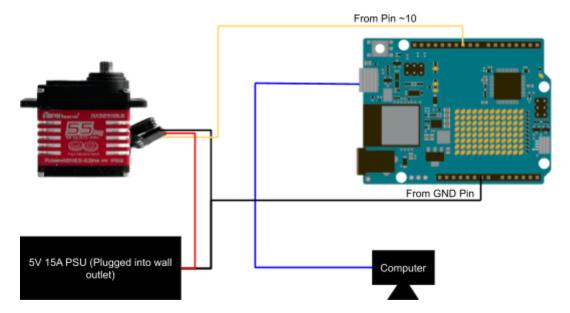


Figure 9: Wiring Schematic

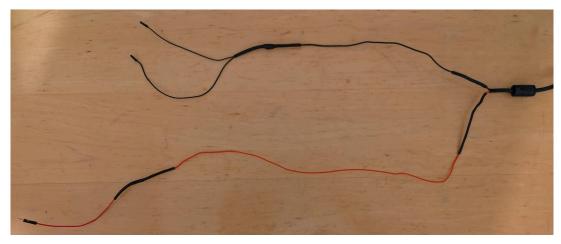


Figure 10: Modified barrel jack adapter. Note the 2 ground pins for the shared ground on the Arduino.

#### **Control Software**

The control logic utilizes the Arduino Servo library. Since a continuous rotation servo acts as a winch, "position" is determined by the duration of the spin rather than a specific angle command.

#### Arduino Code (C++)

#### Octopus Arm Project Code

The Arduino code listens for serial commands to control the motor state. Instead of mapping a joystick position directly to a servo angle (which is impossible with a continuous servo that lacks position feedback), the code implements a state machine with three primary states:

- 1. **Pull (Curl):** The servo spins forward at a slow speed to wind the cable; this can occur in any of the two directions of rotation for curling in 2 different directions.
- 2. **Release (Straighten):** The servo spins backward (relative to the direction it curled in) to straighten the cable.
- 3. **Hold (Stop):** The servo stops to hold the arm's position.

```
// Snippet: Basic Motor Control Logic
if (command == 'F') {
    // Curl: Spin slowly forward (pulling cable)
    motorA.writeMicroseconds(1550);
}
else if (command == 'B') {
    // Release: Spin slowly backward (releasing cable)
    motorA.writeMicroseconds(1450);
}
else if (command == 'S') {
    // Hold: Stop spinning to maintain current curl
    motorA.writeMicroseconds(1500);
}
```

Figure 11: Snippet of Arduino code that controls motor movement

A limitation of the standard Arduino Serial Monitor is that it requires the user to press the Enter key to send commands, which prevents smooth, real-time control. To solve this, a custom interface using Processing was used, a Java-based coding sketchpad.

This Processing script captures keyboard events in real-time. When the W key is held down, it continuously sends the Forward command to the Arduino, when the key is released, it automatically sends a Stop command. This prevents the winch from over-tightening and breaking the arm if the user lets go of the controls, allowing for more precise and safe control of the robot.

### **Challenges & Iterations**

#### The Lifting Issue

Initial tests revealed a significant issue where the winch would pull the entire arm upwards bodily, rather than curling it, due to the friction and weight distribution. A housing was designed to resolve this issue by holding all components in place relative to each other; blocking the arm from being pulled up and allowing it to curl instead. It is shown and discussed <u>further above in the Mechanical Design & Geometry section</u>. The housing is shown integrated with the rest of the arm in the <u>Conclusion section</u>.

### Winch Slip & Torque Transfer

The custom PLA winch was originally designed to press-fit onto the servo spline. However, the 55kg-cm torque proved too great for a plastic-on-metal press fit, causing the spool to slip and strip under maximum load. To resolve this without machining metal parts, a reinforced mounting solution using zip-ties was implemented to mechanically bind the PLA spool to the servo horn, preventing rotation slippage. This fix on the winch can be seen in the complete integration photos below in the Conclusion section.

### **Conclusion & Future Work**

The final prototype successfully demonstrates the bio-inspired curling motion capable of grasping objects. The hexagonal geometry effectively mitigated torsional twisting, and the high-torque actuation provided enough force to curl the full 20-unit assembly.

Improvements that could be made are making the inner tube less stiff, which would help the arm curl tighter (instead of leaving a small gap in the middle, as can be seen in the images below). Another improvement that can be made is designing the housing to be able to hold (and take the force of) the motor while curling the arm without needing zip ties.

Future work could be attaching a battery driven model to a drone to use for object retrieval, and adding another servo and altering the geometry to make it able to curl in three dimensions instead of just two.

Below is a folder with all images and videos of the final assembly working. Some images of the final assembly are included below that:

#### Octopus Project Final Assembly Photos and Videos



Figure 12: Final physical assembly.

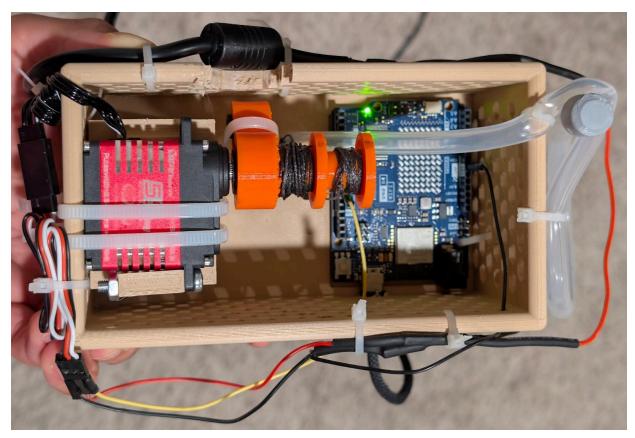


Figure 13: Housing Interior.

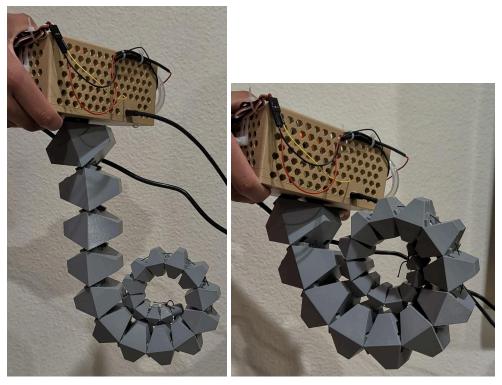


Figure 14: Curling arm (left) and arm fully curled (right)

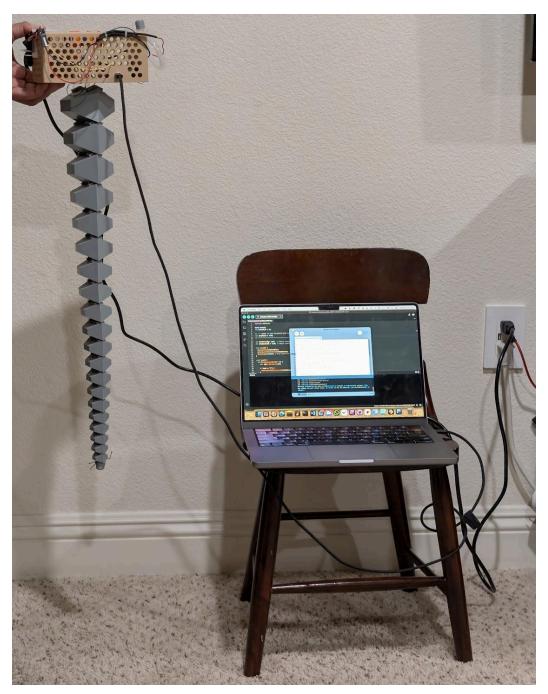


Figure 15: Full setup.