24-775: Bio-inspired robot design and Experimentation
Project Update #2
Team Aquabotics

Lindy Cornwall, Xu Mo, Yibo Chen, Yuanqing Chen, Yufeng Ren
April 22nd, 2022

1. Summary

Fish are able to maneuver quickly and easily in water by using their long, flexible tails. Our project aims to build on research on fish-like underwater robots by designing and building a robotic fish with a tail that uses three servo motors chained together. This robotic fish will have the ability to move forward and turn underwater. Our group will record the relevant data when the fish turn underwater and research on how changing the tail from flexible to rigid affects the robot's maneuverability, as judged by turning radius, angle turned, and turning speed.

2. Progress Overview

- The hypothesis has been changed, as noted in the last line of the summary
- Code design is done.
- Circuit design and assembly are done.
- The internal wiring box of the fish head (including various sensors and PCB boards) has been completed.
- We finished building a version of the soft shell of the fish body, but it has some defects (the tail part is a little broken), but we have finished the CAD design of the new model
- The Project Task Schedule is attached in Appendix A.

3. Detailed Progress

3.1 Code implementation and System Integration (Xu & Yuanqing)

All of the code work was done by two people, so what is described in this section is the joint work of both. All the codes required in the project are done and attached in Appendix B/

After testing all the sensors and assembling the SD card module, we faced a major problem, the flash memory/storage of a single Arduino uno board could not support so many modules running together due to the large storage requirement of the SD module. Therefore, the original design idea turned to the design of using two Arduino uno boards at the same time. The new design was successful, and we did not abandon any of the originally expected modules in the system. All modules cooperated with each other, and the integrated system was in good working condition. What follows is a description of the detailed modules.

- Servo module: For all of the fish tail's movements (swim forward, turn left, and turn right), we designed 5 different speed modes so the operator can use the remote control to perform different speed operations. We also recorded the servo to simulate the rigid body mode for testing purposes, by fixing some of the servos at the initial position to test the influence of each servo on the swimming of the fish.
- Humidity sensor module: We will use this module as a leak sensor. This module is placed inside the head, where the main electronics are. It has two levels of alarm thresholds. Under normal conditions, the relative humidity detected by the DHT humidity sensor is from 26% to 33%. When the humidity in the fish reaches 40%, the first alarm is performed by the LED blinking four times, but the system continues to operate itself. When the humidity reaches 50%, the second alarm is performed and all actions of the robot are stopped. The alarm will be indicated in a different blinking pattern by the LED on the antenna.
- Remote control module: This module allows us to use the remote control to control all servo behavior, including the aforementioned variable speed and shutdown of some servos. The remote control can also control the data reading and writing of the gyroscope and the accelerometer. The remote control module makes it possible to control the Arduino while the robot is under water.
- RGB LED module: This will be implemented on the antenna and work as the signal module for the robot. When the robot is under water, it is difficult to observe the precise situation of the system, so the LED blinks in different patterns to show the user the current condition of the fish. When the robot is in the initialization mode at first, the magenta light will be on, and when the magenta light turns off, the user can start to operate the system with the remote controller. Every time we change the speed mode successfully with the remote controller, the LED will blink twice in blue. When the robot is moving forward, the blue light will be always on, and when the robot is turning left, the red light will be always on, and when the robot is turning right, the magenta light will always be on. When the first humidity warning occurs, the LED will blink four times in different colors and when the second humidity warning occurs (the relative humidity reaches 50%), the LED will blink continuously in different colors.
- SD card module: Each individual test generates a separate text record of gyroscope data, and that data will be stored in the SD card with names in a certain pattern. This data will be used later for data analysis. The gyroscope data is helpful for use to estimate how stable the robot moves under water.
- Arduino: We split all the functional modules on the A board and the B board. Even though there are two boards in use in this project, we will only use one battery to power the whole system. The A board accepts the remote signal and then controls the RGB performance signal, controls the position of the servo, and monitors the humidity sensor at any time. Board B is responsible for all data collection and recording

Lindy took on the task of design and manufacturing of the final circuit on a perf board.

3.2 Fish body manufacturing (Yibo)

According to the update 1 report, the model we designed and molded was found to be too inflexible at its thickest parts and too weak at its thinnest, so was redesigned.

Figure 1: Old version of the mold model (left) and the soft shell (right)

Our old design is shown above. Due to the limitation of the size of the printer, our inner and outer molds were 3D printed in two parts and then combined together. Due to the roughness of the 3D printer, some of the silicone leaked into the bottom of the printed mold when we poured it (as shown in the picture in Appendix C, the black part is the printed mold). This, combined with the drastic changes in thicknesses of the walls, led to a slight breakage of the silicone shell after we removed the mold (as shown in the picture above).

To avoid manufacturing difficulties caused by such curved inner walls, Lindy and Yufeng assisted with redesign of the fish's shell into a long conical table shape.

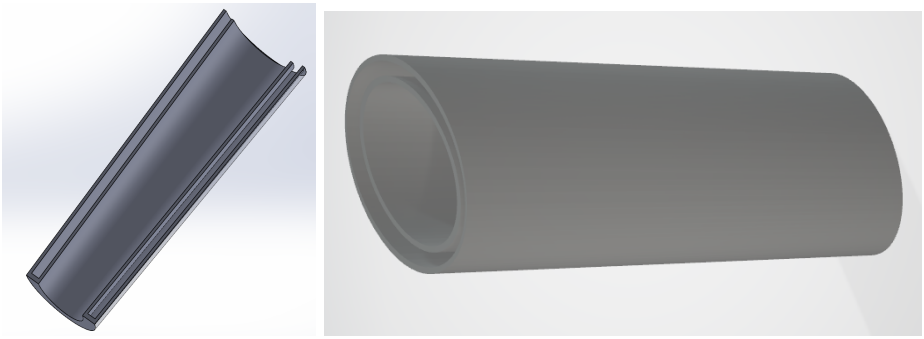

Figure 2. New CAD model of the new mold (left) and new tail shape (right)

We designed the shell this time as a front and back through shape, the front part connected to the head structure, the back part connected to a rigid lasercut tail. Therefore, this time the silicone pouring model is one mold instead of the two parts inside and outside of the previous design, which also greatly reduces the manufacturing difficulty.

3.3 O-Ring Sealing (Yufeng)

The o-ring design is complete and a prototype is built and tested for the water-proof. The o-ring is capable of keeping the water out and letting the inside stay dry. However, due to the design changes of internal electronics, the size of the hole was found to be too small. A new design with similar design is designed and ready to manufacture. The new design will have a rectangular hole with a rectangular o-ring groove so that it is large enough to allow all of the electronics to be removed through.

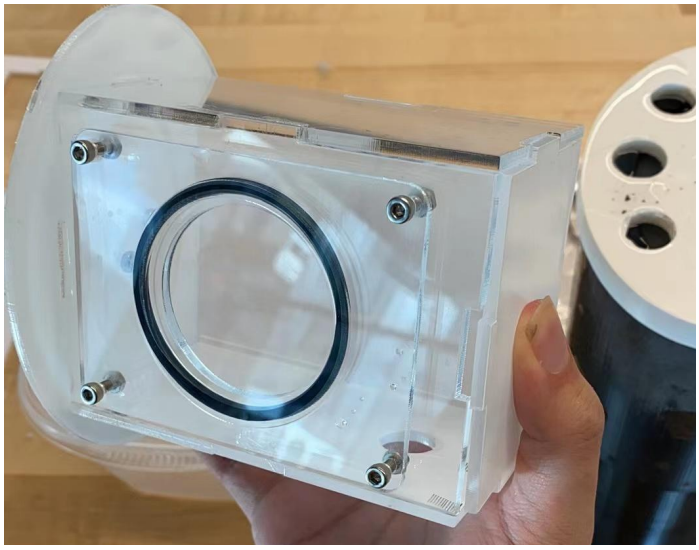

Figure 3: Waterproof box containing all electronics in the head

4. Obstacles Encountered

The main thing slowing down our progress so far is the manufacturing process of soft fish shells, as stated in the detailed progress section. After several failures we adopted a relatively simple shell shape to minimize the inconvenience of the 3D printer's performance. We expect to finish the new fish shell by this Monday morning (the 3D printing molds are lined up to start printing on 4/23, the silicone will be filled on 4/24, and the molds will be disassembled on 4/25 morning)

5. Teamwork

Our team has had no unproductive conflict up to this point. Everyone devotes and enjoys this project.

Appendix A: Current Task Schedule

	A	B	C	D	E
1	**Task Name**	**Duration**	**Start**	**Finish**	**% Complete**
2	Group Meeting & Review 1	0	02/12/22	02/12/22	100
3	Group Meeting & Review 2	0	02/16/22	02/16/22	100
4	Purchasing	6d	02/17/22	02/24/22	100
5	Design	12d	02/17/22	03/04/22	100
6	group proposal	0	02/18/22	02/18/22	100
7	matlab	16d	02/26/22	03/18/22	100
8	3D printing	21d	02/28/22	03/28/22	90
9	arduino	10d	03/18/22	03/31/22	100
10	project update 1	0	04/05/22	04/05/22	100
11	assembly	6d	03/31/22	04/07/22	70
12	test	17d	04/07/22	04/29/22	50
13	project update 2	0	04/22/22	04/22/22	100
14	Final project document	0	05/03/22	05/03/22	0
15	Project video	0	05/05/22	05/05/22	0

Appendix B: Code for the Project

A board:

```
#include <Servo.h>
#include <IRremote.h>
#include "DHT.h"
#define DHTPIN 3     // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);
//setup for remote controller
int IRpin=6;
IRrecv IR(IRpin);
String myCom;
int IRdelay=1000;
int disable_servo1 = 1;
int disable_servo12 = 1;
//setup for RGB led
int red_light_pin= 2;
int green_light_pin = 3; //useless
```

```
int blue_light_pin = 4;
//setup for three servos
float pos1=42;
float pos2=100;
float pos3=96;
int servoPin1=9;
int servoPin2=8;
int servoPin3=7;
int preDelay=10000;
// for forward
int servoDelay=10;
float servoAddTime=0.02;
float ratio1=2./3.;
float ratio2=65./90.*0.9;
float ratio3=64./90.*0.7;
// for turning
float k_turn=1.5;
int servoDelay_t=8;
float servoAddTime_t=0.02;
float ratio1_t=2./3.;
float ratio2_t=65./90.*0.9;
float ratio3_t=64./90.*0.9;
int RGBdelay=200;

float pi=3.14;
float t=0;

Servo Servo1;
Servo Servo2;
Servo Servo3;
bool Warned = 0;
float servoData[5][11]={
      {10,0.02,1.0,0.9,0.7,1.5,8,0.02,1.0,0.9,0.9},
      {25,0.03,1.0,0.7,0.5,1,25,0.03,1.0,0.6,0.6},
      {20,0.025,1.0,0.7,0.5,1.1,20,0.025,1.0,0.7,0.7},
      {15,0.02,1.0,0.8,0.6,1.2,15,0.02,1.0,0.8,0.8},
      {12,0.02,1.0,0.8,0.6,1.3,15,0.02,1.0,0.8,0.8},
      };


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  pinMode(red_light_pin, OUTPUT);
```

```
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);

  IR.enableIRIn();
  dht.begin();
  Servo1.attach(servoPin1);
  Servo2.attach(servoPin2);
  Servo3.attach(servoPin3);
  delay(preDelay);
  Serial.println("initinization done");
  RGB_color(255,255,255);
  delay(25000);
  RGB_color(0,0,0);
}

void loop(){
  RGB_color(0, 0, 0);
  if (IR.decode()){
    IR.resume();
    //Serial.println(IR.decodedIRData.decodedRawData,HEX);
    delay(IRdelay);

    if(IR.decodedIRData.decodedRawData==3175284480) // num 7
    {
      disable_servo1 = 0;
      RGB_color(255,0,0);
      delay(100);
      RGB_color(0,0,0);
      delay(100);
      RGB_color(255,0,0);
      delay(100);
      RGB_color(0,0,0);
      delay(100);
      RGB_color(255,0,0);
      delay(500);
      RGB_color(0,0,0);
    }
    if(IR.decodedIRData.decodedRawData==2907897600) // num 8
    {
      disable_servo12 = 0;
      RGB_color(255,0,0);
      delay(500);
      RGB_color(0,0,0);
      delay(100);
      RGB_color(255,0,0);
```

```
  delay(100);
  RGB_color(0,0,0);
  delay(100);
  RGB_color(255,0,0);
  delay(100);
  RGB_color(0,0,0);
}
if(IR.decodedIRData.decodedRawData==3041591040) // num 9
{
  disable_servo1 = 1;
  disable_servo12 = 1;
  RGB_color(255,0,0);
  delay(200);
  RGB_color(0,0,0);
  delay(100);
  RGB_color(0,0,255);
  delay(200);
  RGB_color(0,0,0);
  delay(100);
  RGB_color(255,0,255);
  delay(200);
  RGB_color(0,0,0);
}
int sc=-1;
if(IR.decodedIRData.decodedRawData==4077715200)sc=1;
if(IR.decodedIRData.decodedRawData==3877175040)sc=2;
if(IR.decodedIRData.decodedRawData==2707357440)sc=3;
if(IR.decodedIRData.decodedRawData==4144561920)sc=4;
if(IR.decodedIRData.decodedRawData==3910598400)sc=0;
if (IR.decodedIRData.decodedRawData==3125149440)
{
            // Stop = 1;
            RGB_color(255,0,0);
            delay(200);
            RGB_color(0,0,0);
            delay(200);
            RGB_color(255,0,0);
            delay(200);
            RGB_color(0,0,0);
            delay(200);
            RGB_color(255,0,0);
            delay(200);
            RGB_color(0,0,0);
            delay(200);
            Serial.println("close file");
```

```
    }

    if(sc!=-1)
    {
      // for forward
      servoDelay=servoData[sc][0];
      servoAddTime=servoData[sc][1];

      ratio1=2./3.*servoData[sc][2];
      ratio2=65./90.*servoData[sc][3];
      ratio3=64./90.*servoData[sc][4];

      // for turning
      k_turn=servoData[sc][5];
      servoDelay_t=servoData[sc][6];
      servoAddTime_t=servoData[sc][7];

      ratio1_t=2./3.*servoData[sc][8];
      ratio2_t=65./90.*servoData[sc][9];
      ratio3_t=64./90.*servoData[sc][10];

      Serial.print("Change mode: speed ");
      Serial.println(sc);
      speed_light();
    }
    bool Stop = 0;
    int mc = -1;
    String Com[3] = {"Go Forward","Turn Left","Turn Right"};
    if(IR.decodedIRData.decodedRawData==3141861120) mc=1; //left
    if(IR.decodedIRData.decodedRawData==3158572800) mc=2; //right
    if(IR.decodedIRData.decodedRawData==3208707840) mc=0; //forward

    if(mc!=-1)
    {
        myCom = Com[mc];
        //myFile.println("L");
        Serial.println(myCom);
        while(1)
        {
            if(Stop==1)
            {
              if((mc==1 or mc==2) and abs(pos3-96)<5 and
abs(pos2-100)<10)
              break;
```

```
            if(mc==0 and (abs(pos2-100)<5))
            break;
          }
          turn(mc);
          float h = dht.readHumidity();
          if(h>40 and h<50 and Warned == 0)
          {
              RGB_color(255,0,0);
              delay(200);
              RGB_color(0,0,0);
              delay(200);
              RGB_color(255,0,255);
              delay(200);
              RGB_color(255,0,0);
              delay(200);
              RGB_color(0,0,0);
              delay(200);
              RGB_color(255,0,255);
              delay(200);
              Warned = 1;
          }

          while(h>50)
          {
              RGB_color(0,0,0);
              RGB_color(255,0,0);
              delay(200);
              RGB_color(0,0,0);
              RGB_color(255,0,255);
              delay(200);
              RGB_color(0,0,0);
              RGB_color(0,0,255);
              delay(200);
          }
          if (IR.decode() ) {
            IR.resume();
            if (IR.decodedIRData.decodedRawData==3927310080) {
              Stop = 1;
              Serial.println("stop");
            }
           }
      }
  }
  }
delay(IRdelay);
```

```
}

void turn(int dir)
{
  int tur;
  if(dir==1)tur=1; else tur=-1;
  if(dir==1 or dir==2) //left
  {
    pos1=tur *
atan(k_turn*(11*sin(2*pi/9+pi*t)/50+67./300.))/pi*180*ratio1_t*disabl
e_servo1*disable_servo12+42;
    pos2=tur *
atan(k_turn*(22*sin(4*pi/9+pi*t)/25+67./75.))/pi*180*ratio2_t*disable
_servo12+100;
    pos3=tur *
atan(k_turn*(99*sin(2*pi/3+pi*t)/50+201./100.))/pi*180*ratio3_t+96;
  }
  if(dir==0) //forward
  {

pos1=atan(2*sin(2*pi/9+pi*t)/3)/pi*180*ratio1*disable_servo1*disable_
servo12+42;

pos2=atan(8*sin(4*pi/9+pi*t)/3)/pi*180*ratio2*disable_servo12+100;
    pos3=atan(6*sin(2*pi/3+pi*t))/pi*180*ratio3+96;
  }
  Servo1.write(pos1);
  Servo2.write(pos2);
  Servo3.write(pos3);
  int RGB[3][2] = {{0,255},{255,0},{255,255}};
  RGB_color(RGB[dir][0],0,RGB[dir][1]);
  delay(servoDelay_t);
  t=t+servoAddTime_t;
}


void RGB_color(int red_light_value, int green_light_value, int
blue_light_value)
 {
  digitalWrite(red_light_pin, red_light_value);
  digitalWrite(green_light_pin, green_light_value);
  digitalWrite(blue_light_pin, blue_light_value);
}
```

```
void speed_light(){
    RGB_color(0, 0, 255);
    delay(RGBdelay);
    RGB_color(0, 0, 0);
    delay(RGBdelay);
    RGB_color(0, 0, 255);
    delay(RGBdelay);
    RGB_color(0, 0, 0);
    delay(RGBdelay);
}
```

B board:

```
#include <TinyMPU6050.h>
#include <SPI.h>
#include <SD.h>
#include <IRremote.h>
File myFile;
MPU6050 mpu (Wire);
#define IRpin 6
IRrecv IR(IRpin);
long randNumber;
String filename;

void setup()
{
  Serial.begin(9600);
  randomSeed(analogRead(0));
  SD.begin(10);
  randNumber = random(3000);
  filename=String(randNumber)+".txt";
  Serial.println(filename);
  myFile = SD.open(filename, FILE_WRITE);
  mpu.Initialize();
  mpu.Calibrate();
  IR.enableIRIn();
  delay(1000);
  Serial.println('d');
}

void loop()
{
  if (IR.decode())
  {
    IR.resume();
```

```
    bool Stop=0;
    if (IR.decodedIRData.decodedRawData==3141861120 or
IR.decodedIRData.decodedRawData==3208707840 or
IR.decodedIRData.decodedRawData==3158572800){//left forw right
      Serial.println('j');
      if (IR.decodedIRData.decodedRawData==3141861120){
      myFile.println('l');
      }
      if (IR.decodedIRData.decodedRawData==3208707840){
      myFile.println('f');
      }
      if (IR.decodedIRData.decodedRawData==3158572800){
      myFile.println('r');
      }

      while(1)
      {
        mpu.Execute();
        delay(1000);
        if (IR.decode() ) {
          IR.resume();
          if (IR.decodedIRData.decodedRawData==3927310080) {
          Stop = 1;
          Serial.println('o');
          break;
          }
        }
          myFile.println("AX   AY   AZ   GX   GY   GZ");
          myFile.print(mpu.GetAccX());
          myFile.print(" ");
          myFile.print(mpu.GetAccY());
          myFile.print(" ");
          myFile.print(mpu.GetAccZ());
          myFile.print(" ");
          myFile.print(mpu.GetGyroX());
          myFile.print(" ");
          myFile.print(mpu.GetGyroY());
          myFile.print(" ");
          myFile.println(mpu.GetGyroZ());
          myFile.println(" ");

      }
      }
      if (IR.decodedIRData.decodedRawData==3125149440) {
        myFile.close();
```

```
          Serial.println('c');
          randNumber = randNumber+1;
          filename=String(randNumber)+".txt";
          Serial.println(filename);
          myFile = SD.open(filename, FILE_WRITE);

        }
    }
}
```

Appendix C: Fish Body

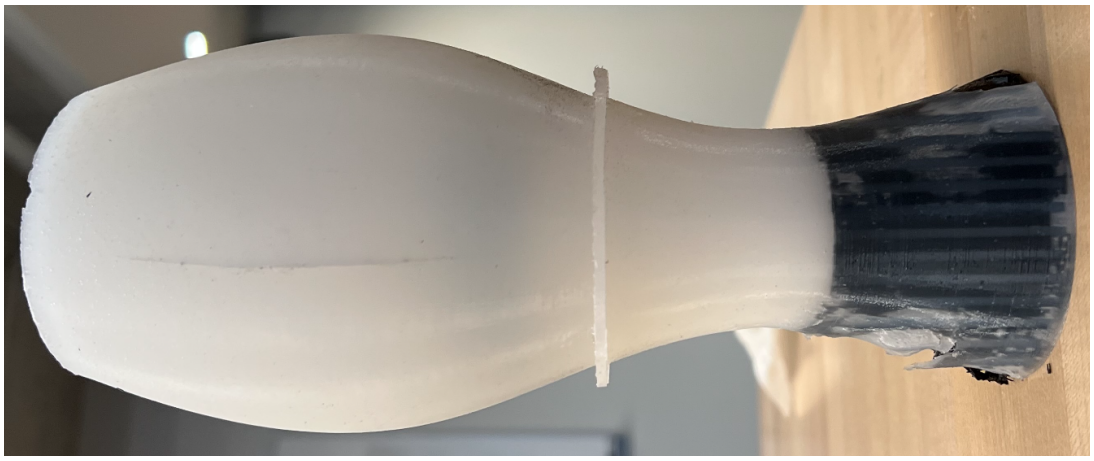

Figure 4: Failed silicone mold