Snake-Mic

Tarun Thathvik
thathvik@nyu.edu

Smrithi Thudi
srt381@nyu.edu

Vedant Desai
vbd223@nyu.edu

New York University- Tandon School of Engineering

ABSTRACT

To alleviate a theatrical experience, a mic that can crawl around like a snake and speak when intended was created. In order to maintain the quality of the sound, a Bose SoundLink Micro Speaker was fitted into a mic-shaped chassis. With the help of an Arduino and a Bluetooth module, this Snake-Mic can be controlled using a user friendly Android App that was developed.

Contents

List of Figures

List of Tables

1 Introduction

An integral part of human entertainment for as long as a man can remember, Theatre has lately been a dying industry, only appreciated by true enthusiasts. In the attempt to take an audience into its world, movies today have progressed to have amongst the best visual and audio effects that imitate theatre, to the best possible extent.

Theatre and movie industries have always incorporated technology to improve the scope of entertainment. One such implementation is the use of Drones to shoot various scenes in movies to create a dramatic effect without interfering with the environment. This gave scope to implement shots that were never thought possible, like performing a transition from ground level to heights physically challenging or chasing cars at high speeds from difficult angles. Primitive technology of a flying drone was first introduced during the first world war, and commercially available for recreational use in 2010, a year after making it legal to commercialize drones. Shortly after this, in 2012, amongst the first movies to use this Drone technology to shoot a chase scene was a James Bond film Skyfall.

As an attempt to innovate in the theatre world, Professor Alejandro Moreno Jashes tried to recreate stage theatrics as performed by several late artists. To actualize this, he used a robotic actuated Mic stand to imitate the performer's movements and patched the audio from the archives from before the death of the performers. The effect of a speaker speaking at the mic creates the inception of a ghost-like presence in this theatrical, rather than just an auditorium with a sound system. This was also made possible by incorporating a speaker into the tiny mic to speak in sync with the performance.

Now Professor Alejandro Moreno Jashes wants to implement a more challenging project to make a mic that can move around in an art gallery responding to people's presence around the mic. As simple as it sounds, it needs a lot of effort to implement this. Technologically, this is no innovation, but the real struggle is to implement a compact model of a simple mobile robot with a speaker in the structure of a mic.

2 Problem Statement

Designing, materializing and controlling a compact mobile robot that can play desired sounds from a speaker by Bose, while still fitting into a mic like structure. This Mic-like structure is supposed to be able to move around in an auditorium during a desired event with less than heavy tail of a wire, in a snake-like fashion.

3 Implementation

3.1 Design Phase

Since the model has to fit a Bose speaker with all its components inside while keeping a compact and mic-like structure, it introduced a lot of unexpected difficulties to the design phase.

3.1.1 Concept generation

Since the problem statement given was particularly clear about the end product, we had to now idealize the possible ways to implement this Snake-Mic

One of the several ideas to make the robot move was to actuate a mouse like structure, as shown in the Fig(1).

Figure 1: Mouse ball actuation

After a lot of discussion, and a lot of other more complicated ways to implement the movement, a simple idea to implement a differential was considered.

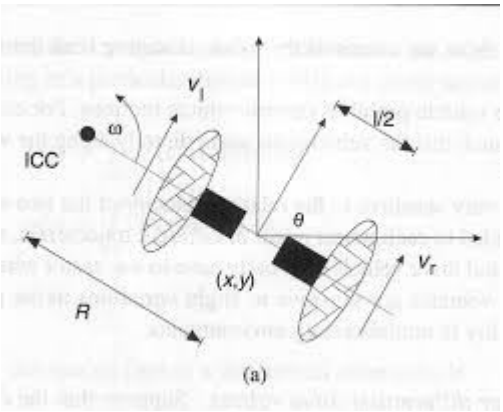

Figure 2: Differential

3.1.2 Designing on CAD

There have been several designs that were made and tested, with only a few meeting both the aesthetic and the size constraints.

3.1.3 Prototyping

Several Prototypes were made and each prototype was an improvement to its predecessor.

3.1.4 Technical Difficulties

Size Factor
One of the main challenges was to fit everything into a compact size. The size of the mic that was used for reference was much smaller than the given Bose speaker. It was quite a challenge to accommodate the even bigger PCB that controlled the Bose Speaker. The motors required to provide the rotational motion had to be fit into this small size of a mic.
Solution: Design a mic frame by taking the measurements of the PCB with

Figure 3: Prototypes

the reference of one of the primitive design models. Since a minimalistic design was desired, several iterations of how to implement the wheels and the motors were done to ensure aesthetics and controllability.

Aesthetics
The final purpose of this project is to make a model that can be presented during a performance in front of an art enthusiastic audience, so the shape and the final design of the model had to mimic that of a mic to the closest sense, with no compromise.
Solution: Since the standard traditional mics are too small to work with, through a thorough research gaming/podcasting mics were found to have amongst the biggest diaphragms to work with and therefore a bigger mesh. And then to mount this on the model, we had to incorporate an aesthetic taper that could mount the mesh of the mic.

3.1.5 First Iteration

One of our primitive designs was to implement the differential with the wheels popping outside with a 3D printed mesh for the speaker as shown

in the fig(4).

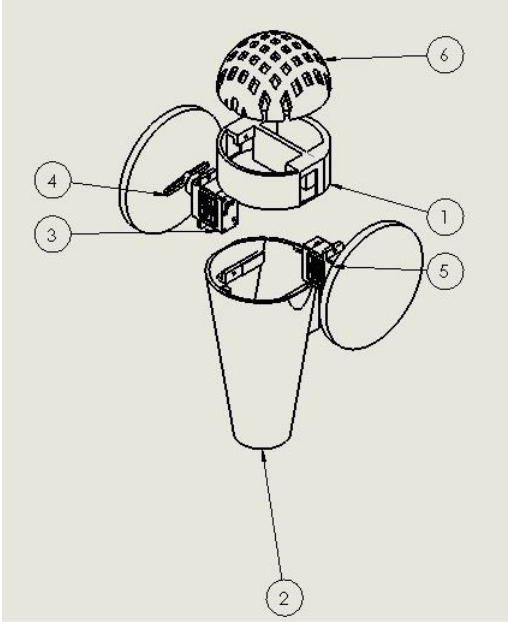

Figure 4: Plan for the first Iteration(Labeled)

Each of the parts on the model as labeled in fig(4) is described in the tab(1).

A 3D print of this model was taken to see the possible improvements to the design. As seen in the fig(5), the model looks more like an Ice-cream cone rather than a mic, and this is when we decided to use an actual mic mesh for the aesthetics, and a put the wheels inside the structure rather than on the outside. This will make the model slightly more complex, but comes with a better aesthetic appeal.

3.1.6 Second Iteration

To make the model look more like a mic, the new design was to have it's wheels inside the trunk with part of the wheels popping out to provide locomotion. Since the whole wheel hub (including motors) with some support need to be mounted completely on the inside of the structure, the model made in two halves, Bottom and Top parts as shown in the fig(6).

Label #	Parts	Description
1	Motor Mount	This part of the body has supports to mount the mini servos on it
2	Cone shaped Hub	This part is to hub all the components, and also act as a caster with the bump in the bottom of the body
3 & 5	Servo Motors	Micro servos were selected for the first iteration because it is one of the smallest motors readily accessible in various Arduino starter kits.
4	Wheel	To provide locomotion, Wheels are attached to the motors to makes a differential drive.
6	Mic Mesh	A mic shaped structure purely to meet the aesthetic requirement of the model

Table 1: Description of parts in the first iteration

Figure 5: Physical 3D model of the first Iteration

This model was then printed to see the possible improvements that can be made. But as seen on the fig(7), the model lacked the strength, so it needed improvement.

Figure 6: 3D model of the second design iteration

Figure 7: Physical 3D model of the top part of the Second Iteration

3.1.7 Third Iteration

During this iteration, it was considered to make and print the body of the mic in as few pieces as possible, especially merging the top and bottom halves from the second iteration. The Labels marked in the fig(8) are described in the tab(2). During this iteration we got our hands on a large mic with a large enough mesh to fit the Bose speaker in it. Assembly was also well considered during the modeling, and described with the aid of pictures:

1. Place the motors into the body of the model as shown in the fig(9).

2. The motors that are placed in the model with the assistance of the guides need to be fixed into place by the use of the Motor Mount Plate as shown in the fig(10).

3. Custom designed wheels used friction coupling to be coupled with the motor shaft and it can be mounted with motors through the slot

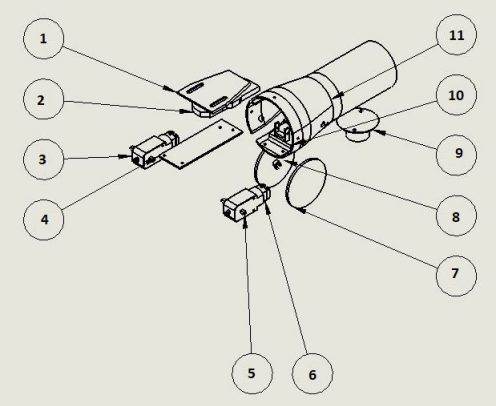

Figure 8: Plan for the third Iteration(Labeled)

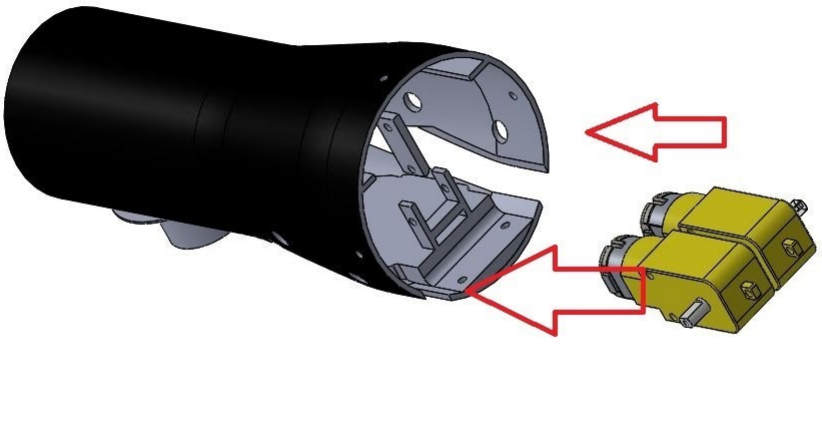

Figure 9: Step 1 of Assembly

as shown in the fig(11).

4. Now the Speaker PCB mount is to be slid into place, as shown in the fig(12).

5. Addressing one of the challenges, a mount was created to help mount the mesh from the mic on to the model. This mount is fixed onto the model, as shown in the fig(13).

Label #	Parts	Description
1	Speaker PCB mount	Designed to hold the Bose Speaker control PCB along with a support for the charging port
2	Speaker PCB	Indicates the position and the placement of the PCB
3 & 5	Torque Gears	Gears that convert the high speed of the motor to the required amount of torque
4	Drive Motor Plate	A plate to hold the motors in place.
6	DC Motor	DC Motors were used for their compact size and ease of control
7 & 8	Wheels	Wheels with a slits (not shown in the fig) for mounting on to the motors
9	Caster Wheel	Provides a point contact for better control of the differential
10 & 11	Body	The upper and lower parts from the second iteration put together as one

Table 2: Description of parts in the third iteration

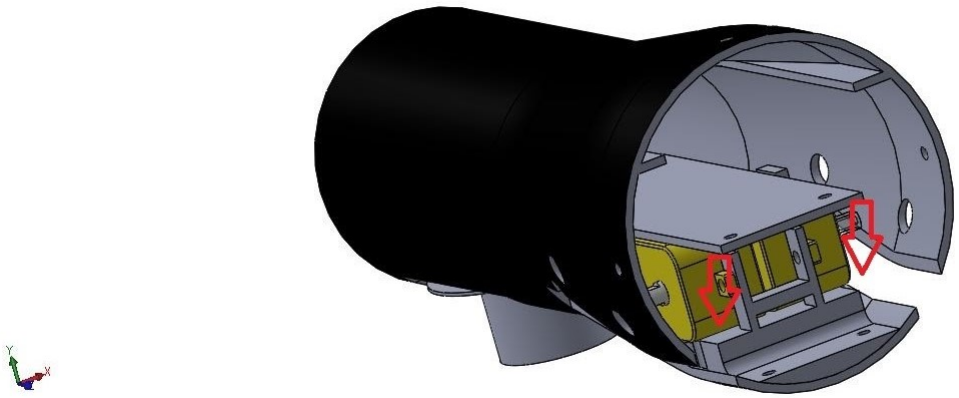

Figure 10: Step 2 of Assembly

6. Last step of mechanical assembly is to place the Mic mesh on the mount as shown in the fig(14).

Once the model was fully assembled as shown in the fig(15), it was noticed that the only issue with this model is the slit-cut wheels. These wheels were technically slightly convenient while mounting, but it was hard to set up a lock to keep the wheels in place (partly because of unavailability of the right size of screws). As it is noticed, there was a necessary hole at the

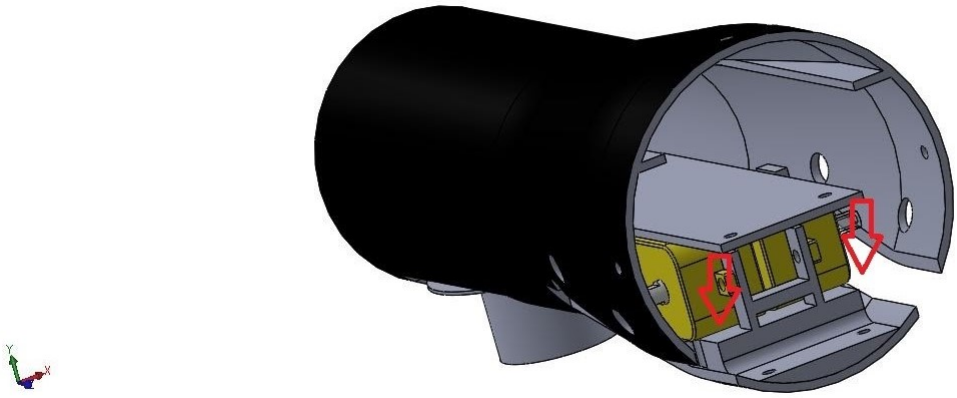

Figure 11: Step 3 of Assembly

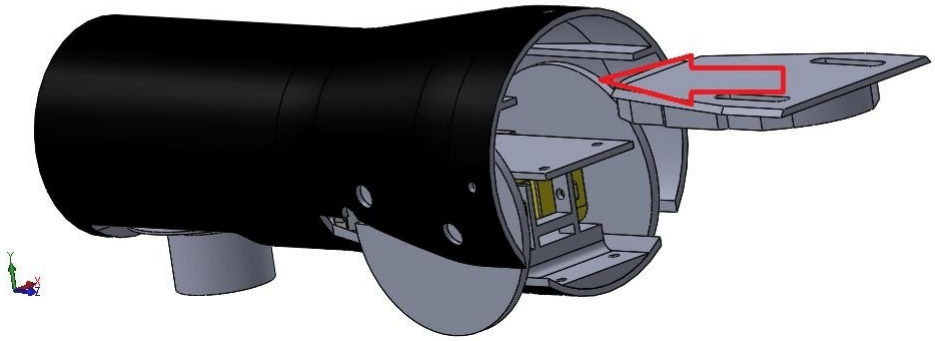

Figure 12: Step 4 of Assembly

top of the model to allow an uninterrupted access to the charging port of the Bose Speaker from the outside, while keeping it in place.

Several Iterations

After all these iterations, the first working prototype was made.

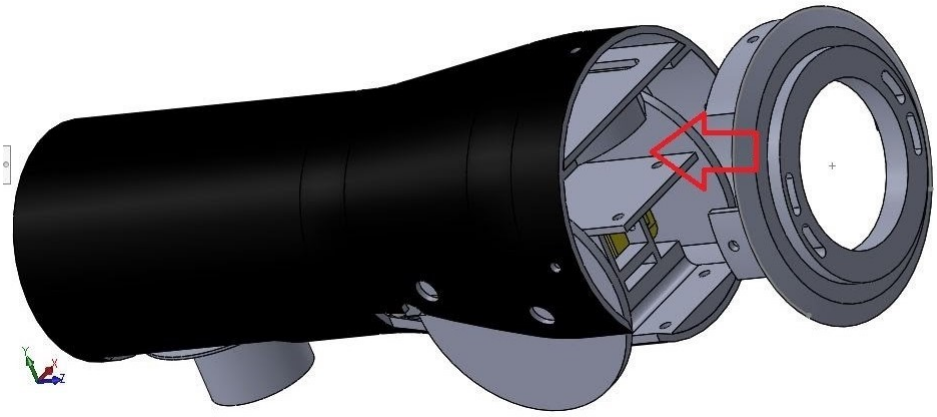

Figure 13: Step 5 of Assembly

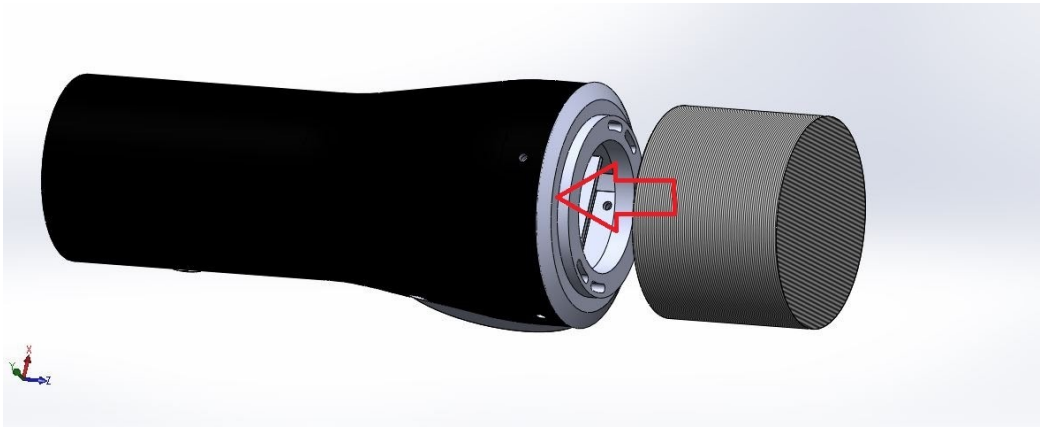

Figure 14: Final step of Assembly

3.1.8 First functioning Prototype

Since the wheels from the third iteration did not give the desired results and since the assembly was rather tricky, it was decided to make a small change to the body as shown in the fig(17).

The assembly of this model almost remains the same as that of the third Iteration, except the motors have the wheels mounted before they are placed in the hub and the top part of the body (as shown in the fig(17)) is to be attached to the rest of the body before step 4 of the assembly.

Figure 15: Fully Assembled model (Third Iteration)

Fig(18) is to show the increase in comfort of assembly as compared to the previous iteration. This model when printed worked quite well and these 3D printed parts were able to support the structure while keeping the structure very lightweight and fit all the components inside the structure. This prototype works as desired- mechanically.

3.2 Electronics

To make this snake-mic work, we need to integrate this mechanical structure with quite a few electronic components, which include:

DC Motor
A simple DC motor that can rotate in both the directions with a low power

Figure 16: Photo of several iterations and a few parts used

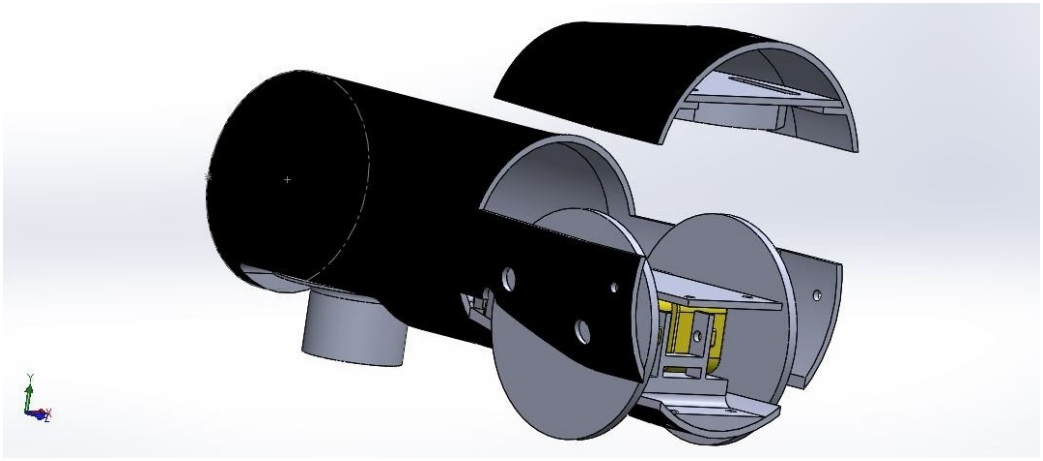

Figure 17: CAD of the Fourth Iteration

rating was used. Typically these motors deliver a lot of speed rather than torque, so the motor was fitted with a few gears to deliver the torque that the model needs.

Figure 18: Almost assembled model of the fourth iteration (green model)

Motor Driver
To control a DC motor at different rotational directions and different speeds while supplying it with the current it needs without straining the microcontroller, a DC motor driver with a H-bridge is needed.

Controller
An Arduino Nano was used as the primary controller, this was powered by

a 9V battery through a power module. Arduino Nano was chosen to keep the size of the controller to the minimum.

Joystick
To test out the controls, initially a joystick was also used, with it connected directly to the Arduino.

Bluetooth Module
A HC-05 Bluetooth Module was used for communication between the controller and the remote control, through an android application.

Bose SoundLink Micro
A Bose Soundlink Micro Speaker was dismantled, and it's components were used as a speaker, as per the problem statement.

3.3 Arduino Code

```
#include <SoftwareSerial.h>
SoftwareSerial Bluetooth(2, 3); // RX, TX

#define en1 5
#define in1 8
#define in2 9
#define en2 6
#define in3 10
#define in4 11

int xDir = 55; //Specifies location of the joystick on screen
    along x axis (additionally used for custom controls)
int yDir = 55; //Specifies location of the joystick on screen
    along y axis (additionally used for custom controls)
//Initialize variables to specify the motor speed of both the
    wheels
int motorSpeed1 = 0;
int motorSpeed2 = 0;

//For curve function
int count = 0;
bool toggleFlag = 0;
void setup() {
  pinMode(en1, OUTPUT);
```

```
  pinMode(en2, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  Serial.begin(9600);
  Bluetooth.begin(9600); // Default communication rate of the
   Bluetooth module
  delay(500);

void loop() {
  // Read the incoming data from the Smartphone Android App
  while (Bluetooth.available() >= 2) {
    xDir = Bluetooth.read();
    delay(10);
    yDir = Bluetooth.read();
    Serial.print(xDir);
    Serial.print(",");
    Serial.println(yDir);
  }
  delay(10);
  // Makes sure we receive correct values
  //Custom Inputs - Straight, Circle, Sine, Stop, Rotate In
   Place
  if (xDir == 5)
  {
    switch (yDir)
    {
      case 172: // Go straight
        motorSpeed1 = 250;
        motorSpeed2 = 250;
        forward();
        break;
      case 174: // Circle
        motorSpeed1 = 250;
        motorSpeed2 = 200;
        forward();
        break;
      case 176: // Sine or Wave motion
        if (toggleFlag)
        {
          motorSpeed1 = 250;
          motorSpeed2 = 200;
        }
        else
```

```
65          {
              motorSpeed1 = 200;
67            motorSpeed2 = 250;
            }
69          curve();
            break;
71        case 178: //Rotate in place
            motorSpeed1 = 250;
73          motorSpeed2 = 250;
            turnRight();
75          break;
          case 180: //Stop
77          Stop();
            break;
79      }
    }
81  else
    {
83    //Joystick Control
      if (xDir > 50 && xDir < 120 && yDir > 50 && yDir < 120) {
85      Stop();
      }
87    if (yDir > 50 && yDir < 120) {
        if (xDir < 50) {
89        turnRight();
          motorSpeed1 = map(xDir, 50, 10, 0, 255);
91        motorSpeed2 = map(xDir, 50, 10, 0, 255);
        }
93      if (xDir > 120) {
          turnLeft();
95        motorSpeed1 = map(xDir, 120, 150, 0, 255);
          motorSpeed2 = map(xDir, 120, 150, 0, 255);
97      }
      }
99    else {
        if (xDir > 50 && xDir < 120) {
101
          if (yDir < 50) {
103         forward();
          }
105       if (yDir > 120) {
            backward();
107       }
          if (yDir < 50) {
109         motorSpeed1 = map(yDir, 50, 10, 0, 255);
```

```
          motorSpeed2 = map(yDir, 50, 10, 0, 255);
        }
        if (yDir > 120) {
          motorSpeed1 = map(yDir, 120, 150, 0, 255);
          motorSpeed2 = map(yDir, 120, 150, 0, 255);
        }
      }
      else {
        if (yDir < 50) {
          forward();
        }
        if (yDir > 120) {
          backward();
        }
        if (xDir < 50) {
          motorSpeed1 = map(xDir, 50, 10, 255, 50);
          motorSpeed2 = 255;
        }
        if (xDir > 120) {
          motorSpeed1 = 255;
          motorSpeed2 = map(xDir, 120, 150, 255, 50);
        }
      }
    }
  }
  //Serial.print(motorSpeed1);
  //Serial.print(",");
  //Serial.println(motorSpeed1);

  analogWrite(en1, motorSpeed1); // Send PWM signal to motor A
  analogWrite(en2, motorSpeed2); // Send PWM signal to motor B
}

void forward() {
  Serial.println("forward");
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}
void backward() {
  Serial.println("backward");
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
```

```
  digitalWrite(in4, HIGH);
}
void turnRight() {
  Serial.println("turnRight");
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}
void turnLeft() {
  Serial.println("turnLeft");
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}
void Stop() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  Serial.println("stop");
}
void curve() {
  Serial.println("curve");
  forward();
  count++;
  if (count >= 200)
  {
    toggleFlag = !toggleFlag;
    count = 0;
  }
  Serial.println(count);
}
```

3.4 User Interface

Since this product will be used later in a performance by artists, it is very important to create a user friendly interface in order to control the Snake-Mic. An Android Application was developed to control the Snake-Mic via

Bluetooth communication. As shown in the fig(19), the app has a Joystick control (right) to move the mic around with speed control and it also has 5 additional commands are provided for more intuitive experience namely:

- **Straight:** Goes forward.
- **Circle:** Moves in the circle.
- **Wave:** Goes forward in a wave like fashion
- **Rotate in-place:** Rotates in place in counter clockwise direction

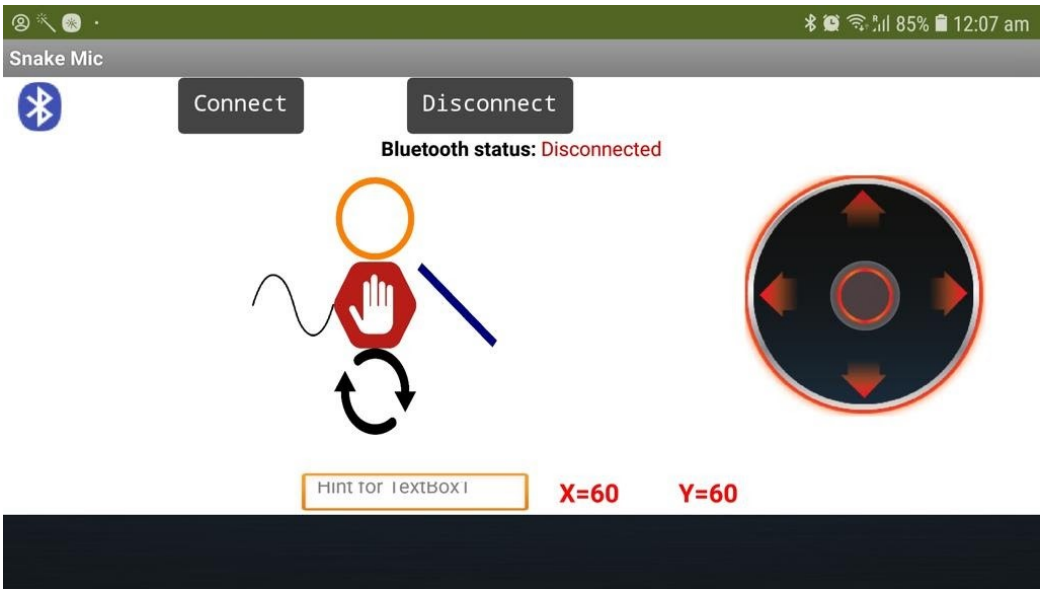

Figure 19: Screenshot of the app

4 Conclusion

To implement a snake mic with a Bose Speaker inside, and all the other electronic components to maneuver the Snake-Mic, although all the technology already exists, there are several design challenges. After several Iterations, the Snake-Mic was successfully moved using a Joystick. After interfacing the model with the Bluetooth, an Android app was developed to operate the Snake-mic with a comfortable user interface.