Snake-Mic

Tarun Thathvik Smrithi Thudi Vedant Desai

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 chassi. 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.

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 robotically 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.

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.

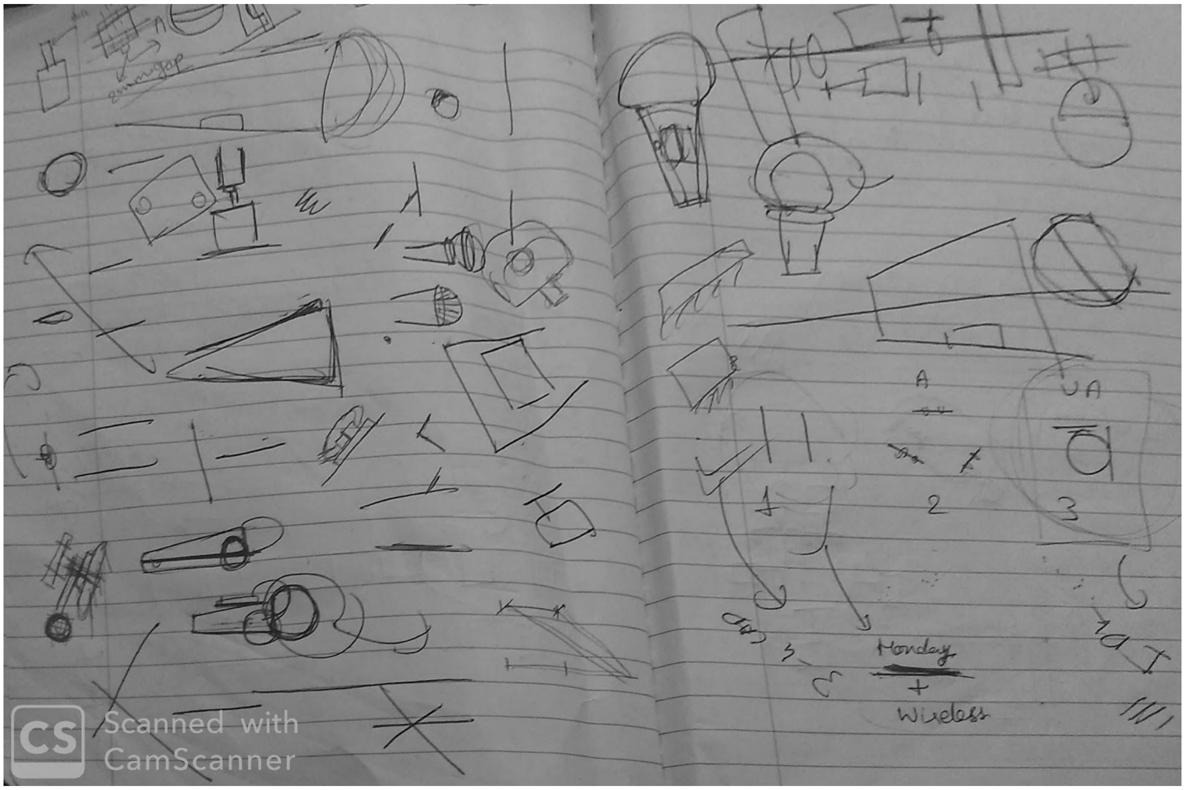
Implementation

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.

Concept generation

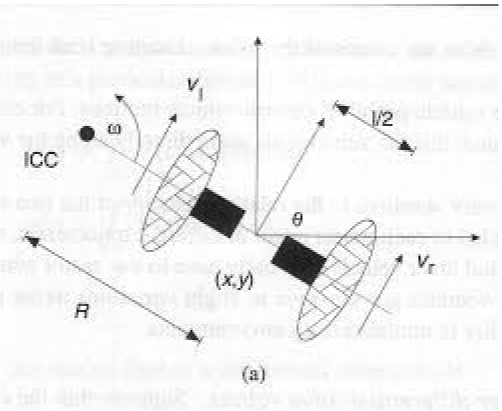
Since the problem statement given was particularly clear about the end product, we had to now idealize the possible ways to implement this Snake-Mic



Ideation

One of the several ideas to make the robot move was to actuate a mouse like structure, as shown in the figure below.

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.

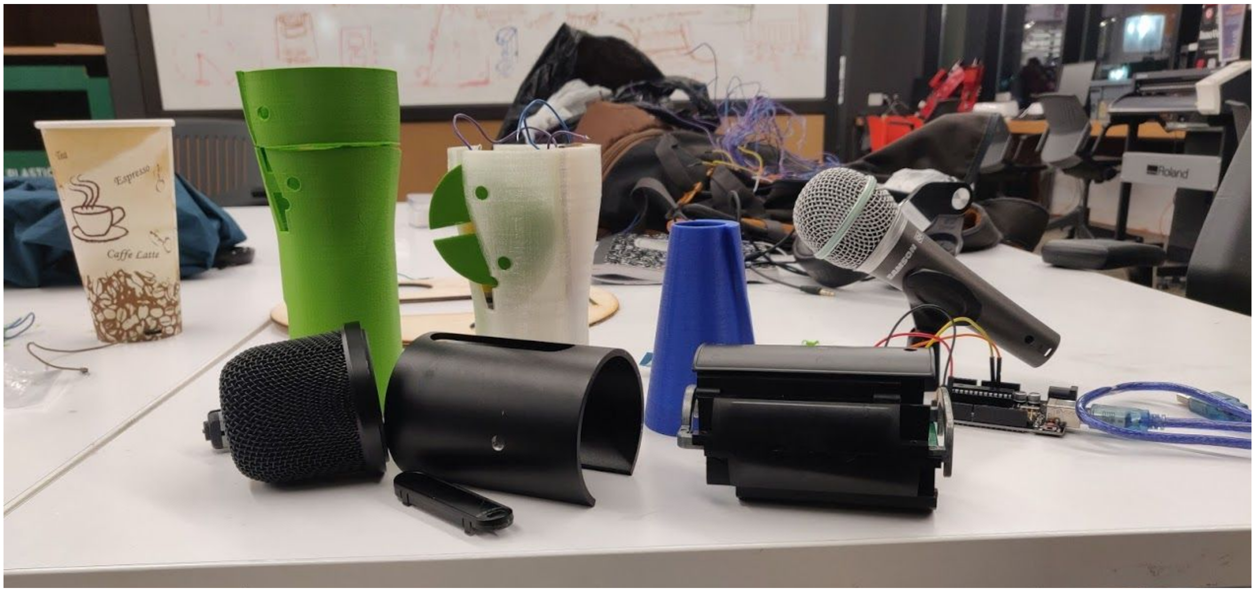


Designing on CAD

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

Concept testing (Prototyping)

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



Several Technical challenges were discovered and overcome during this phase.

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.

<u>Solution</u>:

Design a mic frame by taking the measurements of the PCB with 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:

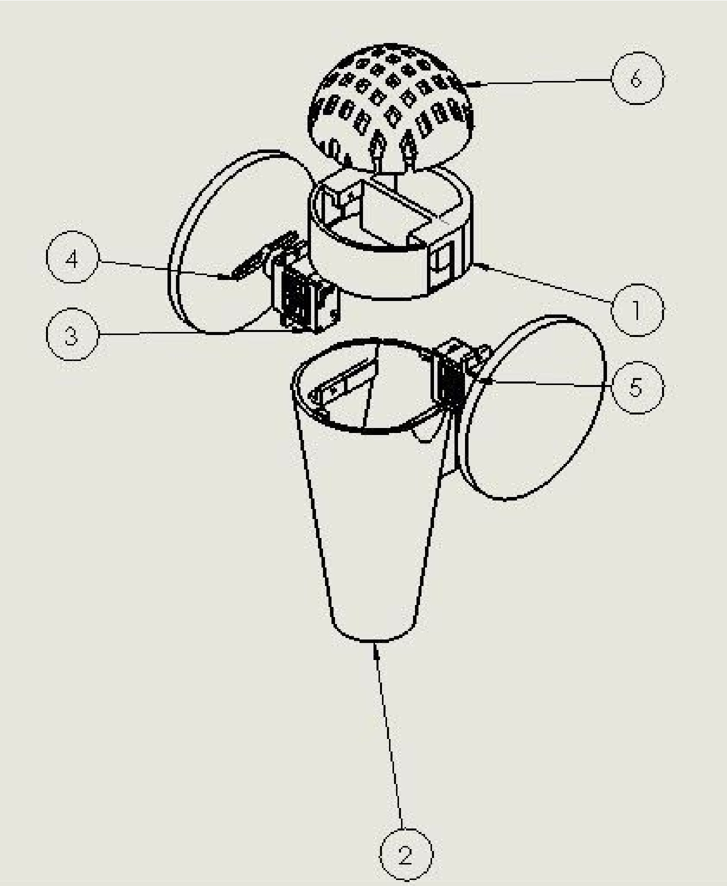
The final purpose of this project is to make a model that can be presented during a performance in front of an art enthusiastic audience, so the shape and the final design of the model had to mimic that of a mic to the closest sense, with no compromise.

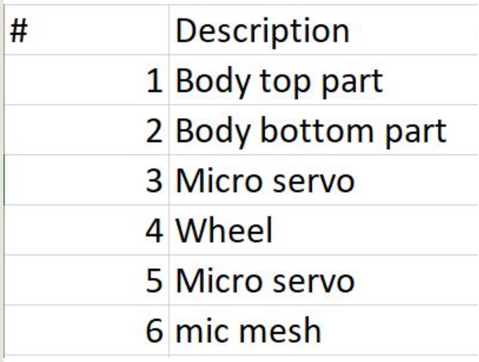
<u>Solution</u>:

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.

First Iteration (Ice-Cream Cone)

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



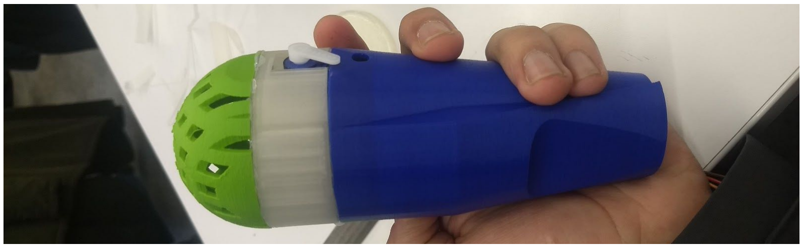


#	Description
1	Body top part
2	Body bottom part
3	Micro servo
4	Wheel
5	Micro servo
6	mic mesh

Description about each of the parts:

Body top part (1): This part has a motor mount and it gets attached with the body bottom part to complete the assembly.
Body bottom part (2) - This part has another half of the motor mount and during assembly it gets attached to the top part providing support to the motor and giving an aesthetic look to the structure as required by the problem statement.
Micro-Servo (3 & 5) - We selected micro servo as our first iteration because it is one of the smallest motors readily accessible in various arduino starter kits.
Wheel (4) - To provide locomotion capabilities to the system. Wheel gets attached to the motors and makes a differential drive.
Mic-Mesh (6) - 3D printed part to give an aesthetic look to the system.

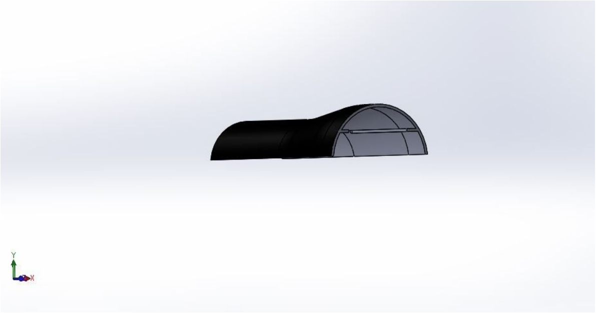
This model was then 3D printed, to check the possible improvements to the design.

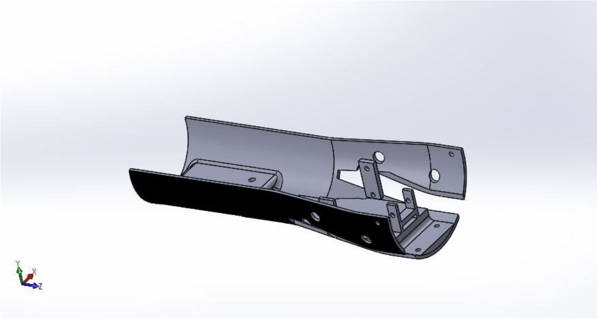


From this image, it is clear that this structure does not satisfy the aesthetic appeal of a mic, and certainly does not fit the huge circuite of the Bose Speaker.

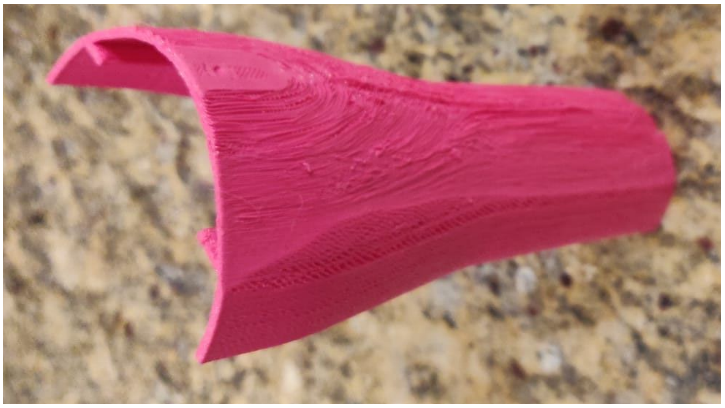
Second Iteration:

In this iteration we tried to make the assembly in two parts as shown below.





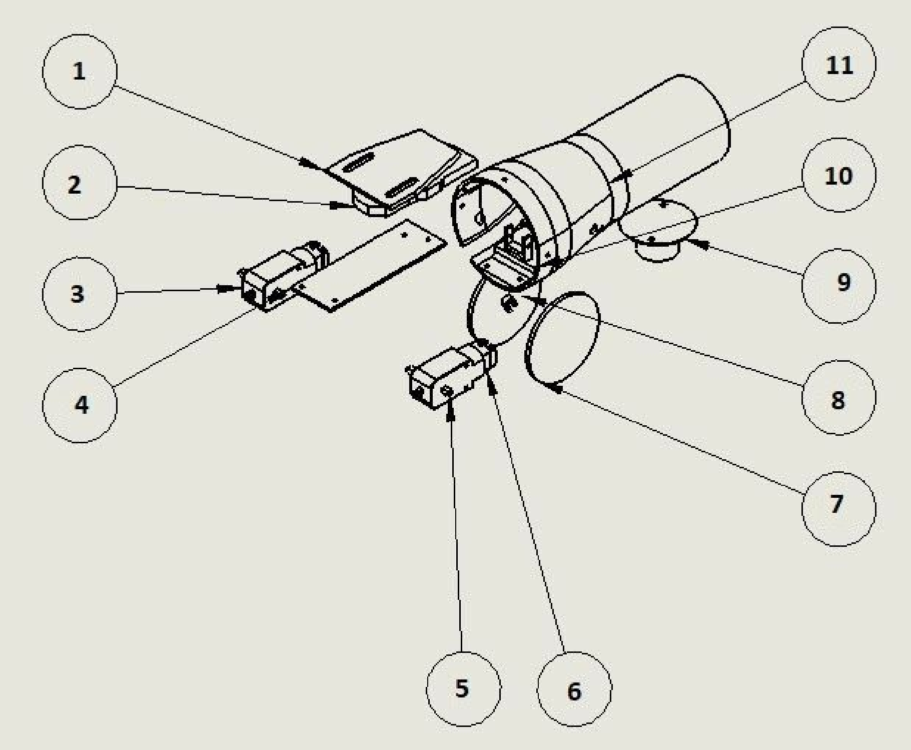
This was then printed to see the possible improvements to the same

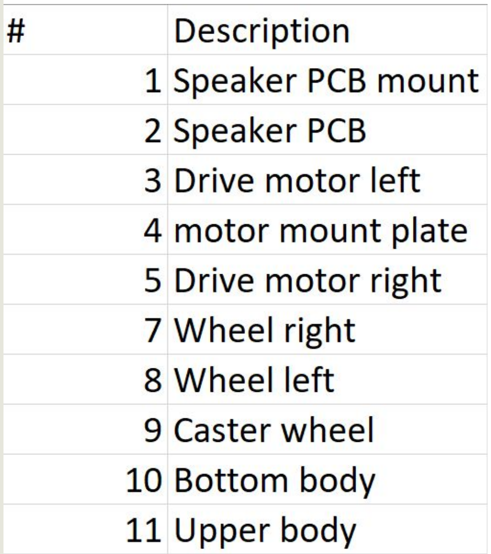


During this Iteration, the structure of the 3D print was very weak, so we decided to modify it slightly differently for the next iteration.

Third Iteration:

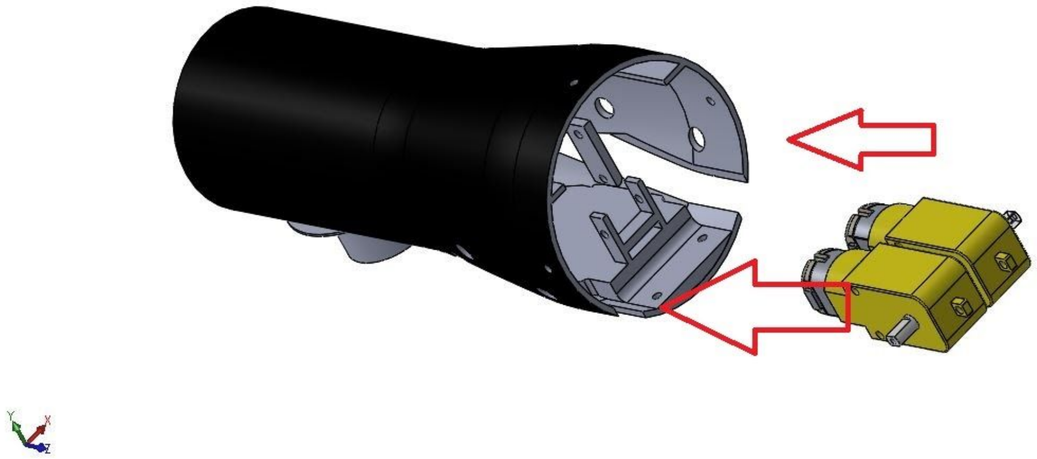
We designed the body of the mic in such a way that it need not have to be printed in parts and one can assemble the whole thing in one part.
The assembly as predicted is shown below.



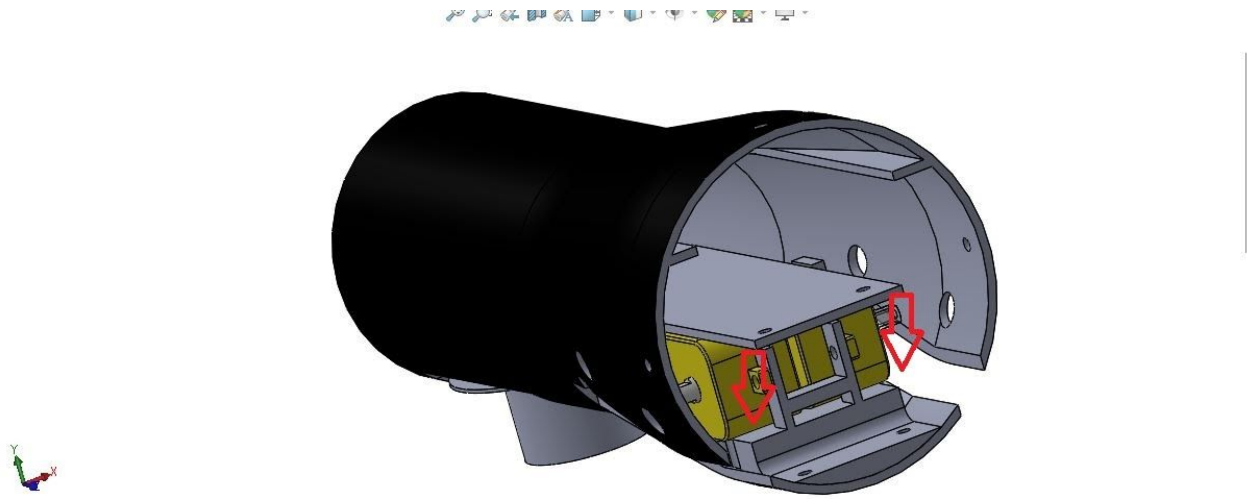


#	Description
1	Speaker PCB mount
2	Speaker PCB
3	Drive motor left
4	motor mount plate
5	Drive motor right
7	Wheel right
8	Wheel left
9	Caster wheel
10	Bottom body
11	Upper body

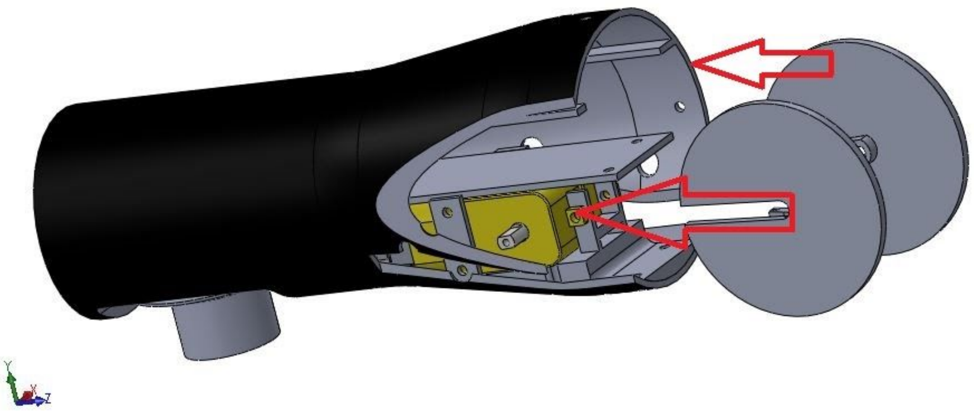
Speaker PCB mount (1) - It is designed such a way that it can hold the Bose speaker PCB along with the charging PCB.
Speaker PCB (2) - It’s the PCB attached to the speaker.
Drive motors (3 & 5) - Motors to drive the system.
Motor mount plate (4) - Plate to fix the drive motors in place.
Wheels (7 & 8) - Wheels to move the system.
Caster wheel (9) - provide planar contact to the assembly for the best control.
Upper body and lower body (10 & 11) - They are shown in the figure above.



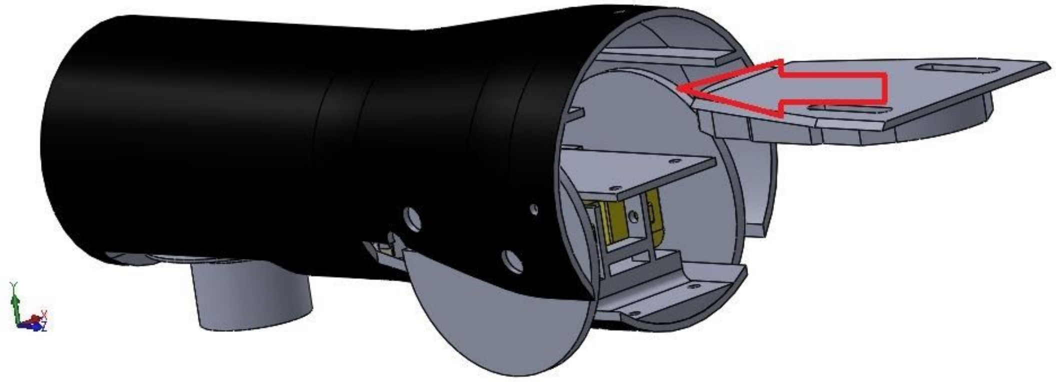
Put motors in the main body laterally.



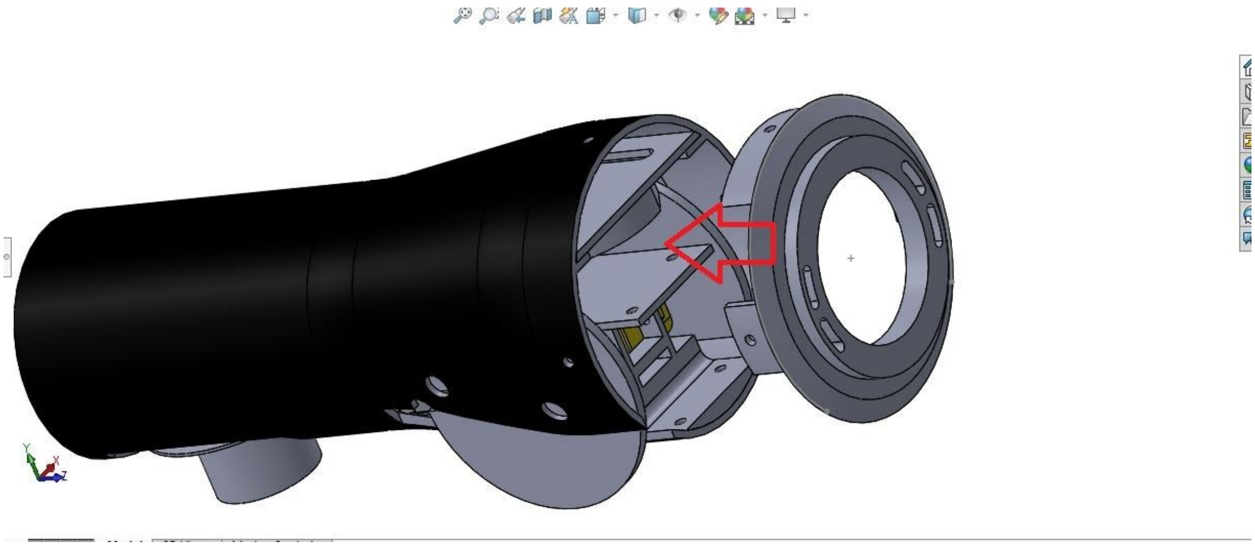
Motor mount plate to fix the motors in position, body has a guide for the motor to facilitate the assembly.



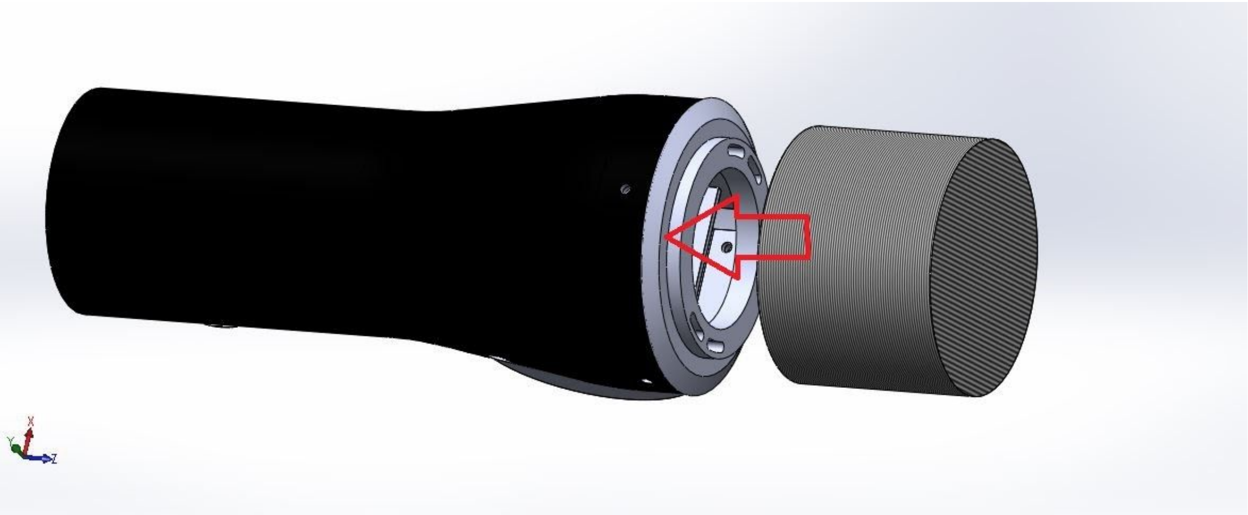
Custom designed wheels used friction coupling to be coupled with the motor shaft and it can be mounted with motors through the slot as shown in the figure above.



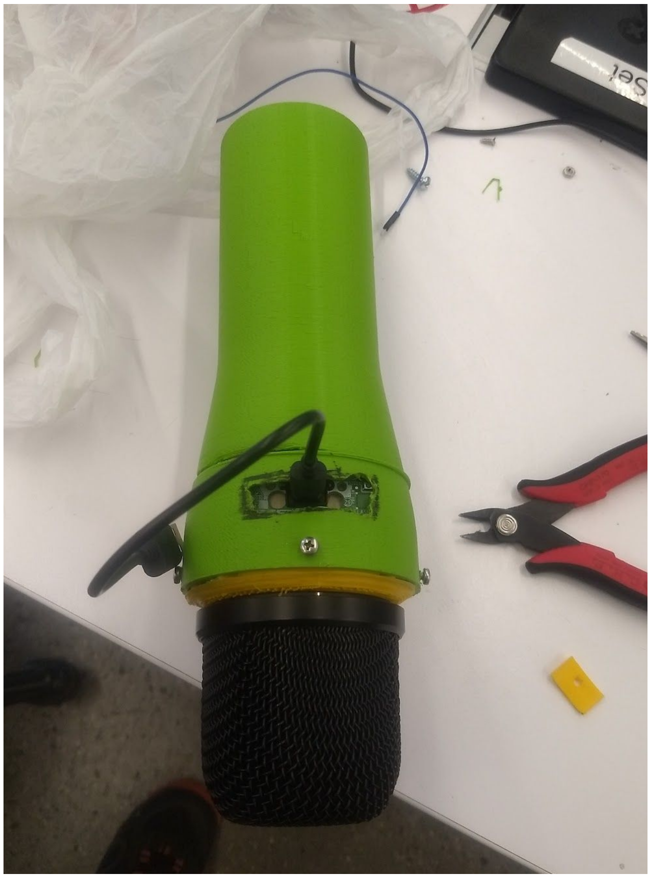
After fixing the motor, fix the speaker PCB mount as shown in the figure above.



As discussed in the challenges section we design a custom mesh mount for mounting the mesh to the body of the mic as shown in the figure above.

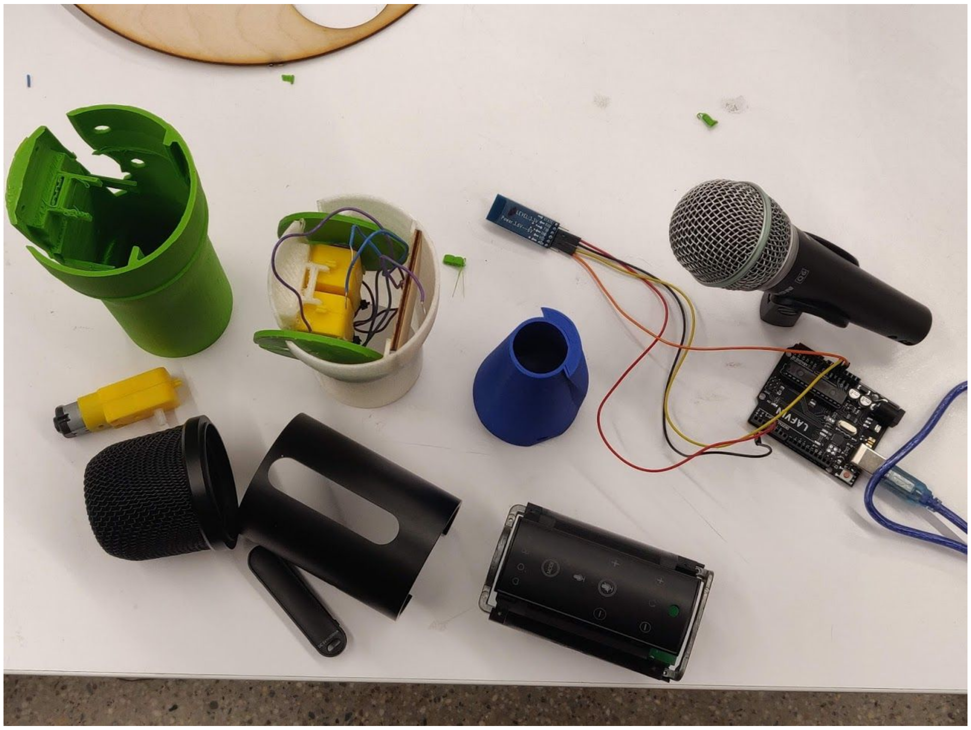


Finally put the speaker in the mesh and mount the mest to the body as shown in the figure above. And this step completes the assembly of the mic.



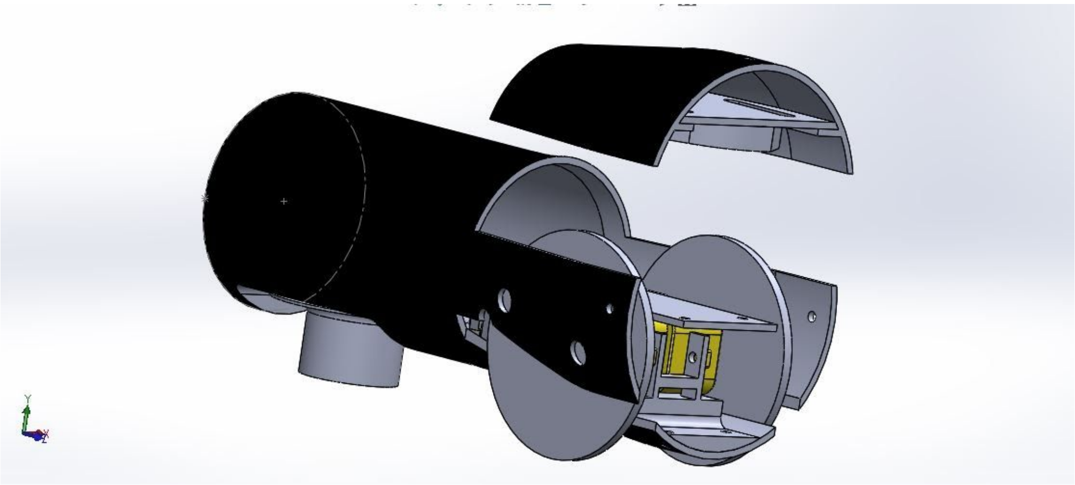
The only issue with this model are the slit-cut wheels. These wheels were technically slightly convenient while mounting, but it was hard to set up a lock to keep the wheels in place (partly because of unavailability of the right size of screws).

Several iterations:



After all these iterations, we finally landed on a working prototype.

First functioning Prototype:



Since the wheels from the third iteration did not work well and assembly was rather tricky we decided to make a new assembly shown below.

Above shown assembly worked quite well and 3-D printed parts are able to support the structure keeping the structure very lightweight and all the components fit inside the structure pretty well. This prototype works as desired- mechanically.

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 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.

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.

Code

Arduino code to control the Snake-mic:

```
#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
        {
          motorSpeed1 = 200;
          motorSpeed2 = 250;

        }
        curve();
        break;

      case 178: //Rotate in place
        motorSpeed1 = 250;
        motorSpeed2 = 250;
        turnRight();
        break;

      case 180: //Stop
        Stop();
        break;
    }
  }
  else
  {

    //Joystick Control
    if (xDir > 50 && xDir < 120 && yDir > 50 && yDir < 120) {
      Stop();
    }


    if (yDir > 50 && yDir < 120) {

      if (xDir < 50) {
        turnRight();
        motorSpeed1 = map(xDir, 50, 10, 0, 255);
        motorSpeed2 = map(xDir, 50, 10, 0, 255);
      }

      if (xDir > 120) {
        turnLeft();
        motorSpeed1 = map(xDir, 120, 150, 0, 255);
        motorSpeed2 = map(xDir, 120, 150, 0, 255);
      }

    } else {

      if (xDir > 50 && xDir < 120) {

        if (yDir < 50) {
          forward();
        }
        if (yDir > 120) {
          backward();
        }

        if (yDir < 50) {
          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);
```

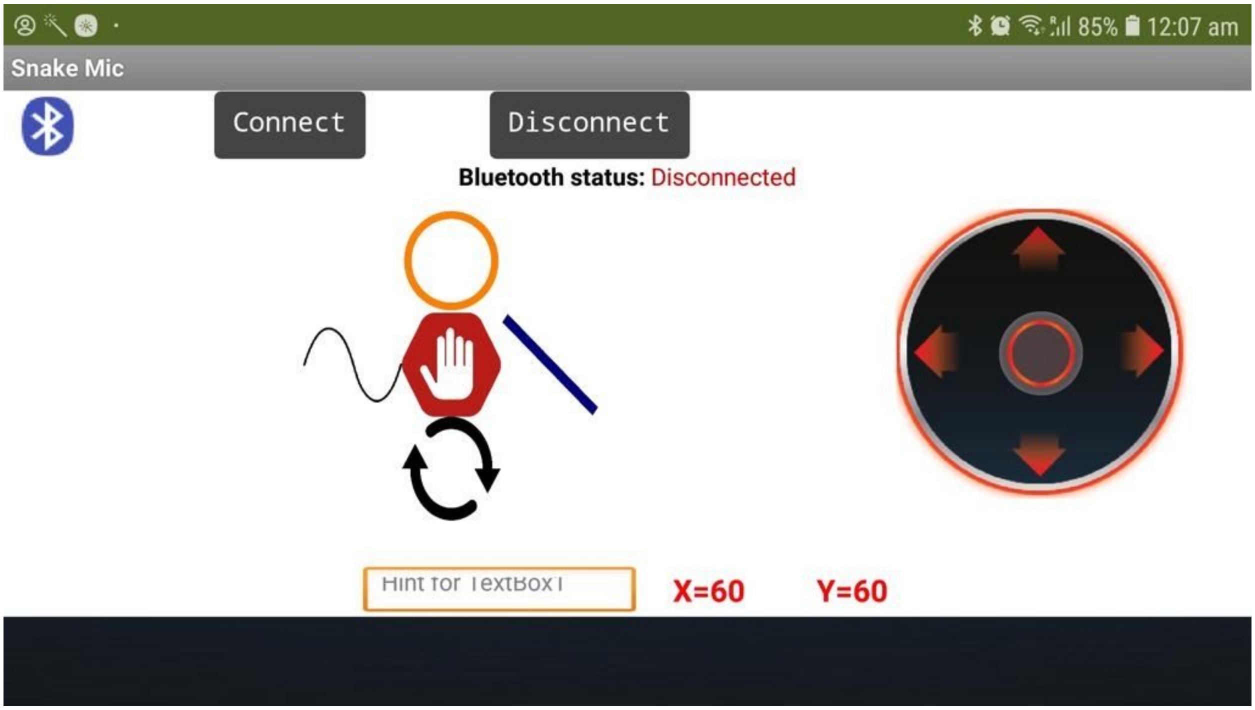
```
}

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);
}
```

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. 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



Bluetooth status: Disconnected

X=60 Y=60

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.