FREQUENCY FOLLOWING BOT

By: Unarv Erinjery, Nathan Gonsalves, Johan Alex, Eric B John

INSPIRED BY LIFE

MANIPAL
ACADEMY *of* HIGHER EDUCATION
(Deemed to be University under Section 3 of the UGC Act, 1956)

Manipal Institute of Technology

Department of Biomedical Engineering

Digital Signal Processing

Faculty: Omkar S Powar

ABSTRACT:

The main purpose or goal of the project is to create a robot that can follow a specific frequency. The robot uses a combination of sensors and control systems to detect and follow the desired frequency. The project involves designing the path following robot, which includes the use of an ultrasonic sensor to measure the distance between the obstacle and the robot and sound sensors to triangulate the location of the source. It's objectives include designing a path follower robot that can navigate a path without any faults, identify the given path, and perform assigned work thoroughly without any errors. The robot is coded to understand the path and create a simple closed loop system for smooth movement. Its methods involve designing and coding the robot's control system, integrating the sensors, and testing the robot's performance on different paths. The project findings are based on the robot's ability to follow a specific frequency and navigate a path without any errors. The conclusion of the project is that the Frequency Following Robot project successfully developed a robot that can follow a specific frequency and navigate a path without any faults, demonstrating its potential for various applications.

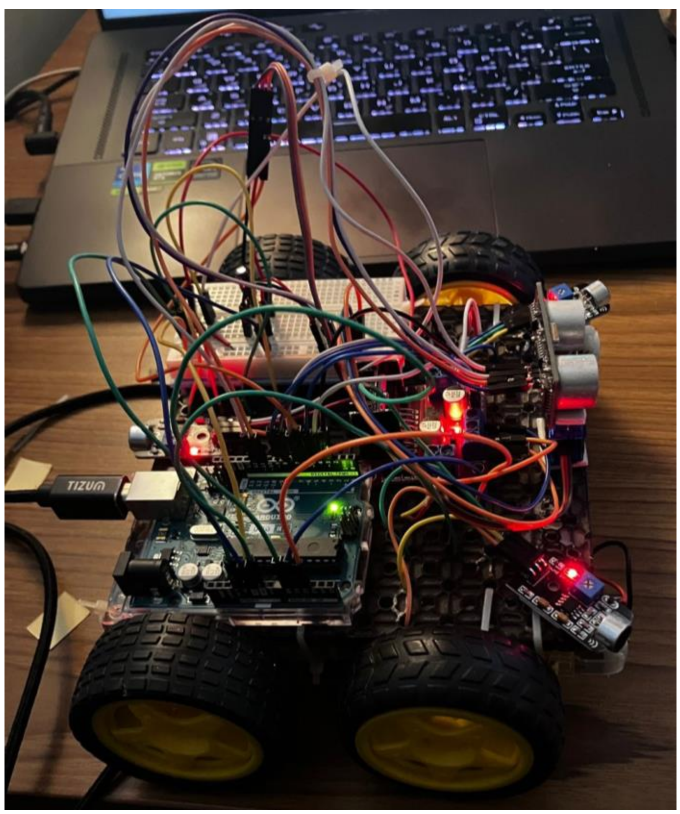


TABLE OF CONTENTS:

INTRODUCTION:

Background:

The frequency-following robot involves the use of sensors and algorithms to detect and follow a specific frequency in the environment. The robot can be controlled by designing an algorithm that determines when it should turn and move forward based on the sound from the microphones. The robot can be built using components like Arduino, ultrasonic sensors, and motors, and the Frequency following two microphones to drive towards a sound source. These robots can be used for a variety of tasks, such as search and rescue, marine exploration and surveillance. They can be made to respond to frequencies, such as a person's voice or other noises of interest.

Objectives and goal of the project:

- Develop a robot that can follow a sound source based on a specific frequency.
- Implements microphone sensors to navigate to the source.
- Create an algorithm that enables the robot to follow a source based on the detected frequency.
- Avoids obstacles using an oscillating ultrasound sensor.
- Continuously improve the robot's design and structure to make it suitable for various applications.
- Make sure the motors provide the necessary torque to carry out the instruction accurately.

Significance and motivation for the project:

- **Improved Performance and Stability:** The project aims to develop a robot that can follow a human more effectively by using a frequency-based approach. This can lead to improved performance and stability in the robot's movements, as it can better adapt to the human's movements and maintain a consistent distance.
- **Advancements in Robotics Technology:** The project contributes to the advancement of robotics technology by exploring new methods for robot interaction. By using frequency-based techniques, the robot can more accurately follow a source.
- **Potential for New Applications:** The project's findings can be applied to various fields, such as search and rescue, marine exploration.
- **Increased Efficiency and Safety:** By improving the robot's ability to follow a source, it can potentially lead to increased efficiency and safety in various applications. For instance, in search and rescue missions, a frequency-following robot will effectively locate and assist individuals in need, reducing risk and improving response times

LITERATURE REVIEW:

We have collected inspirations from previously executed ideas from obstacle avoidance and sound amplitude sensor to make this project

Review:

Frequency-Based Sound Sensing:

One of the fundamental aspects of sound sensing robots is their ability to detect specific frequencies. Research by Zhang et al. (2017) introduced a similar concept where a sound sensing robot was developed to detect and locate specific frequency sources in industrial environments. The robot utilized FFT analysis to identify the frequency signature of machinery malfunctions, allowing for proactive maintenance.

Microphone Sensor Integration:

Integrating microphone sensors with robotic platforms has been explored in various studies. Chen et al. (2019) presented a project where a microphone sensor was used for voice command recognition in a mobile robot.

Arduino-Based Control Systems:

Arduino microcontrollers have become a popular choice for controlling robotic systems due to their versatility and ease of use. In a study by Varma et al. (2019), an Arduino-based obstacle avoiding robot was developed using ultrasonic sensors. While the focus was on obstacle avoidance, the integration of Arduino with sensor modules aligns with the approach proposed in our project.

Ultrasonic Obstacle Detection (Uses Camera):

Obstacle detection plays a crucial role in the navigation of autonomous robots. Research by Kim et al. (2018) presented a novel approach to obstacle avoidance using ultrasonic sensors and machine learning algorithms. The study demonstrated improved obstacle recognition and avoidance capabilities compared to traditional methods. Furthermore, we can use Machine Learning and a Camera capture to accurately avoid obstacles.

Discussion:

The literature review provides insights into the development and implementation of sound sensing-obstacle avoiding robots. By leveraging technologies such as FFT analysis, microphone sensors, Arduino microcontrollers, and ultrasonic sensors, researchers have made significant strides in enhancing the capabilities of robotic systems for intelligent navigation and environmental perception. Building upon the methodologies and findings of previous studies, our project aims to contribute to the advancement of sound sensing-obstacle avoiding robots by integrating multiple sensing modalities and employing robust control strategies for efficient and reliable operation in real-world scenarios.

METHODOLOGY:

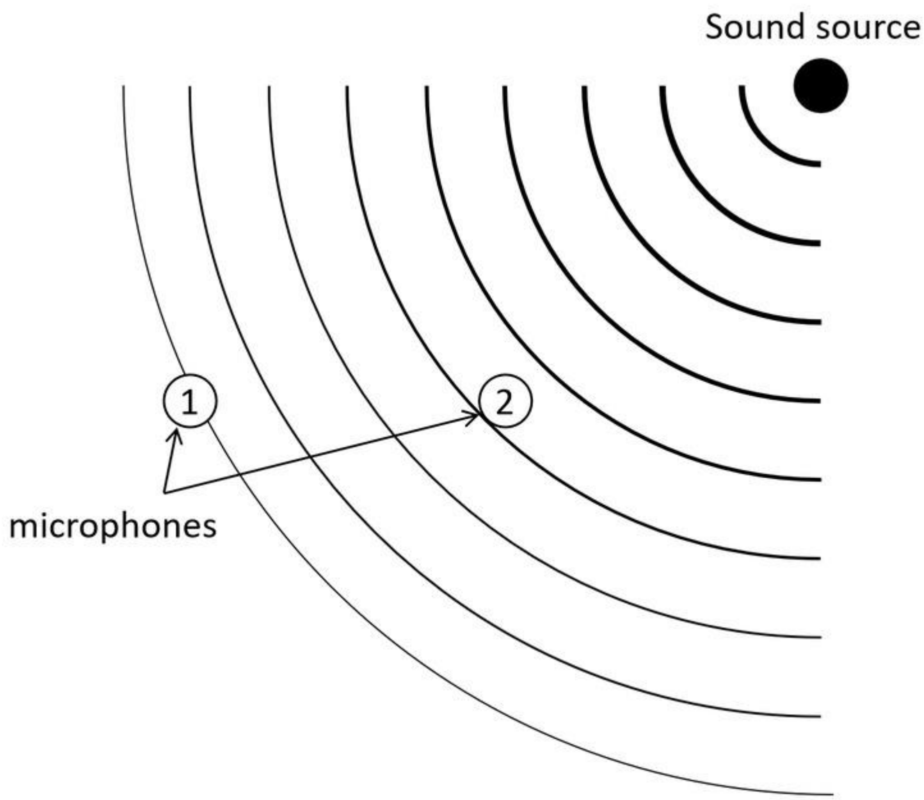
Components:

- Arduino Uno**:** Arduino UNO is a microcontroller board based on the ATmega328p. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.
- Ultrasonic sensor: An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound. Ultrasonic sensors have two main components: the transmitter and the receiver.
- Motor driver (L298N): This **L298N Motor Driver Module** is a high-power motor driver module for driving DC and stepper motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. **L298N Module** can control up to 4 DC motors, or 2 DC motors with directional and speed control.
- Microphone module: It is a sensor or transducer which converts sound to electric signals. It contains an on-board comparator and a potentiometer to adjust sensitivity. There are 2 LEDs present on the module, one being for power while the other provides an indication that the set value has been reached and whenever the output at DO pin is high.
- D.C. Motors: A DC motor is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current and convert this energy into mechanical rotation.
- Servo motors: Servo motors or "servos", as they are known, are electronic devices and rotary or linear actuators that rotate and push parts of a machine with precision. Servos are mainly used on angular or linear position and for specific velocity, and acceleration. Here we use a dc servo motor.
- Breadboard: Used for connection purposes.
- Jumper wires: Used to connect 2 points.
- Wheels: connected to the DC motors for mobility. (Not in schematic).
- Base plate: Acts as our chassis.

Working

Sound sensing and following:

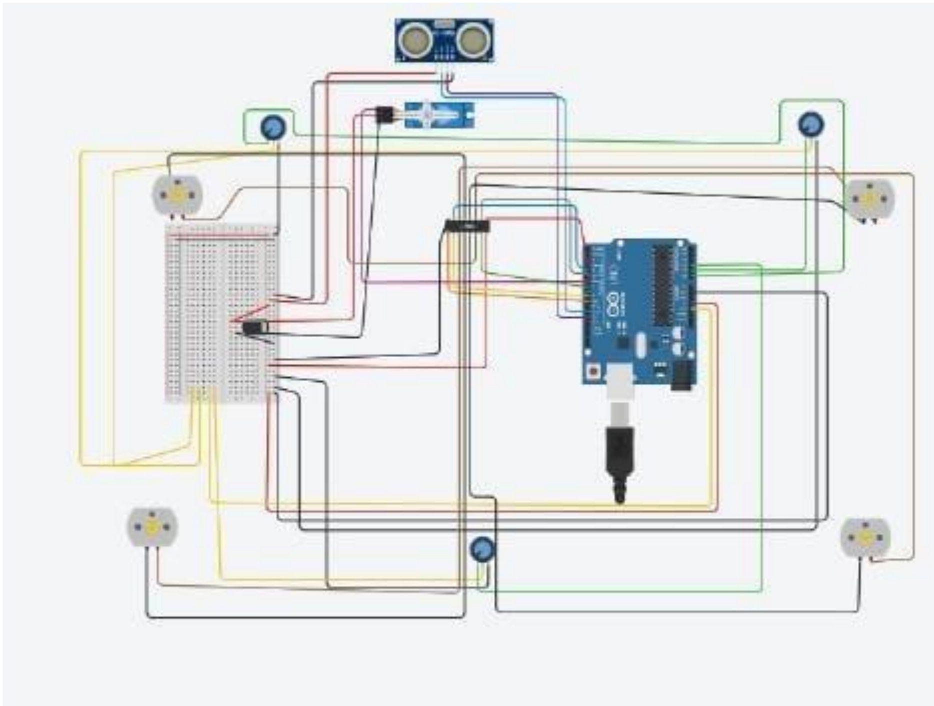
The microphones take in the input sound and output it to the Arduino. The Arduino performs FFT and takes the average of the 3 microphones and then gives priority to the microphone with the highest value. The bot moves in the direction in which the chosen microphone is placed. It continues forward until it either reaches the target or an obstacle is faced. For the obstacle, we have an obstacle avoidance feature.

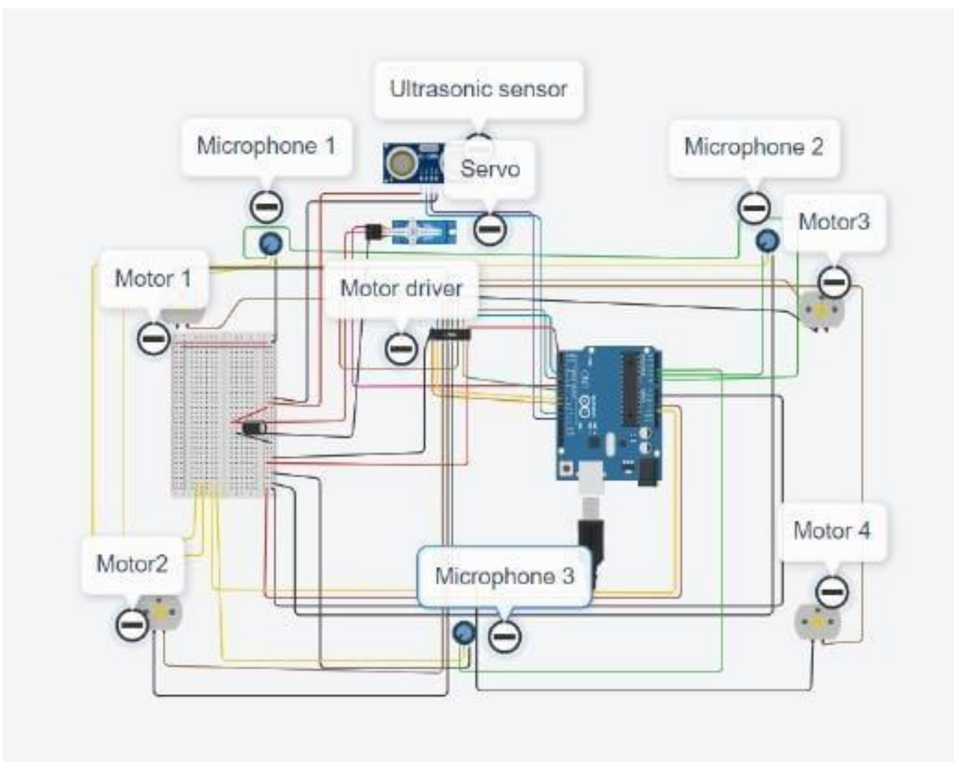


Obstacle avoidance:

When faced with an obstacle which is less than or equal to the specified distance away from the bot, the ultrasonic sensor is activated and sends out an acoustic wave that is activated by the trigger pin connected to the Arduino. It receives a reflected wave that is sent to the Arduino through the echo pin. The servo moves the ultrasonic sensor from left to right in a range of 100 degrees. The sensor takes the input from the entire range and sends it to the Arduino. The Arduino , proportional to the degree of change in the servo moves for a set amount of time(The degree of change in the servo is calculated based on if the ultrasonic sensor reads any obstacles in that angle).Once the move is completed it loops back to the perform FFT function and repeats the sound sensing algorithm.

Arduino-Setup:





Detailed explanation of programming code :

```
// Motor A(RIGHT)
int enA = 9;
int in1 = 8;
int in2 = 7;
// Motor B(LEFT)
int enB = 3;
int in3 = 5;
int in4 = 4;
//-----------------------------------------------------------
//SERVO INISIALISATION
#include <Servo.h>
int trigPin = 11;
int echoPin = 12;
int pos = 50, x, y;
float duration_us, distance_cm;
Servo myservo;//OBJCET MYSERVO
float location[100];
//----------------------------------------------------------
//FFT INISIALISATION
#include "arduinoFFT.h"
const uint16_t samples = 128;
const double samplingFrequency = 44100;  // Sample rate of your sound sample
unsigned int samplingPeriod;
unsigned long microSeconds;
double vReal[samples];
double vImg[samples];
ArduinoFFT<double> FFT = ArduinoFFT<double>(vReal, vImg, samples,
samplingFrequency);
const int readingsCount = 10;
double maxf;
int f;
double fu[3];

//-----------------------------------------------------------
void setup() {
  // put your setup code here, to run once:
  // Set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
```

```
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  //BAUD RATE
  Serial.begin (9600);
  //CONNECT INPUT AND OUTPUT PINS
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  //SERVO TO PIN 6
  myservo.attach(6);
  myservo.write(50);
  samplingPeriod = round(1000000 * (1.0 / samplingFrequency));
}
//--------------------------------------------------------
//FFT FUNCTION
double performFFT(double desiredFrequency, int n) {
    double sum = 0.0;

    for (int j = 0; j < readingsCount; j++) {
        for (int i = 0; i < samples; i++) {
            microSeconds = micros();
            vReal[i] = analogRead(n);
            vImg[i] = 0;
            while (micros() < (microSeconds + samplingPeriod)) {
            }
        }

        FFT.windowing(vReal, samples, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
        FFT.compute(vReal, vImg, samples, FFT_FORWARD);
        FFT.complexToMagnitude(vReal, vImg, samples);

        int desiredBin = (desiredFrequency * samples) / samplingFrequency;
        for(int i=0;i<=10;i++){
        sum += vReal[desiredBin];}
    }

    return sum / readingsCount; // Return the moving average
}
```

```
//-----------------------------------------------------------
//MOTOR FUNCTION
int motor(int f){
  analogWrite(enA, 255);
  analogWrite(enB, 255);
  if(f==0){
  // Turn on motor A & B
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(1000);

  // Now change motor directions


  // Turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);}
  else if(f==1){
  // Turn on motor A & B
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  delay(1000);

  // Now change motor directions


  // Turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);}

  //else{
    // Turn on motor A & B
  //digitalWrite(in1, HIGH);
  //digitalWrite(in2, LOW);
  //digitalWrite(in3, LOW);
  //digitalWrite(in4, HIGH);
```

```
//delay(9800);

// Now change motor directions


// Turn off motors
//digitalWrite(in1, LOW);
//digitalWrite(in2, LOW);
//digitalWrite(in3, LOW);
//digitalWrite(in4, LOW);}
}
//---------------------------------
//motor forward
int motorfor(){
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(1000);

// Now change motor directions


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
//---------------------------------------------
//SERVO-MOTOR
int motorservo(int y){
  if(y>=0 && y<=20){
    // Turn on motor A & B
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(2000);

// Now change motor directions


// Turn off motors
```

```
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);

//straight for 3sec
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(3000);


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);

digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(2000);

// Now change motor directions


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
else if(y>20 && y<50){
   // Turn on motor A & B
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(1500);

// Now change motor directions
```

```
// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);

//straight for 3sec
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(2500);


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);

digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(1500);

// Now change motor directions


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
else if(y>50 && y<=80){
    // Turn on motor A & B
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(1500);
```

```
// Now change motor directions


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);

//straight for 3sec
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(2500);


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(1500);

// Now change motor directions


// Turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
else{
 digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(2000);
```

```
  // Now change motor directions


  // Turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);

  //straight for 3sec
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(3000);


  // Turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);

  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(2000);

  // Now change motor directions


  // Turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  }
  }
void loop() {
  // put your main code here, to run repeatedly:
  double desiredFrequency = 2000;
  for(int n = 0; n < 3; n++) {
```

```
    fu[n] = performFFT(desiredFrequency, n);
    }

  maxf = fu[0];
  f = 0;
  for(int i = 0; i < 2; i++) {
    if(maxf < fu[i+1]){
       maxf = fu[i+1];
       f = i+1;
      }
    }
  motor(f);
  myservo.write(50);
  digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration_us = pulseIn(echoPin, HIGH);
    distance_cm = duration_us/58;
  if(distance_cm<30){
    for (pos = 0; pos <= 100; pos += 1) {
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration_us = pulseIn(echoPin, HIGH);
    distance_cm = duration_us/58;
    location[pos]=distance_cm;
    //Serial.print(pos);
    //Serial.print(" ");
    //Serial.println(location[pos]);
    myservo.write(pos);
    delay(50);
  }
   x=location[0];
  y=0;
  for (pos = 1; pos <= 100; pos += 1){
    if(x<location[pos]){
      x=location[pos];
      y=pos;
    }
  }
  motorservo(y);
  }
  else{
    motorfor();
```

```
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration_us = pulseIn(echoPin, HIGH);
    distance_cm = duration_us/58;
  }
}
```

Here's a breakdown of the code:

1. **Motor Control Setup:**

   - Motor control pins are initialized.

   - Motors are turned off initially.

2. **Serial Communication Setup:**

   - Baud rate is set to 9600 for serial communication.

3. **Ultrasonic Sensor Setup**:

   - Pins for the ultrasonic sensor (trigPin and echoPin) are initialized.

4. **Servo Setup:**

   - A servo motor is initialized on pin 6 and set to the initial position of 50.

5. **FFT Setup:**

   - ArduinoFFT library is included.

   - Parameters for FFT are defined (sample size, sampling frequency, etc.).

6. **performFFT() Function:**

- This function performs FFT on the input signal (analog reading from a sensor) to find the desired frequency.

- It calculates the magnitude of the frequency component corresponding to the desired frequency.

7. **Motor Control Functions:**

- `motor ()` function controls the motors based on the frequency detected.

- `motor servo()` function controls the motors and servo based on the detected frequency and obstacle avoidance.

8. **loop () Function:**

- It continuously executes the main logic.

- FFT is performed to detect the desired frequency (2000Hz).

- The maximum magnitude frequency component is determined.

- Motors are controlled based on the detected frequency and obstacle avoidance.

- Ultrasonic sensor measures the distance to obstacles.

- If an obstacle is detected (distance < 30 cm), the servo rotates to find the clear path, and the robot moves accordingly.

9. **Obstacle Avoidance:**

- When an obstacle is detected, the servo sweeps to find the direction with the maximum clear distance.

- Based on the direction with the maximum clear distance, the robot moves forward, backward, or turns to avoid obstacles.

Overall, the code integrates motor control, sensor input, servo control, and signal processing techniques like FFT to create a robot that can navigate towards a specific frequency while avoiding obstacles in its path.

RESULTS:

Outcomes:

In the presence of the required frequency, the two sound sensors in front of the robot will catch the signals. When comparing the signals from both sensors, one will have received more frequency than the other due to the shorter distance. This difference in frequency reception will make the robot move in the direction of higher frequencies. The servo and the ultrasound will ensure that the robot avoids obstacles on the way to the source. The servo motor will rotate the ultrasound sensor to check for obstacles in different directions, and the ultrasound sensor will detect the distance to the obstacle. If an obstacle is detected, the servo motor will rotate the sensor to check for other directions, and the robot will move in the direction with the fewest obstacles. This process will continue until the robot reaches the source of the frequency.

Challenges during the project:

- Motor wasn't getting enough power
- Due to lose connections many of the components were not operational
- Free range of motion was not being attained by server motor and ultrasound during wiring problems.
- Subpar wheels make it difficult to create traction.
-

DISSCUSION AND CONCLUSION:

In conclusion, the use of sound sensors and ultrasonic technology in robots has proven to be an effective solution for detecting and navigating towards a specific frequency source. By comparing the signals from two sound sensors, the robot can determine the direction of a higher frequency and move toward it. The servo and ultrasonic sensors ensure that the robot avoids obstacles on its path to the source, enabling it to navigate complex environments with ease.

The use of sound sensors and ultrasonic technology in robots has led to improved maintenance practices, enhanced robotics capabilities, and more effective sensing solutions across various industries. By harnessing the power of sound and ultrasonic waves, robots can navigate complex environments with ease, detecting and avoiding obstacles along the way.\\

IMPROVEMENTS:

- Better microphones can be added to detect sound waves outside the audible range (20Hz-20000Hz).
- Higer torque motors for different terrains
- IR sensors to detect terrains for irregularities
- Camera for visual feedback

REFERENCES:

B. Zhang, K. Masahide and H. Lim, "Sound Source Localization and Interaction based Human Searching Robot under Disaster Environment," *2019 SICE International Symposium on Control Systems (SICE ISCS)*, Kumamoto, Japan, 2019, pp. 16-20, doi: 10.23919/SICEISCS.2019.8758766.

Chen, Guoliang Xu, Yang 2019 ,A Sound Source Localization Device Based on Rectangular Pyramid Structure for Mobile Robot,463985,2019

Varma, Arjun & A, Ashwath & Verma, Ayush & Annasamy, Bagubali & Krishnan, Kishore. (2019). Implementation of Obstruction Avoiding Robot using Ultrasonic Sensor and Arduino UNO. International Journal of Recent Technology and Engineering (IJRTE). 8. 7335-7339. 10.35940/ijrte.D5295.118419.

Kim, Pil & Park, Chang & Jong, Yun & Yun, Jea & Mo, Eun & Kim, Chin-Su & Jie, Min & Hwang, Soo & Lee, Kang. (2007). Obstacle Avoidance of a Mobile Robot Using Vision System and Ultrasonic Sensor. 4681. 545-553. 10.1007/978-3-540-74171-8_54.

Obstacle Avoiding Robot (youtube.com)

Sound-Following Arduino Robot | Science Project (youtube.com)

https://www.fujielectric.com/about/column/detail/servo_01.html#01

https://www.tinkercad.com/things/4R0DI6Zpz96-smashing-bombul-uusam

https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

https://circuitdigest.com/microcontroller-projects/obstacle-avoiding-robot-using-arduino-and-servo-motor