Sai Adithyaa Basavaboina

Mechanical Engineering (B.Tech, 2018–2022) — Robotics & UAV Propulsion (CFD)

Anurag Group of Institutions, Hyderabad (Department of Mechanical Engineering)

Contact: [email] • [phone] • [LinkedIn/GitHub]

Profile

Mechanical engineering graduate focused on robotics and aerial propulsion. Built a working ESP32-CAM–based multipurpose surveillance robot (hardware + firmware) and completed a literature-backed CFD study of quadrotor propeller aerodynamics. Hands-on with AutoCAD, Arduino (embedded C), sensor integration and ANSYS Fluent.

Core Skills

- CAD/CAE: AutoCAD; ANSYS Fluent (k–ω SST, overset/rotating mesh); CATIA
- Programming & Boards: Arduino IDE (embedded C), ESP32-CAM, FTDI USB–TTL
- Electro-mechanical: DC motors, relays, IR & metal sensors, power systems (12V, 7.5Ah)
- Other: Basic circuit design, prototyping, testing, documentation

Selected Projects

Projectile Motion — Numerical & Analytical Study (ME 5890, Apr 2024)

- Solved projectile motion with and without linear air resistance using Laplace transforms; derived closed-form expressions for range/height and verified parabolic trajectory.
- Implemented MATLAB simulations to generate trajectories and visualize effects of complementary angles (θ and 90°−θ) on equal sub-maximum range.
- Worked examples at v0=300 ft/s for θ=38° and θ=52°; compared analytical results with numerically simulated paths to validate theory.
- Deliverables: clean derivation write-up, MATLAB scripts, and publication-quality plots.

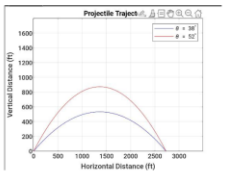


Fig:1 Projectile without air resistance

4. With Linear Air Resistance

4.1 Solving the Modified System:

The system of linear differential equations describing the motion of the projectile with air resistance, we'll use Laplace transforms. The given equations are:

$$m\frac{d^2x}{dt^2} = -\beta\frac{dx}{dt}$$
$$m\frac{d^2y}{dt^2} = -mg - \beta\frac{dy}{dt}$$

We'll apply Laplace transforms to both equations and then solve for $X(s)$ and $Y(s)$, which represent the transformed functions of $x(t)$ and $y(t)$, respectively. Then, we'll invert the Laplace transforms to obtain $x(t)$ and $y(t)$.

Step 1: Take the Laplace Transform
Taking the Laplace transform of equations (5) and (6), and applying the initial conditions $x(0) = 0, x'(0) = v_0\cos\theta, y(0) = 0$, and $y'(0) = v_0\sin\theta$, we get:

$$ms^2X(s) - msx(0) - mx'(0) = -\beta sX(s) + \beta x(0)$$
$$ms^2Y(s) - msy(0) - my'(0) = -mg - \beta sY(s) + \beta y(0)$$

Substituting the initial conditions and rearranging, we get:

$$ms^2X(s) - v_0\cos\theta = -\beta sX(s)$$
$$ms^2Y(s) - v_0\sin\theta = -mg - \beta sY(s)$$

Step 2: Solve for $X(s)$ and $Y(s)$
Solving for $X(s)$ and $Y(s)$, we get:

Projectile Motion — Figure 1

$$X(s) = \frac{v_0\cos\theta}{s(ms+\beta)}$$
$$Y(s) = \frac{v_0\sin\theta}{s(ms+\beta)} - \frac{mg}{s(ms+\beta)}$$

Step 3: Invert the Laplace Transform
Now, we'll invert the Laplace transforms to obtain $x(t)$ and $y(t)$. This involves finding the inverse transforms of $X(s)$ and $Y(s)$, which we can do using standard Laplace transform tables or software.

To find $x(t)$ and $y(t)$, we need to compute the inverse Laplace transforms of $X(s)$ and $Y(s)$. Using standard Laplace transform tables or software, we find:

$$\mathcal{L}^{-1}\left\{\frac{v_0\cos\theta}{s(ms+\beta)}\right\} = \frac{v_0\cos\theta}{m}\left(1 - e^{-\frac{\beta}{m}t}\right)$$
$$\mathcal{L}^{-1}\left\{\frac{v_0\sin\theta}{s(ms+\beta)} - \frac{mg}{s(ms+\beta)}\right\} = \frac{v_0\sin\theta}{m}\left(1 - e^{-\frac{\beta}{m}t}\right) - \frac{mg}{\beta}t$$

These are the expressions for $x(t)$ and $y(t)$ in terms of v_0, θ, m, β, and t, where e is the base of the natural logarithm.

4.2 Calculation:

Now, you can substitute the given values of v_0, θ, m, and β to get the specific solutions for $x(t)$ and $y(t)$.

To compute the specific solutions for $x(t)$ and $y(t)$, we'll substitute the given values:

- $v_0 = 300\text{ft/s}$
- $\theta = 38°$
- m = mass of the projectile
- β = constant proportional to air resistance into the expressions we derived previously:

$$x(t) = \frac{v_0\cos\theta}{m}\left(1 - e^{-\frac{\beta}{m}t}\right)$$
$$y(t) = \frac{v_0\sin\theta}{m}\left(1 - e^{-\frac{\beta}{m}t}\right) - \frac{mg}{\beta}t$$

Given that $\theta = 38°, v_0 = 300\text{ft/s}$,

1. For $\theta = 38°, v_0 = 300\text{ft/s}, m = \frac{1}{4}$ slug, $g = 32\text{ft/s}^2$, and $\beta = 0.02$: Substituting these values into the expressions:

Projectile Motion — Figure 2

Arduino Tuner — Audio Frequency Analyzer & Uncertainty (ME 5890, May 2023)

- Built an Arduino-based audio acquisition system (MAX4466 electret mic + SD module) to capture 10-s tones at 110, 261 and 3520 Hz; 35 readings per tone in a controlled room (source at 5 cm).
- Processed spectra and statistics in MATLAB; confirmed near-normal samples via Q-Q plots; performed two-tailed t-tests at 95% CI to evaluate sensor accuracy vs. true tone.
- Found that random error grows with frequency while systematic error is lowest near the speech band (~100–250 Hz); overall ≤0.01% full-scale error at low/mid frequencies and <1% at high frequency.
- Documented limitations (UNO 10-bit ADC and serial link throughput) and mitigations (SD logging at 16 kHz).

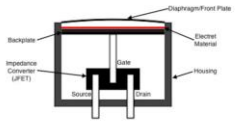


Figure 2: Diagram of an electret microphone

Once an user presses the push button, the LED light will turn on and the sensor will start recording. Due to sampling limitations, the sound signal is saved int an SD card to maximize sampling. The signal is then transferred into Matlab for spectral analysis. A schematic of the sensor is presented below:

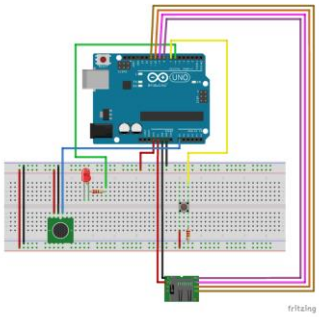
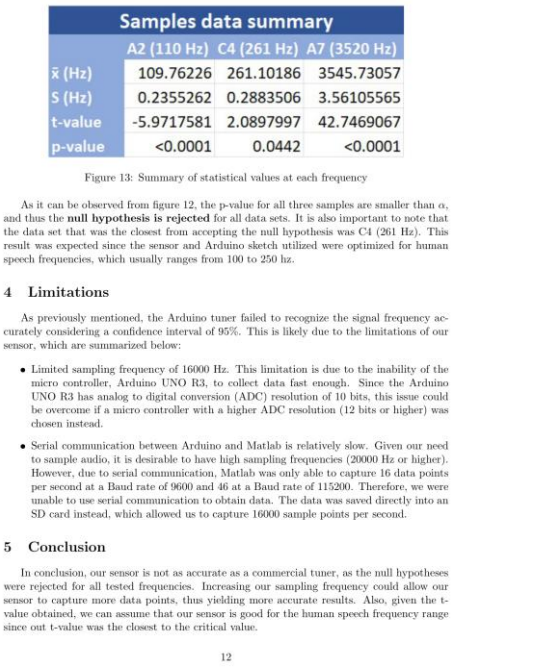


Figure 3: Schematic of Arduino tuner

The electret microphone features adjustable gain, 25x to 250x, which is controlled via a screw on the board. However, the adjustable gain did not affect the sound capturing capabilities of our sensor.

5

Arduino Tuner — Figure 1



Samples data summary			
	A2 (110 Hz)	C4 (261 Hz)	A7 (3520 Hz)
$\bar{x}$ (Hz)	109.76226	261.10186	3545.73057
S (Hz)	0.2355262	0.2883506	3.56105565
t-value	-5.9717581	2.0897997	42.7469067
p-value	<0.0001	0.0442	<0.0001

Figure 13: Summary of statistical values at each frequency

As it can be observed from figure 12, the p-value for all three samples are smaller than α, and thus the **null hypothesis is rejected** for all data sets. It is also important to note that the data set that was the closest from accepting the null hypothesis was C4 (261 Hz). This result was expected since the sensor and Arduino sketch utilized were optimized for human speech frequencies, which usually ranges from 100 to 250 hz.

4 Limitations

As previously mentioned, the Arduino tuner failed to recognize the signal frequency accurately considering a confidence interval of 95%. This is likely due to the limitations of our sensor, which are summarized below:

- Limited sampling frequency of 16000 Hz. This limitation is due to the inability of the micro controller, Arduino UNO R3, to collect data fast enough. Since the Arduino UNO R3 has analog to digital conversion (ADC) resolution of 10 bits, this issue could be overcome if a micro controller with a higher ADC resolution (12 bits or higher) was chosen instead.
- Serial communication between Arduino and Matlab is relatively slow. Given our need to sample audio, it is desirable to have high sampling frequencies (20000 Hz or higher). However, due to serial communication, Matlab was only able to capture 16 data points per second at a Baud rate of 9600 and 46 at a Baud rate of 115200. Therefore, we were unable to use serial communication to obtain data. The data was saved directly into an SD card instead, which allowed us to capture 16000 sample points per second.

5 Conclusion

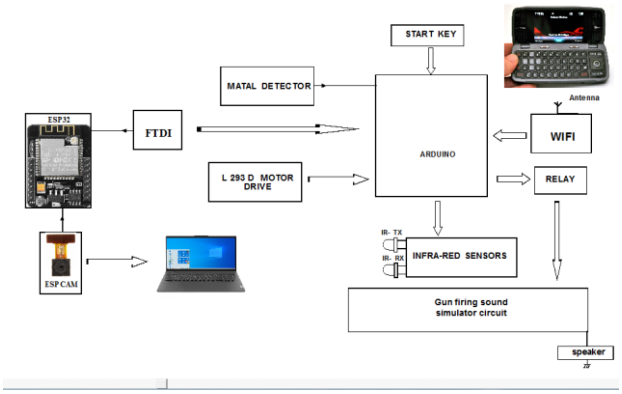
In conclusion, our sensor is not as accurate as a commercial tuner, as the null hypotheses were rejected for all tested frequencies. Increasing our sampling frequency could allow our sensor to capture more data points, thus yielding more accurate results. Also, given the t-value obtained, we can assume that our sensor is good for the human speech frequency range since out t-value was the closest to the critical value.

12

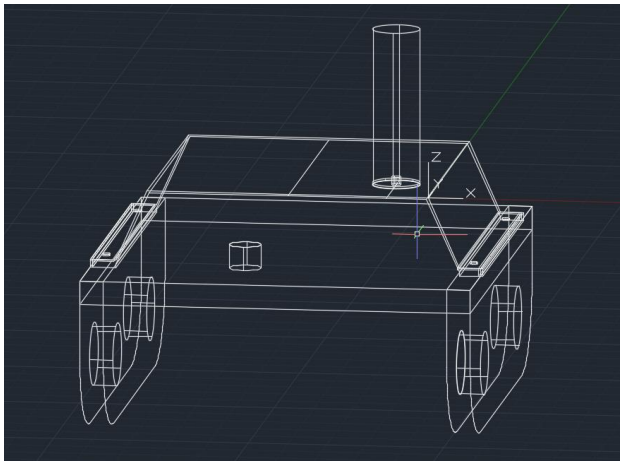
Arduino Tuner — Figure 2

Modeling & Fabrication of a Multipurpose Surveillance Robot (Major Project, Jun 2022)

- Modeled a 3D chassis in AutoCAD and fabricated a metal frame; integrated ESP32-CAM for live streaming with Wi-Fi/Bluetooth control.
- Implemented drive (DC motors), sensors (IR motion, metal detection), and relay-based actuators; designed wiring harness and power system around 12 V/7.5 Ah battery.
- Prototyped, assembled and validated the robot; documented schematics, bill of materials, and test runs.



Robotics — Figure 1



Robotics — Figure 2

Turbulence Model of Drone Propeller (Mini Project — Dec 2021)

- Modeled a DJI-Spark-style propeller in CATIA; exported STEP; built fluid domain and ~200k-element mesh in ANSYS.
- Ran Fluent cases at 500/2500/5000 RPM; post-processed pressure contours, velocity vectors/streamlines and turbulence kinetic energy to study thrust trends.


Certificate UID : SRI-OT01-214913

certificate
of completion

This is to certify that

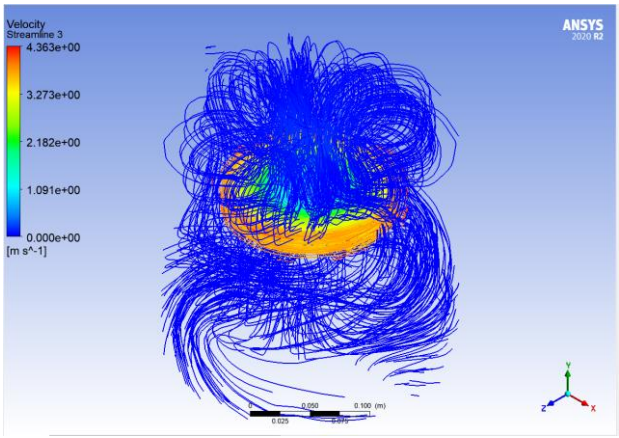
Basavaboina Sai Adithyaa

From Anurag Group of Institutions has actively participated and completed the project work on Turbulence model of Drone Propellor During his/her Internship held from 7th Sep 2021 to 5th Oct 2021 at Skyy Rider Institutions.

Program Manager

CEO
Skyy Rider Institutions

Mini Project — Figure 1



Mini Project — Figure 2

Programming/Tools: MATLAB; Arduino IDE (embedded C); ESP32-CAM

Education

B.Tech in Mechanical Engineering — Anurag Group of Institutions, Hyderabad • Major Project submitted June 2022