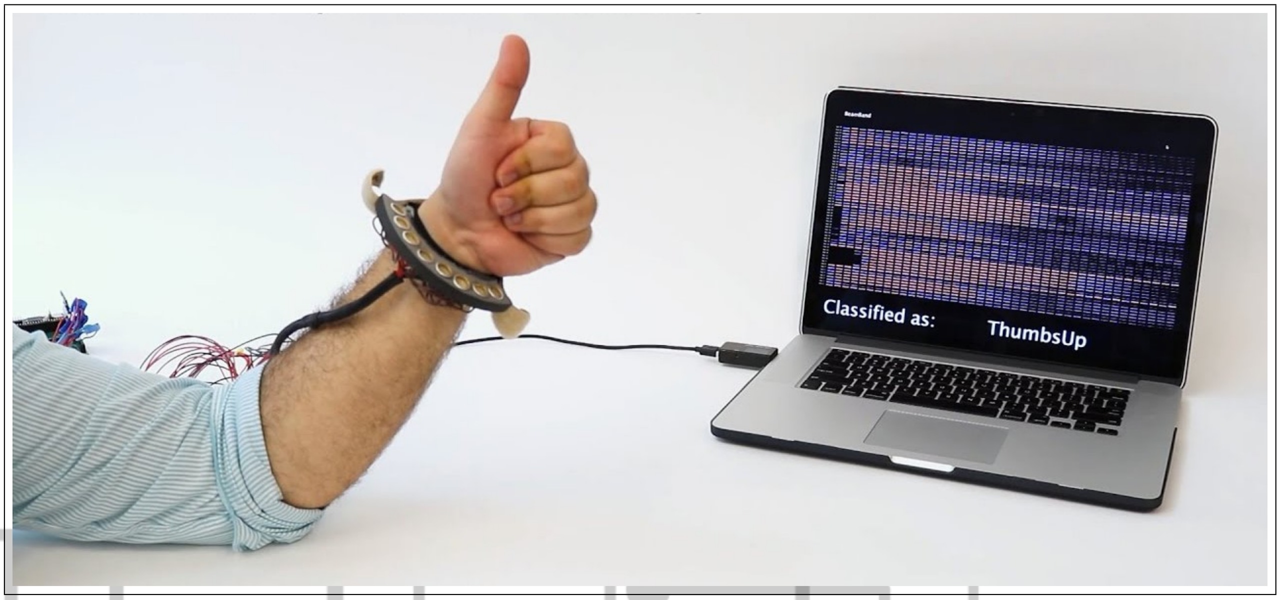
Indian Institute of Technology, Kanpur

Science and Technology Council

AUDIOBAND

Hand Gesture Sensing with Ultrasonic Beamforming



Final Report

Contents

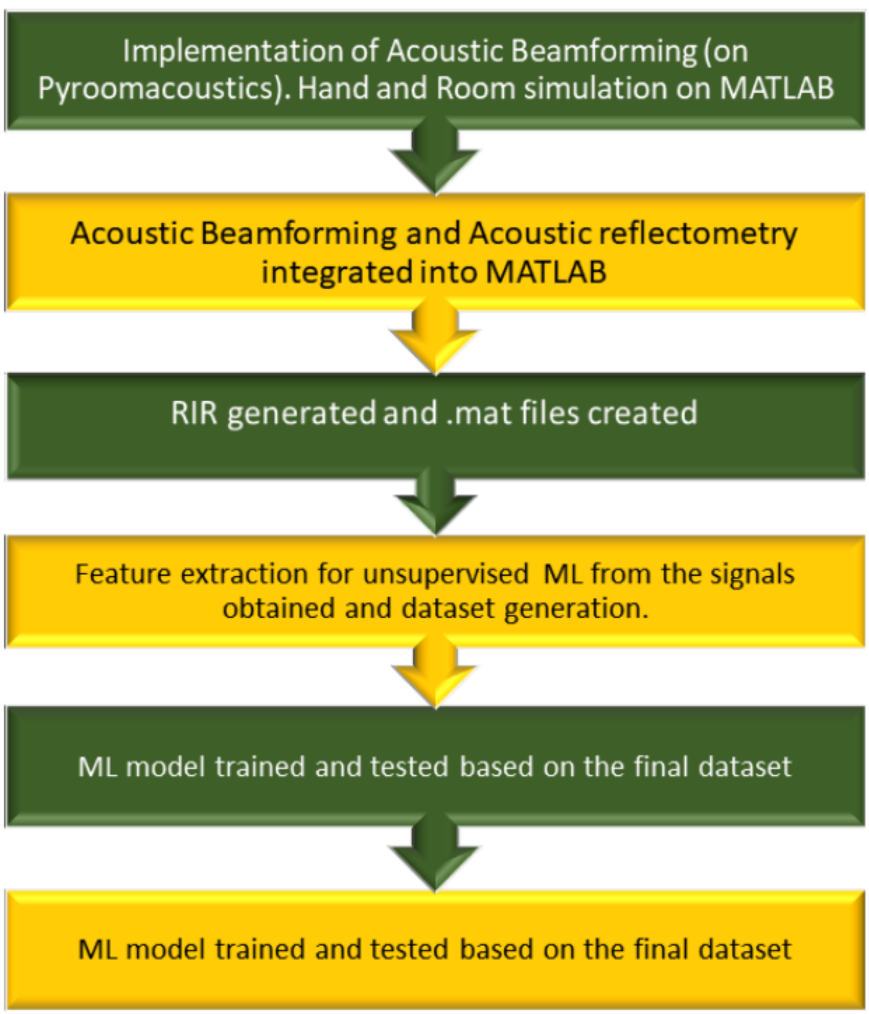
1. Acknowledgement

Our team would like to express their gratitude to Mr. Mudit Agrawal, General Secretary, Science and Technology Council, Student's Gymkhana, for giving us the golden opportunity to do this project. We are highly indebted to the Electronics Club and its coordinators Mr.Afzal Rao, Mr.Anshul Rai, Mr.Netravat Pendsey, and Mr.Utkarsh Gupta for providing us invaluable guidance and support in doing this project.

2. Inroduction

The main objective of the project is to achieve hand gesture sensing leveraging the Ultrasonic-Beamforming and Acoustic Reflectometry technique. This involves the recognition of a signal set consisting of a fixed number of specific signals received through reflection at hand. We inspect the geometry of the hand and the placement of fingers with reference to three axes, two at the palm and one at fingers, subsequently leading to the identification of the gesture. The project thus aims to provide an insight into the human-machine interaction system with the robust implementation of the required framework, consisting of proper simulations and processing. In a broader aspect, this can be used separately or integrated into smartwatches or controllers for better user interaction.

WORKFLOW

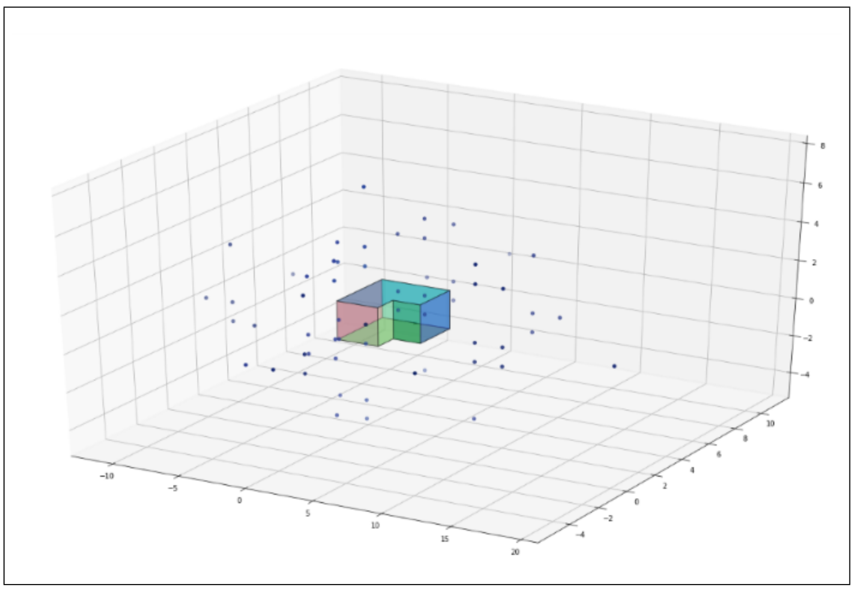


3. Initial Tasks

3.1 Room Simulation

3.1.1 On Pyroomacoustics

We used the Pyroomacoustics library of Python to create a virtual environment for reflection and Beamforming of the signals emitted from the transducers set on our Audioband. Here we used the shoebox() and from_corners() functions to create 3D rooms. The transducer array was set inside the room using add_source() and add_microphone_array() functions.

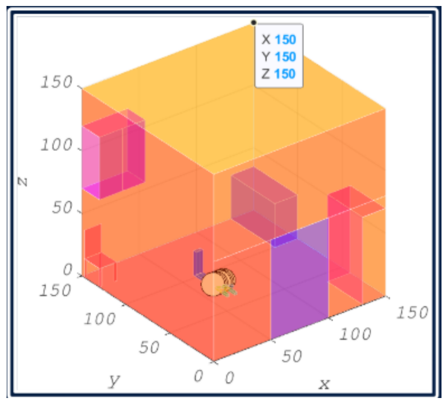


Simulation using Pyroomacoustics

During virtual environment creation on pyroomacoustics, we came across the problem that we cannot create objects or hand gestures inside the room. This was a significant issue for our simulation. Thus we shifted our work of simulation on Matlab and used its Indoor Ultrasonic Simulator Toolbox.

3.1.2 On MATLAB

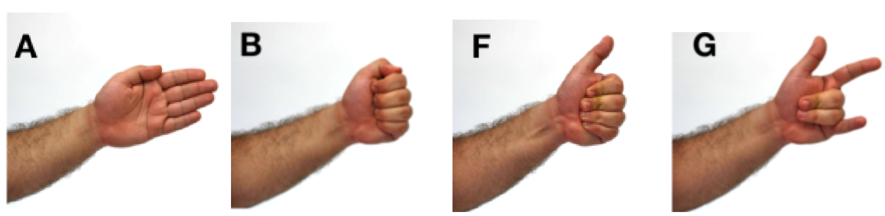
Surface Modelling:
We have used MATLAB Indoor Simulator toolbox for creating a virtual room reflectors for the signals emitted from the transducer array. The addWall function used can create quadrilateral surfaces by specifying four coordinates along with a reflection coefficient. To make a room, we require the addition of 6 surfaces, which is done by passing their coordinates to the addWall function.



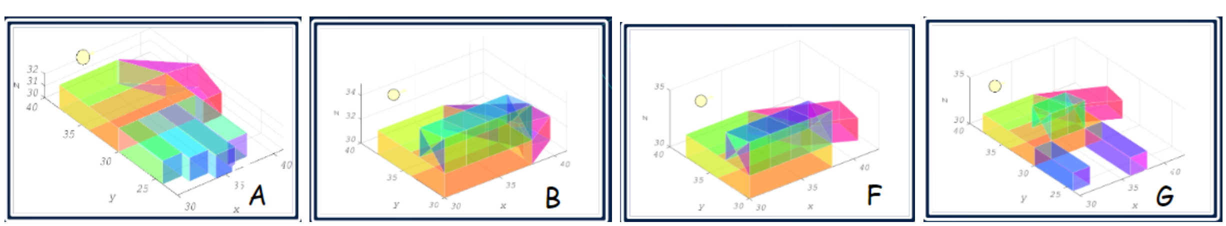
Our simulated room in MATLAB

3.2 Hand Simulation

We have adopted a set of four hand gestures, i.e., relax(A), fist(B), thumbs up(F), and spider-man(G) for our project. We simulated these gestures using the addWall function and setting the reflection coefficient same as that of our skin. The base of the wrist is kept at a distance of 2 units and the base of fingers at 8 units, as mentioned in the paper.



Hand gestures



Gestures simulated in MATLAB

[1]

[1]Link to Toolbox https://www.mathworks.com/matlabcentral/fileexchange/61364-indoor-ultrasonic-simulator

3.3 Beamforming

The transducer array sends ultrasonic beams to the hand, which on reflection reaches the receiver. The input signal received is transferred to the recognition model after further processing. The set of the linearly placed 8 transducer array iterates through a cycle in which each one of these transducers becomes the receiver, and the rest 7 are transducers. The process consists of two parts: the acoustic reflectometry at the fingers and the beamforming by the 7 transducers.

For beamforming, we performed an online simulation on Google Collaboratory using the room simulator: pyroomacoustics. 7 kinds of beams are produced from the source array: 5 beams steered at angles 0, 22.5,45,-22.5,-45 and 2 beams focused at 2 cm and 8 cm from the centroid of the linear array. We performed the mathematics to achieve such configurations through initial phase differences between the waves produced at each source. We validated the mathematics used using a microphone near the source array and making conclusions based on intensity plots of the signal obtained using various libraries. We emitted radial planar sinusoidal waves from each of the sources of the source array.

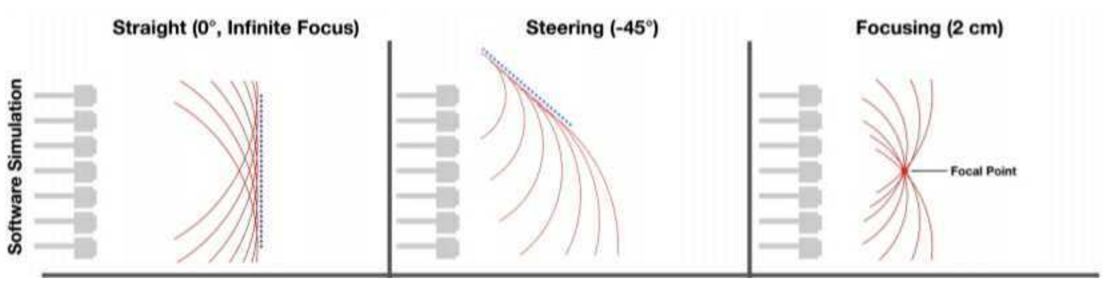


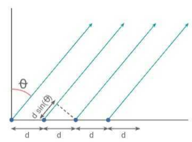
Figure showing the various beamforms we generated (Red curves join points where the total phase of the sine wave from the source in consideration has a fixed given value)

2

[2]Link to library `https://pyroomacoustics.readthedocs.io/en/pypi-release/`

3.3.1 Steered-Beam

For steered beams we put $\theta = 45, 22.5, 0, -22.5, -45$ and get the successive phase differences between the adjacent pairs of sources. We give an initial phase 0 to the 1st source and add ψ to the initial phase of each succeeding source, thus producing the same beamform as denoted by the diagram above



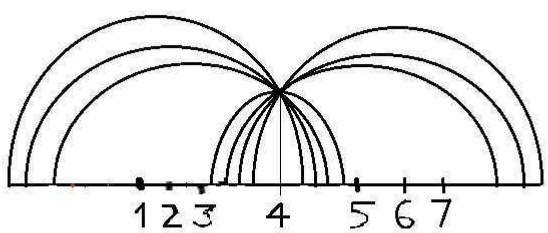
Application of antenna beam-forming as used with mobile or cellular telecommunications

$$\Psi = \frac{2\pi d}{\lambda}(\sin\theta)$$

Where

Ψ = phase difference between two adjacent beams.

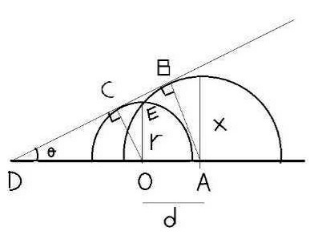
3.3.2 Focused-Beam



We have two distances to focus the beam at: 2cm and 8cm.

We first take the case of 2cm.

The centroid of the linear source array is the 4th source, hence the focus is situated at 2cm from the 4th source on the perpendicular bisector of the linear source array



Consider each of the other sources one by one. We find all points where the total phase of the sine wave from the source in consideration has the same value as the total phase due to the sine wave emitted by the fourth source at the focus. What we get is a pair of semicircles intersecting at the focus. We then draw the tangent to this pair of semicircles. That shall intersect the linear source array at an angle θ.

Calculations:

Consider a semicircle with centre at 4 and that at 5.

$$OC = r, AB = X$$
$$\angle CDO = \angle BDA = \theta$$
$$X = r + d\sin\theta$$

Equations of circles with centres of O and A

$$S_1 : x^2 + y^2 = r^2$$
$$S_2 : (x-d)^2 + y^2 = X^2$$

Equation of OE(common chord) is

$$S_1 - S_2 = 0$$
$$2dx - d^2 = r^2 - X^2$$

Common chord passes through O(0,0)

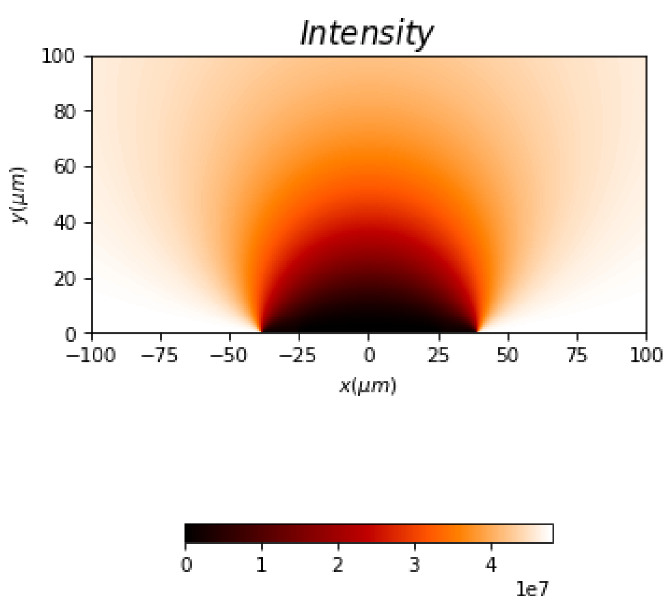
$$d^2 = X^2 - r^2$$
$$d^2 = (r + d\sin\theta)^2 - r^2$$
$$d^2\cos^2\theta = 2rd\sin\theta$$
$$r = \frac{d\cos^2\theta}{2\sin\theta}$$
$$\psi = \tfrac{2\pi d\sin\theta}{\lambda},$$

hence calculate ψ for the transducer pair $(4, 5)$

Similarly calculate θ for each pair and then ψ for each

Hence calculate θ for each pair and calculate ψ for each pair. Assign those values of ψ to the signals from each of the sources. Hence form the beam.

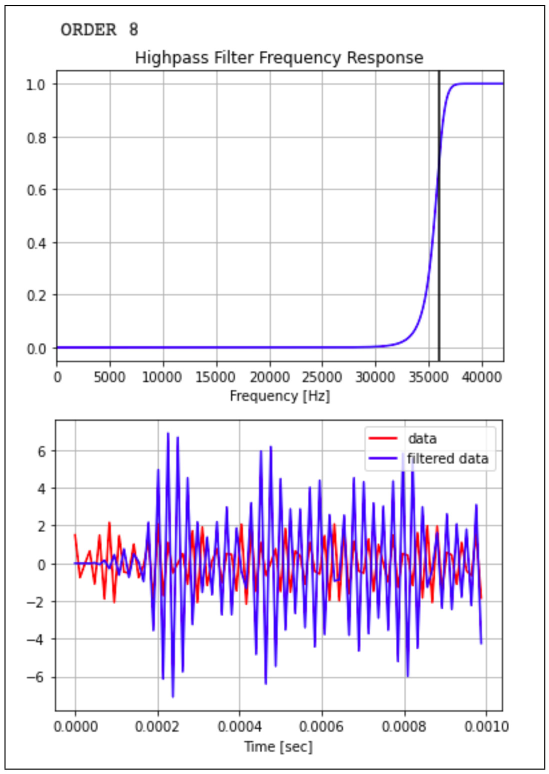
Intensity plots for the focused beams to validate our calculations (simulated using the module diffractio): Intensity plots will determine whether the beam has a focus at 2cm as the intensity will have a local maximum at that point.



Intensity plots for Focused Beams

3.4 Filtering of Signal

After obtaining data from room simulations , we pass the signals through an active high pass filter with fixed gain(cutoff frequency=39KHz and Gain=5) to remove the noise.
We use the signal processing library of Scipy. Scipy.signal.butter and scipy.signal.lfilter are used to create a high-pass filter which outputs digital data. We apply the filter with varying orders and plot the output for each of them. As the order increases, slope of frequency response graph increases. We prefer the filter with order 8.

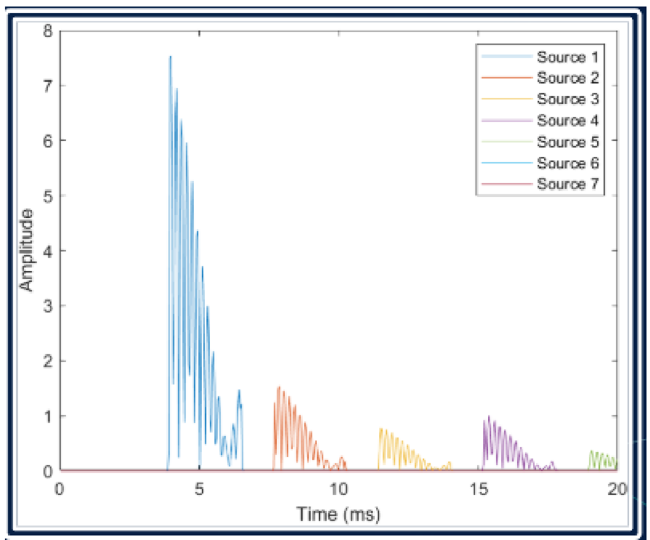


Graph of Filtering of Signals

4. Later Tasks

4.1 RIR Generation

We used ROOM IMPULSE RESPONSE (RIR) of the signals reflected from room and hand to distinguish the four hand gestures, i.e., Relaxed hand Gesture, Fist Gesture, Thumps Up gesture, Spiderman Gesture. In the MATLAB toolbox, RIR is generated by using the impr() function.

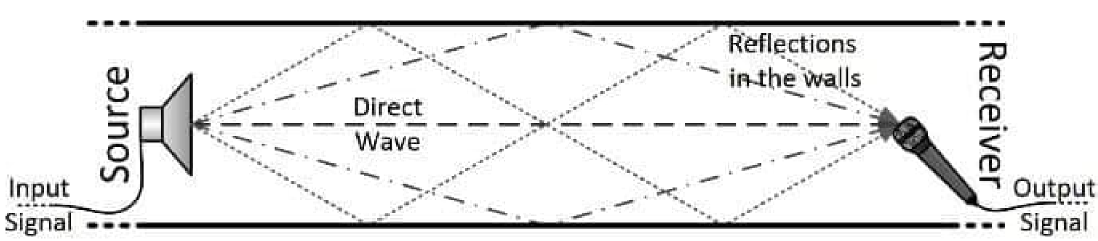


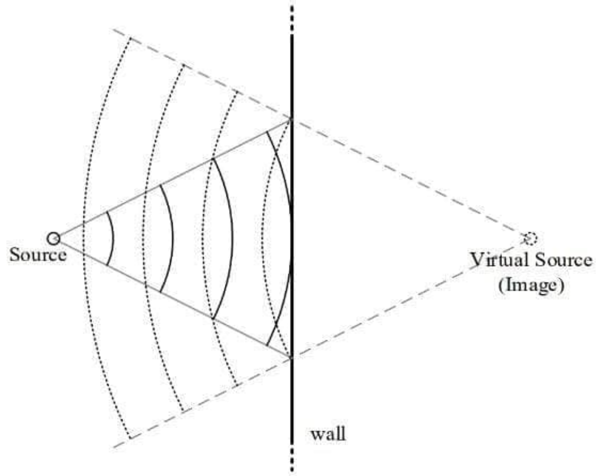
Room Impulse Response(RIR) Graph

The concept of RIR generation used in the toolbox is as follows:

4.1.1 Reflection Modelling

Ultrasonic wave travel is Omni-directional (i.e., it spreads radially in all directions), and whenever it finds some obstacle (e.g., wall, etc.), it reflects some of its portion of and rest is absorbed into the reflecting surface. The figure below shows an ultrasonic wave propagation in a room with two walls.



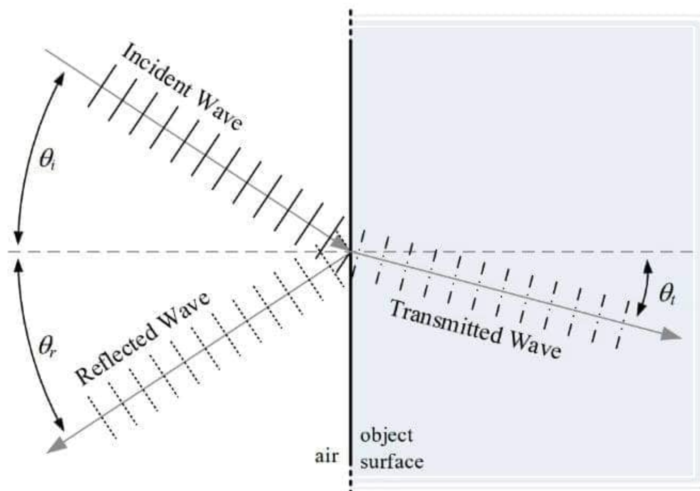


Ultrasonic wave reflection in a wall from a punctual source.

The reflection of the ultrasonic wave is similar to the way light is reflected from a mirror. So the well-known Snell's law of refraction can be applied here too to find the direction of propagation of sound wave after reflection within the room.

Snell's law is given as:

$$\theta_i = \theta_r$$
$$\frac{\sin\theta_i}{c_i} = \frac{\sin\theta_t}{c_t}$$



In the simulation, we discarded the transmitted wave due to its low energy. We assumed that this transmitted wave is completely absorbed by the reflection surface of the room and the object present on the path of the ultrasonic wave.
After the reflection in our simulation, we consider that the reflected wave has less energy than the incident wave, and this energy decreases even more with the increasing order of reflection and goes to zero after a certain number of reflections which is the same as the real-life scenario.

In our simulation, we even set the reflection coefficient for different objects present in the room (e.g., wall, ceiling, marble, carpet, wood, glass, skin, etc.), which indicates the amount of reflected sound from the surface. For some of the objects like wood, metal, plastic, glass, etc. the reflected wave has almost the same energy as that of the incident wave. This is because of the high reflection coefficient and for some soft objects like curtains, cork, carpet, etc. The reflected wave has very low energy due to the small reflection coefficient and thus a large amount of absorption of sound waves on these surfaces.

The absorbing coefficient of different objects used in our simulation is shown in the table below:

SOUND-ABSORBING COEFFICIENTS FOR VARIOUS MATERIALS

MATERIAL	125 Hz	250 Hz	500 Hz	1,000 Hz	2,000 Hz	40,000 Hz	NRC
Marble	0.01	0.01	0.01	0.01	0.02	0.02	0.00
Gypsum board, 1/2" (13 mm)	0.29	0.10	0.05	0.04	0.07	0.09	0.05
Wood, 1" (25 mm) thick, with air space behind	0.19	0.14	0.09	0.06	0.06	0.05	0.10
Heavy carpet on concrete	0.02	0.06	0.14	0.37	0.60	0.65	0.30
Acoustical tile, surface mounted	0.34	0.28	0.45	0.66	0.74	0.77	0.55
Acoustical tile, suspended	0.43	0.38	0.53	0.77	0.87	0.77	0.65
Acoustical tile, painted (est.)	0.35	0.35	0.45	0.50	0.50	0.45	0.45
Audience area: empty, hard seats	0.15	0.19	0.22	0.39	0.38	0.30	0.30
Audience area: occupied, upholstered seats	0.39	0.57	0.80	0.94	0.92	0.87	0.80
Glass fiber, 1" (25 mm)	0.04	0.21	0.73	0.99	0.99	0.90	0.75
Glass fiber, 4" (100 mm)	0.77	0.99	0.99	0.99	0.99	0.95	1.00
Thin fabric, stretched tight to wall	0.03	0.04	0.11	0.17	0.24	0.35	0.15
Thick fabric, bunched 4" (100 mm) from wall	0.14	0.35	0.55	0.72	0.70	0.65	0.60

If Absorbing coefficient is A and Reflection coefficient is R then:

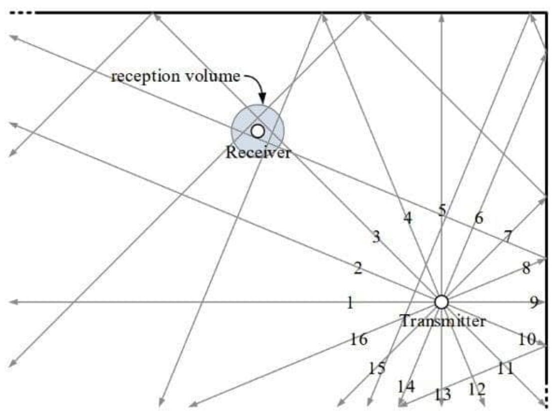
$$A = 1 - R^2$$

is used to find the reflection coefficient of the objects mentioned above.

There are two ways to simulate the acoustic reflections in large rooms and that are, 1) Ray Tracing Method and 2) Image Source Method :

Ray Tracing Method:

As mentioned previously that the ultrasonic wave is Omni-directional. It spreads all over the room, producing a large number of rays inside the room due to reflection from the objects present in the path. Hence, the ray tracing method uses a very high number of rays. Here the path of each ray is calculated, and then a certain volume is taken around the receiver, and all the rays passing through that volume are considered as received by the receiver. With this method, it isn't easy to obtain good accuracy.
Figure below shows an example of a transmitter and a receiver in a room with two walls. It is assumed that the source emits 16 rays and the receptor only receives ray 3 directly, rays 4 and 8 are reflected once and ray 7 twice. The resultant received wave is the combination of the four received waves.



Ray tracing method (example with 16 rays).

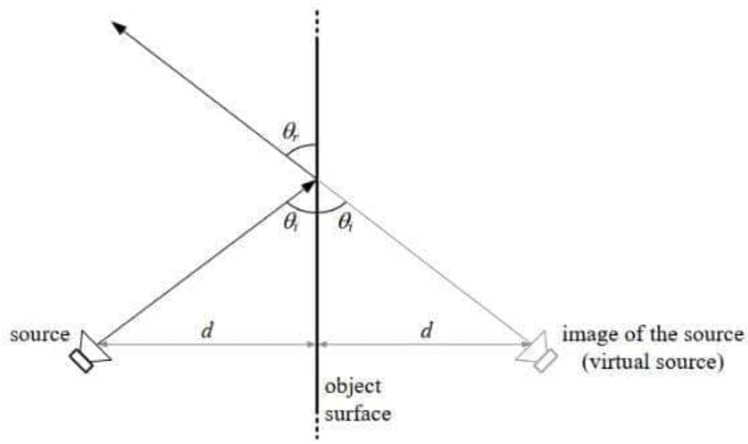
Due to large number of rays it is difficult to trace the path of each of them with good accuracy so there is high risk of receiving false reflections in this approach.
The probability of a ray to intersect a surface with area A after traveling t seconds with a wave front less than A/2 is high. Therefore, the minimum number of rays N_{rt} that the system must consider is:

$$N_{rt} \geq \frac{8\pi c^2}{A} t^2.$$

As a result of this the minimum number of rays necessary in a common room is usually very high.

Image Source Method:

The Image Source Method (ISM) uses the ultrasonic wave property that the wave reflects from the surface as a ray of light is reflected from the mirror. So in this approach we consider that the wave reflected from the surface is actually coming from a virtual source present at the same distance behind the surface of reflection as that of the actual sound source present in front of that.



Virtual Source Method principle ($\theta_i = \theta_r$).

Consider a rectangular room having volume V and the total number of virtual sources, N_{VS}, confined to a circumference with radius ct and centered in the source can be easily calculated using the formula:

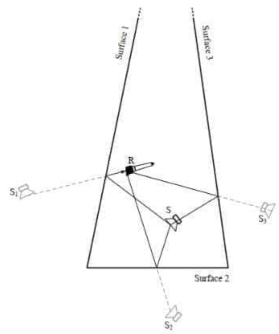
$$N_{vs} = \frac{4\pi c^3}{3V} t^3$$

Since like ray tracing method here we don't have to trace the path of each ray so it is relatively more accurate but the complexity of this method increases with the order of reflection as that will increase the number of virtual sources.

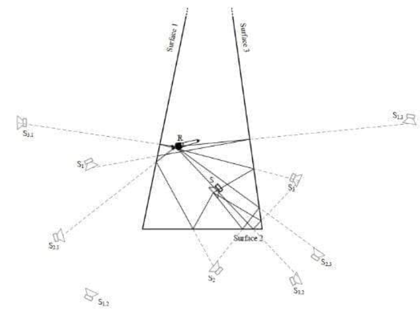
For example, In a room with N different surfaces, there are N virtual sources that produce reflections of first order and N - 1 that produce reflections of second order. Therefore, summing up all the total number of virtual sources N_T that create all the reflections up to order r in a room is:

$$N_T = 1 + \frac{N(N-1)^r - N}{N-2} \approx (N-1)^r.$$

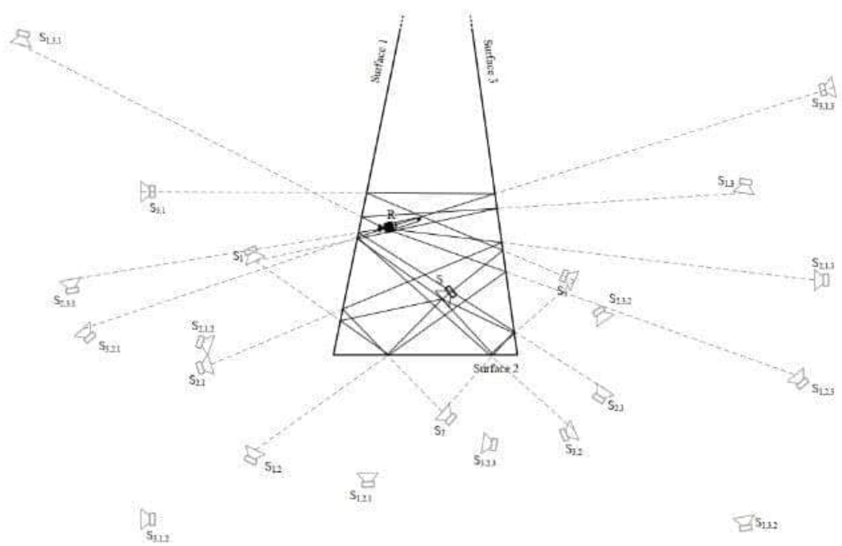
The figure below shows the complexity of virtual sources on increasing reflection order.



Example of Virtual Sources that produce reflections of first order.



Example of Virtual Sources that produce reflections of second order.



Example of Virtual Sources that produce reflections of third order.

4.2 Sorting Differences between Physical and Virtual Simulations

As cited in the limitations of a MATLAB simulation, the Room Impulse Response is used to form the MATLAB data file instead of the original signal at the receiver. The Room Impulse Response is essentially the original signal deconvolved into its constituents using the Image Source Model (ISM). During the physical simulation, the Room Impulse response can be generated from the original signal by means of deconvolution. The mathematics of the procedure to be used are described as follows:
The objective of deconvolution is to find the solution f of a convolution equation of the form:

$$f * g = h$$

Here, h is the recorded signal, and f is some signal that we wish to recover, but has been convolved with another signal g (which was formerly h in the previous stage of deconvolution), before we recorded it.
Deconvolution is then performed by computing the Fourier transform of the recorded signal h and the other function g. Deconvolution is then performed in the time domain (in the absence of noise) using:

$$F = H/G$$

where F, G, and H are the Fourier transforms of f, g, and h respectively. Finally, the inverse Fourier transform of the function F is taken to find the estimated deconvolved signal f.
We implement continuous deconvolution in stages on the received signal, until we receive all the constituents to finally form the RIR. Subsequent operations work identically as the online simulation.

4.3 Feature Extraction

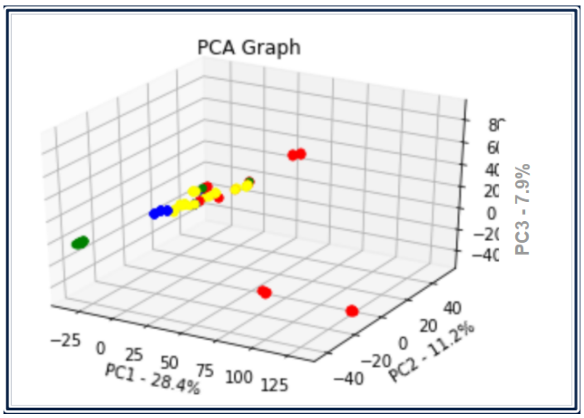
In the feature extraction process, the data stored as a 5D array of numbers in the MATLAB data file is extracted and unfolded into individual datasets of shape (8,7,7,500). Each signal generated, corresponding to eight arrangements of the transducer array, seven modes for each such arrangement, and seven sources transmitting the signals, is of size 500.

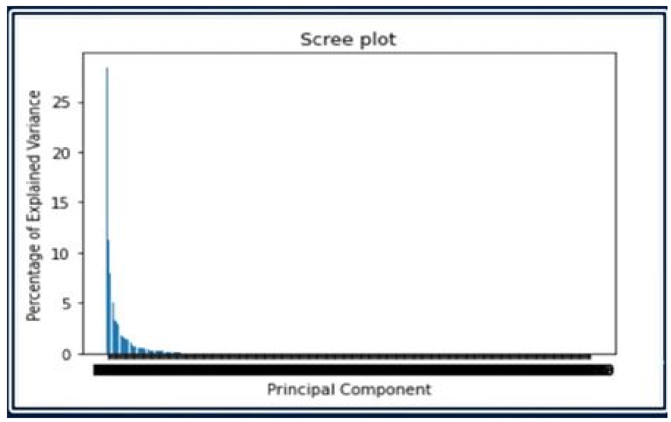
We performed a series of operations on these signals which involved binning them into 20 bins and obtaining the standard deviation of each. The standard deviations corresponding to each such signal is collected in an array. All these arrays are merged into one through concatenation, and that merger gets written as a row in a CSV file with the label of the gesture it belongs to. The CSV file thus created is fed into the Machine Learning model for training and gesture recognition.

4.4 Data Visualisation

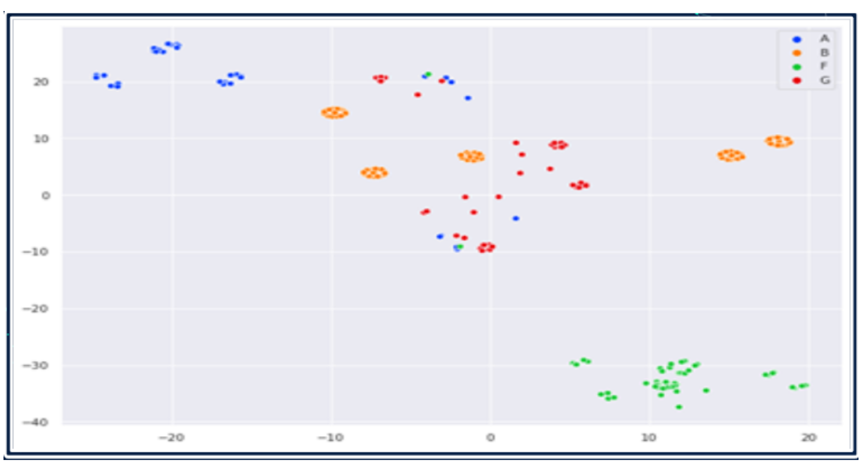
Since our data have 7840 features, visualisation of which requires the implementation of Machine Learning dimensionality reduction algorithm **Principal Component Analysis** (PCA) and **t-Distributed Stochastic Neighbor Embedding**(t-SNE) which helps us in visualising the data using two or three features out of 7840 features that have the maximum property of dataset and is able to explain most of the characteristics. The impact of different features of our dataset can be seen from the **scree plot**.

PCA is a linear algorithm and will not be able to interpret complex polynomial relationship between features whereas t-SNE is based on probability distributions with random walk on neighborhood graphs to find the structure within the data.





t-SNE Plot



4.5 Machine Learning Model

After the RIR generation and the feature extraction part, our task was to classify the four gestures from these datasets and thus was the need to train the Machine Learning model. We used four classification algorithms for our tasks that were K Nearest Neighbours, Random Forest, Support Vector Machine, and Neural Networks.

Out of these four algorithms, we got the best results with Neural Networks and thus used this on our user interface. A Neural Network model with one hidden layer was used for Classification. Average accuracy of 98.6% is reported on performing 5-fold cross validation.

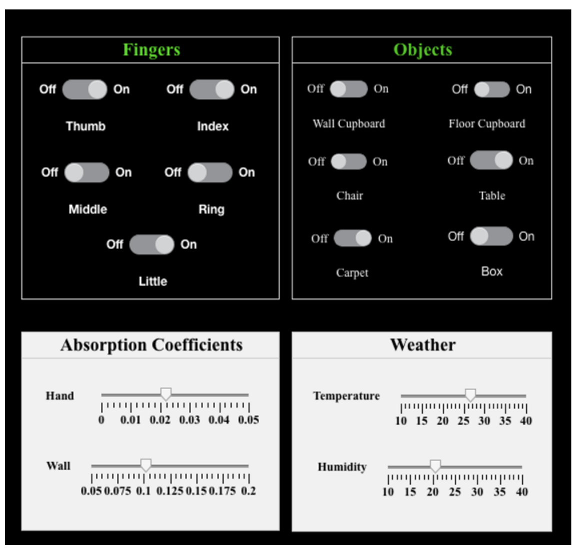
ML MODEL TABLE

Classification Model	Description	Average Score
Neural Networks (NN)	Neural networks can adapt to changing input so the network generates the best possible result without needing to redesign the output criteria.	98.66
K Nearest neighbours (KNN)	In KNN we indentify the k nearest neighbours around the sample and classify it to the class having maximum votes.	97.74
Support Vector Machine (SVM)	It uses a technique called the kernel trick to transform the data and then based on these transformations it finds an optimal boundary.	98.82
Random Forest (RF)	Here we estimate probabilities by making a class prediction for each tree and counting the fraction of trees that vote for a certain class.	98.86

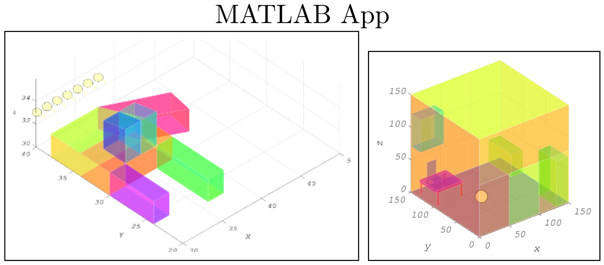
On verification of models on new test data we get the best results by NEURAL NETWORKS so we proceeded with this classification algorithm on user interface.

5. User Interface (MATLAB App)

A MATLAB app designed by us allows the user to design a variety of hand gestures by relaxing and folding each of the fingers. The obstacles around the hand in a typical real-world room may include tables, chairs, cupboards and the walls. The app provides to the user the functionality to insert/remove any of these objects and design a typical room environment around the hand.
The Audioband works on all types of hands, as it predicts the gesture accurately for all values of the absorption coefficient of the hand (within a certain universally accepted range). The absorption coefficients of the wall, the temperature and humidity in the room can also be changed to a certain degree. The number of reflections at each wall can also be adjusted by changing the reflection order. Finally, the app gives the option of adjusting the bandwidth of the signals to the user in a certain range (approximately 1-2% of the original signal frequency), thereby proving that the app is suitable to work under any real-world circumstance.
After the design has been made, the 'Simulate' button on the app directs the control to a python script to form the data file corresponding to the currently created gesture, and subsequently identify the gesture using a previously trained neural network. The result gets displayed to the user in an approximately 4-6 minutes.



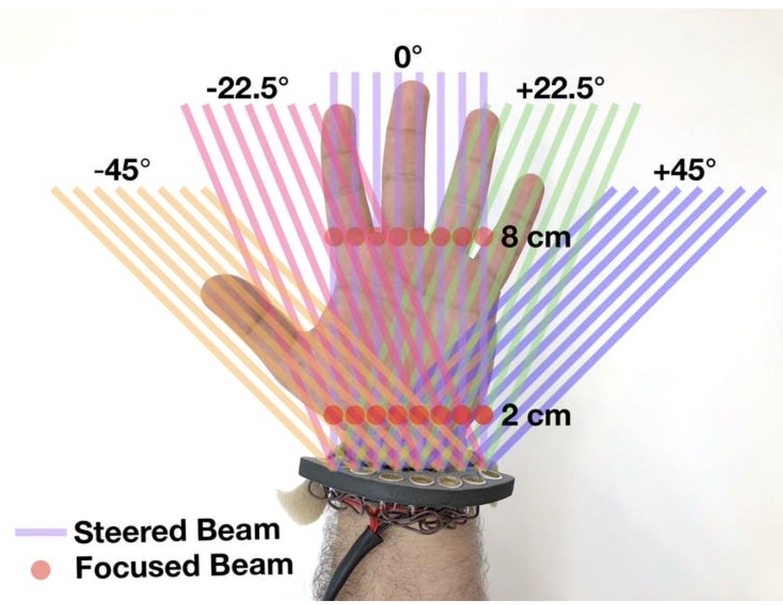
MATLAB App



Hand and room simulation result

6. Scope of Future Work

We have presented AudioBand, a novel wrist-worn sensing method that uses Ultrasonic Beamforming for on-body hand gesture recognition. AudioBand projects ultrasonic wavefronts at different angles on the user's hand and measures the reflected waves. We hope our effort will act as a catalyst for deeper investigation into ultrasonic beamforming for enabling novel human-computer interactions. We envision AudioBand may sit behind an acoustically-transparent plastic window on the side of smartwatches, similar to medical ultrasound wands.
Optimization of running time of MATLAB function is also included in our future work since currently the run time for our simulation is about 20 minutes which is sufficiently large.



7. Team

7.1 Project Mentors

- Afzal Rao
- Anshul Rai
- Netravat Pendsey
- Utkarsh Gupta

7.2 Project Members

- Aditi Singla
- Akshay Mehta
- Ananya Singla
- Astha Pant
- Himanshu PS
- Manish Mayank
- Sarvesh Chandra
- Shubhi Kesarwani
- Tanishka Agrawal
- Videeta Sharma

8. References

- Yasha Iravantchi, Mayank Goel, Chris Harrison. BeamBand: Hand Gesture Sensing with Ultrasonic Beamforming. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI'19), 2019.
- Daniel Albuquerque, "Indoor Acoustic Simulator for Ultrasonic Broadband Signals with Doppler Effect," Applied Acoustics.
- The reference links used in this project can be found at `https://docs.google.com/document/d/1JgmBB6_dpZVbEFwVVbJV7EAdhiodrbbWyRuplBF2EOs/edit?usp=sharing` .