Topic: Analysis of signal processing used to measure the distance between smart devices

(1) Summary of the project

Different methods have been used to find the distance between two smart devices using information generated from GPS, Gyroscope, Bluetooth, and compass. In this project, we are working on the signal processing part of a model which uses sound and works for any device which has a microphone and a speaker.

In this model, round trip sound time of two note sequence is calculated. First, the two devices communicate and then the pinger plays the first note - PING. The second device receives the ping and waits for a delay of 'd' and then replies with the second note - PONG. Once the pinger detects the pong at a certain time, the distance between the two devices can be calculated using the speed of sound.

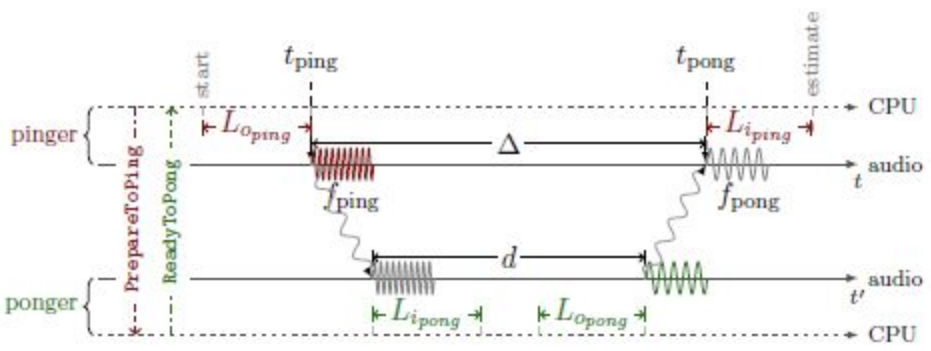


Fig. Timeline of the pairwise distance measurement. [1]

Here, the first task is to detect the ping signal. But using a tuned frequency detector would be too slow for the purpose. So we need to use an onset detector which will then trigger the tuned frequency detector. In this project, we have made matlab models for the two detectors and analysed their working.

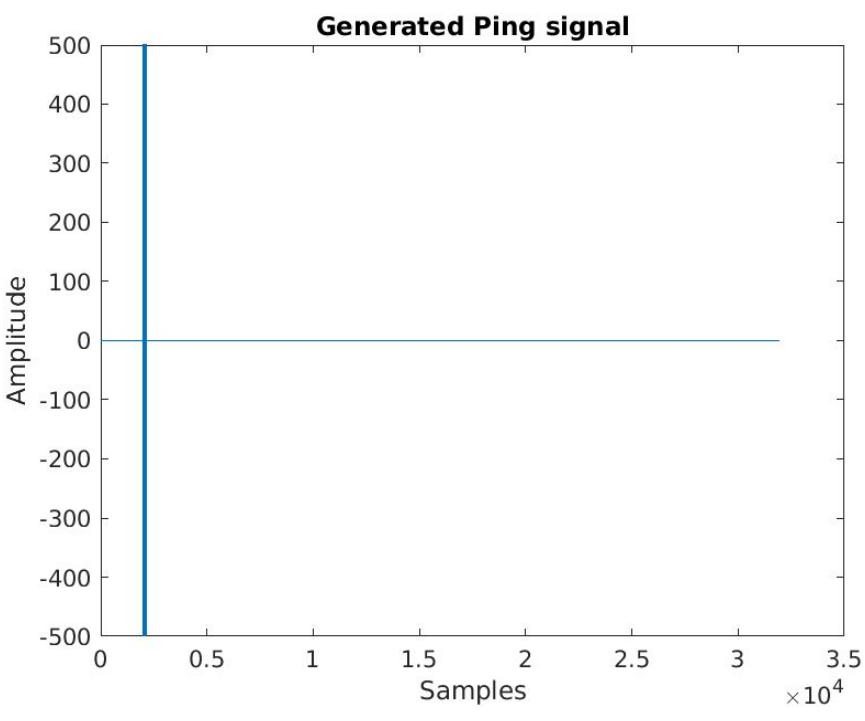
(2) Work done in the project:

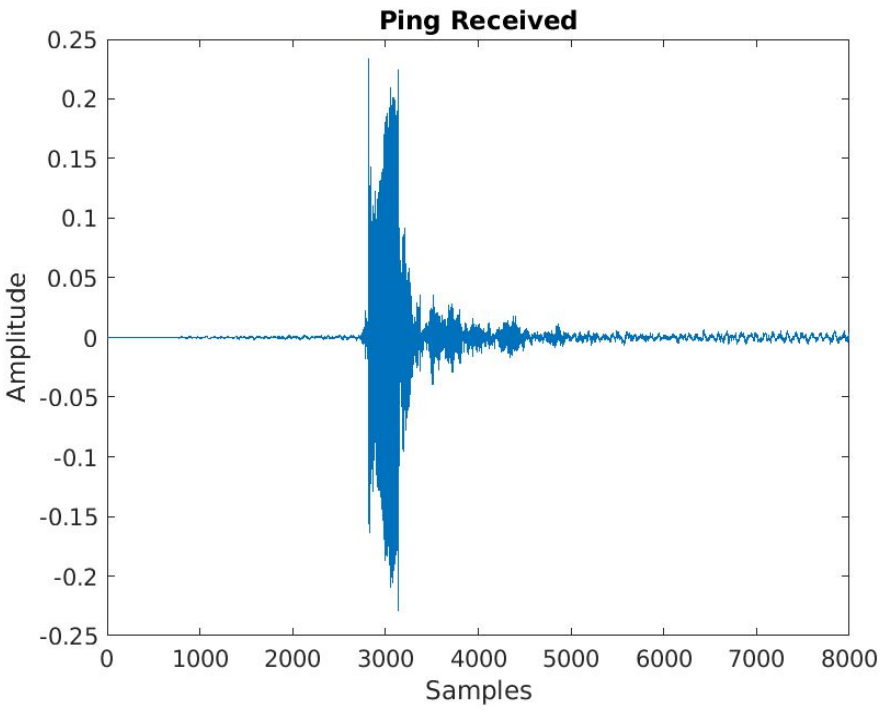
First, we generated ping signals using **ping_signal_generate.mlx**, and recorded them with our smartphones to add noise so as to obtain a realistic representation of the signal.

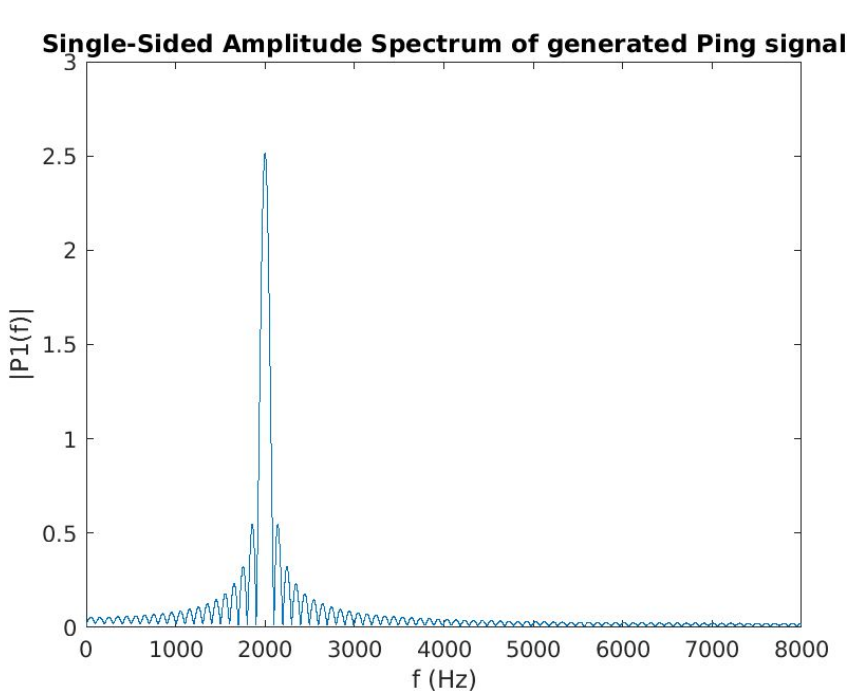
The recorded signals were fed into **onset_detection.mlx.** In this script, short term over long term energy ratio has been calculated, then a threshold was applied to find the onsets.

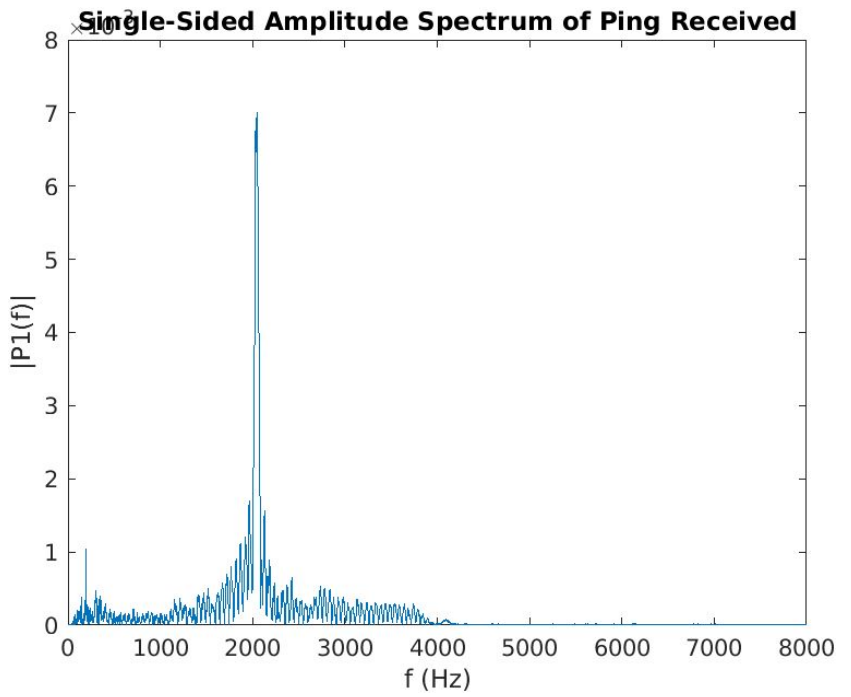
Technically the **signal_detection.mlx** should be run after an onset has been identified in real time but since we are doing an analysis we directly applied the filter onto the whole signal.

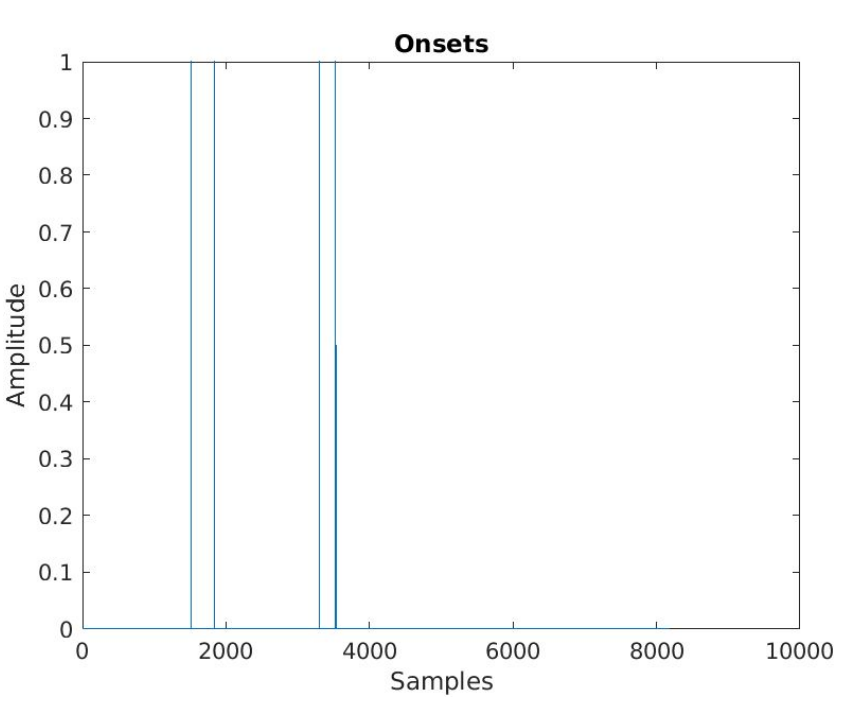
In this script, we use a combination of notched and peaked comb filters tuned to the ping frequency and take the ratio of their energies to calculate the confidence of the required signal being present.

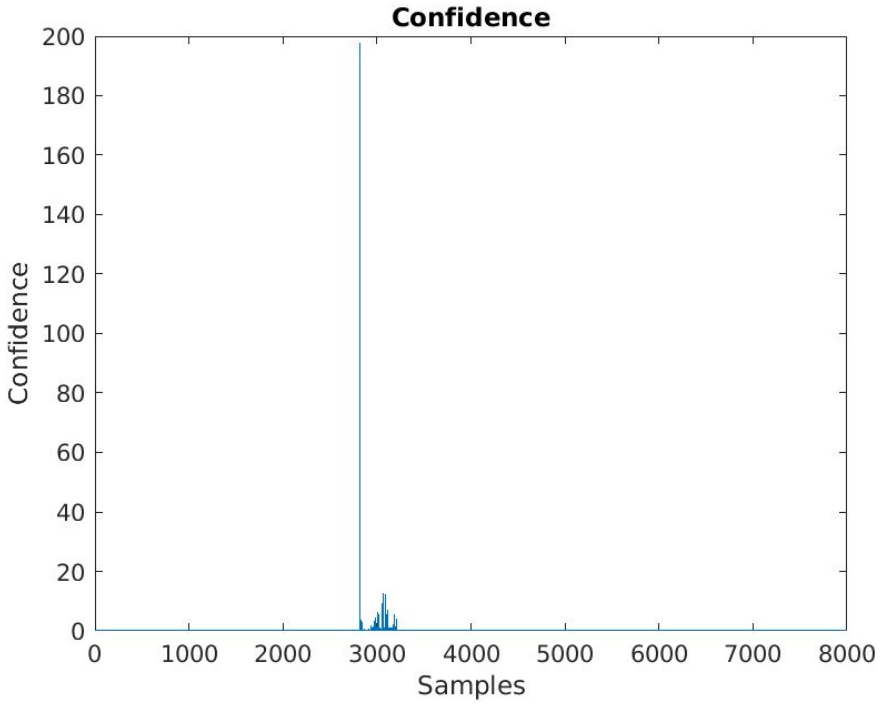


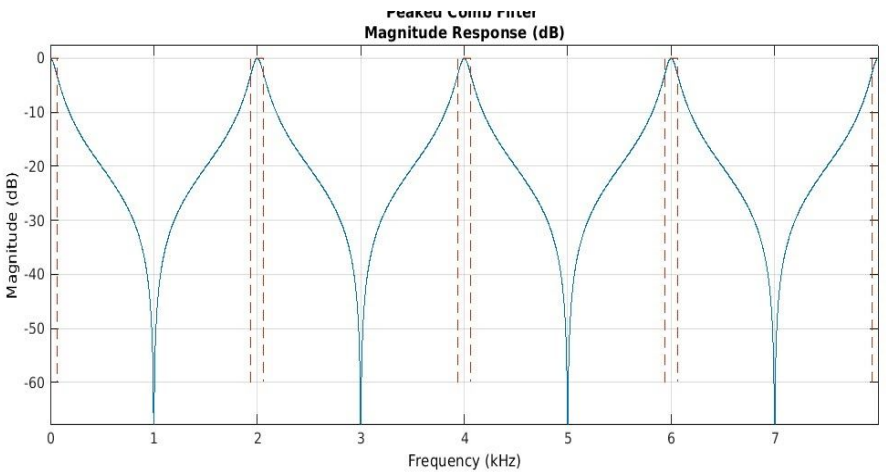


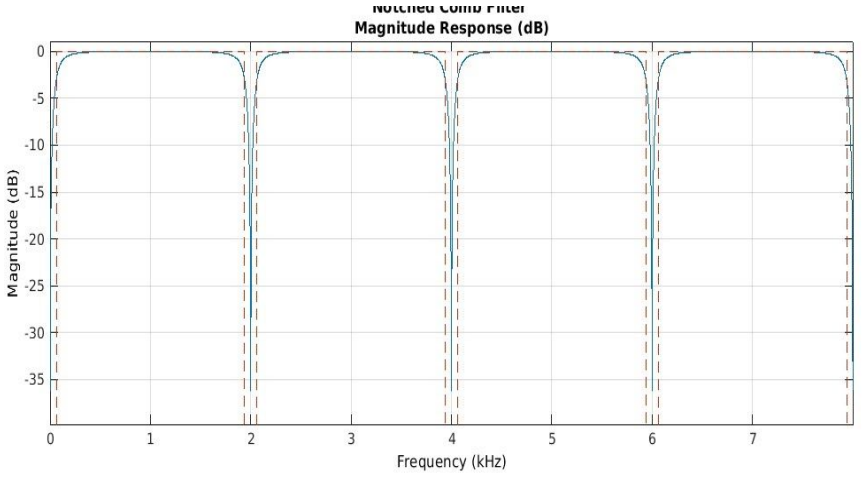












(3) Results

The system worked well under low noise circumstances. 4 onsets were detected, 3 of which were false positives, i.e. . 75%. Which agrees with the suggestions of Herra and Kim. Confidence was accurate and was able to find the ping signal accurately.

In noisy signals, many onsets were detected (30 in one case) and confidence was very high for multiple (2) samples. Inferring from this we can say the system does not work very effectively for high noise signals.

Other limitations of the system are the tuning required for comb filters for some frequencies requires use of fractional delay and gets complicated, Inability to detect reflections, no mechanism to deal with noise spike when ping is received.

(4) Future Scope.

It can be used to make a 3D positioning system of devices which can then be implemented in social distancing apps, gaming systems, safety systems, or speaker systems to reduce interference. Cross platform software needs to be developed to implement this. Can be scaled using an android application.

References:

[1] Ping-Pong: Using Smartphones To Measure Distances And Relative Positions, Jorge Herrera and Hyung Suk Kim, Stanford University, December 2, 2013

Group Members:

1. Jubin Nagpal - 170002022
2. Satyajeet Deshmukh - 170002042
3. Shreyansh Thakur - 170002048
4. Sarthak Chincholikar - 170003013