# Abstract :

Traditional analog hearing aids are similar to a simple radio. They can be tuned and adjusted for volume, bass and treble. But hearing loss is not just a technical loss of volume. Rather, hearing deficiency can increase sensitivity and reduce tolerance to certain sounds while diminishing sensitivity to others. For instance, digital technology can tell the difference between speech and background noise, allowing one in while filtering out the other. Approximately 10% of the world's population suffers from some type of hearing loss, yet only a small percentage of this statistic use a hearing aid. The stigma associated with wearing a hearing aid, customer dissatisfaction with hearing aid performance, and the cost associated with a high performance solution are all causes of low market penetration. Through the use of digital signal processing, digital hearing aid now offers what the analog hearing aid cannot offer. It proposes the possibility of performing signal-to noise enhancement, flexible gain-processing, digital feedback reduction, etc. In this paper, the simulation of simple digital hearing aid was developed using MATLAB programming language.

# Hearing Aid Demo in MATLAB

## Screen Display Settings

%Measure Screen Size of the device  
%Calculate position values of figure windows  
scrsz = get(0,'ScreenSize');  
%P1 = [50 300 scrsz(3)/2 scrsz(4)/2];  
%P2 = [50 80 scrsz(3)/3 scrsz(4)/3];  
%P3 = [620 500 scrsz(3)/3 scrsz(4)/3];  
P4 = [620 80 scrsz(3)/2 scrsz(4)/2];

## Initialise Microphone Parameters

SamplesPerFrame=1024;  
Fs=44100;  
T=10;%Specify Algorithm run time in seconds here  
numplays = (T\*Fs)/SamplesPerFrame;

## Initialise Audio recorder toolbox Parameters

mic=dsp.AudioRecorder();  
mic.SamplesPerFrame=SamplesPerFrame;  
mic.DeviceName='Microphone (Realtek High Definition Audio)';  
mic.NumChannels=1;  
mic.SampleRate=Fs;  
mic.OutputDataType='double';  
mic.BufferSizeSource='Property';  
mic.QueueDuration=2;  
mic.BufferSize=128;

## Create and Configure an Audio Player System Object

audplyer = dsp.AudioPlayer('SampleRate',Fs);

## Scope Parameter Initialisation

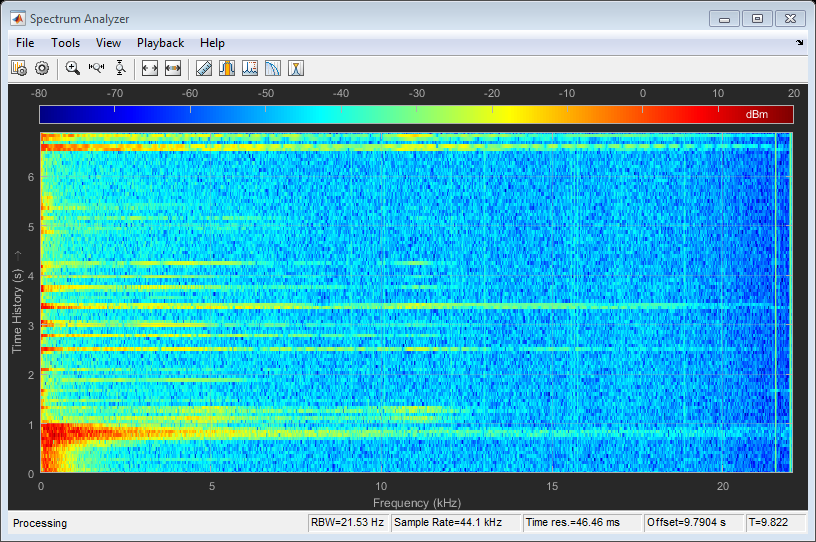
scope = dsp.TimeScope('SampleRate',Fs,'TimeSpan',0.1,...  
 'Position',P1,'YLimits',[-1 1]);  
scope.ShowGrid = 1;  
scope.ShowLegend = 1;

## Spectrum Analyser Specification

SpectroGraph = dsp.SpectrumAnalyzer('SampleRate',Fs,'Position',P4,...  
 'PlotAsTwoSidedSpectrum',false);  
SpectroGraph.SpectrumType = 'Spectrogram';

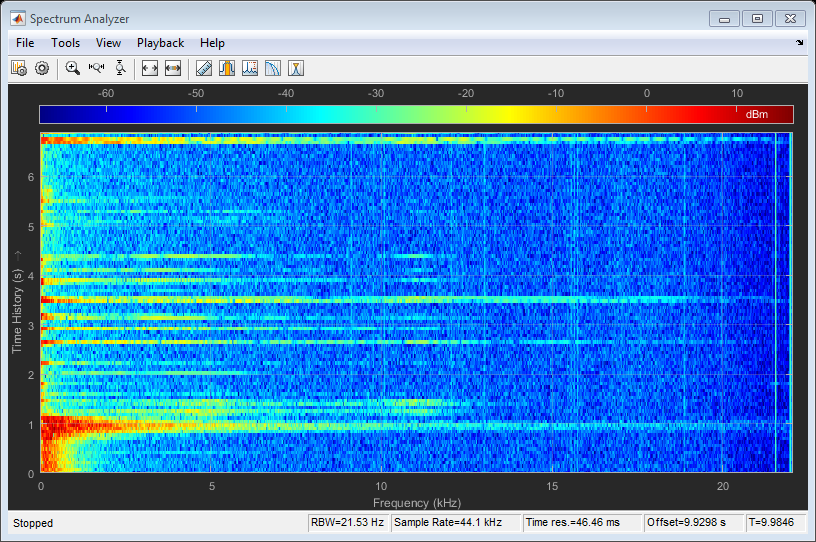
## Stream Processing Loop

while numplays>0  
  
 data=step(mic);  
 step(audplyer, double(data));%Play the Output Signal  
 %step(scope,data);  
 step(SpectroGraph,data);  
 numplays = numplays-1;  
  
end



## Release Hardware

release(mic);  
release(audplyer);  
release(scope);  
release(SpectroGraph);



## End of Program