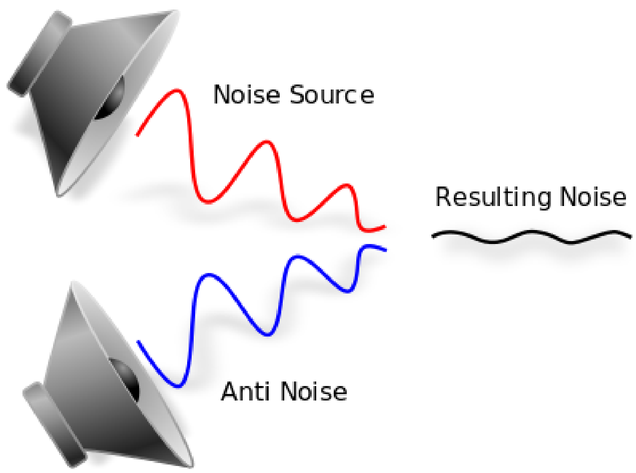
# ADAPTIVE NOISE CANCELLATION




KARAMJIT SINGH BEDI

102006179 3ECE4 UEC502

# FINITE IMPULSE RESPONSE FILTER

* In signal processing, the FIR filter is a filter whose impulse response is of finite duration, because it settles to zero in finite time. This in constrast to IIR filters, which may have internal feedback and may continue to respond indefinitely

* PROPERTIES :

1. require no feedback
2. are inherently stable
3. are easy to implement
4. can easily be designed to be linear phase by making the coefficient sequence symmetric

* The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required as compared to an IIR filter with similar sharpness or selectivity, especially when low frequency cutoffs are needed.

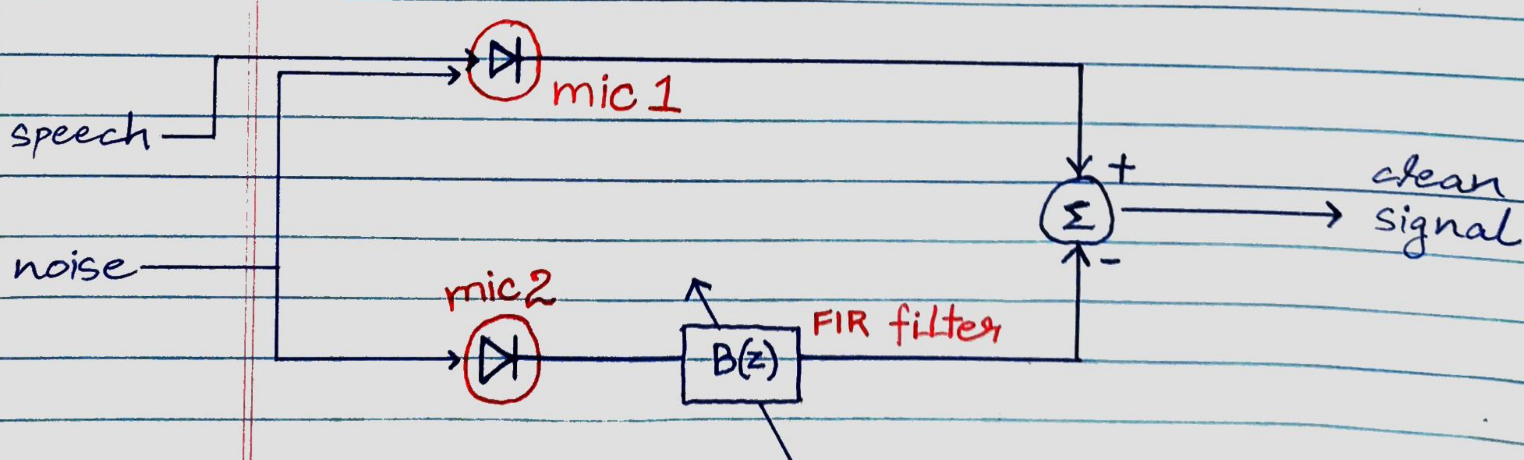
$$h[n] = \sum_{i=0}^{N} b_i \, \delta[n-i]$$

## NOISE CANCELLATION

* Noise Cancellation is a method for reducing unwanted sound by the addition of a second sound specifically designed to cancel the first. This technology is widely used in road vehicles and mobile phones.

* Modern noise cancellation is generally achieved through the use of analog circuits or digital signal processing

* Since Adaptive Noise Cancellation requires higher-order FIR filters, the Fast Fourier Transform algorithm is used to perform the calculations entirely in the frequency domain



# EXPERIMENTAL SETUP

* The primary microphone picks up speech and background noise from a noise source. A secondary microphone close to the noise source also acquires the noise signal. Since this microphone is located sufficiently far from the primary microphone, it does not receive the speech signal.

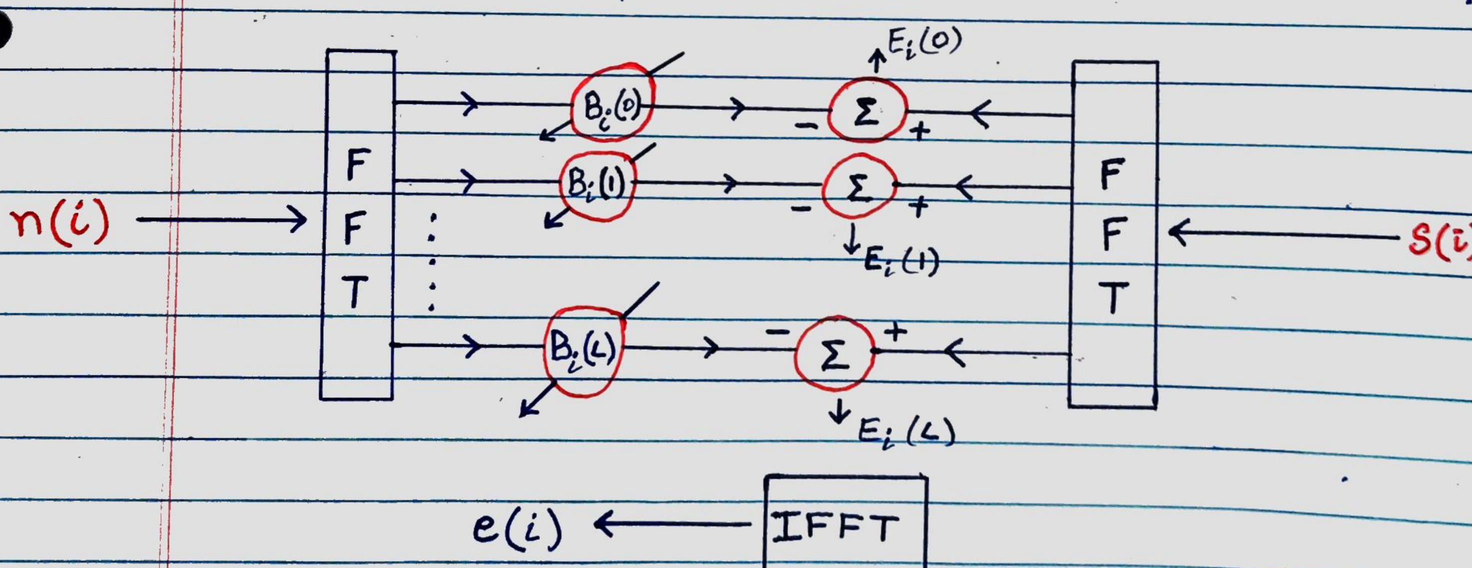
* The idea is to acquire an approximation of the noise signal, which is then subtracted from the original signal. If this approximation is equal to the noise signal retrieved by the secondary microphone, then the noise reduction is perfect and the output of this configuration will be clean speech. However, this is practically impossible, given the involvement of room acoustics like reflections and delay.

* The FIR filter is needed to model the reflections and delays in the acoustic path between the noise source and the primary microphone. Since the acoustic path is

usually time-varying, the filter must be continuous estimated. In this project, an adaptive FIR filter is proposed, the coefficients of which are estimated using a frequency-domain adaptive algorithm.

# ALGORITHM

- The FFT of the original signal and the noise signal are calculated.
- The frequency spectrum of the noise signal is multiplied by the frequency response of the FIR filter and then subtracted from the original signal.
- The IFFT of this difference signal is taken to obtain an approximation of the speech signal.
- The Adaptive algorithm uses the error to update the filter coefficients iteratively.



# EVALUATION CRITERIA

* The Signal-to-Noise Ratios (SNR) of the original signal and the filtered output signal are calculated with respect to the clean speech signal.

* These ratios are used to calculate the improvement in SNR and evaluate the FIR filter.

* The SNR improvement is used to test the filter for different values of the adaption coefficient to find the ideal value.

$$SNR_{dB} = 10 \log_{10} \left( \frac{\sum_{1}^{N} s^2(n)}{\sum_{1}^{N} (s(n) - mic1(n))^2} \right)$$

$$SNR'_{dB} = 10 \log_{10} \left( \frac{\sum_{1}^{N} s^2(n)}{\sum_{1}^{N} (s(n) - e(n))^2} \right)$$

$$SNR_{dB\ \text{improvement}} = SNR'_{dB} - SNR_{dB}$$

## RESULTS

| N | $\mu$ | SNR improvement |
|---|---|---|
| 256 | 0.003 | 12.7709 |
| 256 | 0.004 | 12.5740 |
| 256 | 0.005 | 12.2379 |
| 256 | 0.001 | 12.0078 |
| 256 | 0.0026 | 12.7944 |

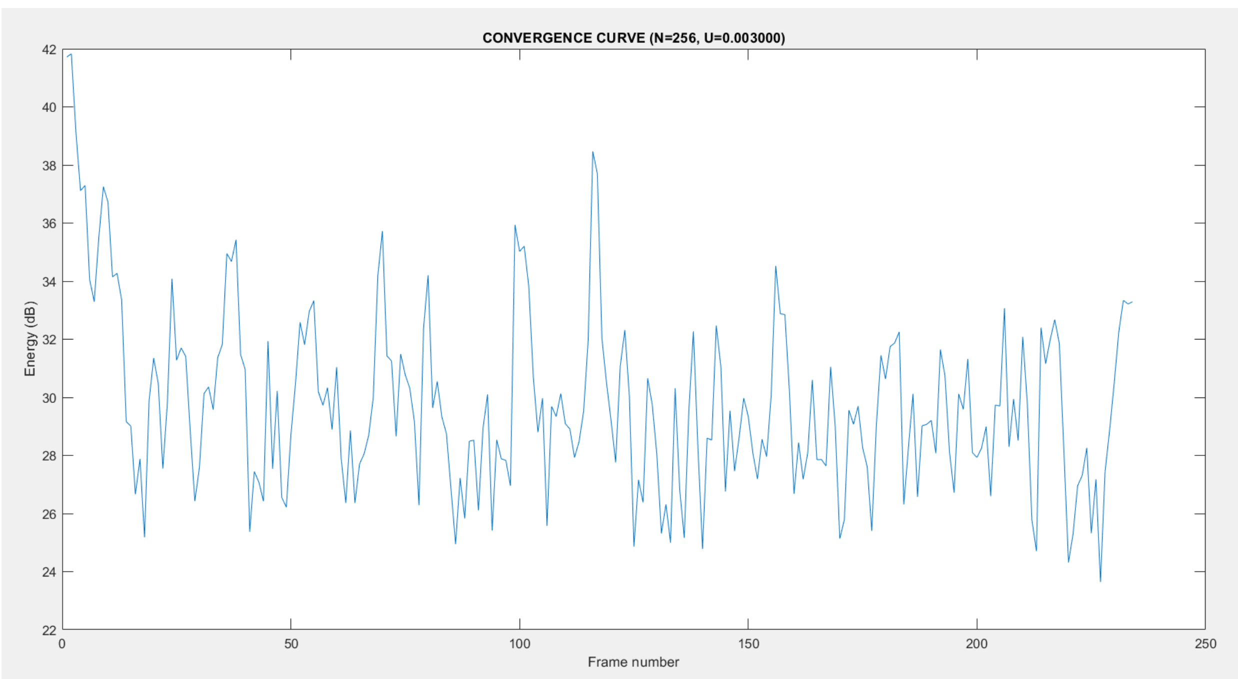
After conducting manual experiments using different values of the adaption coefficient it is observed that the improvement in SNR is greatest for $\mu = 0.0026$

18/10

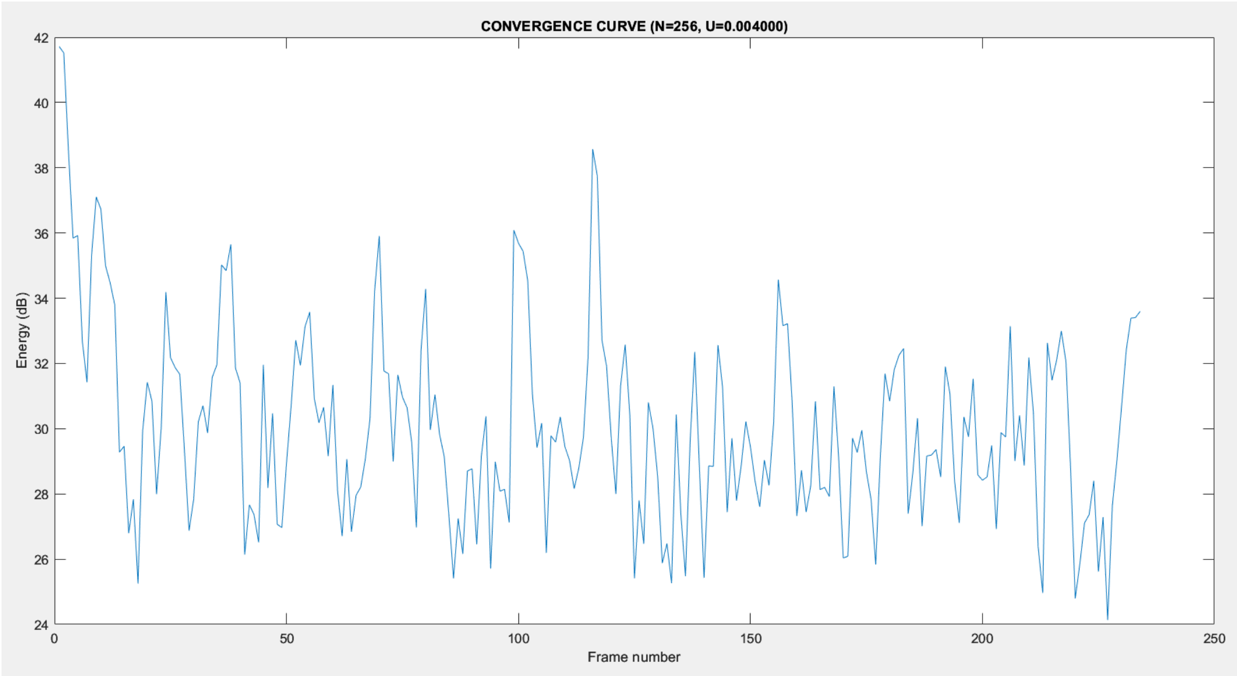
Karamjit Singh Bedi
(102006179) (3EC4)

```matlab
clc;
clearvars;
close all;

[mic_1, Fs_1] = audioread("Corrupted_Speech.wav");
[mic_2, Fs_2] = audioread("White_Noise.wav");
[clean, Fs_3] = audioread("cleanspeech.wav");

win_size = 2^8;
adapt = 0.003;
nFrames = fix(min(length(mic_1), length(mic_2))/win_size);
initialFreqResp = zeros(win_size, 1);

% Fixing length of reduced noise signal same as 'cleanspeech' signal
opSig_f = zeros(length(clean), 1);
opSig_t = zeros(length(clean), 1);

for k = 1 : nFrames

   % Compute indices for current frame
   j = (1:win_size) + (win_size*(k-1));
   D = fft(mic_1(j), win_size);
   X = fft(mic_2(j), win_size);
   X_diag = diag(X);

   if k == 1
       % filtered output signal
       oS_f(1:win_size, k) = D - (X_diag*initialFreqResp);
       oS_t(1:win_size, k) = ifft(oS_f(1:win_size, k));

       % We now again calculate reduced noise speech signal as a
       % column-vector as that would help with the SNR calculation

       opSig_f(j, 1) = D - (X_diag*initialFreqResp);
       opSig_t(j, 1) = ifft(opSig_f(j, 1));

       % frequency response of the filter
       B(:, k) = initialFreqResp + (2*adapt*X_diag'*oS_f(1:win_size, k));

   else
       oS_f(1:win_size, k) = D - (X_diag*B(:, k-1));
       oS_t(1:win_size, k) = ifft(oS_f(1:win_size, k));

       opSig_f(j, 1) = D - (X_diag*B(:, k-1));
       opSig_t(j, 1) = ifft(opSig_f(j, 1));

       B(:, k) = B(:, k-1) + (2*adapt*X_diag'*oS_f(1:win_size, k));
   end
end
```

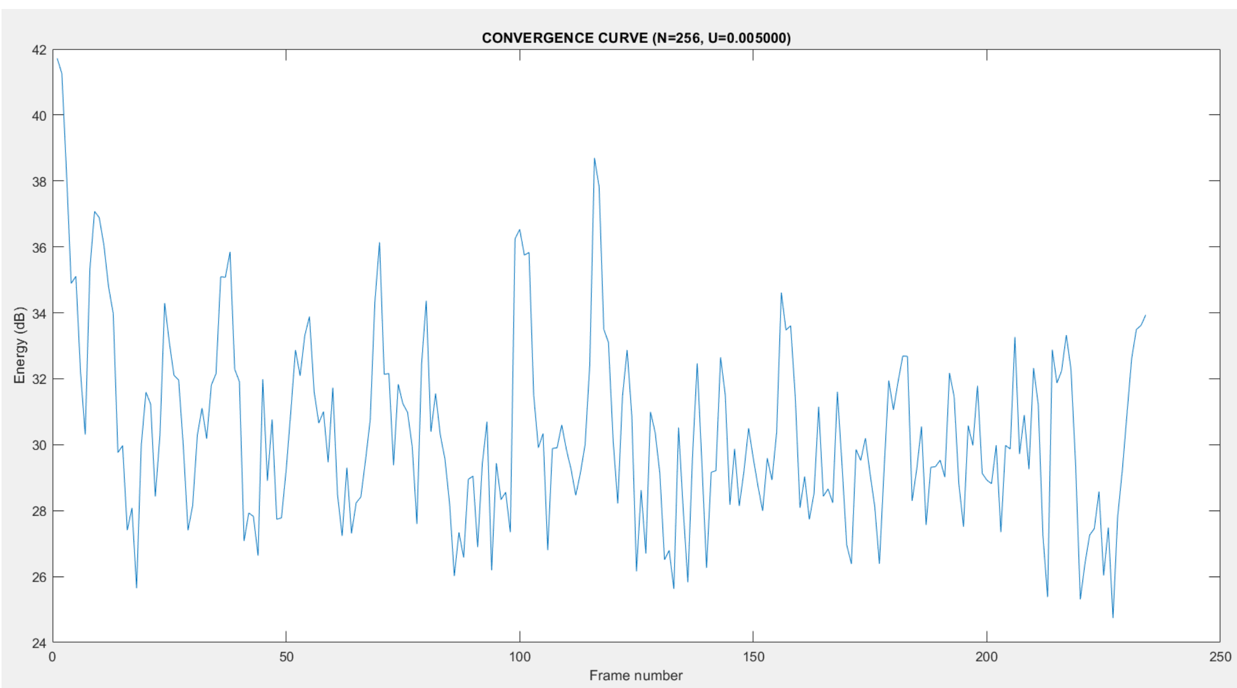
```
SNR_before = 10 * log10((clean'*clean)/((clean-mic_1)'*(clean-mic_1)));
SNR_after = 10 * log10((clean'*clean)/((clean-opSig_t)'*(clean-opSig_t)));
SNR_change = SNR_after - SNR_before;

for p = 1:nFrames

   % Calculating energy of each frame to find energy convergence
   E(1, p) = 10 * log10(oS_f(:, p)'*oS_f(:, p));
end

figure;
plot(E);
msg = sprintf("CONVERGENCE CURVE (N=%d, U=%f)", win_size, adapt);
title(msg);
xlabel("Frame number");
ylabel("Energy (dB)");

audiowrite("Filtered_Speech.wav", opSig_t, Fs_1);
```



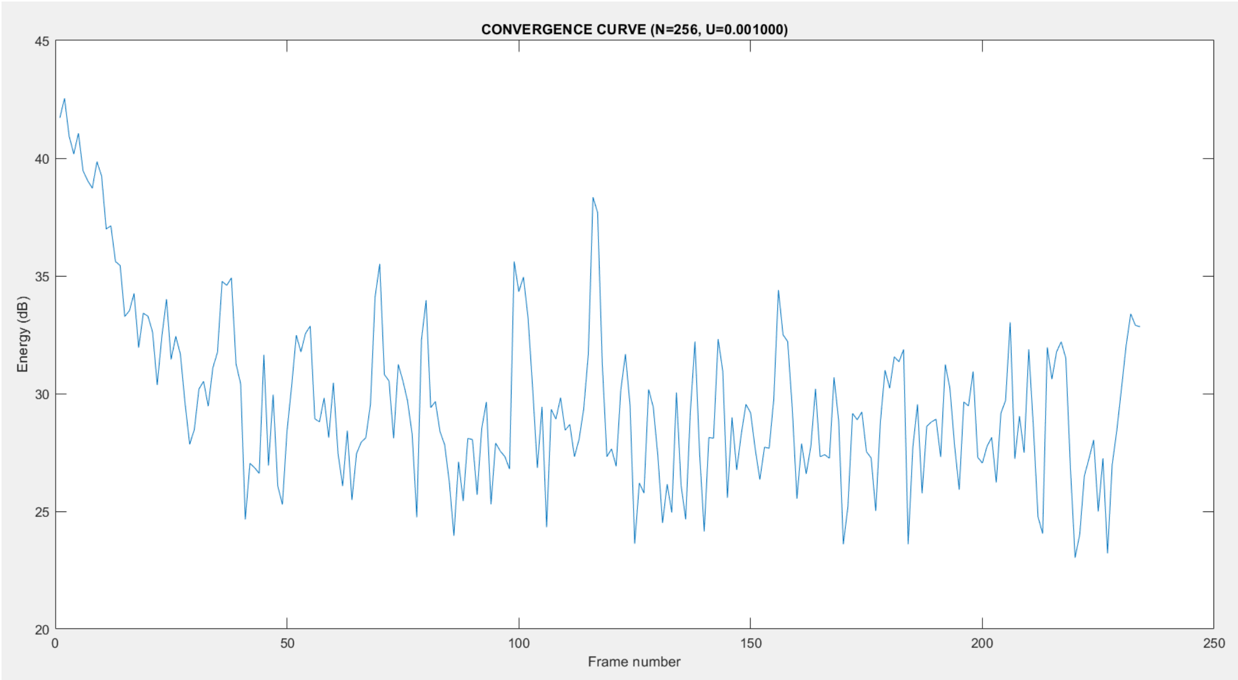

**SNR_change = 12.7709**



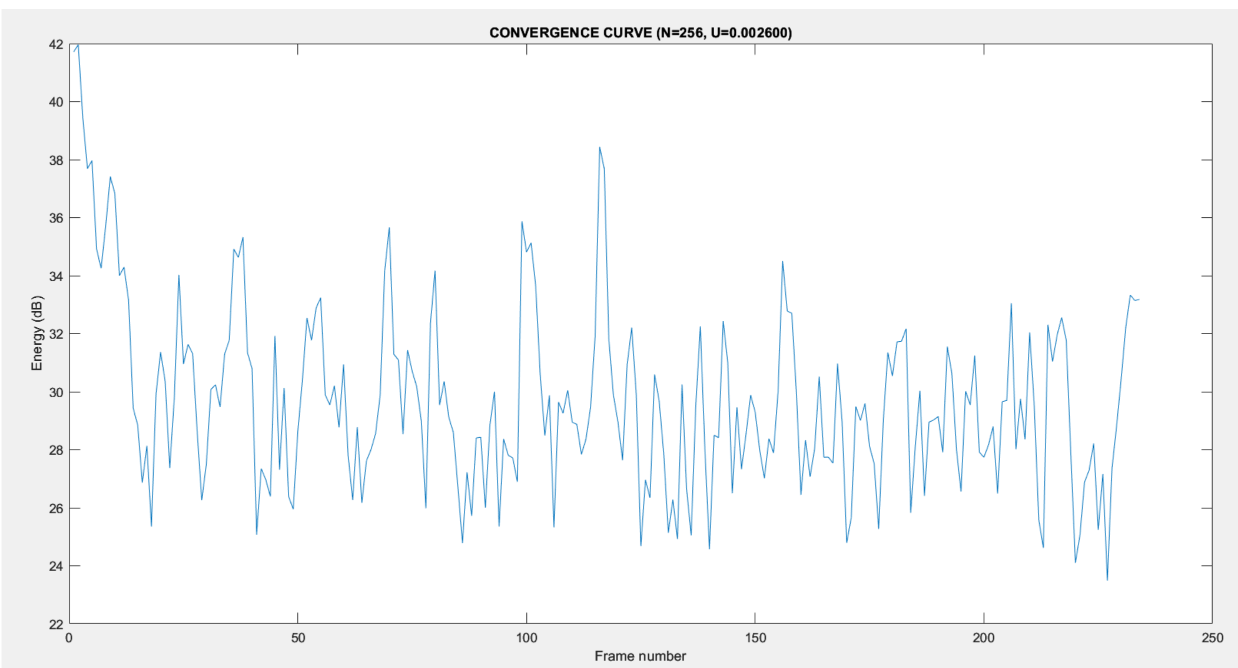

**SNR_change = 12.5740**




**SNR_change = 12.2379**




**SNR_change = 12.0078**




**SNR_change = 12.7944**