1. We followed below mentioned steps to get the desired results:

   (a) Imported the file using *soundfile* library's *read* function and stored it in *data* variable.To find the quantization levels and bits/sample:

   - We found the minimum difference between any two data points to obtain the *step_size*.
   - To get total number of quantization levels in the data

   $$no\_of\_levels = 2V_p/step\_size$$

   - To calculate the bits required to encode one Quantization level

   $$bits/sample = log_2(no\_of\_levels)$$

   - To find all the quantization levels we found the minimum value in the data set and then incremented each by *step_size*.
   - Another simplistic approach could also be to find the bit depth of the audio file which is an implicit property. This provides us with the number of quantization levels. Hence the *step_size* would be :

   $$step\_size = 2V_p/2^{bit\_depth}$$

   (b) We created *my_func* to do all the operations.

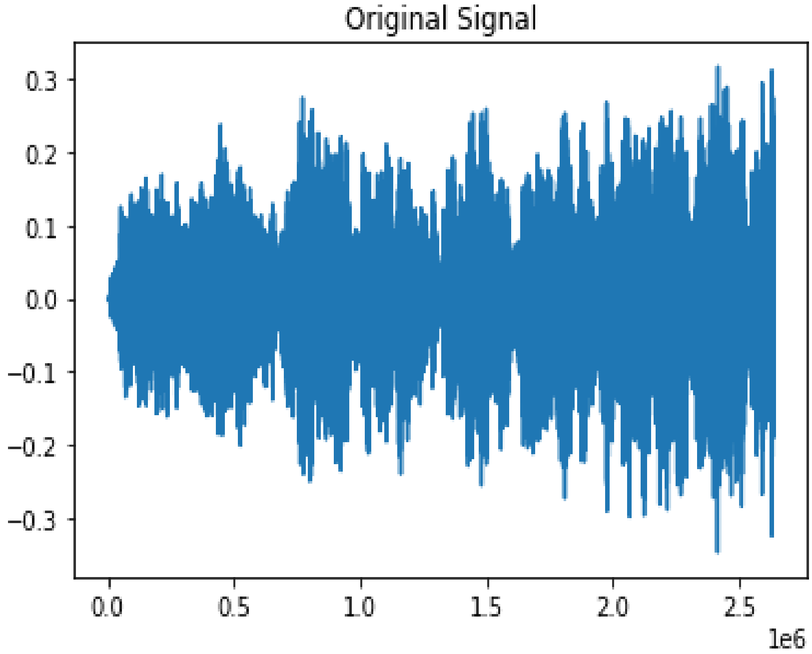
   - To do further steps we scaled the values by *P* = *32768* = 2^{15} and converted them to integer values to make the conversion of decimal to binary and vice-versa easier.
   - After that we convert it to NRZ-L PCM giving us *arr* which is then sampled and noise is added to it using *awgn* function.
   - After adding the noise we obtain *arr_s* as the output of the channel.
   - To begin with the detection process we pass the above obtained signal through the matched filter i.e., *pass_tgh_filtr* function.
   - The output of the filter is then divided by P to counter the effect of scaling.
   - Ideally the number of bits required for encoding the output of the matched filter is:

   $$bits\_required = log_2((V_{max} - V_{min}) * 10^{precision\_point})$$

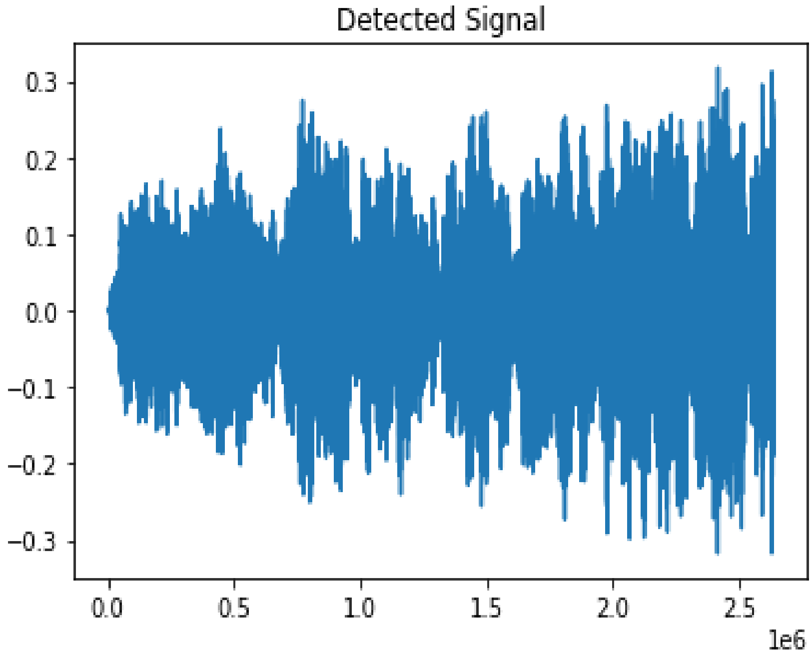
   where Vmax and Vmin are the observed maximum and minimum values in the data array of the audio file and precision point is an integer value which signifies the number of digits after the decimal point to be considered for any calculation.

2. **Observations and Conclusion:**

- We obtained the following audio signal from the file provided.

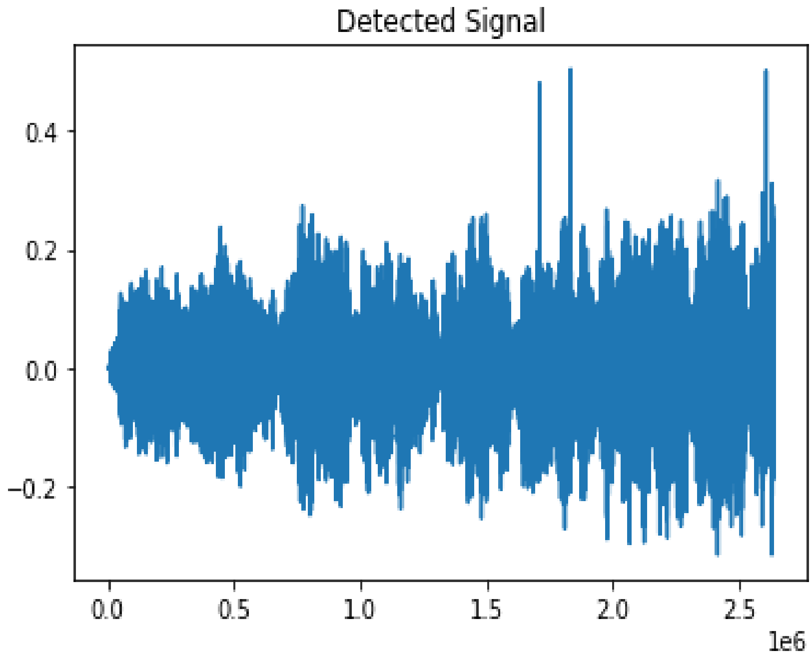


- After adding noise with variance=1 and then following the above steps for detection we were able to retrieve the following signal.



The Average Error in detection was found to be 6.82468554988411e-08.

- When we added noise with variance =10 we retrieved the following signal.



  The Average Error in detection was found to be 8.491695112323169e-07.

- On increasing the variance of the input noise the final output file kept on getting noisier. This can be seen from the audio files new1.wav and new10.wav which represent the output files for variance = 1 and 10 respectively[1].
- If the number of encoding bits for the matched filter output is less than the theoretical value then the error value rises significantly. On increasing the number of bits with respect to the theoretical value, the error value reduces but the computational cost significantly increases.

[1]We have provided aotj1.wav, new1.wav and new10.wav audio files in the zip file that is supposed to be submitted.