Course: MSc Electronic Systems Engineering

Unit Title: Advanced DSP Techniques

Unit Code: EG71ADSP

Faculty / Department: Engineering

Facilitator / Assessor: Dr Chliveros

Student ID No:

Submission Date:

I confirm that no part of this coursework, except where clearly quoted and referenced, has been copied from material belonging to any other person - e.g. from a book, handout or another student. I am aware that it is a breach of regulations to copy the work of another without clear acknowledgement, and that attempting to do so renders me liable to disciplinary actions.

I understand that my submission to the assessment is only valid if it has been submitted electronically to the Unit Moodle submission Portal, within the deadline, and that this constitutes my submission receipt.

_____

Signature

Table of Contents

Abstract

The following paper discusses the implementation of a noise removal adaptive filter. An audio file with unknown frequencies, is introduced. Firstly, the audio signal's frequencies spectrum is estimated by means of discrete Fourier. Afterwards a White Gaussian noise signal is generated and added to the original audio signal and the new spectrum of frequencies present in the noised signal are estimated, by implementing discrete Fourier. The noised signal is then filtered by means of a FIR bandpass filter in order to attempt to remove the random noise from the original signal. However, since the audio signal has random behavior, that property also follows the noise since is additive on the signal. The calculation of the power spectral density is implemented by means of autocorrelating the noised signal and then performing the discrete Fourier transform. The final stage of the algorithm is the implementation of an adaptive filter by means of the Least Mean Square (LMS) algorithm.

Introduction

In order to estimate the PSD, the FFT can be utilized. The general form of the FFT is $X(k) = \sum_{n=0}^{N-1} x(n) w_N^{nk}$; as Monson H. (1999) notes the FFT implementation is the 'split' of an N-point DFT into smaller length DFTs, performing $\frac{1}{2}N^2$ multiplies and adds in comparison with the $\frac{N^2}{2}$ required by the DFT.

The AWGN, as Vaseghi S. (2008) states, is "an uncorrelated random noise process", its mean is equal to zero, has stable standard deviation; presents equal power at the bandwidth that it occupies.

Notable properties of the FIR filters are, they are always stable, have linear phase in the frequency domain and are time-invariant. The function of the Hamming window is defined, according to Monson H. (1999), as $w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right), & 0 \le n \le N \\ 0, \quad Else \end{cases}$. The window function is multiplied by the frequency response of the filter, and the function of the FIR filter is defined as $h(n) = h_d(n)w(n)$.

The Power Spectral Density (PSD) can also be calculated by the autocorrelation function, defined as $rxx(k,l) = E[x(k) \cdot x(l)^*] = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} x(k)x(k+m)$. By implementing the FFT on the autocorrelated signal, the PSD can be calculated.

The LMS algorithm, is utilized in order to estimate the Wiener-Hopf equation, defined as $r_{yx} = R_{xx} \cdot \hat{h}$ by means of the steepest descent algorithm, defined as $h_{n+1} = h_n + \mu\nabla_h(E\{e^2(n)\})$.

Implementation

The audio signal was introduced by the 'audioread' function that inputs an audio file and outputs the signal along with the sampling frequency 'Fs'. The frequency spectrum is estimated by utilizing the 'fft(input_signal)' function, its values normalized by raising each value to the power of two and outputting its absolute values. The code for implementing this is presented below.

```
[y, Fs] = audioread(fullfile(pathname, filename),samples);
yff = fft(y); % Perform FFT
yff = abs(yff).^2; % Normalize FFT values
yff_len = length(yff); % Number of frequencies present
freq_y = (0:yff_len-1)*Fs/yff_len; % Normalize frequency
values
% Implement for the first half of the frequency spectrum
yff_h = yff(1:(yff_len/2)); % Show half of the FFT spectrum
freq_y_h =freq_y(1:(length(freq_y)/2)); % Show half of the
Normalized frequency values
```

The noise is additive, therefore added to the input signal. The noise signal is created by using the function randn(), that generates random Gaussian distributed numbers.

```
% Generate Random Gaussian White noise
r_noise = randn(size(y));

% Normalize and Add the generated Random Gaussian White noise to
the signal
s = y + (r_noise./100);

% Perform FFT on the signal with the added Random Gaussian White
noise,
% Normalize FFT values, Show Half of the present frequencies &
Define Proper
% Frequency values
sff = fft(s); % Perform FFT
sff = abs(sff).^2; % Normalize FFT values
sff_len = length(sff); % Number of frequencies present
freq_s = (0:sff_len-1)*Fs/sff_len; % Normalize frequency values
% Implement for the first half of the frequency spectrum
sff_h = sff(1:(sff_len/2)); % Show half of the FFT spectrum
freq_s_h = freq_s (1:(length(freq_s)/2)); % Show half of the
Normalized frequency values
```

A FIR bandpass filter, that utilizes the Hamming window is implemented. This is achieved by utilizing the function fir1(), in order to create an 14^{th}order filter. The filter allows frequencies that range from 60Hz up to 7940Hz, while rejecting any other frequency. The function 'filter()' is responsible for the filtering.

```
% Define bandpass window
Wn = [(f_low./Fs), (f_high./Fs)];

% Create a 20th-order FIR bandpass filter. The bandpass argument
not
% necessairy but utilized to highlight the type of filter
fir1_bp = fir1(fir_order, Wn, 'bandpass');

% Apply the filter to the signal with AWGN
fir_out = filter(fir1_bp,1,s);
```

The autocorrelation of the FIR output is then calculated in order to compute the PSD of the signal, utilizing the 'xcorr' function. The PSD of the autocorrelated signal is calculated by using the FFT algorithm.

```
[acf_r, lags] = xcorr(fir_out,fir_out, window_size);

% Perform FFT on the autocorrelated signal
a_ff = fft(acf_r); % Perform FFT
a_ff = abs(a_ff).^2; % Normalize FFT values
a_ff_len = length(a_ff);
freq_a = (0:a_ff_len-1)*Fs/a_ff_len;
```

The implementation of the LMS algorithm was achieved by firstly initializing all parameters such as the number of coefficients to be calculated, the initial filter coefficients, the definition of the desired and the noised signal, the length of the samples present in the desired signal and the initial definition of the error rate vector. For the LMS to operate, the noised signal must have the same length with the desired signal. The algorithm repeats the following operations for the number of coefficients that have been implemented. Firstly, the noised signal is segmented into parts, equal to the number of coefficients, implementing a window. An initial estimation is performed by the predefined coefficients and then the error rate is calculated by subtracting the desired signal with the estimated signal. Then, by utilizing the steepest descent algorithm, the algorithm estimates the next sample's coefficients to be used in the next iteration.

```
% Get input of desired signal, signal with noise & initial FIR
% coefficients
d = y(:);
x = fir_out(:);
w = zeros(fir_order,1);
% Define error rate variable
e = zeros(1, length(d));
% Pre-allocate adaptive filter coefficients
wn = zeros(fir_order, 1);
% Assuming step size: 0.0004
mu = 0.0005;

% Determine the value of present samples
N = length(d);
M = fir_order;
% Loop through every sample of the signal
for n = M:N
    xN = x(n:-1:n-M+1); % Implement window
    est(n)= w'*xN; % Calculate estimated filter coefficients
    e(n) = d(n) - est(n); % Error rate
    wn = w + 2*mu*xN*e(n); % Calculate filter coefficient for next
sample
end
```

Based on the above description of the implementation, a block diagram of this system is presented below.

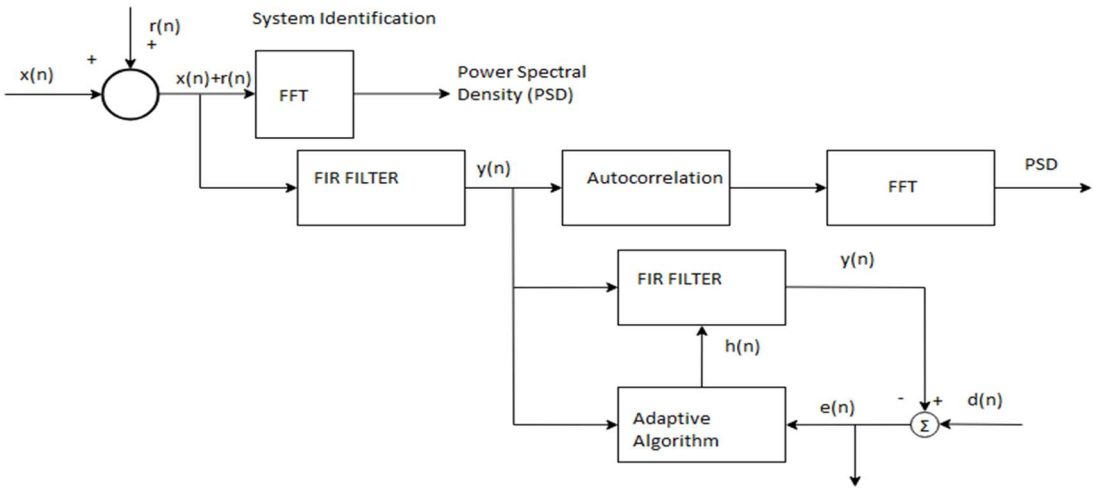

Figure 1 Block Diagram of implemented system

Results

Results: Un-noised signal.

Based on the above implementation and calculations, the following results were obtained.

The un-noised signal plots both in time and frequency domain are presented below.

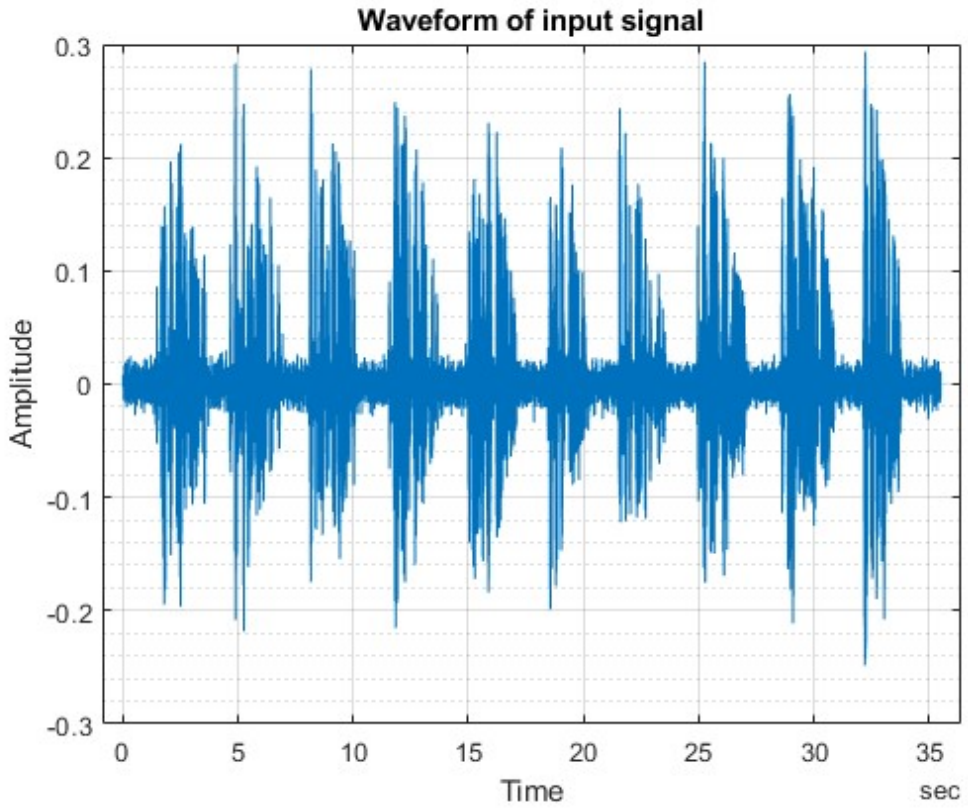

Figure 2 Input signal waveform in time domain

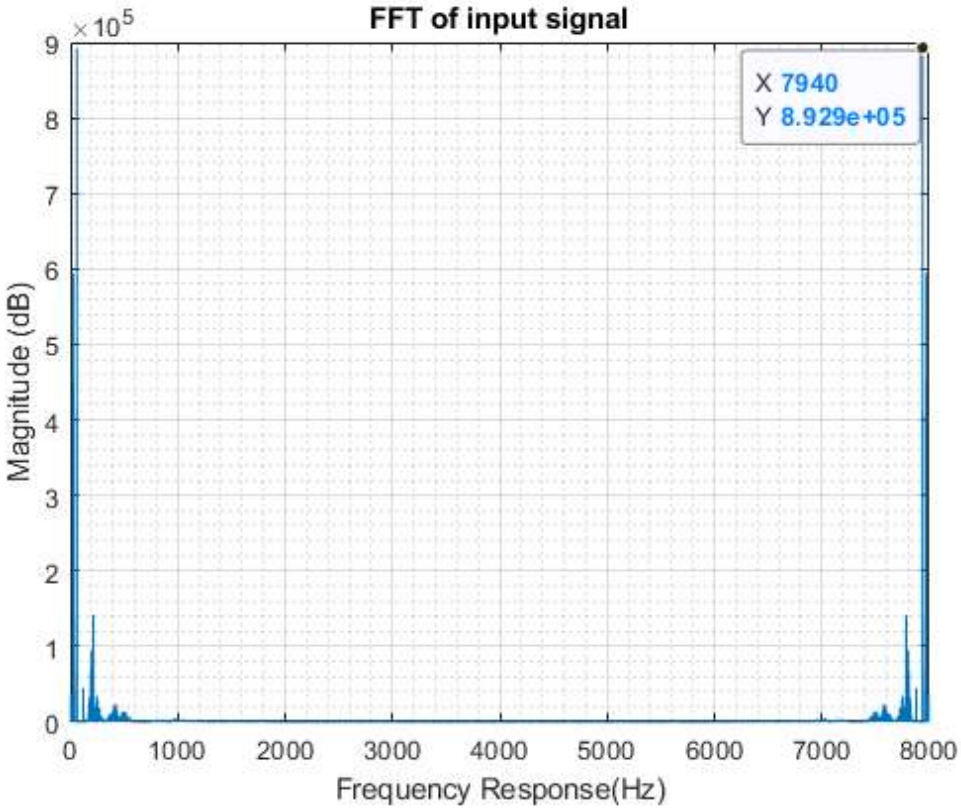

Figure 3 Input signal FFT calculations and the highest dominant frequency

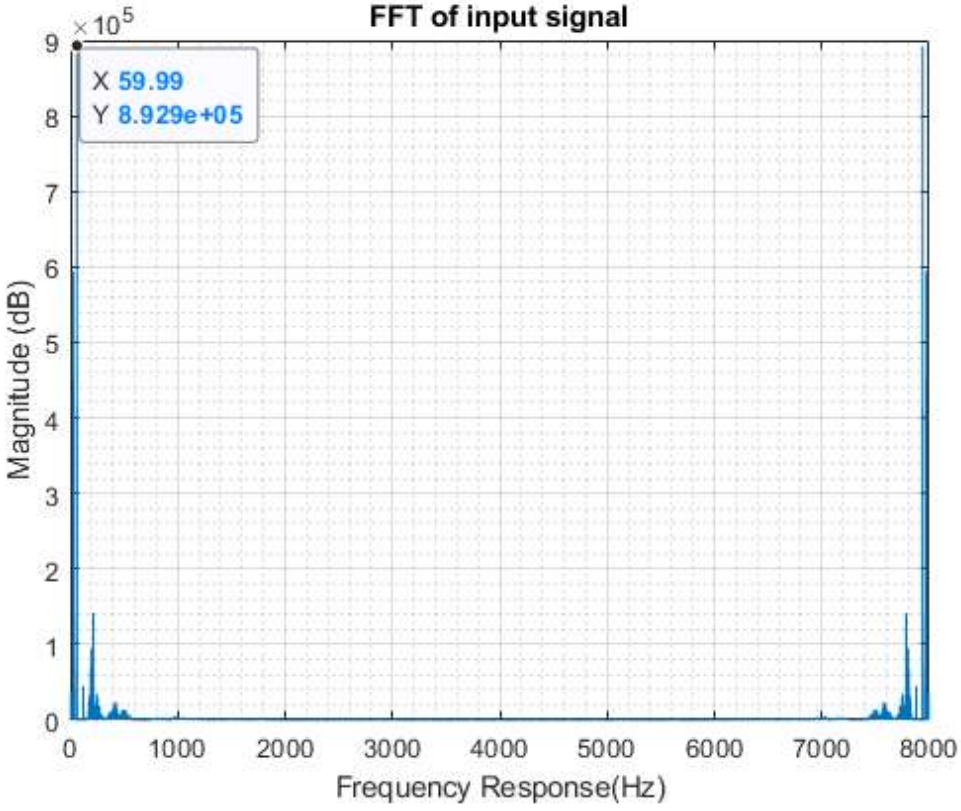

Figure 4 Input signal FFT calculations and the lowest dominant frequency

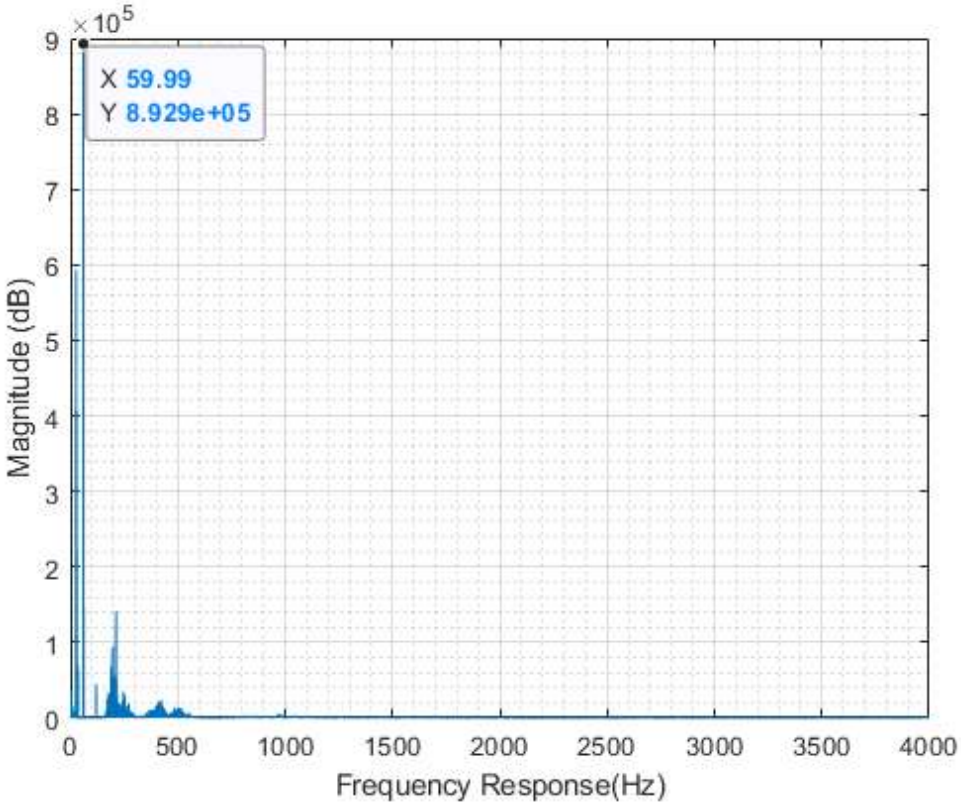

Figure 5 First half of input signal FFT calculations

Based on the above results, given by the FFT of the input signal, the frequencies of 59.99Hz (implemented as 60Hz) and the 7940Hz seem to be the most frequent.

Results: Noise infected signal.

The results of the signal infected with the AWGN are presented below.

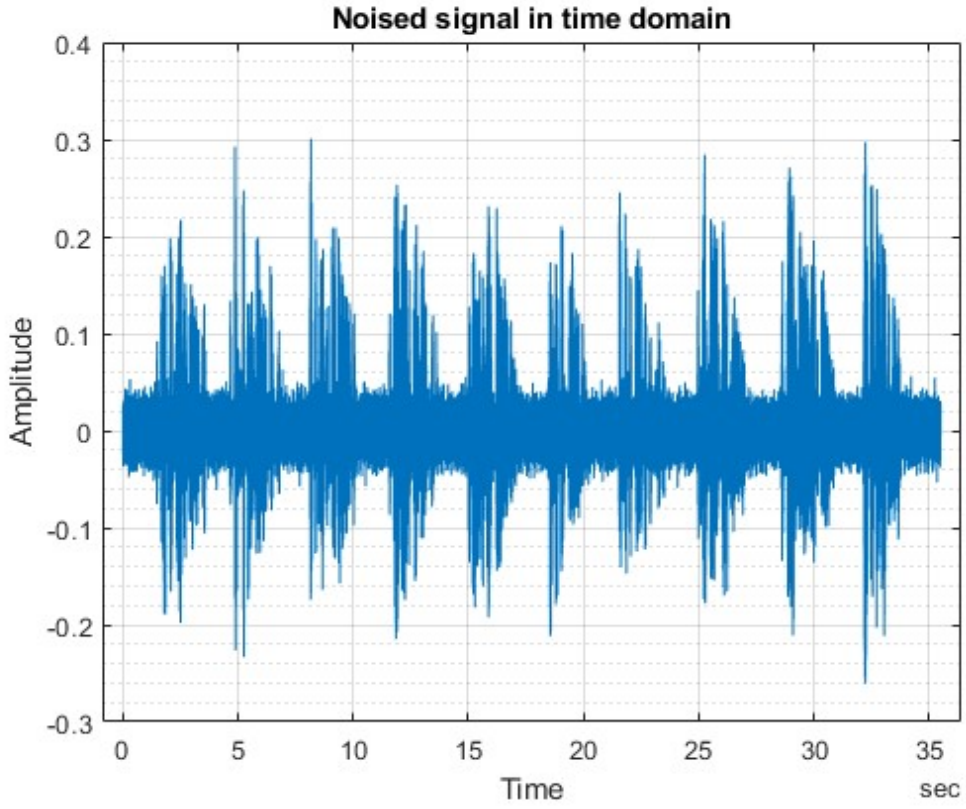

Figure 6 Signal with AWGN waveform

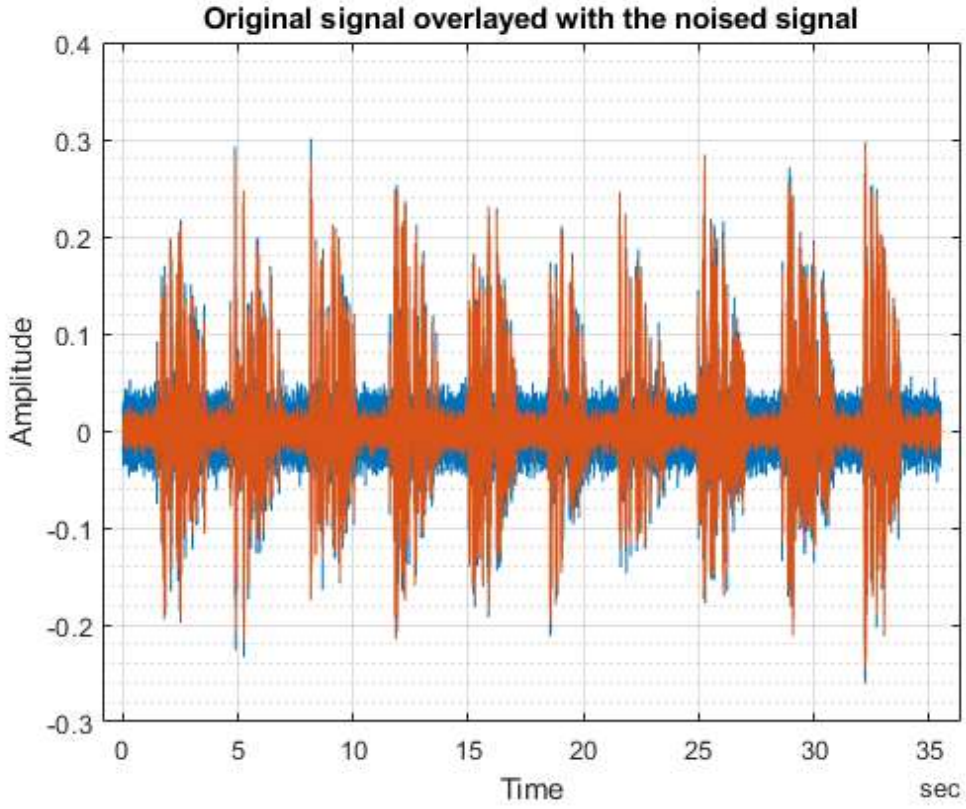

Figure 7 Un-noised signal plotted over the noised signal for comparison.

The FFT of the noised signal is presented below.

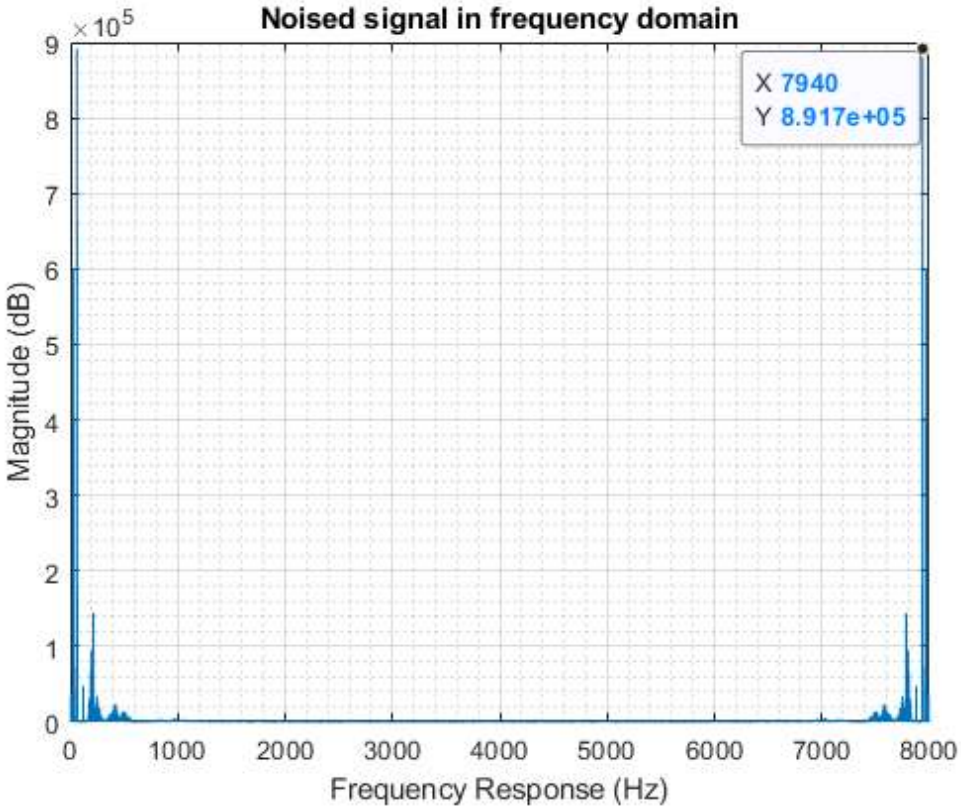

Figure 8 FFT results with the highest dominant frequency

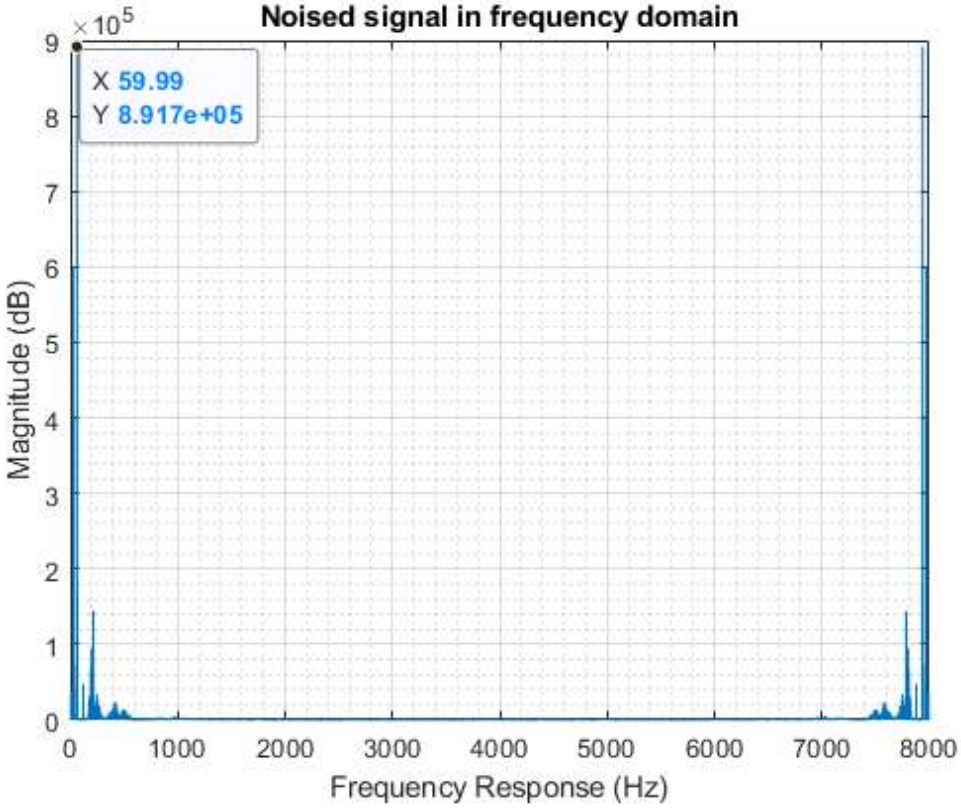

Figure 9 FFT results with the lowest dominant frequency

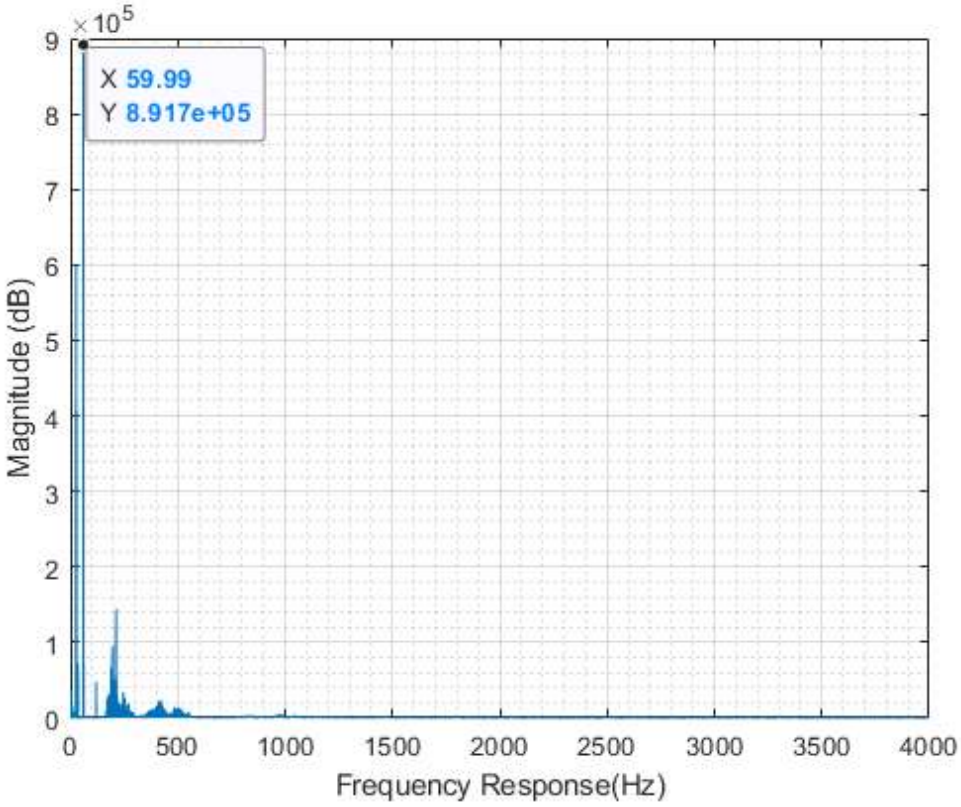

Figure 10 First half of the FFT calculations

Based on the above plots and results, it can be concluded that the noise has infected the original signal at the most dominant frequencies, both high and low.

Results: FIR filter & filtered signal

The results of the filter response and the filtered signal by FIR are presented below.

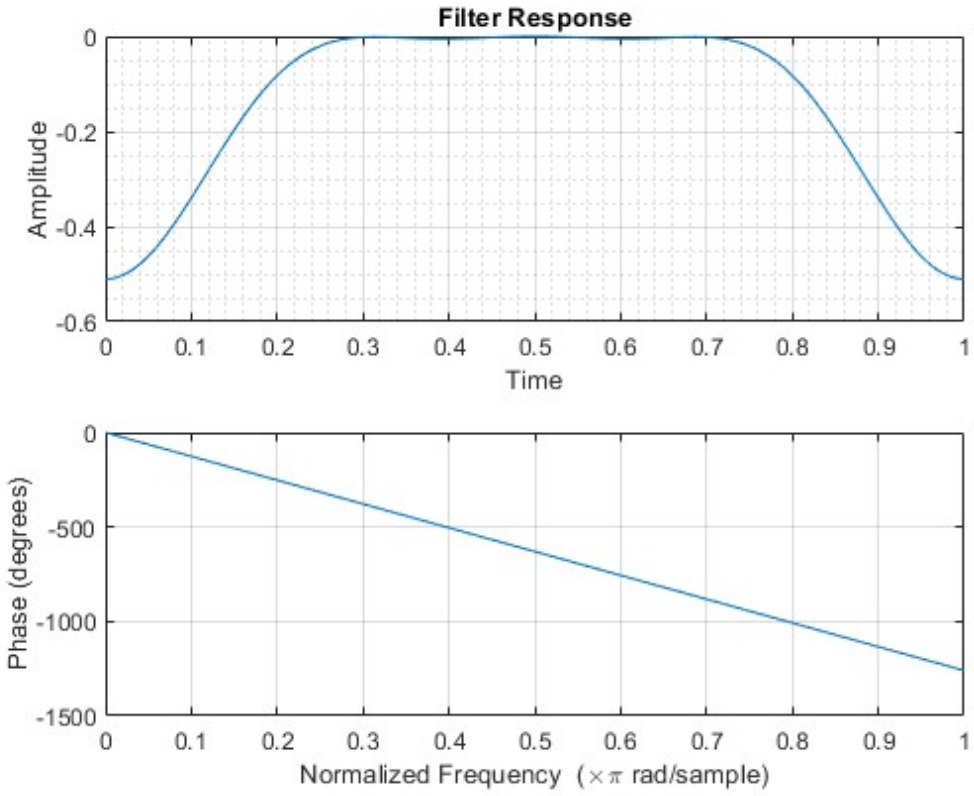

Figure 11 FIR filter response

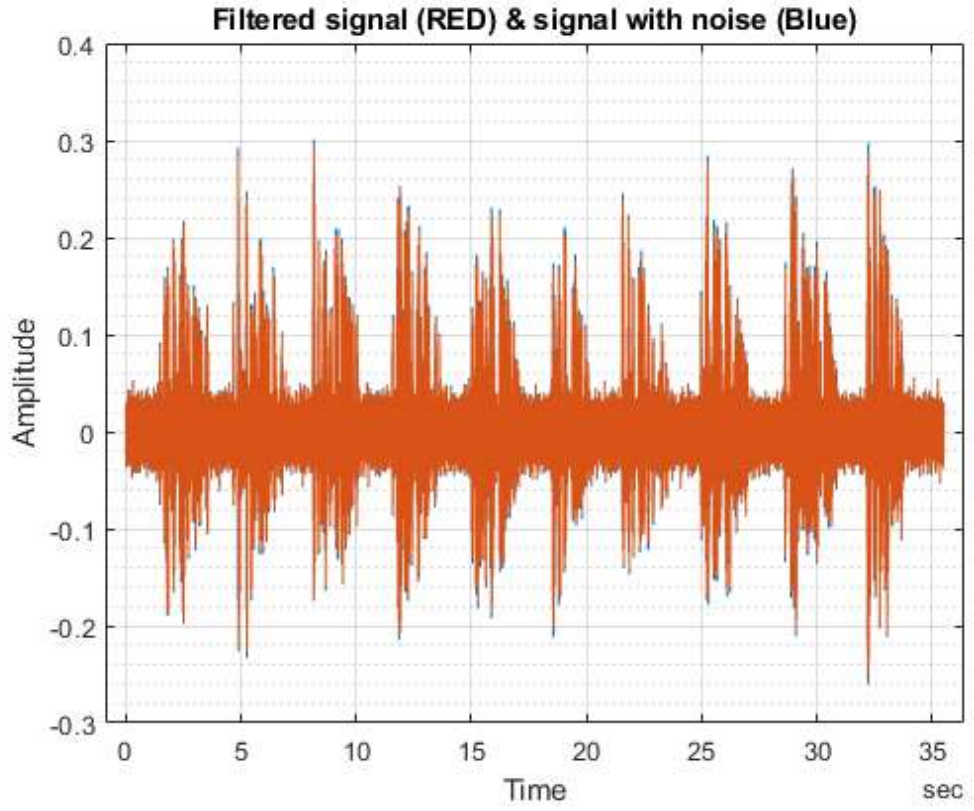

Figure 12 Filtered signal (Red) plotted on top of noised signal (Blue)

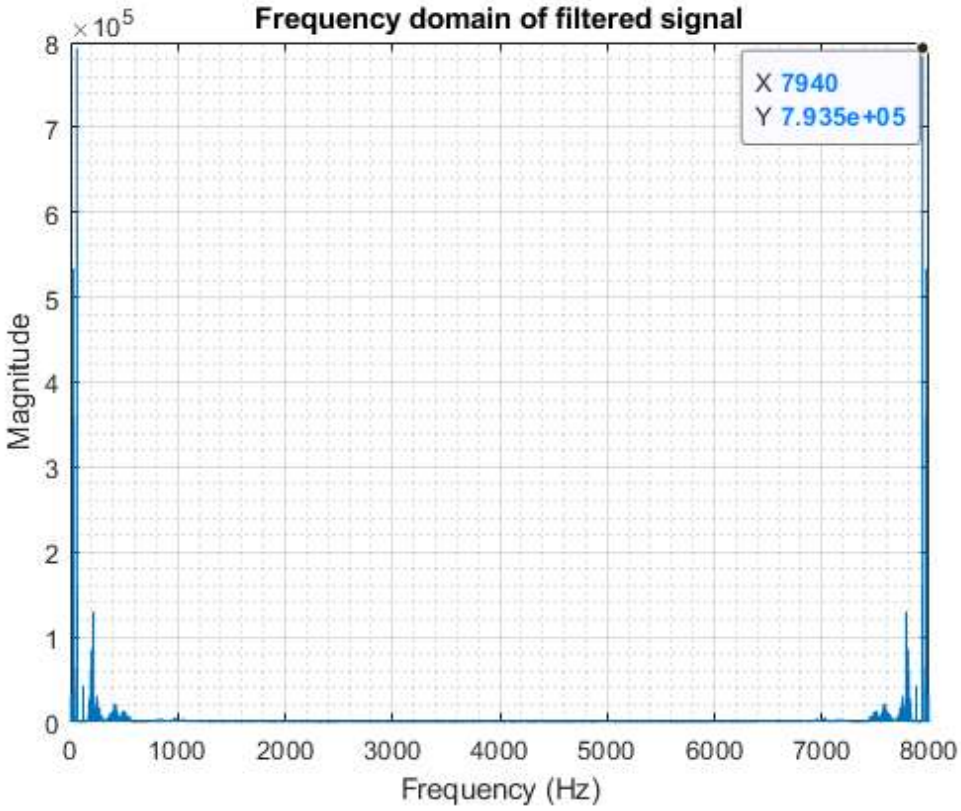

Figure 13 FFT results with the highest dominant frequency.

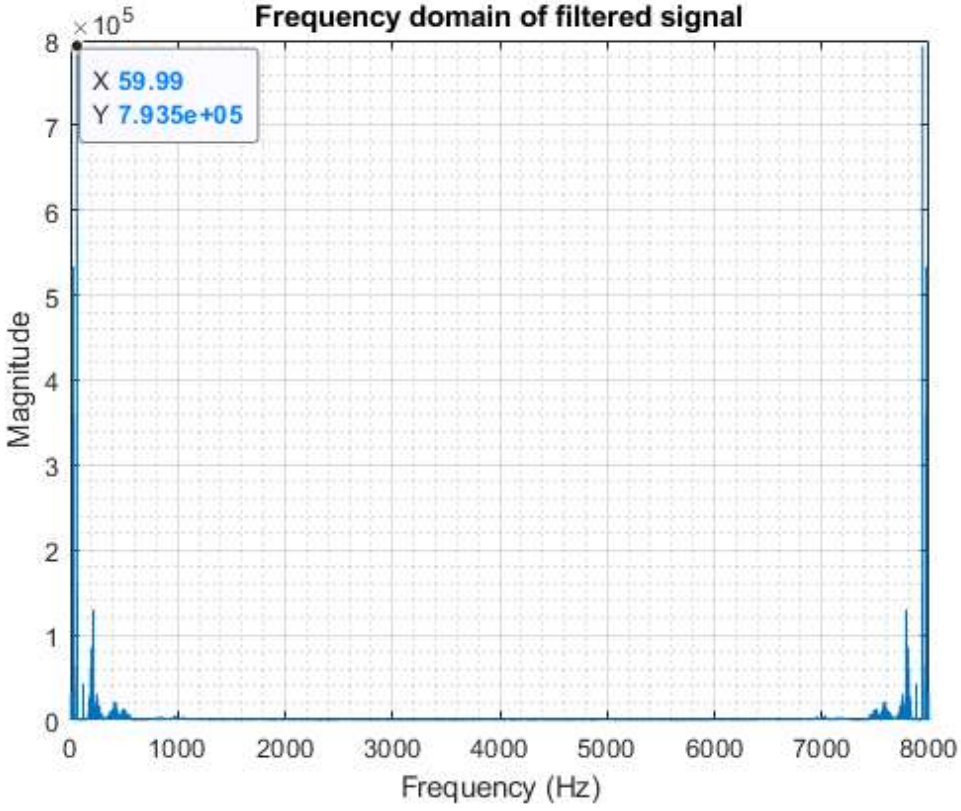

Figure 14 FFT results with the lowest dominant frequency

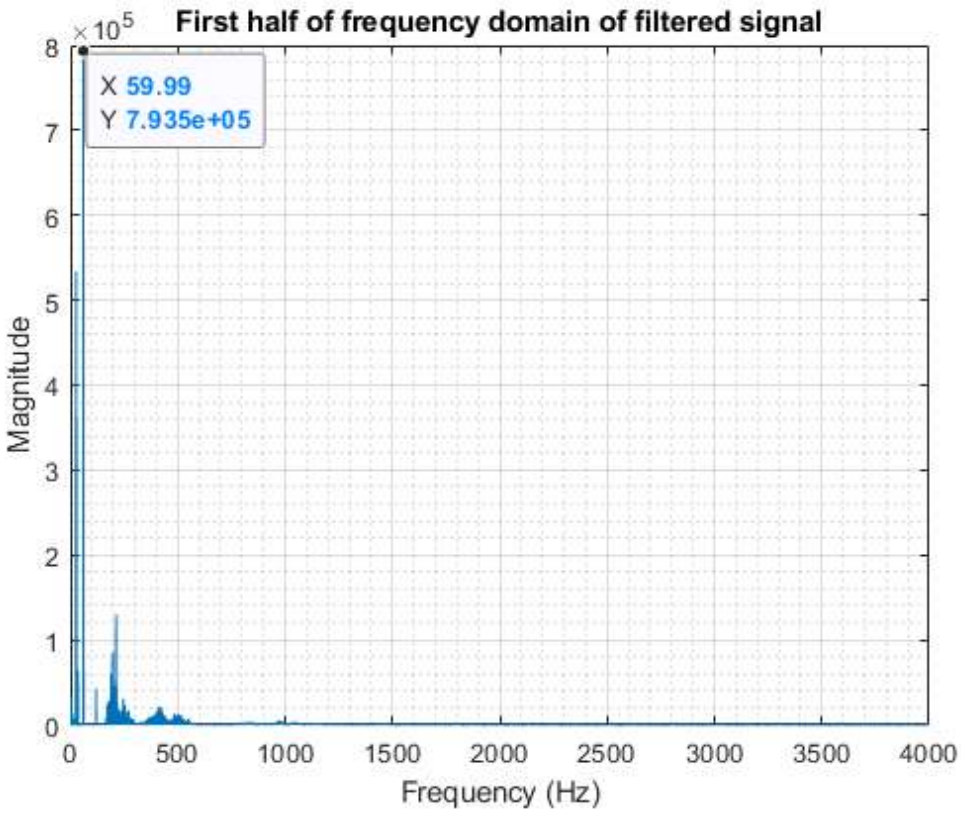

Figure 15 First half of the FFT calculations with the lowest dominant frequency

Based on the above results, it can be noted that although the filter response implies that the FIR filter operates correctly, the noise is still present, and a very small amount has been removed.

Results: Autocorrelation & Power Spectral Density.

The results of the autocorrelation and its PSD are presented below.

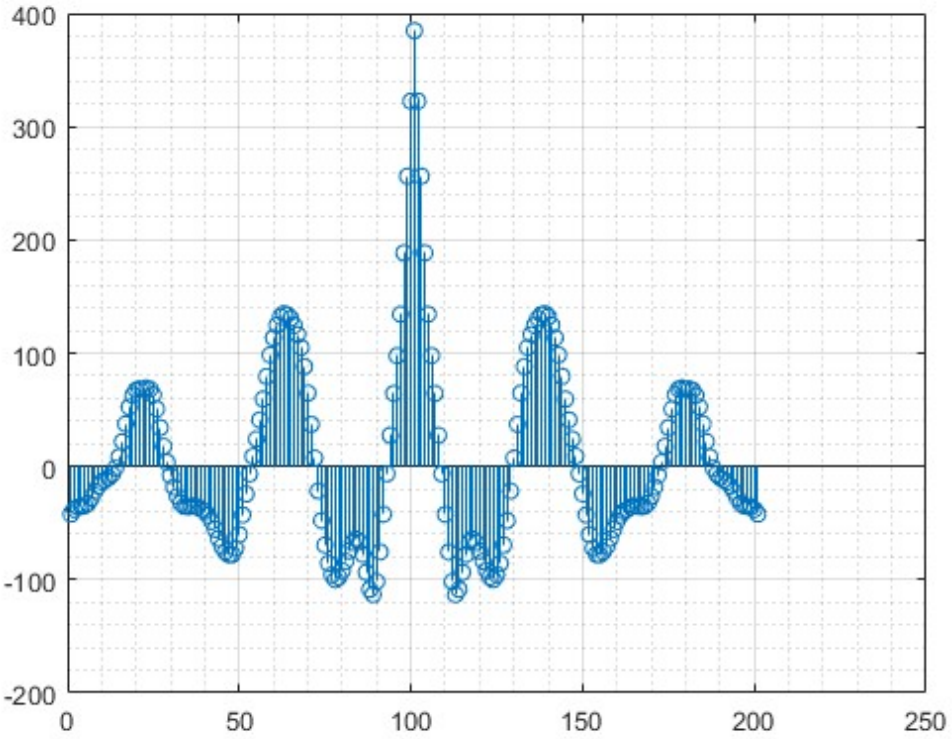

Figure 16 Autocorrelation plot of the filtered signal.

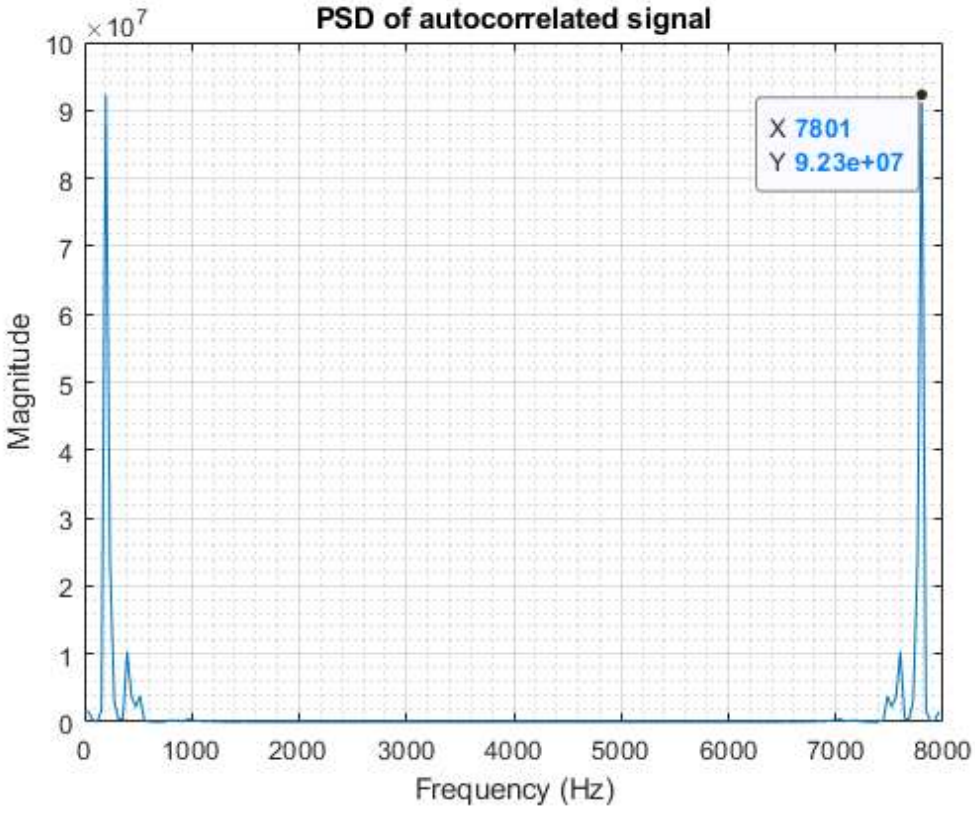

Figure 17 PSD plot of the autocorrelated signal with the highest dominant frequency.

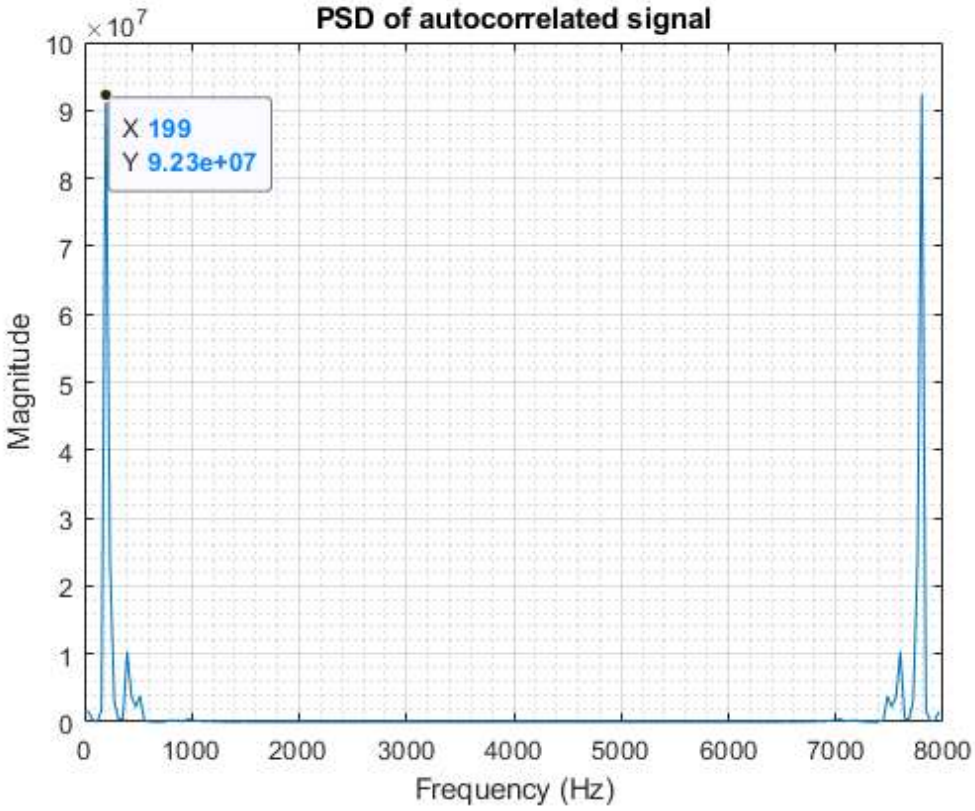

Figure 18 PSD plot of the autocorrelated signal with the lowest dominant frequency.

The above plots confirm the correct calculation of the PSD by statistical means and present the behavior of the signal.

Results: LMS Implementation.

For the LMS implementation the following results were obtain for different step sizes.

Step size = 0.005

For step size equal to 0.005 the following results were obtained.

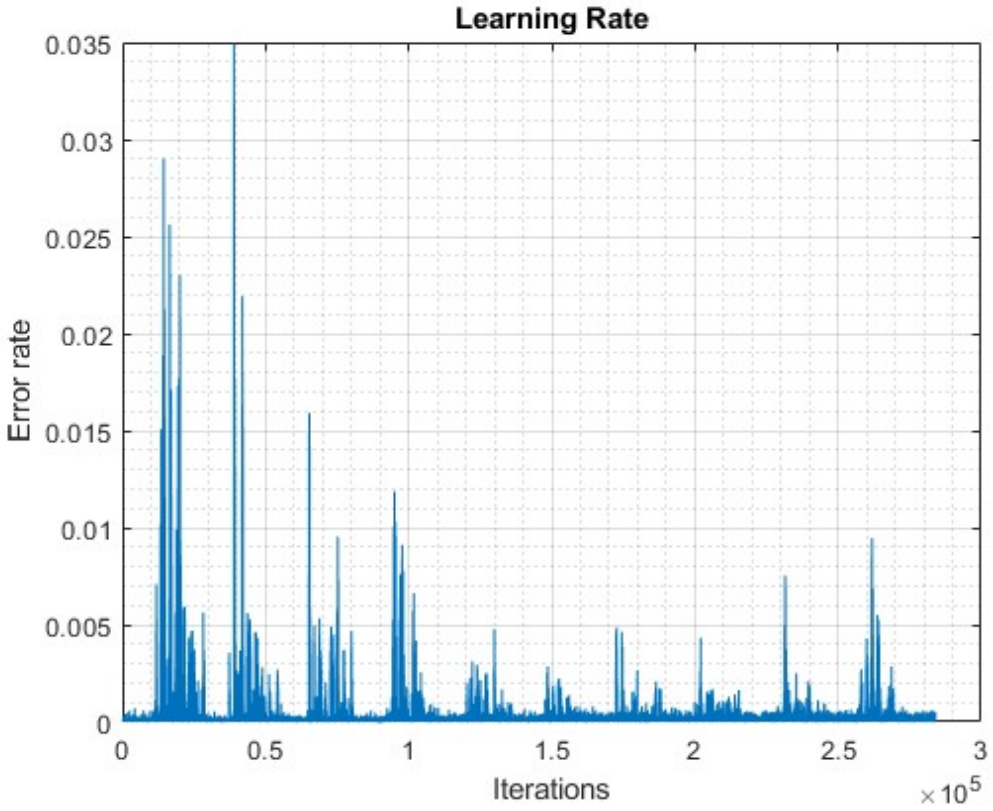

Figure 19 Learning rate of LMS with step size=0.005

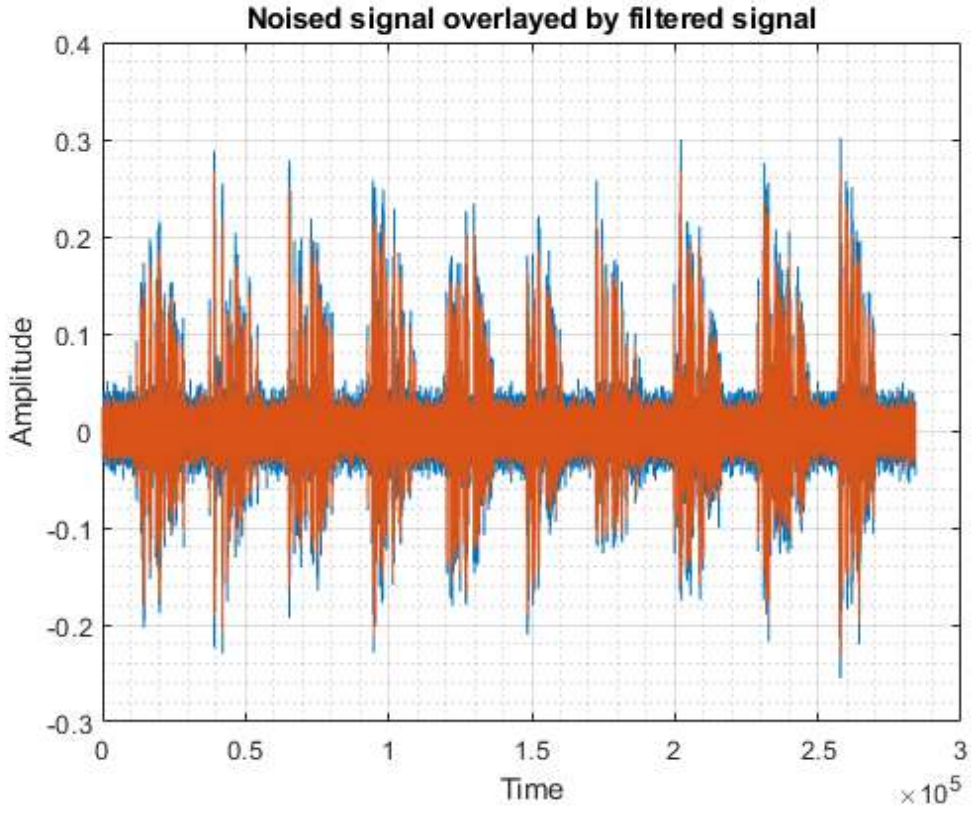

Figure 20 Signal filtered by LMS with step size=0.005 (Red) plotted on top of the noised signal (Blue).

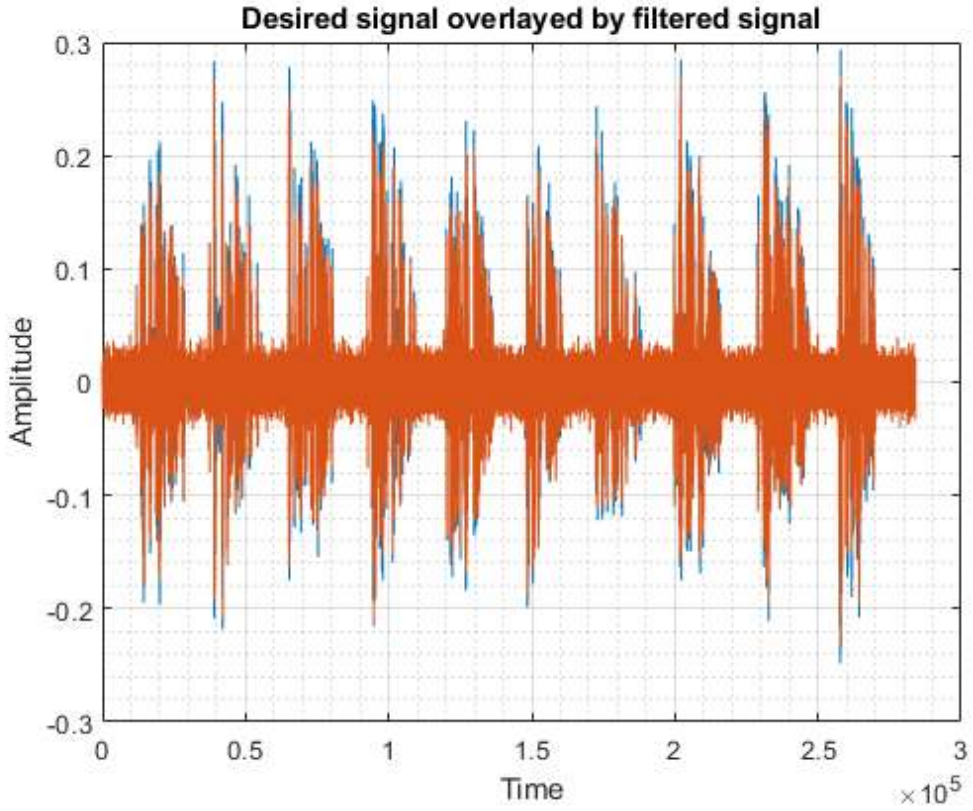

Figure 21 LMS filtered signal with step size=0.005 (Red) plotted on top of the desired signal (Blue).

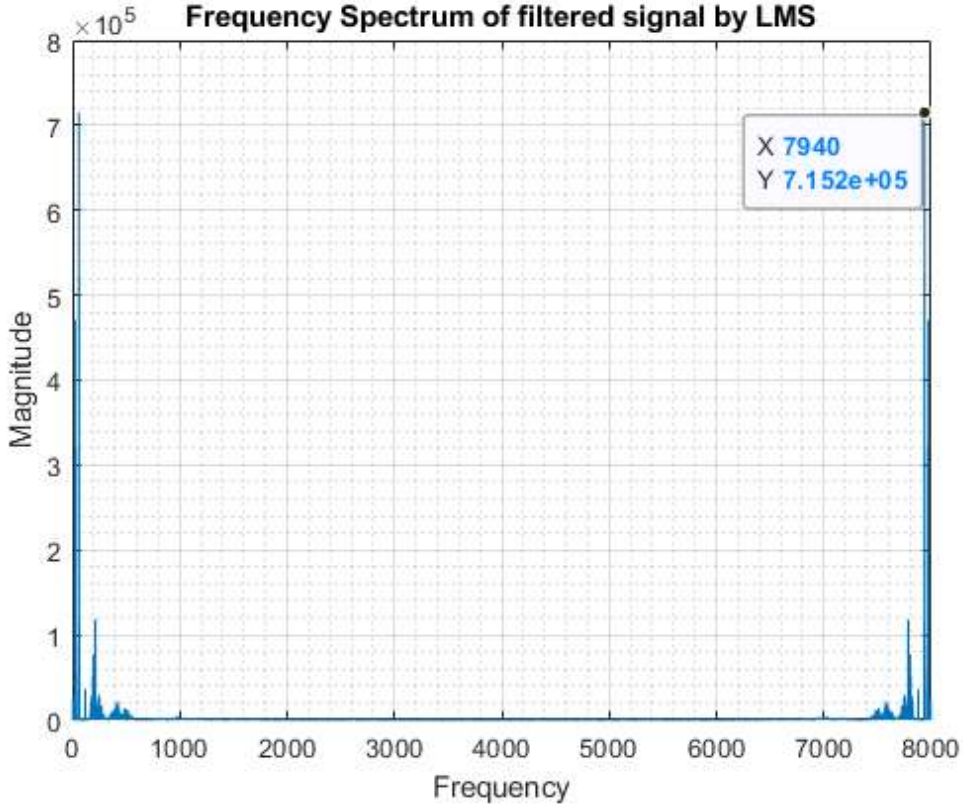

Figure 22 Frequency spectrum of LMS filtered signal with step size=0.005 and the highest dominant frequency.

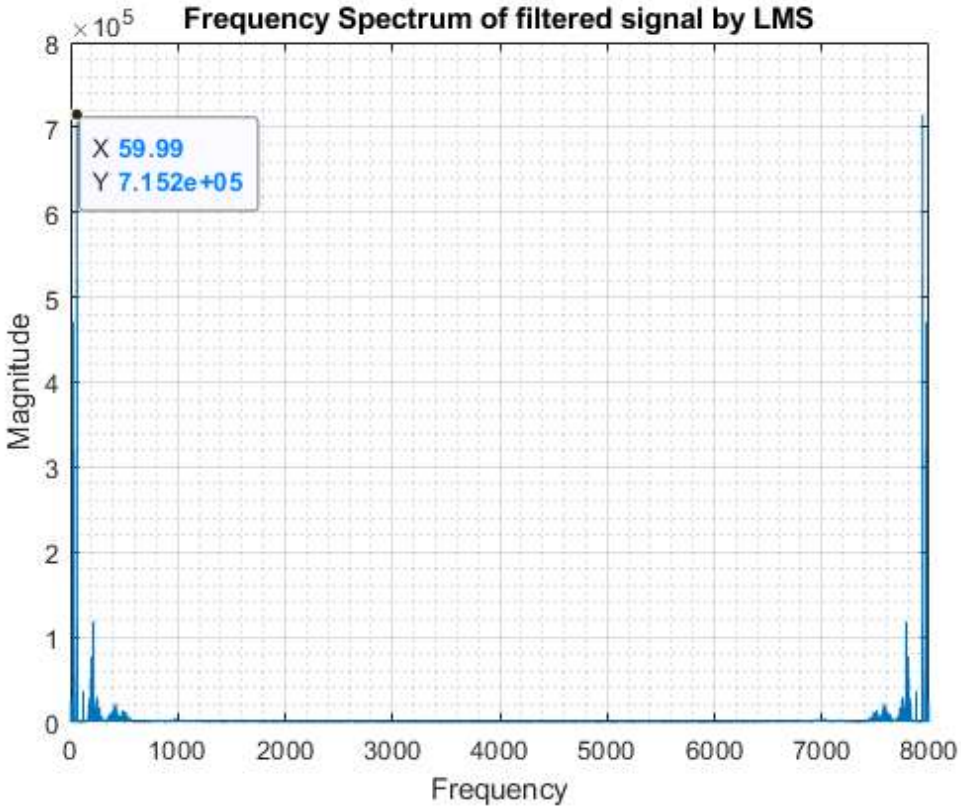

Figure 23Frequency spectrum of LMS filtered signal with step size=0.005 and the lowest dominant frequency.

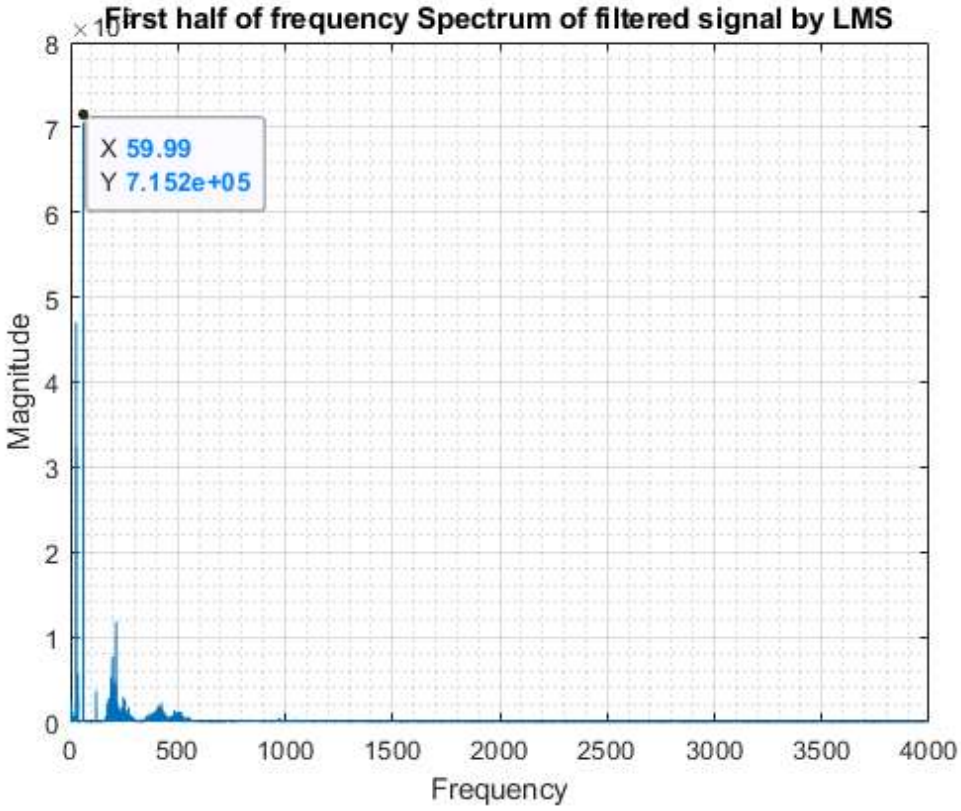

Figure 24 First half of frequency spectrum of LMS filtered signal with step size=0.005 and the lowest dominant frequency.

Based on the results presented above, it can be noted that while the learning rate fluctuates, the algorithm is operational. The removal of noise is noticeable and the filtered signal while still has some noise, is closer to the desired in comparison with the FIR filtered signal.

Step size= 0.0005

The results of step size equal to 0.0005 are presented below.

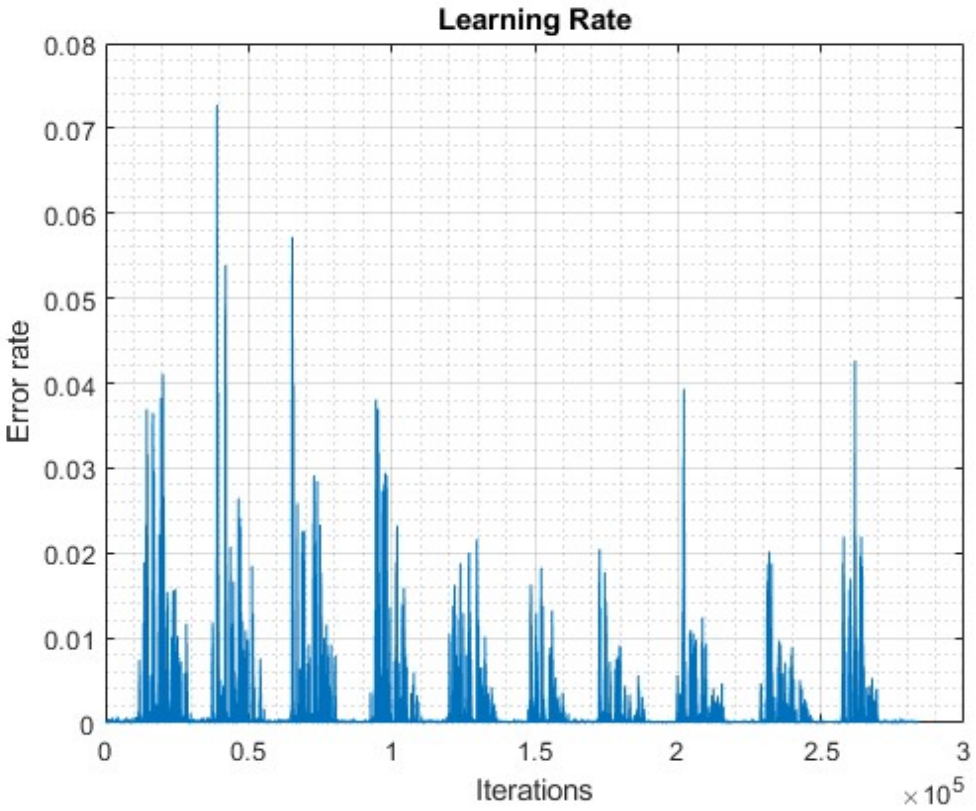

Figure 25 Learning rate of LMS with step size 0.0005.

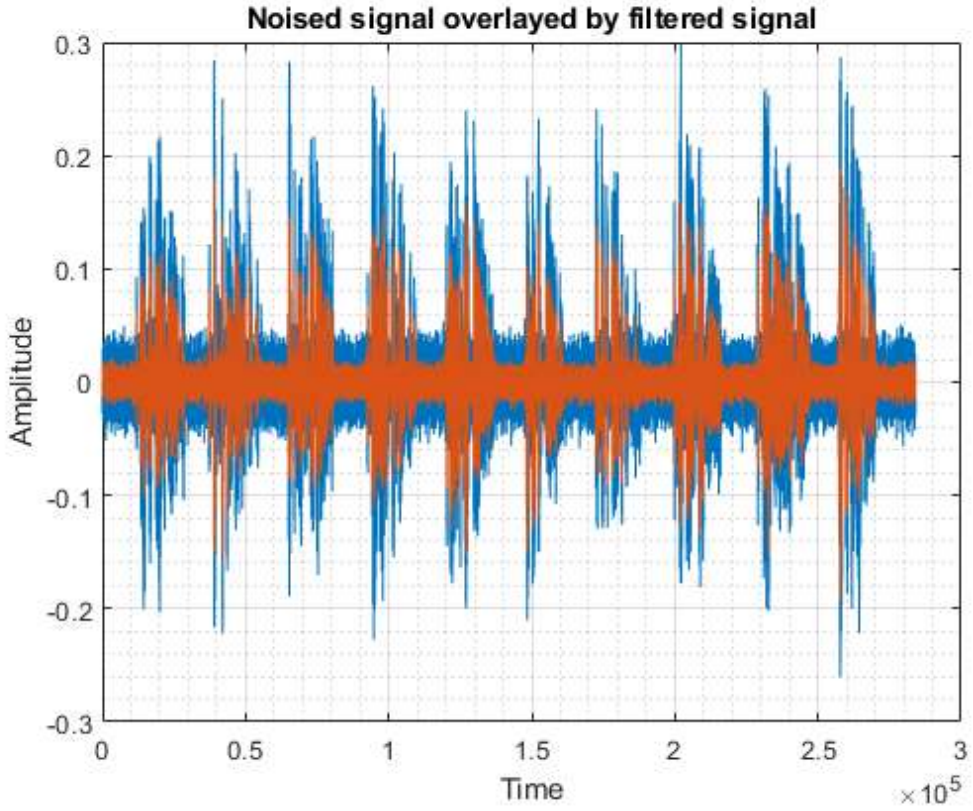

Figure 26 LMS filtered signal with step size 0.0005 plotted on top of the noised signal.

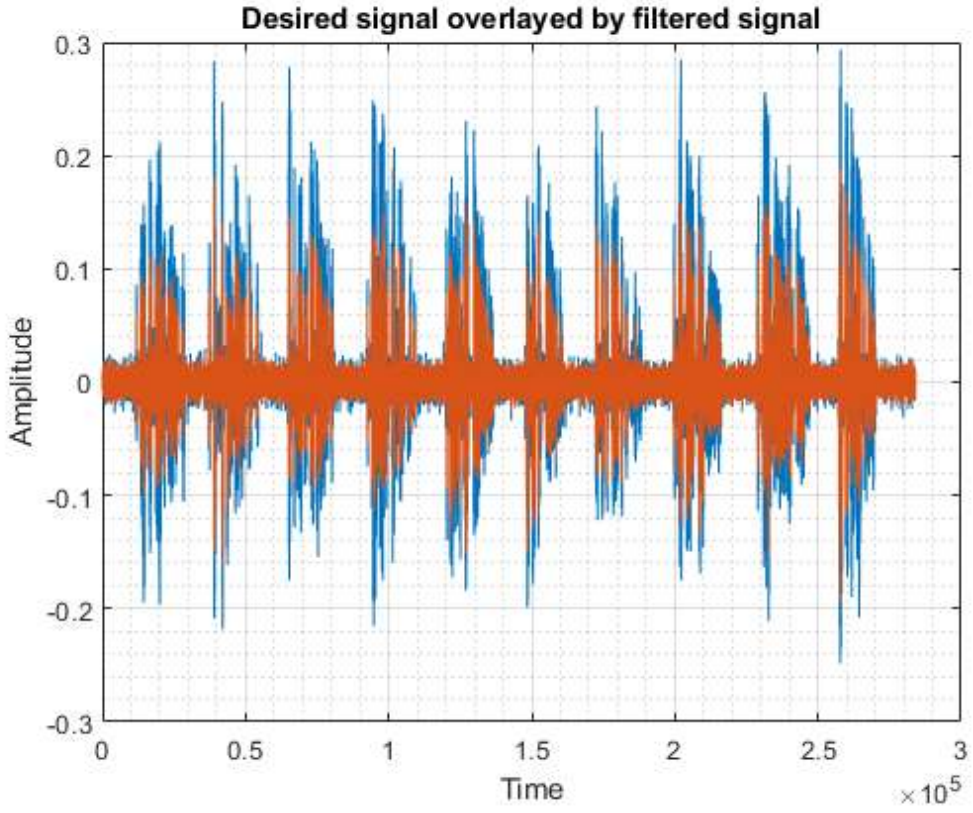

Figure 27 Filtered LMS with step size 0.005 plotted on top of the desired signal.

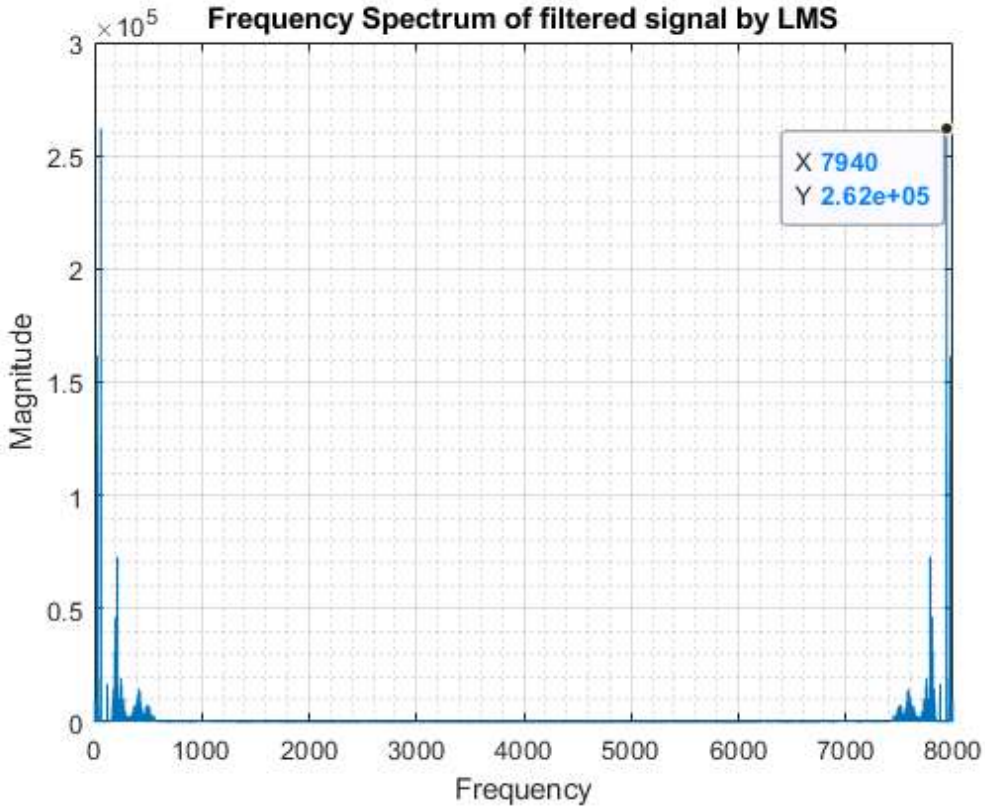

Figure 28 Frequency spectrum of LMS with step size 0.0005 and highest dominant frequency.

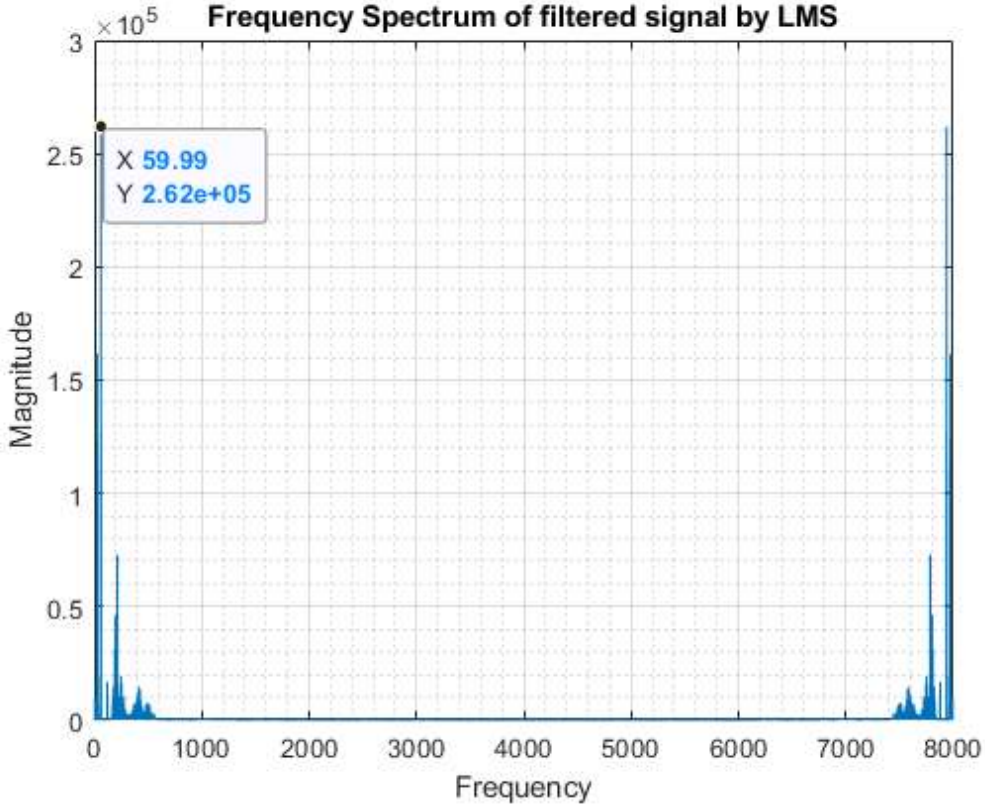

Figure 29 Frequency spectrum of LMS with step size 0.0005 and lowest dominant frequency.

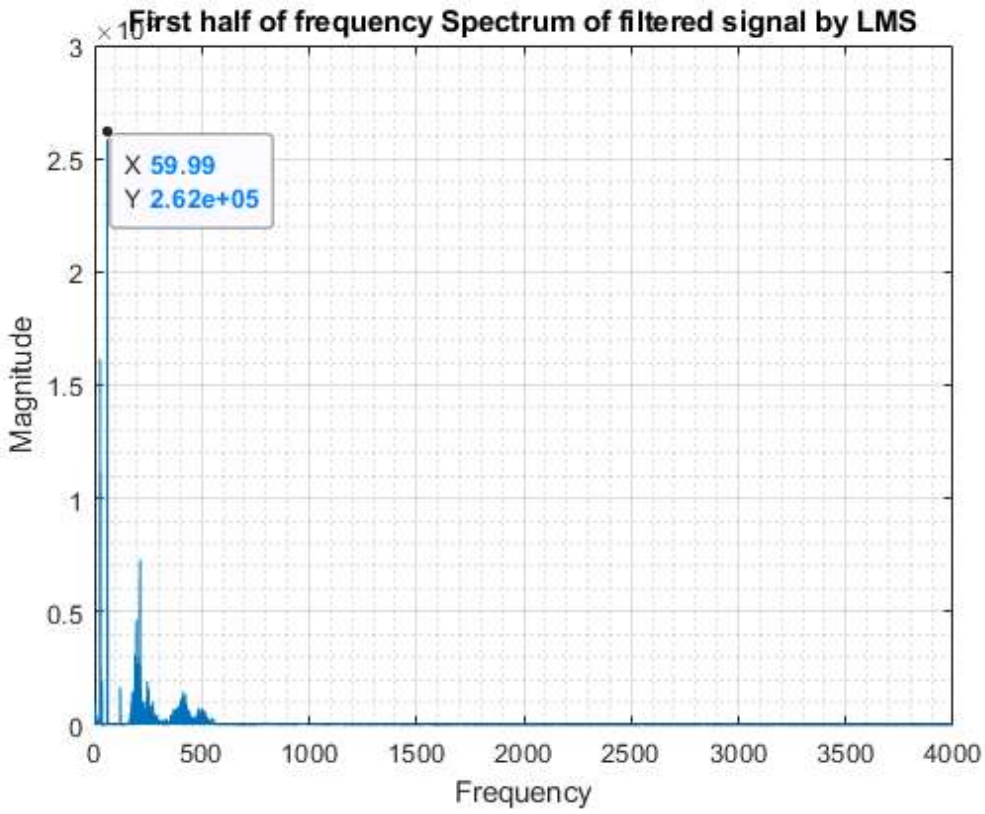

Figure 30 First half of frequency spectrum of LMS with step size 0.0005 and lowest dominant frequency.

Based on the above results, it can be noted that the LMS error rate is higher in comparison to the step size= 0.005 but it produces signal closer to the desired. However, the magnitude appears to be lower than the magnitude of the step size of 0.005.

Step size=0.00005

The results of the LMS with step size equal to 0.00005 are presented below.

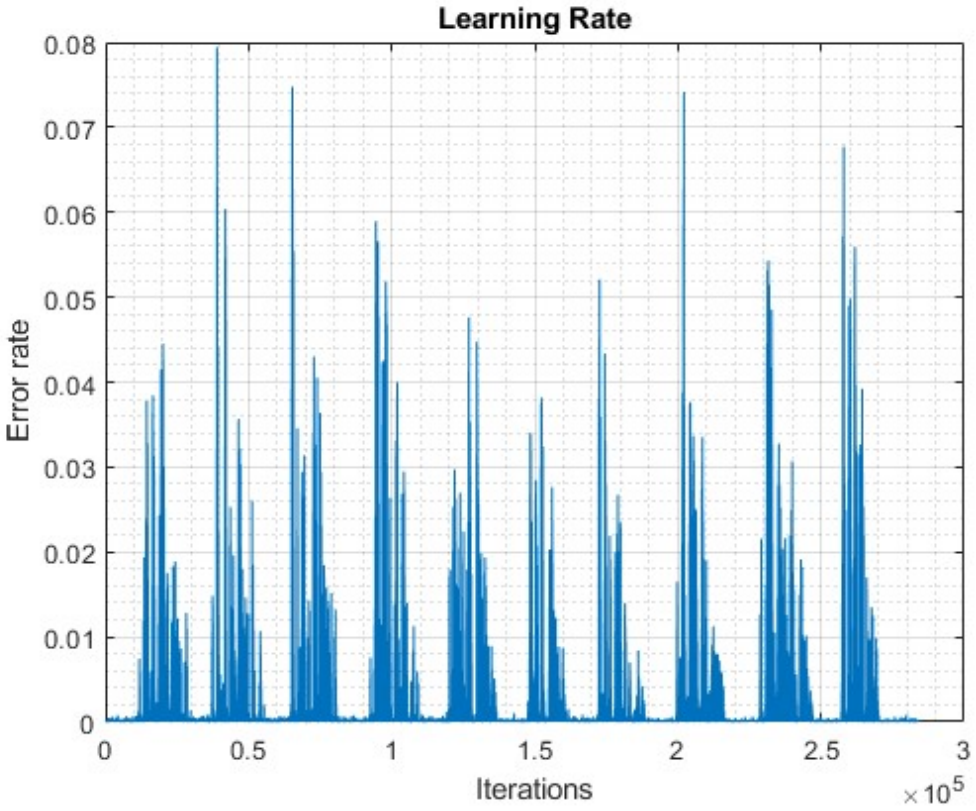

Figure 31 Error rate for LMS with step size 0.00005

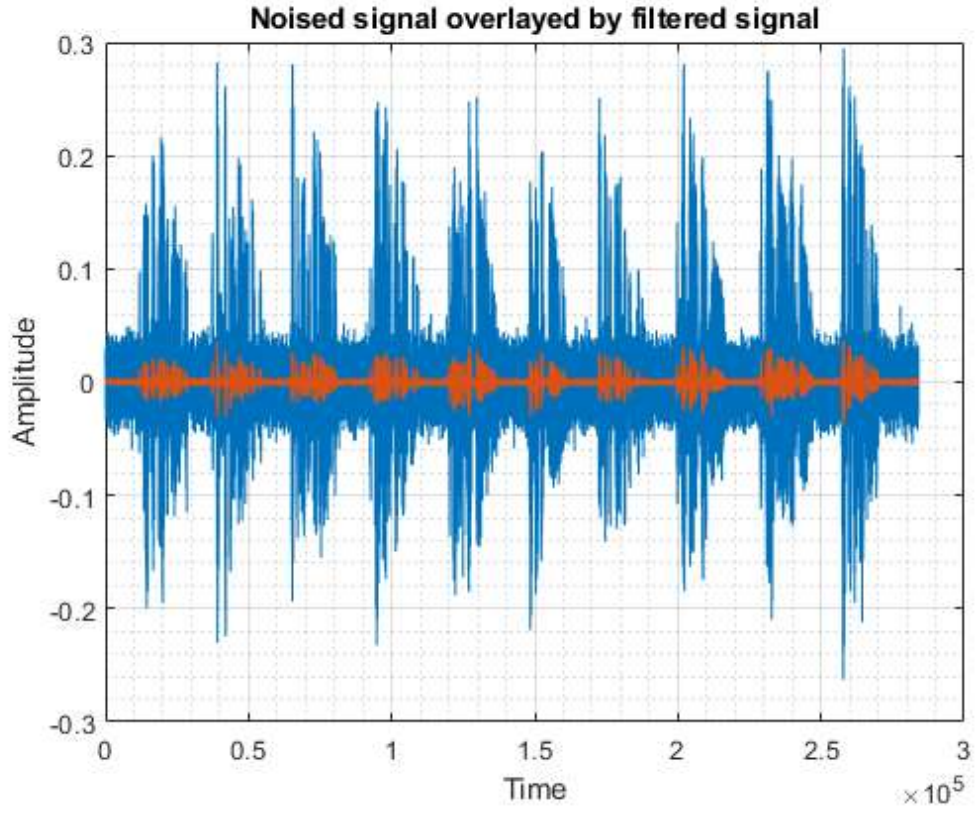

Figure 32 LMS signal with step size 0.00005 plotted on top of the noised signal.

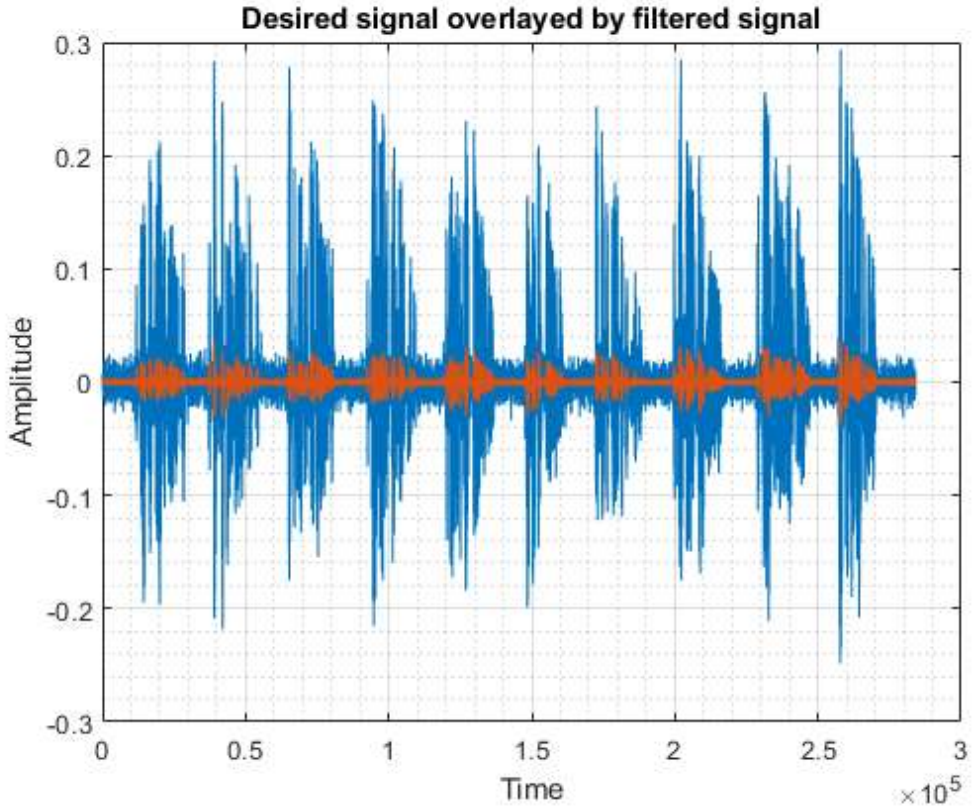

Figure 33 LMS signal with step size 0.00005 plotted on top of the desired signal.

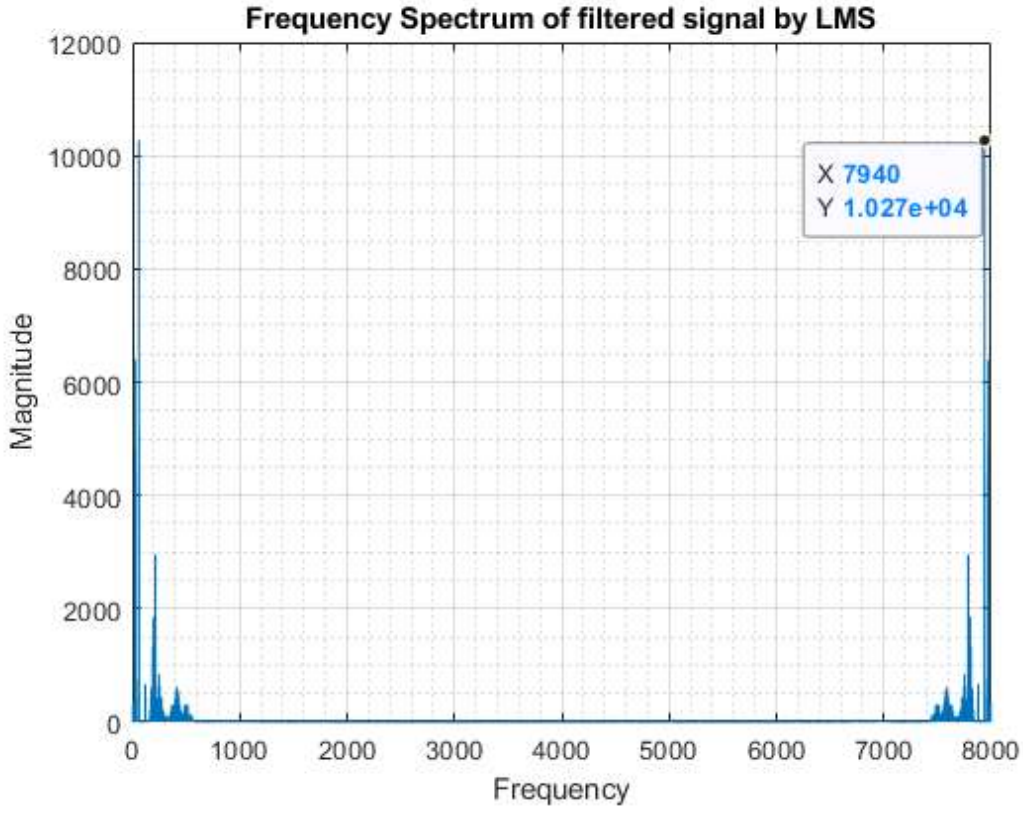

Figure 34 LMS frequency spectrum with step size 0.00005 and highest dominant frequency.

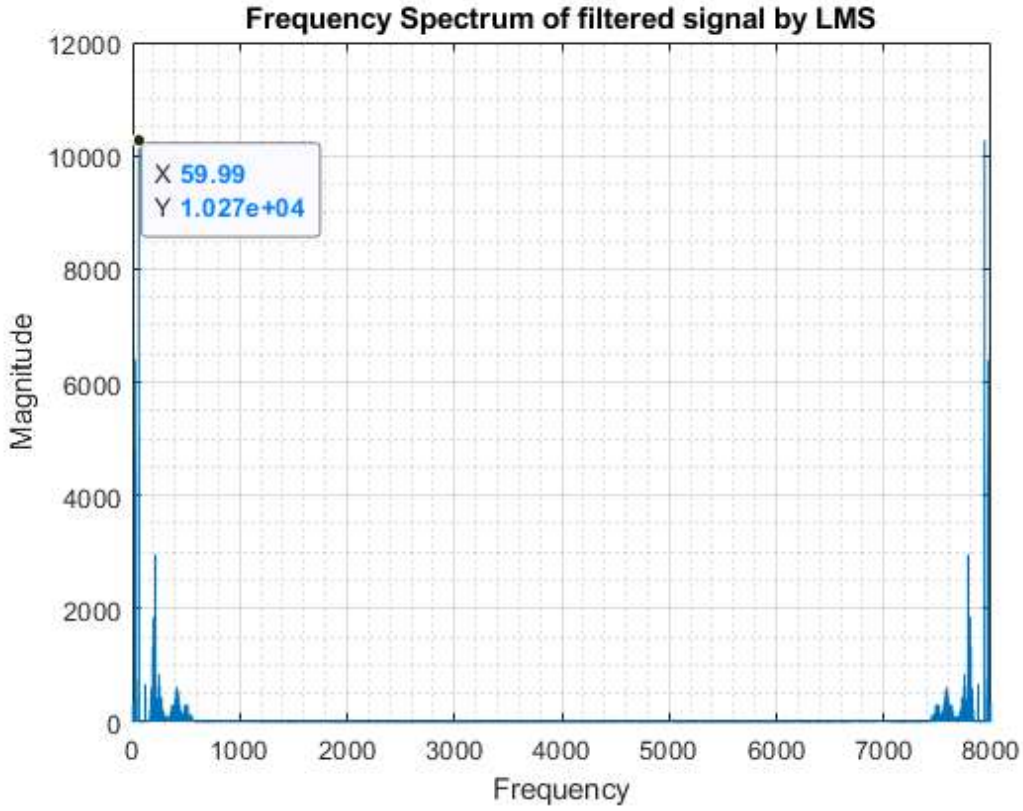

Figure 35 LMS frequency spectrum with step size 0.00005 and lowest dominant frequency.

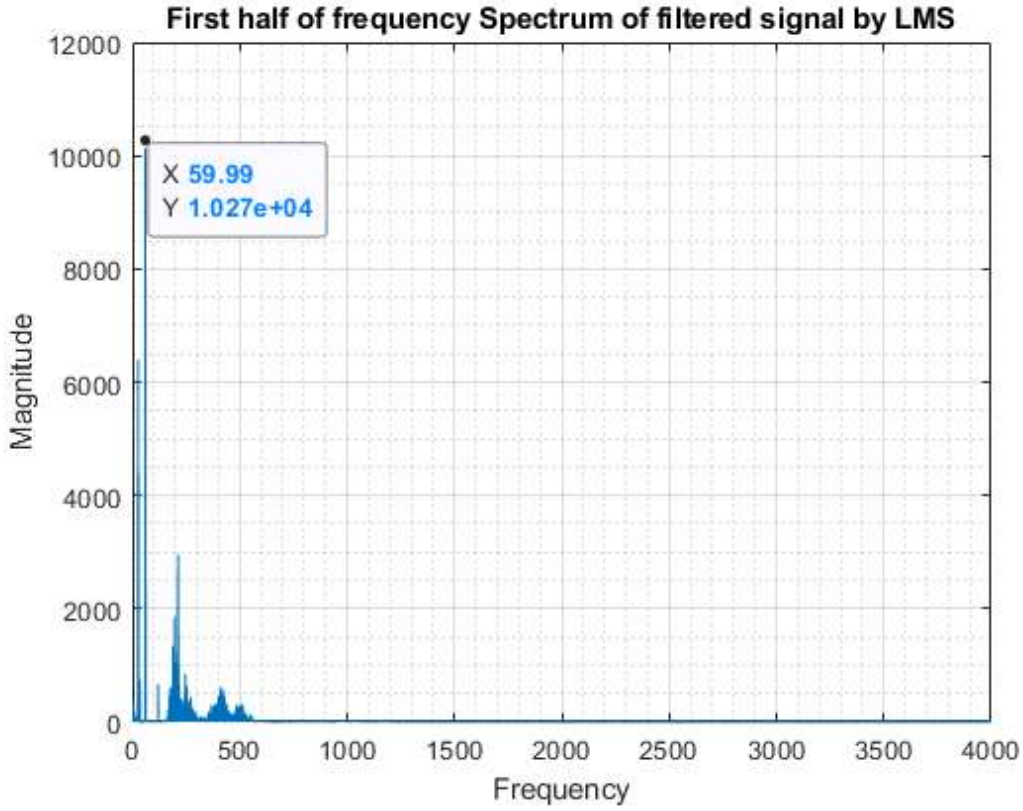

Figure 36 First half of frequency spectrum of LMS with step size 0.00005 and lowest dominant frequency.

Based on the above results it can be noted that the filtered signal is significant lower in its magnitude and the error rate is even higher than the previous step sizes.

Analysis

Regarding the first task of this paper, the input signal is a stochastic since its contents are a person talking. The injection of random normally distributed noise, therefore, also becomes a stochastic signal, since it inherits the same signal properties of the un-noised signal. The FFT calculations that have been performed on both signals return two dominant frequencies, the 60Hz and 7940Hz. The 60Hz frequency is perhaps due to equipment noise that was utilized to record the audio and the 7940Hz is due to the sampling frequency being 8000Hz. Based on those observations, it can be noted that the noise rests on top of those two frequencies, while ignoring the intermediate frequencies.

The FIR bandpass filter utilized those two frequencies in order to attempt to filter out the noise from the signal. While the filter's response was the expected one, having linearity in its phase and allowing the defined bandwidth, it failed to filter out the noise from the signal. That is to be expected, though, because as mentioned above, the noise is present on the most frequent frequencies and the filter coefficients are static for every sample of the signal. While that implementation would work on a deterministic signal, the FIR filter fails on random signals.

The autocorrelation reveals that the system is an ARMA process, meaning Autoregressive Moving Average. That is also to be expected since the speech signal is an autoregressive process and the FIR filter is a Moving Average process. The utilization of the autocorrelation in order to calculate the PSD is also proven since it correlates with the results obtained by the FFT implementation.

The LMS algorithm implementation presents better results in comparison with the FIR filter, since it implements adaptive coefficients for the filtering, by estimating the Wiener-Hopf equation by means of the steepest descend algorithm, as discussed above. While it does not completely remove the noise from the signal, its product is very close to the original (desired) signal.

Conclusions

The filtering of noise present in stochastic signals is a complex subject; filtering techniques such as the FIR filter fail to produce an acceptable result, in most cases failing altogether to remove the noise. While the extraction of the frequency characteristics can assist with the filtering, nothing more can be done with the static filtering techniques. The implementation of feature extraction by statistical means can be very effective in order to implement adaptive filters. While adaptive filters are not necessarily optimal, they can be very efficient in filtering. In the specific implementation that this paper has presented, the LMS algorithm was implemented and has removed an AWGN from a random signal successfully.

Further development on this specific implementation that this paper has presented, are the implementation of an adaptive step size parameter on the LMS algorithm, in order to calculate the new coefficients more efficiently, the implementation of overlapping windows in order to have even smaller error rate and more “optimal” filter coefficients.

References

Hayes, M. (1996). *Statistical digital signal processing and modeling*. New York: Wiley.

Ingle, V., & Proakis, J. (2012). *Digital signal processing using MATLAB²*. Stamford, Conn: CENGAGE Learning.

Manolakis, D., Ingle, V., & Kogon, S. (2005). *Statistical and adaptive signal processing*. Boston: Artech House.

Monson, M. (1999). *Schaum's outline of theory and problems digital signal processing*. New York: McGraw-Hill.

Mulgrew, B., Grant, P., & Thompson, J. (2001). *Digital signal processing*. Houndmills: Palgrave.

Poularikas, A., & Ramadan, Z. (2006). *Adaptive filtering primer with MATLAB*. Boca Raton: Taylor & Francis.

Vaseghi, S. (2008). *Advanced digital signal processing and noise reduction*. Chichester, U.K.: J. Wiley & Sons.