Study on the Denoising of Sounds and Images

Yeonho Jung

Abstract

Using computational signal analysis, this paper executes noise removal through a Low Pass Filter (LPF) and various windows. By executing noise removal with various combinations of filters and windows, we search for efficient models. We substantiate the efficiency of the filter and the windows through a comparison of the pure sample, pure voice, and noise removal outputs. Application of LPF reduced the amplitude of pure noise, while application of windows augmented the amplitude of pure voice. By applying both LPF and windows, we can execute efficient noise removal.

Also, for analysis of denoising of bioimage, further experiments with the proposed filter showed that it increased the resolution of the image while taking less time compared to other filters tested before to form the final image. The proposed filter was created by multiplying three parts: LPF, new filter equation, and full K-space. Using the selected filter, vast data that was collected from a patient were transformed into a final image using mathematical transformations.

Introduction

Noise reduction is essential to produce audible sound files and proceed with precise communication; it is also of vital importance in creating images. Various noises disrupt and distort the original sounds and signals, puzzling the audience. Especially, when noise interferes with speech, the noise obscures the speech and results in the speech in slightly degraded to completely unintelligible, depending on its amount and type. Because these noises are inevitable, accurate noise removal is necessary for clear communication. Because we are relatively more sensitive to the higher frequency sounds than lower frequency sounds, we are vulnerable to noises, that mainly consisted of high-frequency sounds compared to main sound sources such as voice. To emphasize and enhance the original signal, we diminish the noise and amplify the signal.

Numerous filters are designed to reduce the effects of noise and enhance the original sound. These noise-removing systems are improved by implicating various filters and windows, also applying some algorithms. Discrete-wavelet transform-based algorithm is used for speech denoising and the outputs are calculated using thresholding methods. [1] Other algorithms, such as Least Mean Square, Normalized

Least Mean Square, and Sign-Data Least Mean Square, are used to remove the noise efficiently. [2] Also, a stand-alone noise suppression algorithm is investigated for its reduction of spectral effects of acoustic noise. [3] Not only the algorithms but also the filters, such as Finite Impulse Response filters using the Frequency Response Masking technique, are designed to remove high-frequency noise or randomized noise. [4-5] Karam and Verteletskaya investigated using spectral subtraction to remove noise from noisy speech signals in the frequency domain. [6-7]

The main objective of part A of this research is to find the best-fit filter design in order to reach the highest efficiency in noise reduction.

The main purpose of part B of this research was to develop a better algorithm that would both enhance the quality of the final image and decrease the amount of time taken to produce it. An ideal LPF would be able to increase the resolution of an image as well as decrease the Ringing Artifact. In this experiment, various functions were tested as LPFs: Square function, Gaussian function, and circle function. All the functions showed their distinct features. When a square function was used as an LPF, it created the Sinc function over the image domain. When the domain of the square function increased, more data in the K-space was captured, resulting in images of a high resolution. When the domain of the square function decreased, fewer data in the K-space was captured, resulting in a lower resolution. In addition to the observation of the consequential resolution of the images produced, when a wide square function was used, a narrow Sinc function formed as a result.

*All graphs were produced by MatLab.

Materials and Methods

In this paper, we employ combinations of Low Pass Filter and some windows, such as Hanning, Bartlett, Hamming, and Blackman window, to remove noise. We implicate the filter design at simple trigonometric functions and realistic samples to substantiate the noise-removing efficiency of the filter design. We used MATLAB to analyze the audio files and execute Fast Fourier Transform.

A. Sound Analysis (*All the graphs are from implementing MatLab)

Noise Removal Using Simple Functions

Sample 1.

N = 400;
n = [0:N];
fm = 20;
fn = 100;
T = 0.001;

Voice: cos(2*pi*fm*n*T)
Noise: cos(2*pi*fn*n*T)

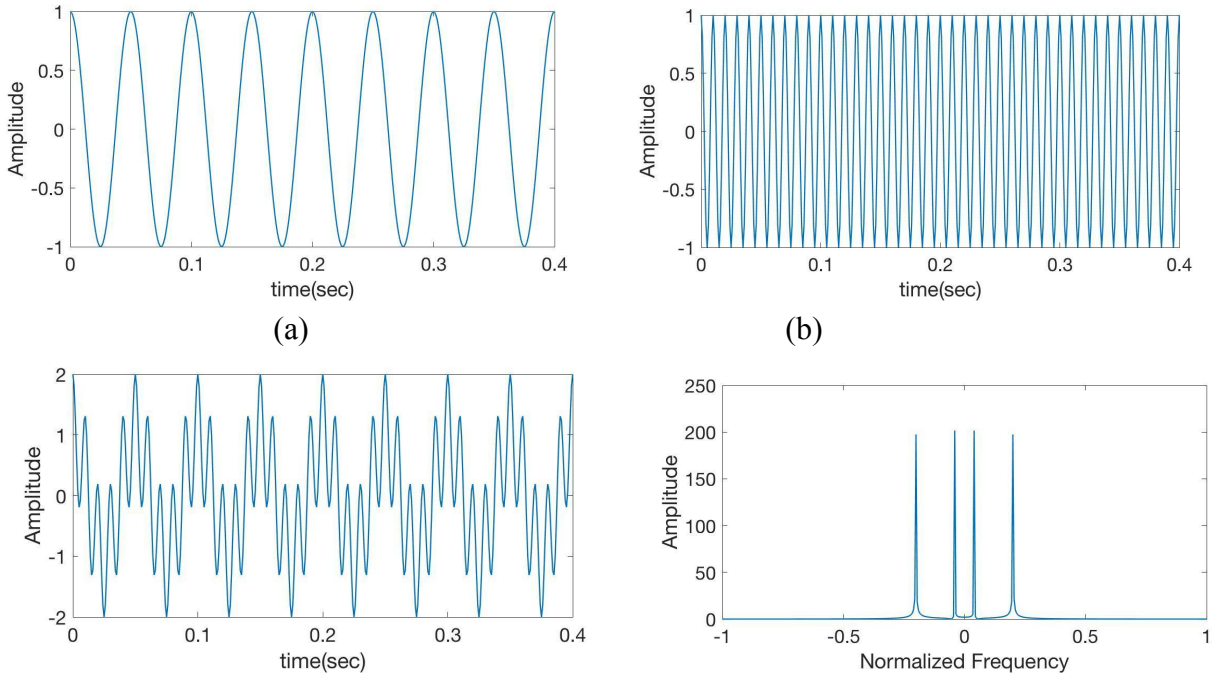


Fig 1 (a). Voice v(t) (b). Noise n(t) (c). Input signal = Voice v(t) + Noise n(t) (d). Fourier transformation of x(t) to find magnitude |FX| in the frequency domain

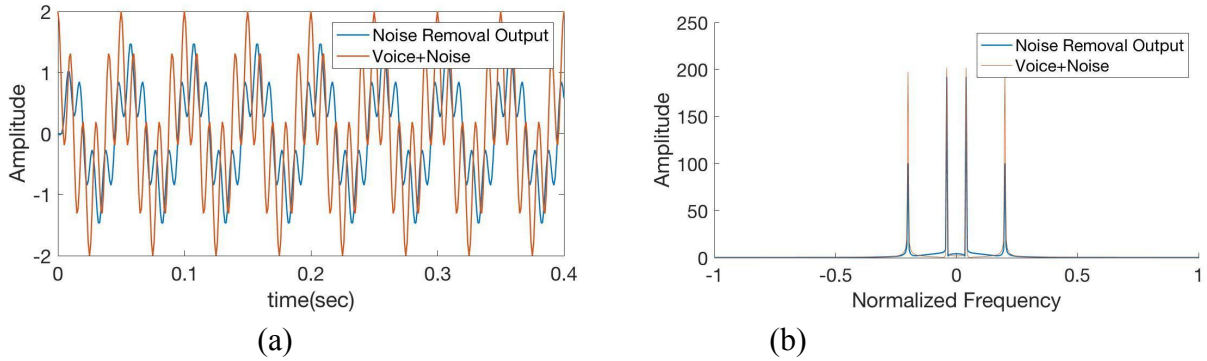
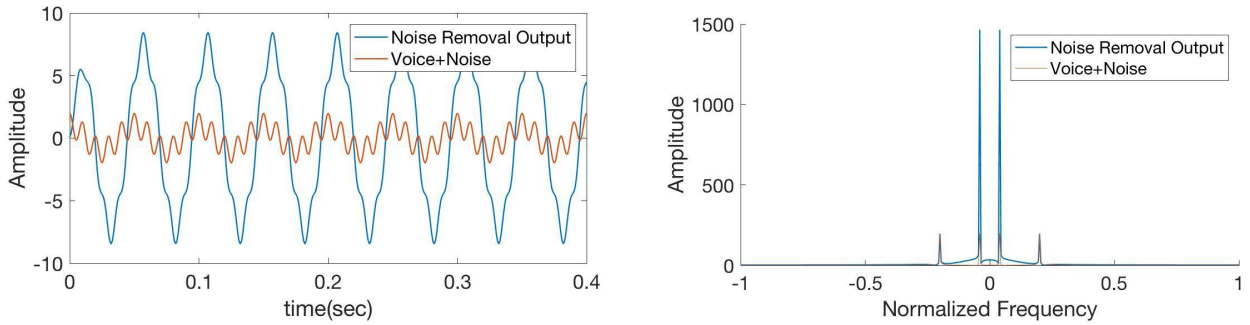
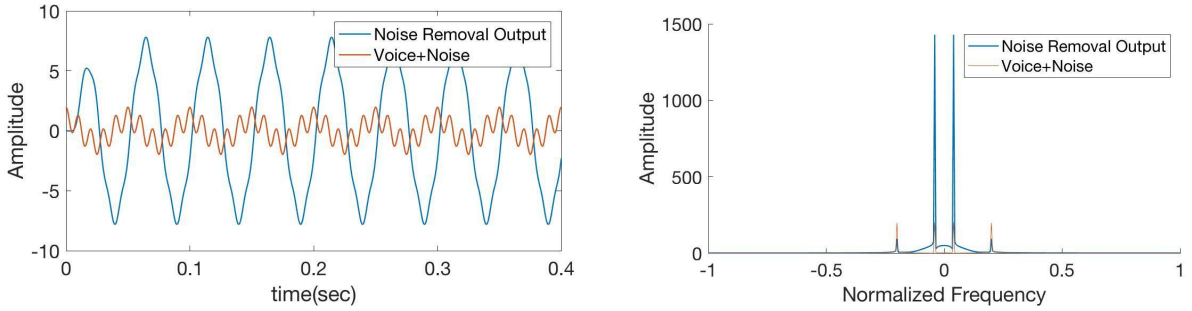


Fig 2 (a). Comparison of the graph y(t) after the use of the FIR filter vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice decreased a little while the amplitude of noise notably decreased when noise removal process is executed.



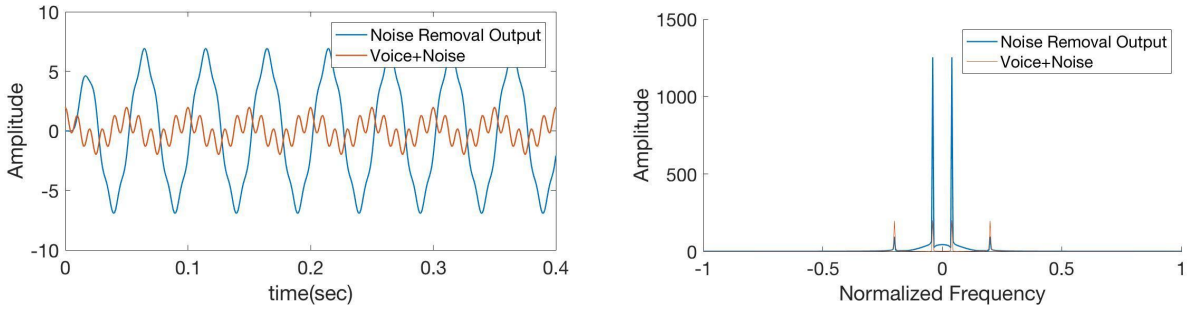
(a) (b)

Fig 3 (a). Comparison of the graph y(t) after the use of the **Hanning window** vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise stayed the same when noise removal process is executed.



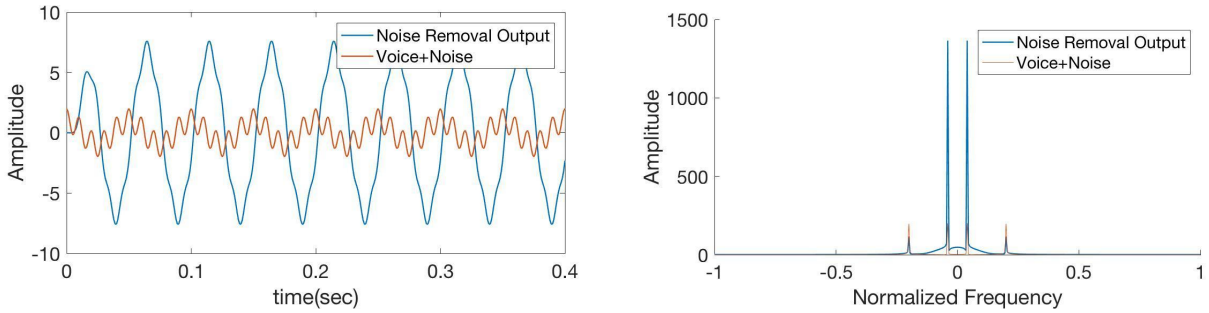
(a) (b)

Fig 4 (a). Comparison of the graph y(t) after the use of the FIR filter and Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



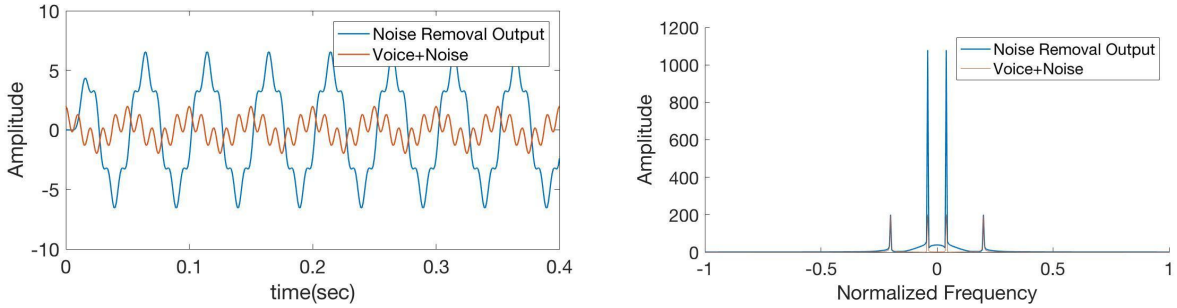
(a) (b)

Fig 5 (a). Comparison of the graph y(t) after the use of the FIR filter and Bartlett window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



(a) (b)

Fig 6 (a). Comparison of the graph y(t) after the use of the FIR filter and Hamming window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



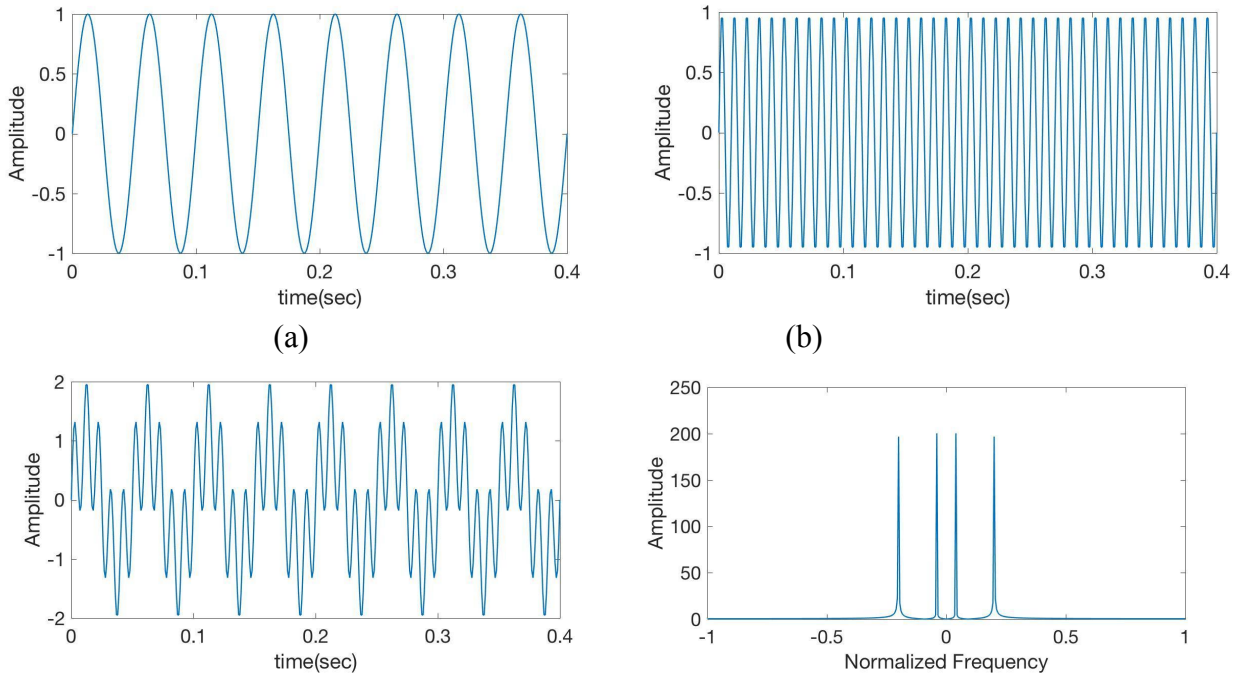
(a) (b)

Fig 7 (a). Comparison of the graph y(t) after the use of the FIR filter and Blackman window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.

Sample 2.

N = 400;
n = [0:N];
fm = 20;
fn = 100;
T = 0.001;

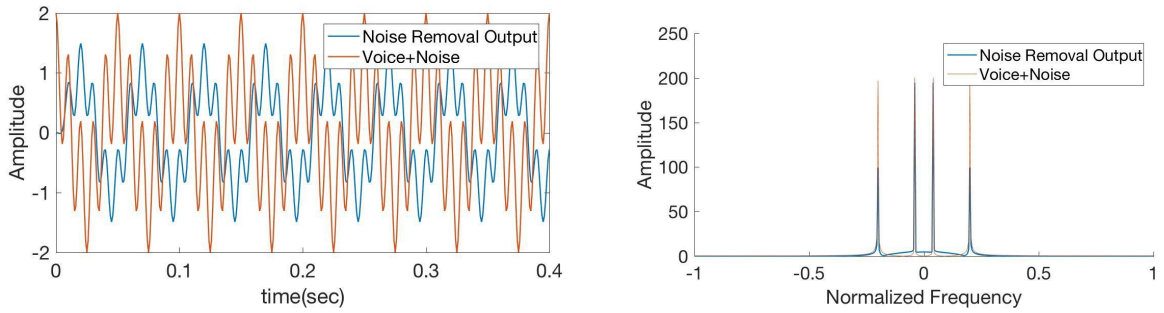
Voice: sin(2*pi*fm*n*T)
Noise: sin(2*pi*fn*n*T)



(a) (b)

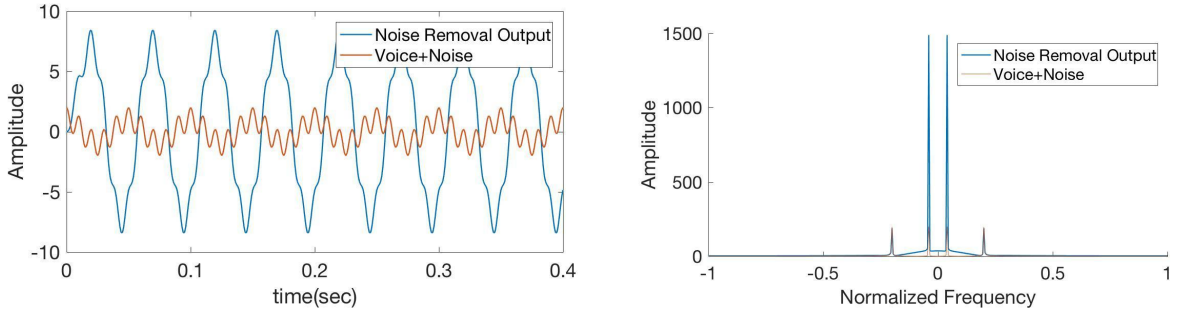
(c) (d)

Fig 8 (a). Voice v(t) (b). Noise n(t) (c). Input signal = Voice v(t) + Noise n(t) (d). Fourier transformation of x(t) to find magnitude |FX| in the frequency domain



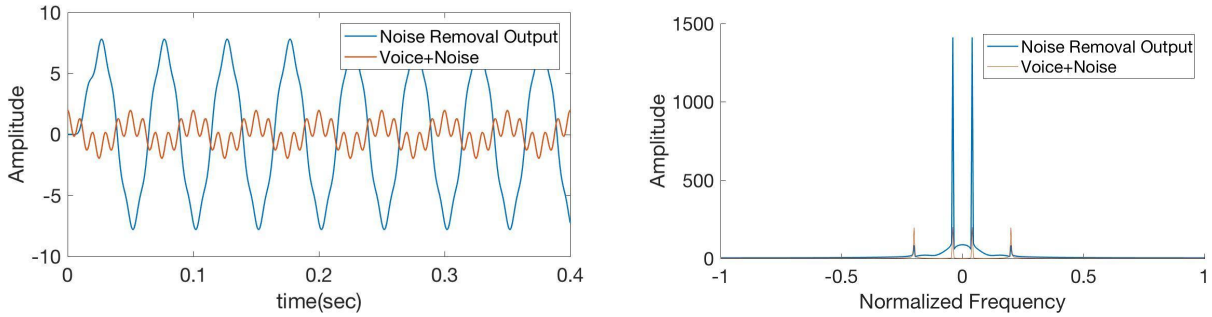
(a) (b)

Fig 9 (a). Comparison of the graph y(t) after the use of the FIR filter vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice decreased a little while the amplitude of noise notably decreased when noise removal process is executed.



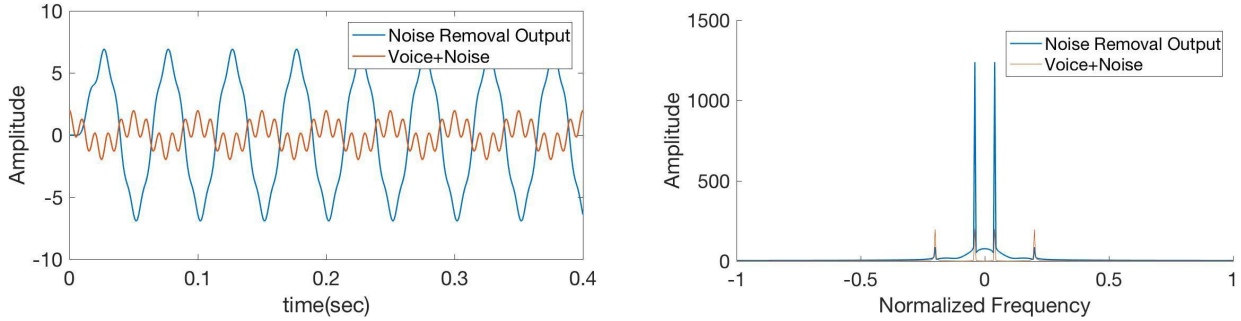
(a) (b)

Fig 10 (a). Comparison of the graph y(t) after the use of the Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise stayed the same when noise removal process is executed.



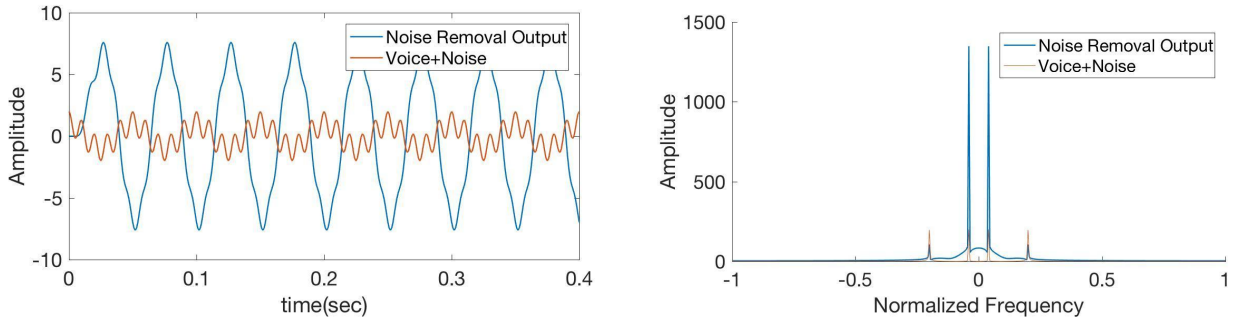
(a) (b)

Fig 11 (a). Comparison of the graph y(t) after the use of the FIR filter and Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



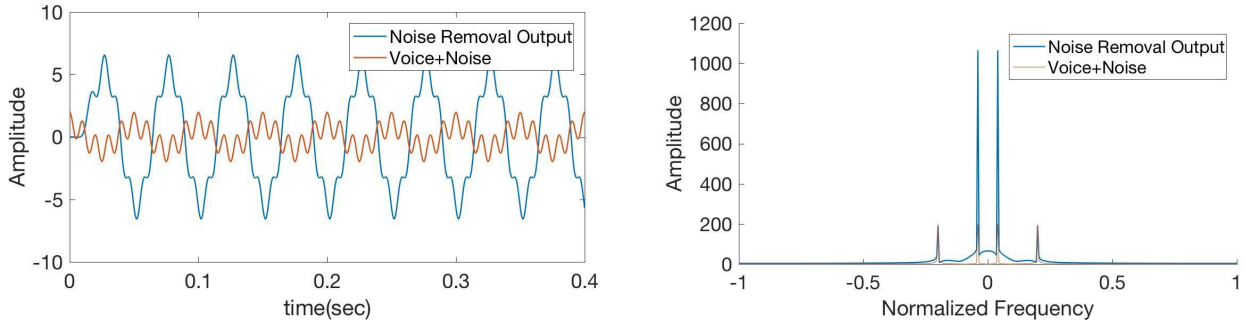
(a) (b)

Fig 12 (a). Comparison of the graph y(t) after the use of the FIR filter and Bartlett window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



(a) (b)

Fig 13 (a). Comparison of the graph y(t) after the use of the FIR filter and Hamming window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



(a) (b)

Fig 14 (a). Comparison of the graph y(t) after the use of the FIR filter and Blackman window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise stayed the same when noise removal process is executed.

Sample 3.

N = 400;
n = [0:N];
fm = 20;
fn = 100;
T = 0.001;

Voice: cos(2*pi*fm*n*T)
Noise: 0.6*cos(2*pi*fn*n*T)+0.8*sin(2*pi*fn*n*T)

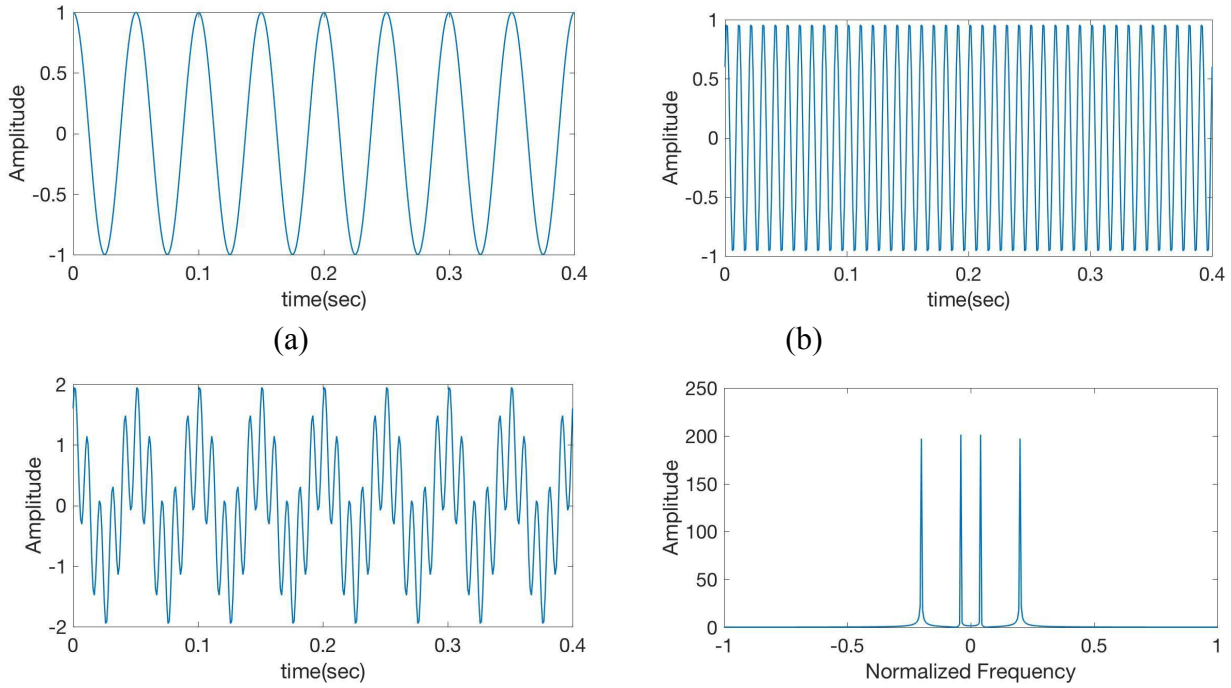


Fig 15 (a). Voice v(t) (b). Noise n(t) (c). Input signal = Voice v(t) + Noise n(t) (d). Fourier transformation of x(t) to find magnitude |FX| in the frequency domain

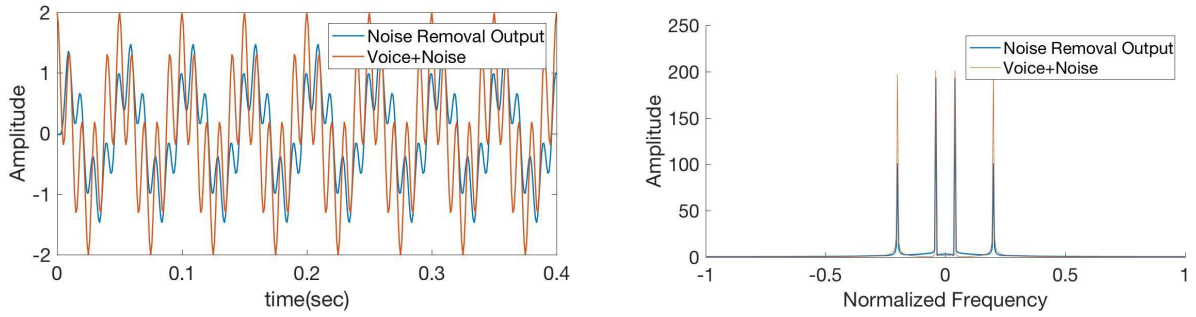
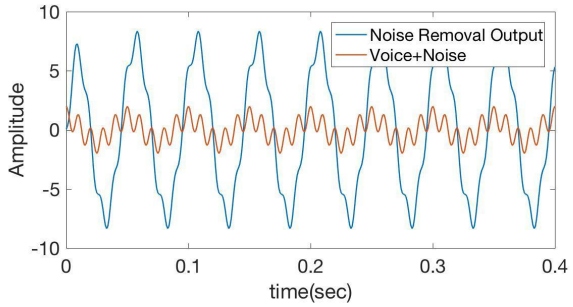
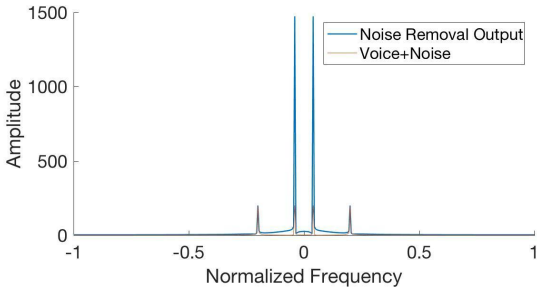


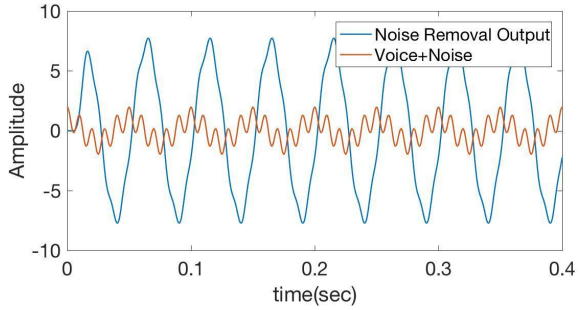
Fig 16 (a). Comparison of the graph y(t) after the use of the FIR filter vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice decreased a little while the amplitude of noise notably decreased when noise removal process is executed.

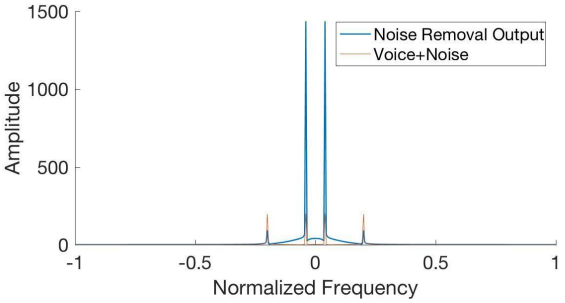




(a) (b)

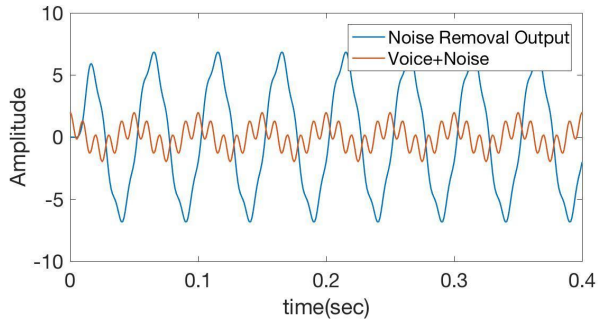
Fig 17 (a). Comparison of the graph y(t) after the use of the Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise stayed the same when noise removal process is executed.

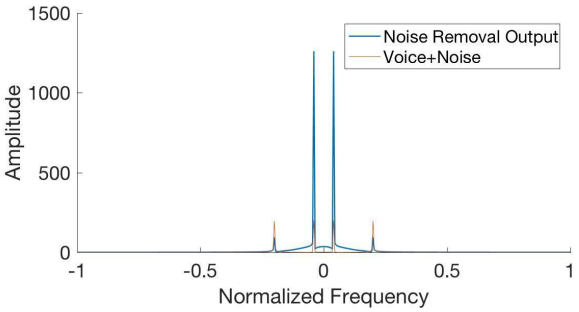




(a) (b)

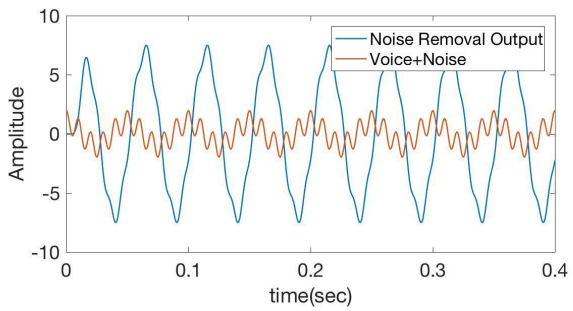
Fig 18 (a). Comparison of the graph y(t) after the use of the FIR filter and Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.

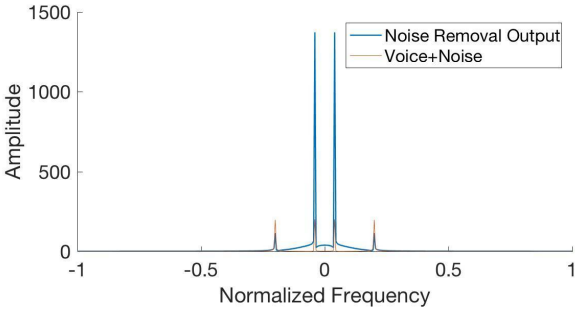




(a) (b)

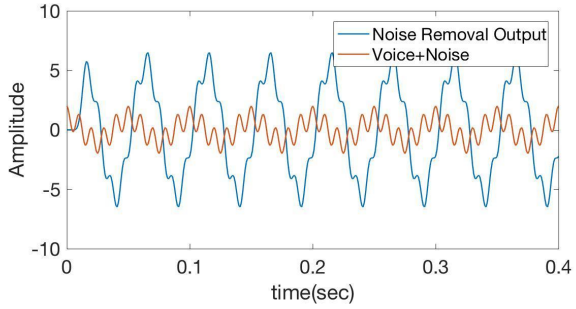
Fig 19 (a). Comparison of the graph y(t) after the use of the FIR filter and Bartlett window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.

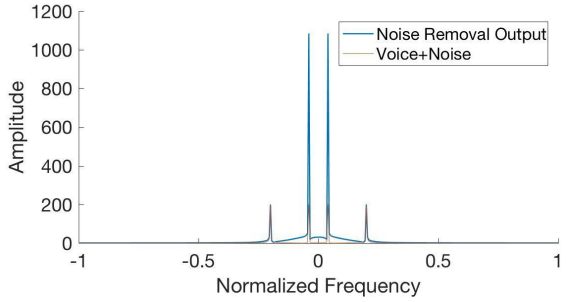




(a) (b)

Fig 20 (a). Comparison of the graph y(t) after the use of the FIR filter and Hamming window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.





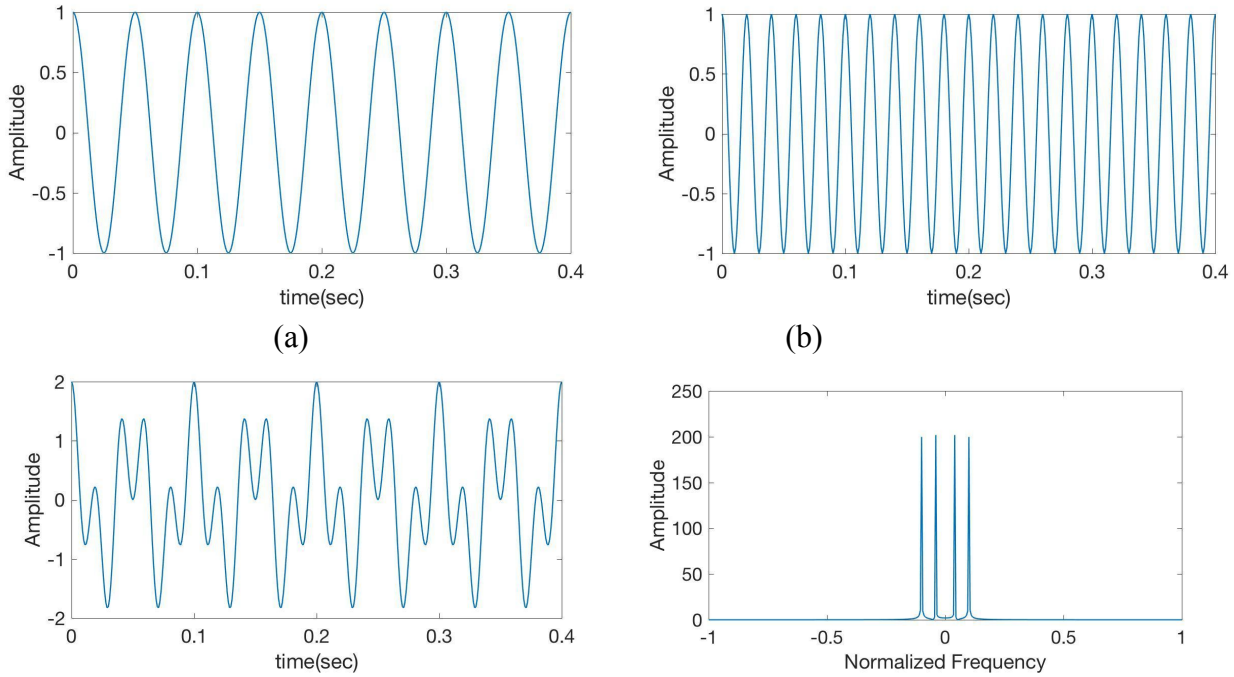
(a) (b)

Fig 21 (a). Comparison of the graph y(t) after the use of the FIR filter and Blackman window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise stayed the same when noise removal process is executed.

Sample 4.

N = 400;
n = [0:N];
fm = 20;
fn = 50;
T = 0.001;

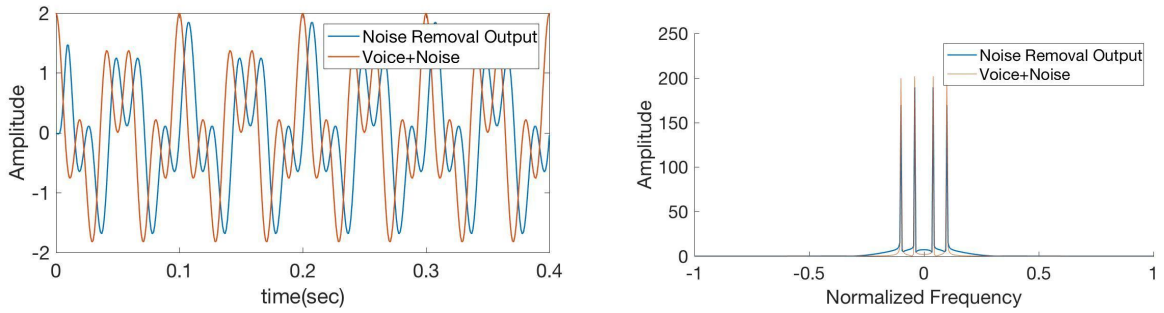
Voice: sin(2*pi*fm*n*T)
Noise: sin(2*pi*fn*n*T)



(a) (b)

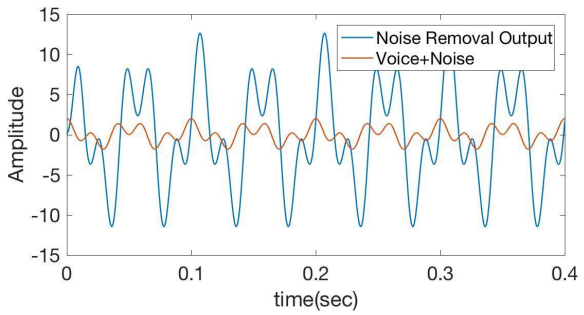
(c) (d)

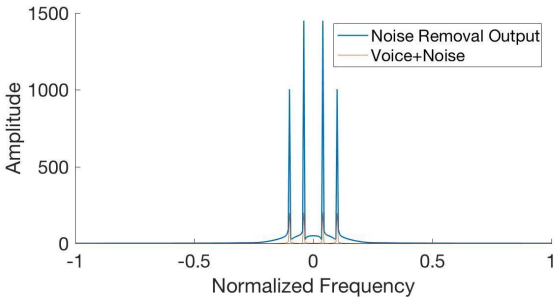
Fig 22 (a). Voice v(t) (b). Noise n(t) (c). Input signal = Voice v(t) + Noise n(t) (d). Fourier transformation of x(t) to find magnitude |FX| in the frequency domain



(a) (b)

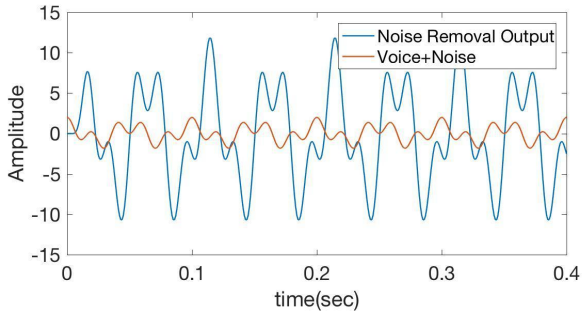
Fig 23 (a). Comparison of the graph y(t) after the use of the FIR filter vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice decreased a little while the amplitude of noise decreased a little more when noise removal process is executed.

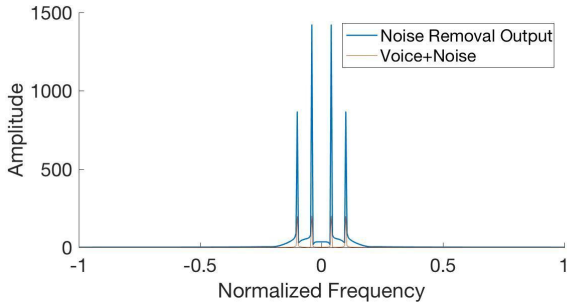




(a) (b)

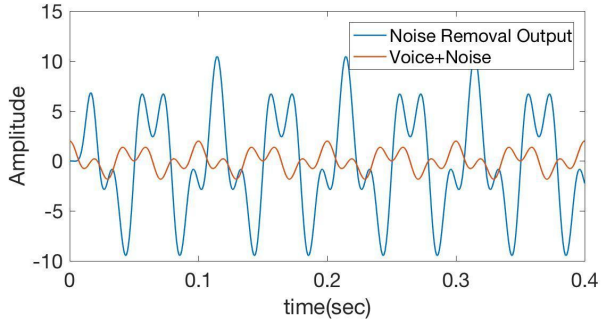
Fig 24 (a). Comparison of the graph y(t) after the use of the Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased more, compared to the amplitude of noise.

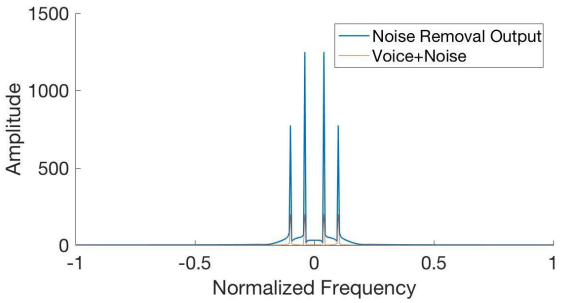




(a) (b)

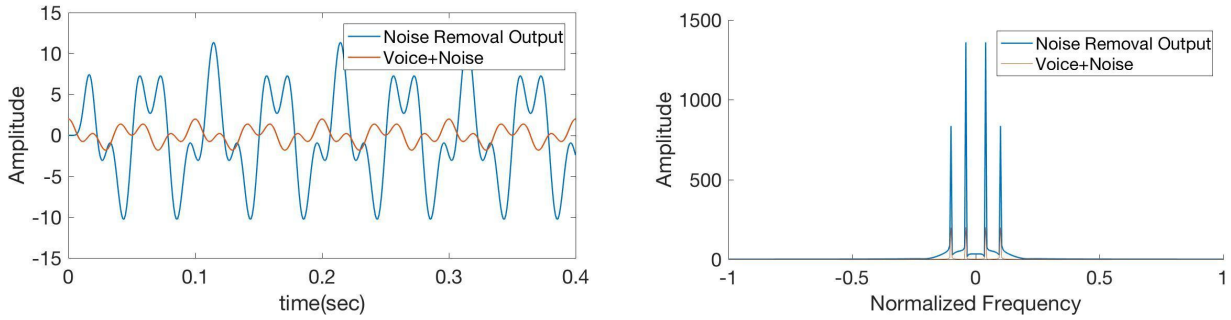
Fig 25 (a). Comparison of the graph y(t) after the use of the FIR filter and Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased more, compared to the amplitude of noise.





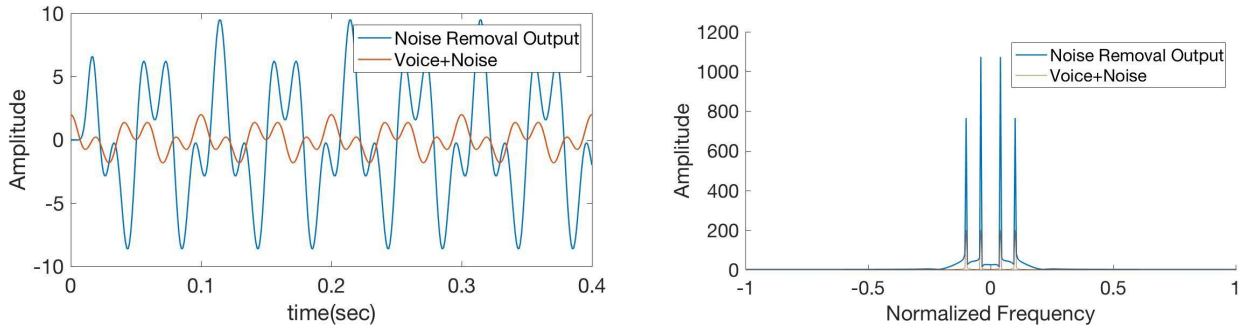
(a) (b)

Fig 26 (a). Comparison of the graph y(t) after the use of the FIR filter and Bartlett window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased more, compared to the amplitude of noise.



(a) (b)

Fig 27 (a). Comparison of the graph y(t) after the use of the FIR filter and Hamming window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased more, compared to the amplitude of noise.



(a) (b)

Fig 28 (a). Comparison of the graph y(t) after the use of the FIR filter and Blackman window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased more, compared to the amplitude of noise.

Sample 5.

N = 400;
n = [0:N];
fm = 20;
fn = 200;
T = 0.001;

Voice: sin(2*pi*fm*n*T)
Noise: sin(2*pi*fn*n*T)

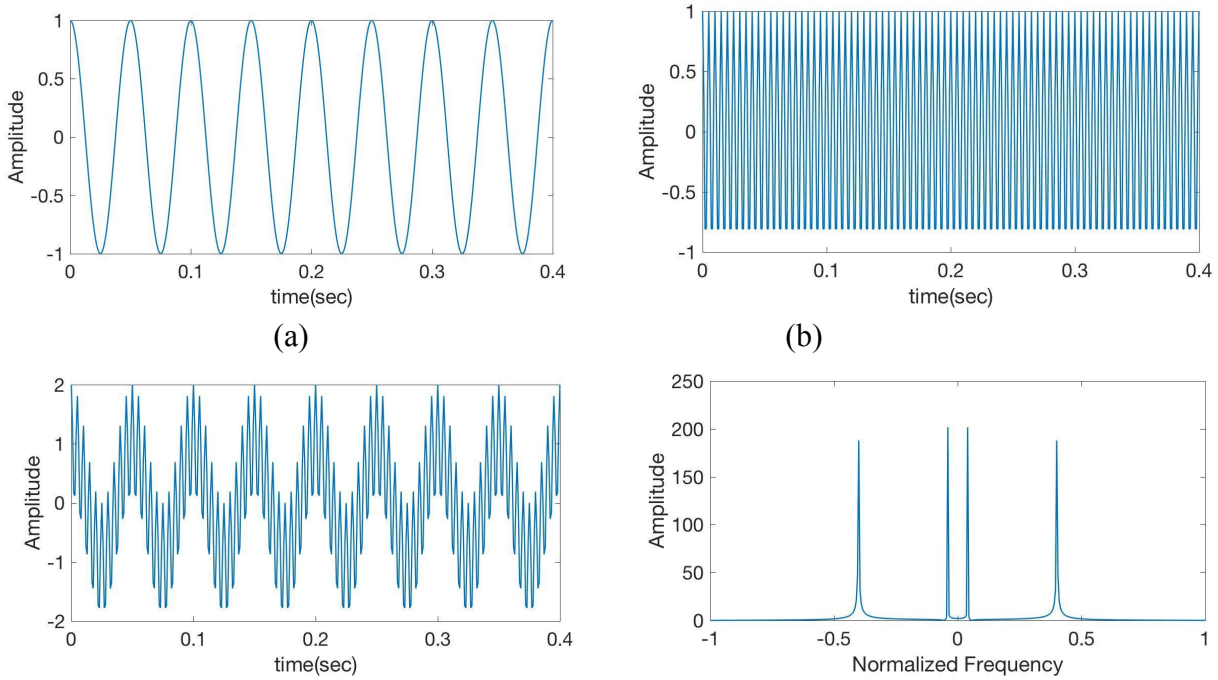


Fig 29 (a). Voice v(t) (b). Noise n(t) (c). Input signal = Voice v(t) + Noise n(t) (d). Fourier transformation of x(t) to find magnitude |FX| in the frequency domain

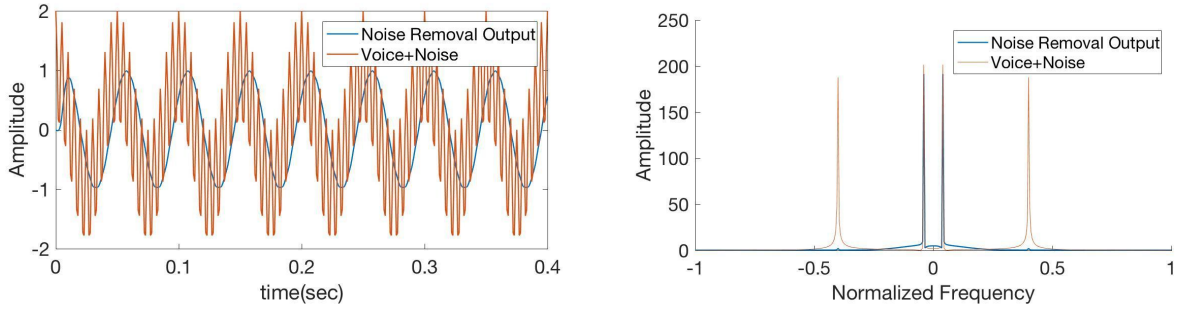
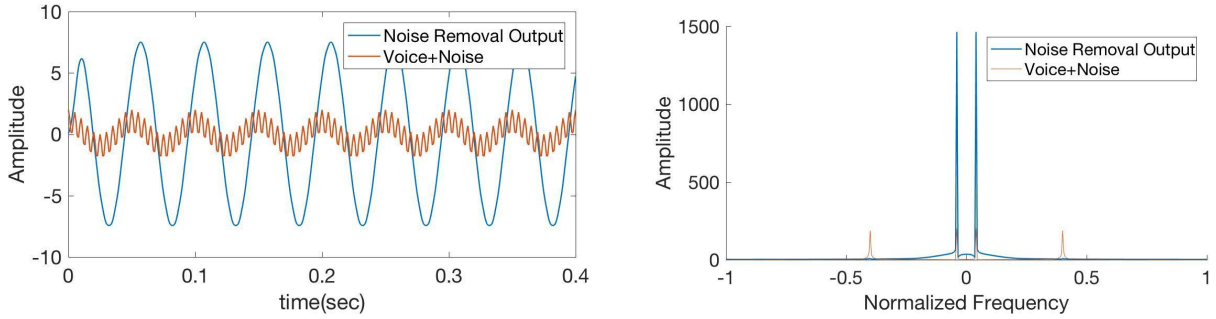
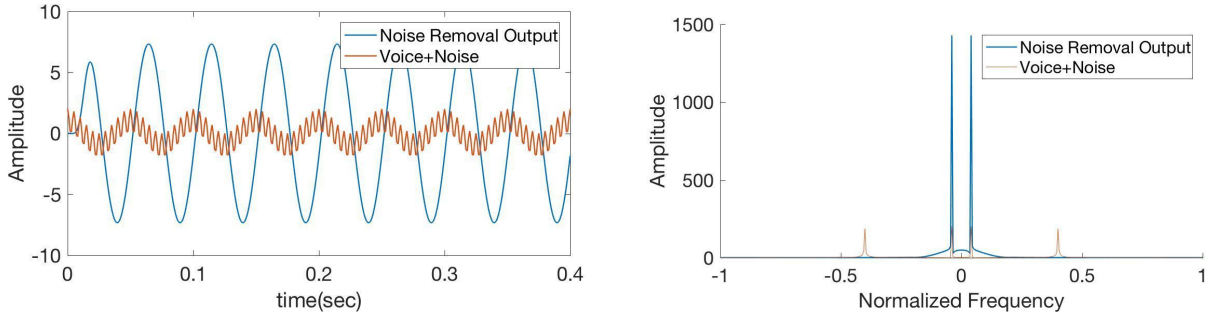


Fig 30 (a). Comparison of the graph y(t) after the use of the FIR filter vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice decreased a little while the amplitude of noise notably decreased when noise removal process is executed.



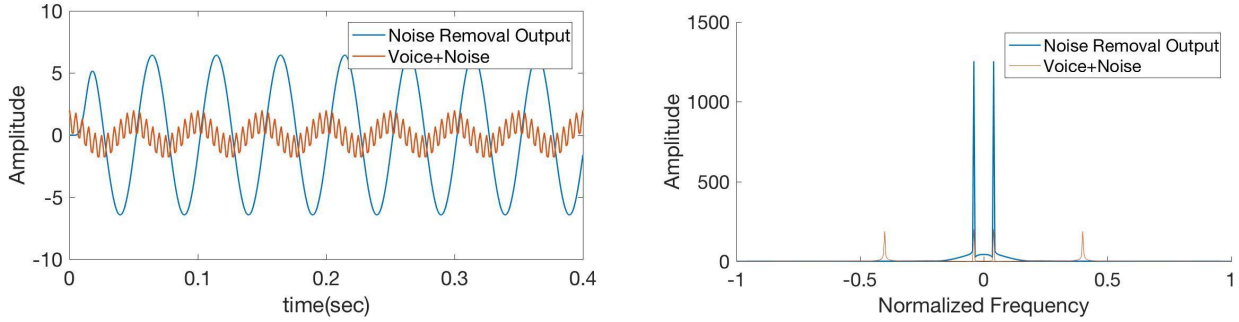
(a) (b)

Fig 31 (a). Comparison of the graph y(t) after the use of the Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise decreased drastically when noise removal process is executed.



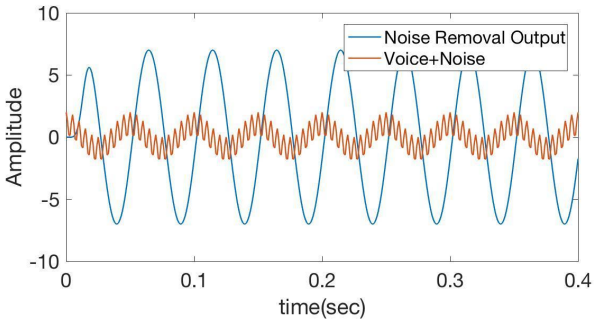
(a) (b)

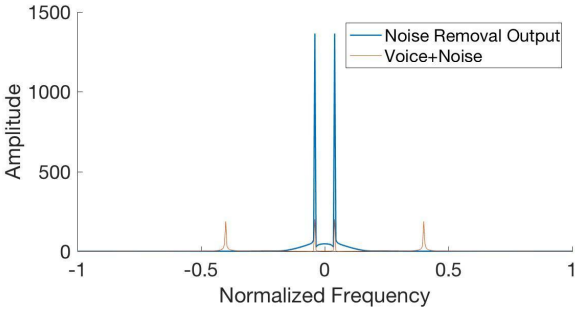
Fig 32 (a). Comparison of the graph y(t) after the use of the FIR filter and Hanning window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



(a) (b)

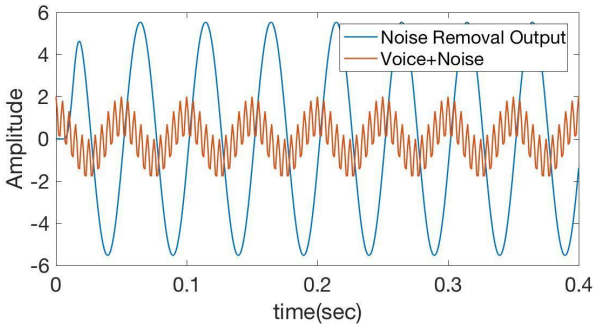
Fig 33 (a). Comparison of the graph y(t) after the use of the FIR filter and Bartlett window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.

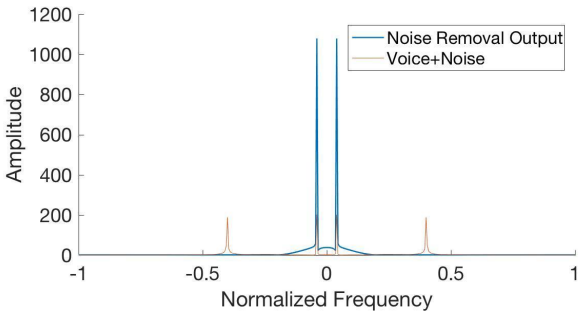




(a) (b)

Fig 34 (a). Comparison of the graph y(t) after the use of the FIR filter and Hamming window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.



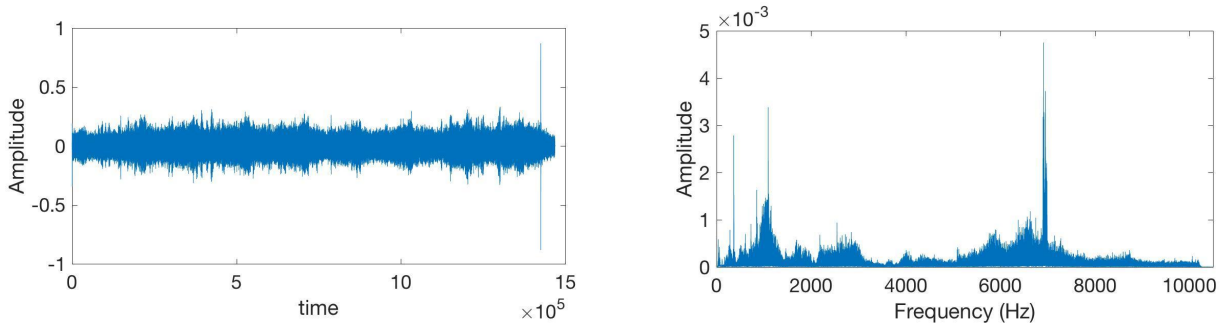


(a) (b)

Fig 35 (a). Comparison of the graph y(t) after the use of the FIR filter and Blackman window vs the graph of the input signal. (b). Comparison of the Fourier transform of noise removal output vs input signal. The amplitude of voice increased while the amplitude of noise notably decreased when noise removal process is executed.

Noise Removal Using Realistic Samples (*All graphs are from implementing MatLab)

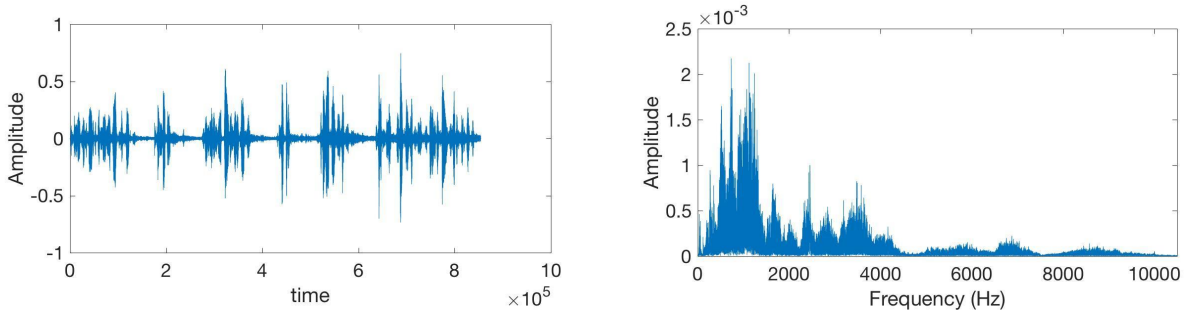
Realistic Sample



(a) (b)

Fig 36 (a). Input signal = Voice v(t) + Noise n(t) (b). Fourier transformation of input signal to find magnitude in the frequency domain

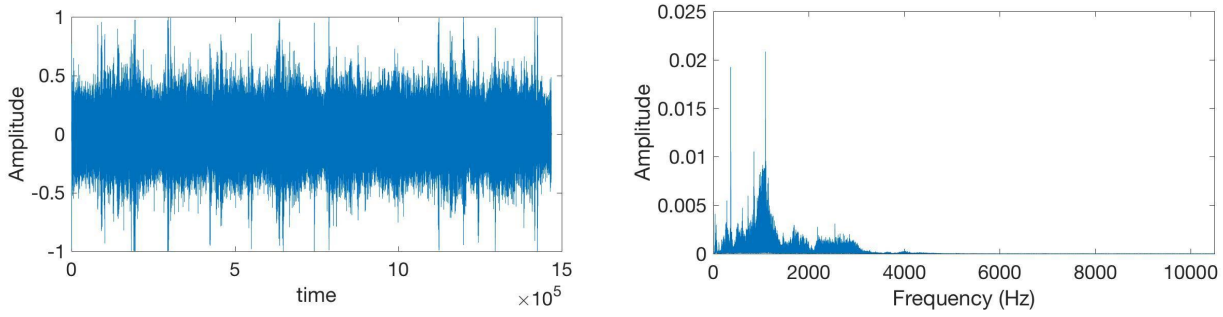
Voice Noise



(a) (b)

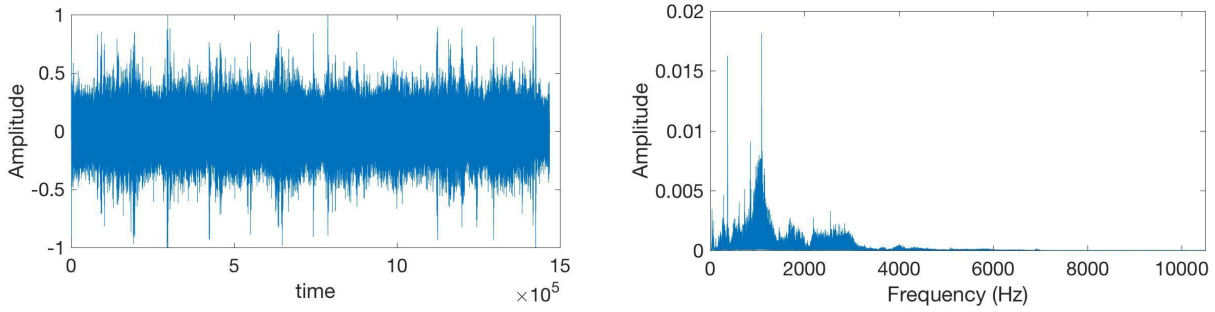
Fig 37 (a). Voice v(t) (b). Fourier transformation of voice v(t) to find magnitude in the frequency domain.

Only Voice



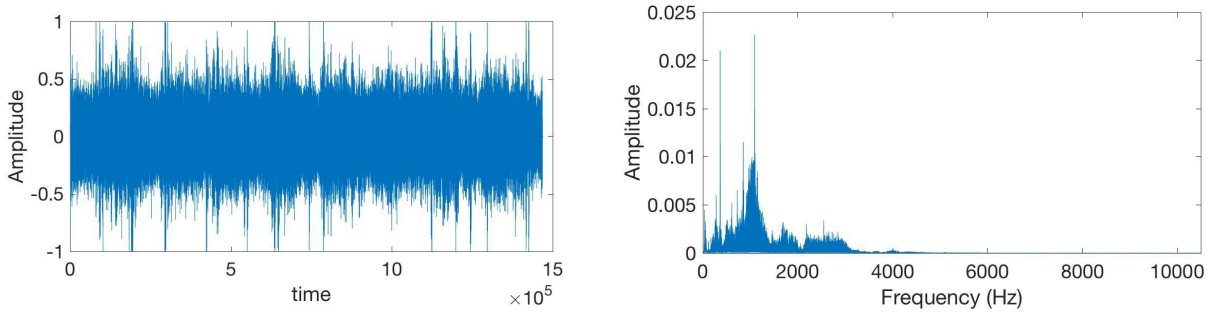
(a) (b)

Fig 38 (a). Noise removal output after the use of the FIR filter and Bartlett window. (b). Fourier transform of noise removal output to find magnitude in the frequency domain.



(a) (b)

Fig 39 (a). Noise removal output after the use of the FIR filter and Blackman window. (b). Fourier transform of noise removal output to find magnitude in the frequency domain.



(a) (b)

Fig 40 (a). Noise removal output after the use of the FIR filter and Hamming window. (b). Fourier transform of noise removal output to find magnitude in the frequency domain.

B. Denoising of Bioimaging Using LPF (*Images are produced from student by using Paintbrush)

Magnetic Resonance Image is one of the most widely used technologies to detect, diagnose, and study various diseases. The images produced by MRI are accurate, and clear, however, there exist some drawbacks to the technology. In order for MRI to produce clear, and rich representations of the area imaged, it takes a long production time. Time consumption is mainly caused by MRI's use of every data in spatial frequency.

To get the image from MRI, frequency has to be transferred to the image using mathematical and computational transformations. An ample amount of the frequency data are obtained from an MRI process; however, all the frequency information is not needed to determine the final image.

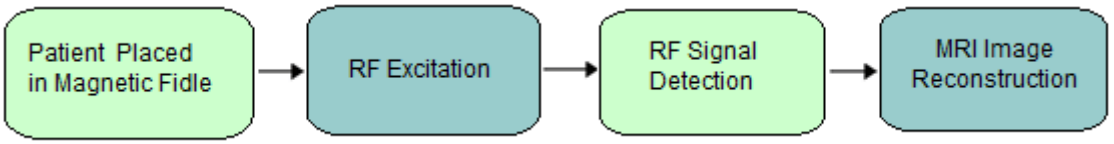


Figure 1. The process of the transformation from frequency data to image domain

Often, the process of transformation from the frequency domain to image domain requires time because Inverse Fourier Transformation takes every frequency point to determine the final output image. However, if a proper function is multiplied to K-space, it results in reduced domains of frequency, which will be used to determine output images (Figure 2).

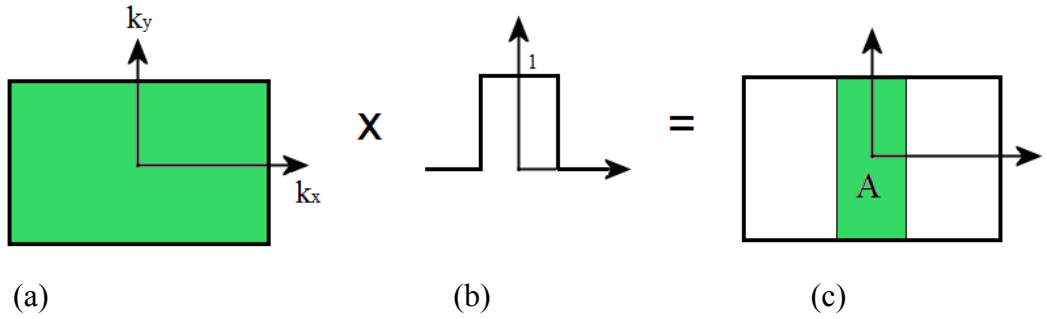


Figure 2. Multiplying proper function(b) to K-space results in a reduction of frequency domain(a)

The purpose of the present research is to develop a more efficient low pass filter or filter function, in order to increase the resolution of the image and, at the same time, decrease the time required to produce the image.

The task of MRI is to acquire a K-space image, then transform it into a spatial-domain image. K-space is a matrix, which is composed of M rows and N columns, where kx is interpreted in real-time to give N samples and ky is adjusted to M samples. This conversion of axes allows K-space in N-M matrix to be converted into kx vs. ky plane, which can be used to create an image in an x vs. y plane. Therefore, it is valid to say that an MRI image is the magnitude of the Fourier transform of the K-space image.

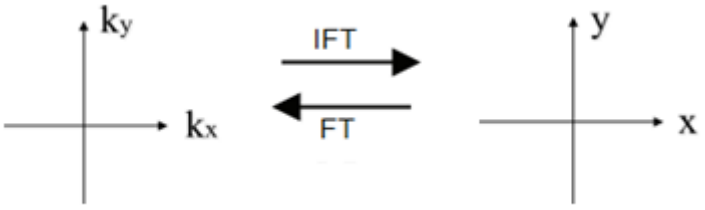


Figure 3. K-space is transformed into image space through Inverse Fourier Transformation, and vice versa

Magnetic Resonance Image Reconstruction from Raw Data

In order to produce the image domain from MRI, there is a complex computational process that requires an intensive analysis. In Part A of this paper, K-space was constructed from an MRI image of a human brain using MATLAB software. Different proposed filters were applied on the full K-space in order to find the most efficient filter, which can be used to produce the best MRI image. In Part B of this paper, a sample brain image is constructed from a K-space provided from raw data from a patient. In this experiment, 12 different K spaces were obtained from 12 corresponding coils. Multiplying the best filter determined from Part A reduced the 12 K-spaces, and they were transformed to 12 images of a human brain using Inverse Fourier Transform. Finally, a comprehensive brain MRI image was obtained by taking a Root Mean Square of 12 different MRI images. The overall flowchart is shown in Figure 4 and Figure 5.

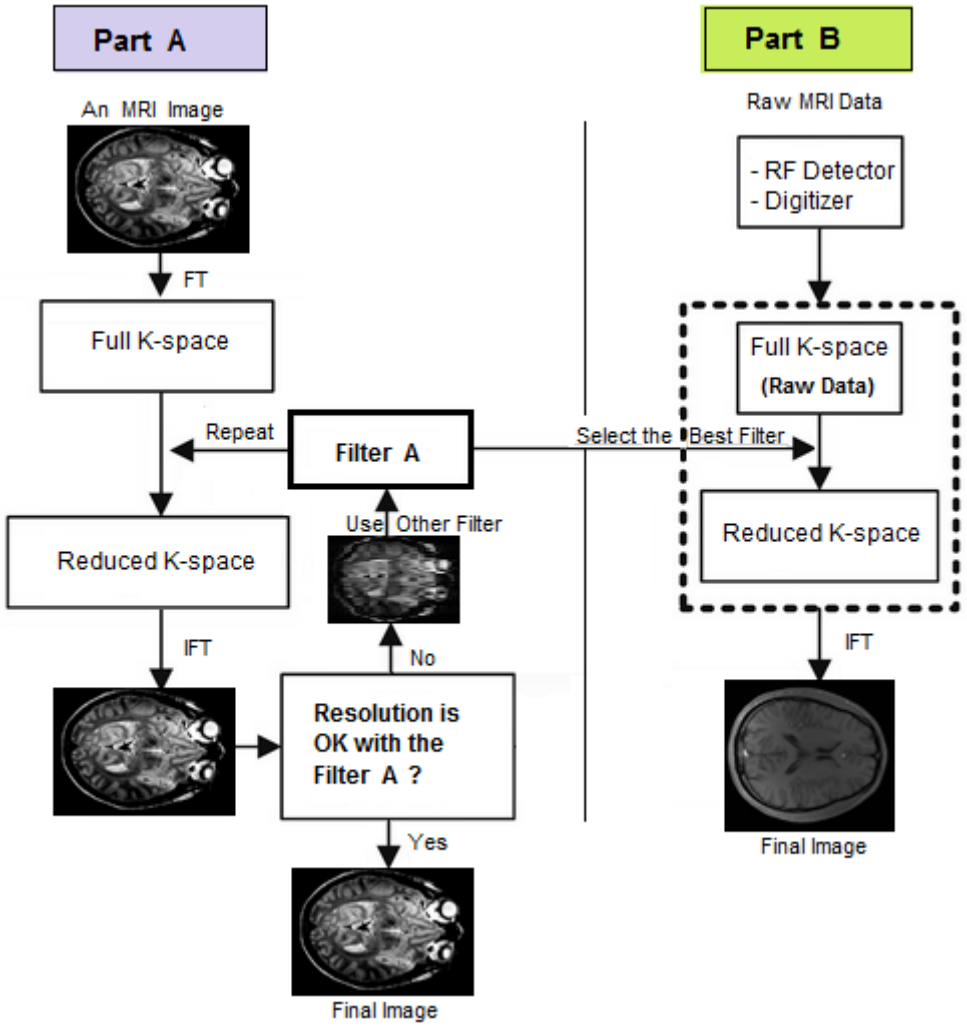


Figure 4. Data before and after the Fourier Transformation

*MRI images are from MRI Research Lab in UIUC and others are drawn by Paintbrush

This paper presents the selectivity of proper K-space as shown in Figure 5 and Figure 6 by removing different amounts of high or low frequencies to create the most optimal images.

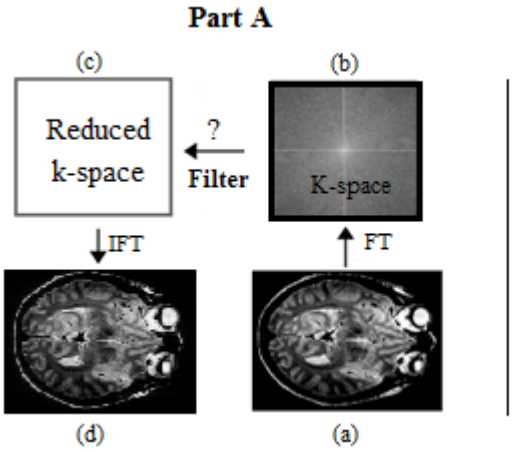


Figure 5. Find an efficient filter

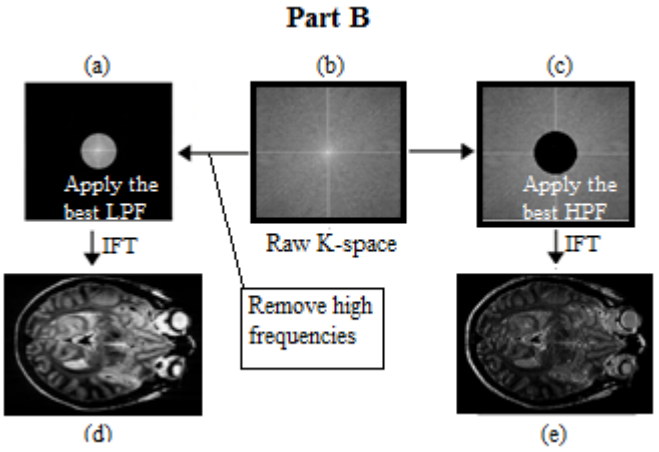


Figure 6. Test the filters obtained from Part A

*Images (A)-a, (A)-d, (B)-d, and (B)-e are from MRI Research lab in UIUC) and student using MatLab. Others(images on top) are from Google.

Filter Design (*Images are produced from student - by using Paintbrush)

Changing different variables in low pass filters can change the function produced over the image domain. Originally, square functions are used during Fourier Transformations. However, various filters were tested in creating images to find an efficient and proper filter.

- **Rectangular Functions Applied on MxN K-space as LPF**

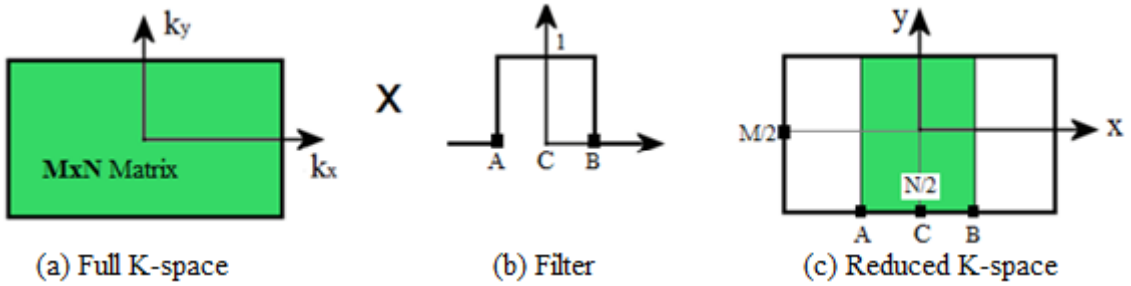


Figure 7. Multiplication of K-space and square function causes a reduction in the frequency domain

(Tested Rectangular function in the MATLAB: LPF(N/2-N/10:N/2+N/10)=1, AC=BC= N/10**)**

- **Gaussian functions as LPF**, Y = exp(-((l-L/2).^2)/10^n)
- **Circle equations as a filter**, r = sqrt((x-M/2)^2+(y-N/2)^2)
- **Proposed even function as a filter**, 1-abs((x/(width/2)))^n

Results (*Original MRI was obtained from MRI Research lab in UIUC and graphs were obtained by using Matlab)

Part A. K-space reconstruction using MRI image (nonconventional method)

In Part A of this paper, K-space was constructed from an MRI image of a human brain shown in Figure 8 using MATLAB software. Different proposed filters were applied on the full K-space in order to find the most efficient filter, which can be used to produce the best possible MRI image.

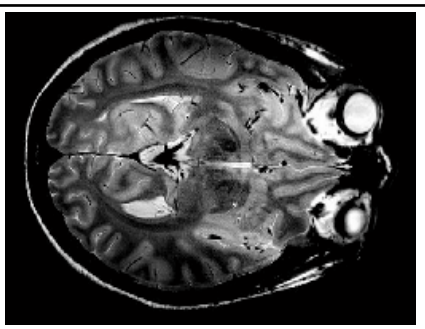
Original MRI image (Full K-space is used / No filter is used)	

Figure 8. Original MRI image

Square function as LPF (*Original MRI was obtained from MRI Research lab in UIUC and graphs were obtained by using Matlab)

Two different shapes of rectangular functions produced different MRI images as follows:

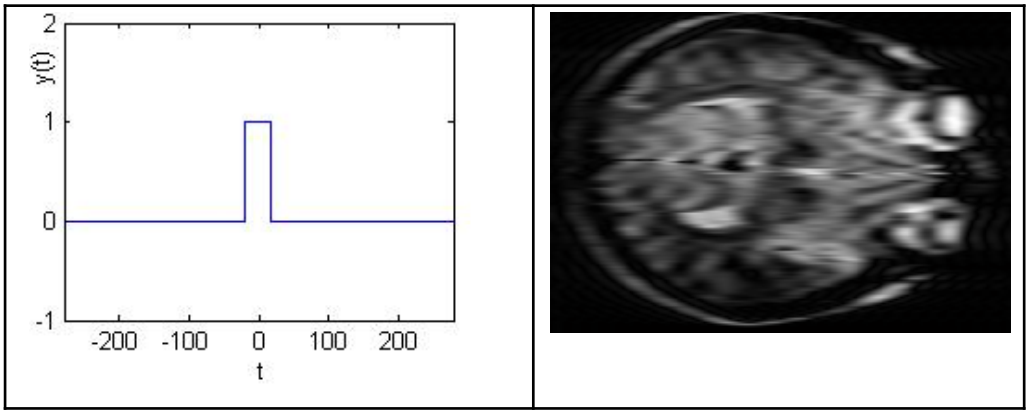


Figure 9. Narrow ({x|-15<x<15}) square function filter and its image

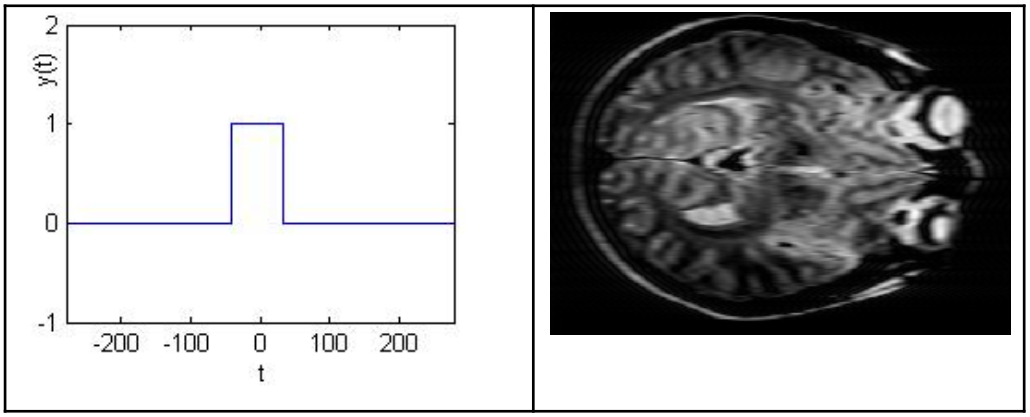


Figure 10. Narrow ({x|-30<x<30}) square function filter and its image

Figure 9 and Figure 10 showed that when a square function was used to construct the final image, it caused the image to be blurry due to the ringing effect. A narrow square function as the low pass filter caused the resolution of the image to decrease because the narrow low pass filter did not collect enough information.

Gaussian function as LPF (*Original MRI was obtained from MRI Research lab in UIUC and graphs were obtained by using Matlab)

Figure 11 is of using a narrow exponential function, or Gaussian function as a filter function.

`y = exp(-((l-L/2).^2)/a^2), where L=2*7*N/40, N=557`

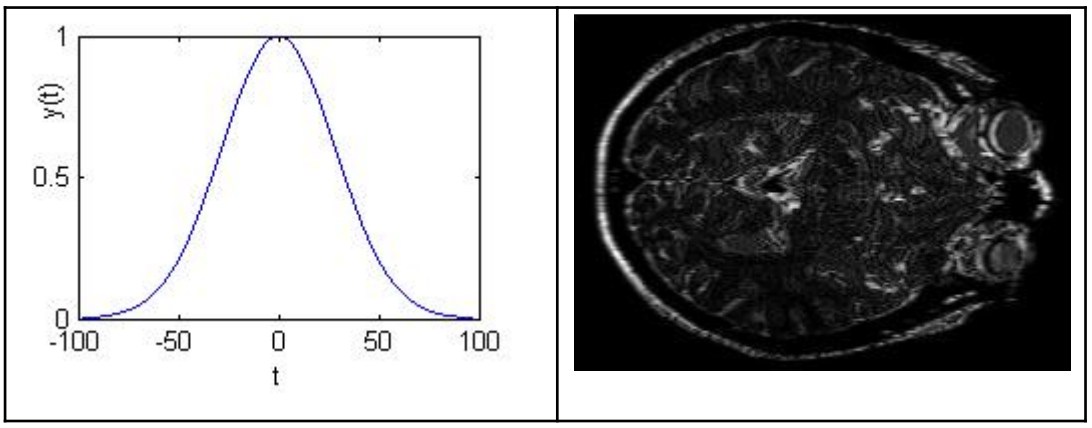


Figure 11. Gaussian function (a=40) filter and its image

The final image on the right showed that a narrow exponential function was able to collect only minimal amount of information, causing the resulting image to be deprived of the original details. Multiplying the exponential function to K-space decreased the ringing effect, but the image lacked in clarity due to a lower resolution.

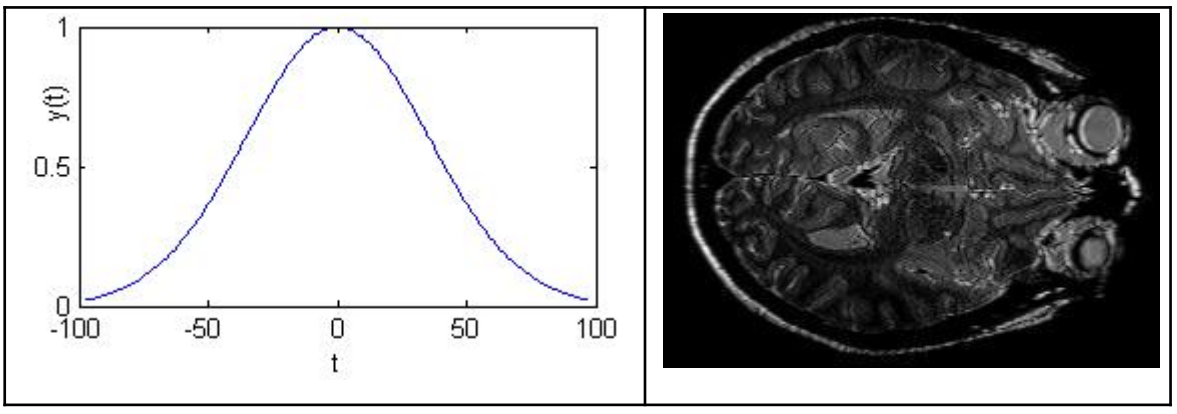


Figure 12. Gaussian function (a=50) filter and its image

A wider Gaussian function was used in Figure 12. The wider width over the frequency domain included more high-frequency points, thus, the image had more details, including clearer edges, compared to the image in Figure 11.

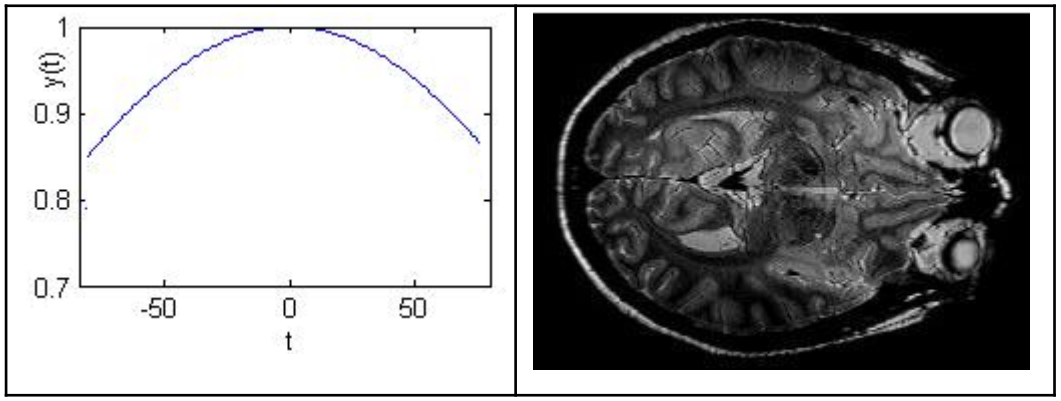


Figure 13. Gaussian function (a=60) filter and its image

In Figure 13, the domain increased even more than that of Figure 12, thus, more information was collected, which was used in producing a more detailed image. In this image, there was almost no ringing effect; the image showed clear edges and other important details.

Proposed LPFs (*All graphs were produced from MatLab)

In this section, different types of graphs that can be potentially used as LPF were considered. Powers with the fractions with an even numerator and an odd denominator are continuous throughout all four quadrants. The equation is:

$$x^n + y^n = 1$$

When the power n is less than 1, the graph caves inwards. When the power n is greater than 1, the graph curves outward, and finally, it becomes a rectangular-shaped graph as n becomes infinite.

$$x^{\frac{2}{5}} + y^{\frac{2}{5}} = 1 \quad (1)$$

$$x^{\frac{4}{5}} + y^{\frac{4}{5}} = 1 \quad (2)$$

$$x^{\frac{8}{5}} + y^{\frac{8}{5}} = 1 \quad (3)$$

$$x^{\frac{16}{5}} + y^{\frac{16}{5}} = 1 \quad (4)$$

$$x^{\frac{104}{5}} + y^{\frac{104}{5}} = 1 \quad (5)$$

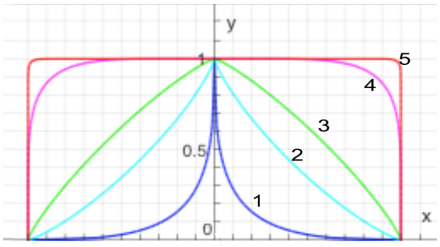


Fig. 10. Proposed filter functions

A trend or pattern was found: Our computational simulation showed that if the graph becomes sharp or pointed like 1, 2, or 3 in Fig. 14, the RA(Ringing Artifact) gets bigger so it does not seem desirable to use them as LPFs. The equations (1), (2), (3) produce a lower quality of bio-images, whereas the equation (4) produced the best quality of bio-images, so it will be used as the ideal LPF among the five equations above.

If the graph becomes more rounded, it gets more information from the k-space and shows less of the ringing artifact in bio-images. Usually, when the domain of the function becomes wider, the resolution of the image produced becomes better, but the computer running time for the image process increases.

Appendix (*The matrix is obtained using MatLab)

k space

This is a portion of the sample matrix, taken 262~275 columns and 335~348 rows from the whole 365x557 matrix. The Gaussian filter with p=40 was used to create the matrix in the image domain. This sub-matrix is produced by sampling at the lower midpoints of the image domain, 365x557 matrix.(and then it can be normalized using a technique). Note that the bottom left elements at around [9, 3] have the large values, which means bright in MRI image, and decreasing gradually as distance from that point increases to the bottom right.

0	0	0	2	0	0	0	0	0	0	0	0	0	0
2	3	0	0	0	1	0	0	0	0	2	3	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	7	1	0	1	5	1	2	0	0	0	0
1	2	1	0	0	1	1	2	0	0	1	3	0	0
5	3	0	0	0	0	0	0	5	0	1	3	2	0
0	11	22	38	41	37	37	44	55	12	0	1	1	0
68	98	143	152	128	113	116	136	114	41	5	0	0	26
176	134	151	119	80	80	63	80	138	107	48	19	48	84
160	116	61	30	22	52	14	10	73	132	56	34	87	50
130	80	13	0	0	22	0	0	3	37	14	5	28	9
33	14	2	2	8	5	0	2	8	0	3	6	0	6
0	0	3	0	8	1	0	0	4	0	2	2	0	2
5	4	1	0	0	0	4	1	0	6	1	0	7	3

Discussions and Conclusions

A. Acoustics: Denoising Sound Algorithm

A combination of a low pass filter and windows effectively removed the noise from the audio files. Low pass filters reduced the amplitude of noise, while the windows enhanced the amplitude of pure signals. The combination of the filter and window amplifies the pure sound and diminishes the noise, emphasizing the pure signal. The effectiveness of noise removal depends on the frequency of the original sound file and the noise. As the difference between noise and sound is greater, the noise reductions become more effective, even filtering completely inaudible sound files into intelligible signals. The combination of filter and window does reduce the noise of the sound file when the difference between noise and sound is insignificant, but it is less efficient.

Because this study focused on the higher frequency noise, removing lower frequency noise should be determined. While the windows examined in this research, Hanning, Bartlett, Hamming, and Blackman window, were efficient to an equivalent extent, the difference in each of their characteristics in the noise removal process should be determined thoroughly.

B. Denoising the Bioimage

Many filters are good at reducing the size of K-space, but most of the filters had the ringing effect, which blurred the sharp "edges" of the images as a side effect. In this research, new experiments were carried out with several modified filters to reduce the ringing effect and improve the resolution of an MRI image to a degree. The data produced finally proposed an efficient function as a new filter.

1. The proposed filter is different from the rectangular (square) function, Gaussian function, and circle function, but the trial and error method was done on the new filter to confirm that it encompasses all the advantages or properties of the 3 functions.

2. The best efficiency occurred when the exponent n in the proposed filter equation is in between 2 and 10, but a new algorithm is needed to find the exact number since numbers can exist in other forms other than integers.

3. A vast raw data that was collected from a patient was tested to create 12 sub-images using the proposed filter. The final image was constructed successfully using the coordinate transformations and the least square method. As other methods are also possible to visualize the image, it is desirable to compare the methods to each other.

Reference

1. Aggarwal, Rajeev, et al. "Noise reduction of speech signal using wavelet transform with a modified universal threshold." *International Journal of Computer Applications* 20.5 (2011): 14-19.

2. Kumar, T. Lalith, and K. Soundara Rajan. "Noise Suppression in speech signals using Adaptive algorithms." (2012).

3. Boll, Steven. "Suppression of acoustic noise in speech using spectral subtraction." *IEEE Transactions on acoustics, speech, and signal processing*27.2 (1979): 113-120.

4. Hymavathy, K. P., and P. Janardhanan. "Noise Filtering in Speech Using Frequency Response Masking Technique." *International Journal of Emerging Trends in Engineering and Development* 2.3 (2013): 208-213.

5. Muangjaroen, Supavit, and Thaweesak Yingthawornsuk. "A Study of Noise Reduction in Speech Signal Using FIR Filtering." *Proceedings of the International Conference on Advances in Electrical and Electronics Engineering*. 2012.

6. Karam, Marc, et al. "Noise removal in speech processing using spectral subtraction." *Journal of Signal and Information Processing* 5.02 (2014): 32.

7. Verteletskaya, Ekaterina, and Boris Simak. "Speech distortion minimized noise reduction algorithm." *Proceedings of the World Congress on Engineering and Computer Science*. Vol. 1. 2010.

8. Rabiner, Lawrence R., and Ronald W. Schafer. *Digital processing of speech signals*. Prentice Hall, 1978.

9. Allen, Jonathan. "Short term spectral analysis, synthesis, and modification by discrete Fourier transform." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25.3 (1977): 235-238.

10. Vijaykumar, V. R., P. T. Vanathi, and P. Kanagasapabathy. "Modified adaptive filtering algorithm for noise cancellation in speech signals." *Elektronika ir elektrotechnika* 74.2 (2007): 17-20.

11. Lim, Yong Ching, and Yong Lian. "Frequency-response masking approach for digital filter design: Complexity reduction via masking filter factorization." *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*41.8 (1994): 518-525.

12. Koike, Shin'Ichi. "Convergence analysis of adaptive filters using a normalized sign-sign algorithm." *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 88.11 (2005): 3218-3224.

13. Saric, Matko, Luki Bilicic, and Hrvoje Dujmic. "White noise reduction of audio signal using wavelets transform with the modified universal threshold." *University of Split, R. Boskovica b. b HR* 21000 (2005).