Department of Electronics and Communications Engineering,

Institute of Technology,

Nirma University, Ahmedabad.

BASIS PURSUIT DENOISING

Special Assignment Report

Submitted for the partial fulfilment of the

requirements for completion of the course on

2EC502 DIGITAL SIGNAL PROCESSING

Submitted To:

Dr. Mehul R Naik (Course Faculty)

Dr. Rachna Sharma (Lab Faculty)

Submitted By:

Arya Tripathi 20BEC130 | Yash Purohit 20BEC137

Semester V

I N D E X

SPARSITY IN SIGNALS :

Signal is considered sparse if most of its information is contained within very few numbers of non-zero samples as compared to its' dimensions. Consequently, a signal reconstruction algorithm has to find a sparse vector that best represents the measured signal after it is received. This can be done using least square solution or sparse approximation.

The method of Least Square Solution :

Given an inconsistent matrix equation **Ax = b**;

*where **A** is a m x n matrix*

***b** is a vector in* $\boldsymbol{R^m}$.

Ax = b is solved as closely as possible such that the sum of the squares of difference ***(b – Ax)*** is minimized. A least square solution of ***Ax = b*** is a vector $\hat{x}$ in R^n such that,

$$\mathbf{dist(b, A\hat{x}) \le dist(b, Ax).}$$

Least square comes from the fact that

$$dist(b, Ax) = || b - A\hat{x} ||$$

is the square root of the sum of the squares of the differences between the entries of $\boldsymbol{A\hat{x}}$ and ***b***.

The method of Sparse Approximation :

Sparse approximation or sparse representation theory deals finds solutions for systems of linear equations. Techniques for finding these solutions and exploiting them are widely used in image processing, signal processing and much more.

For noiseless observations :

$$\min_{\alpha \in \mathbb{R}^p} \|\alpha\|_0 \text{ subject to } x = D\alpha,$$

where $||\boldsymbol{\alpha}||_0 = \# \{ i : \alpha_i \neq 0, i = 1, \ldots p \}$ is the $\boldsymbol{l_0}$ pseudo-norm, which counts the number of non-zero components of **α**

D is an undetermined **m** x **p** (m<p) matrix assumed to be full rank.

For noisy observations :

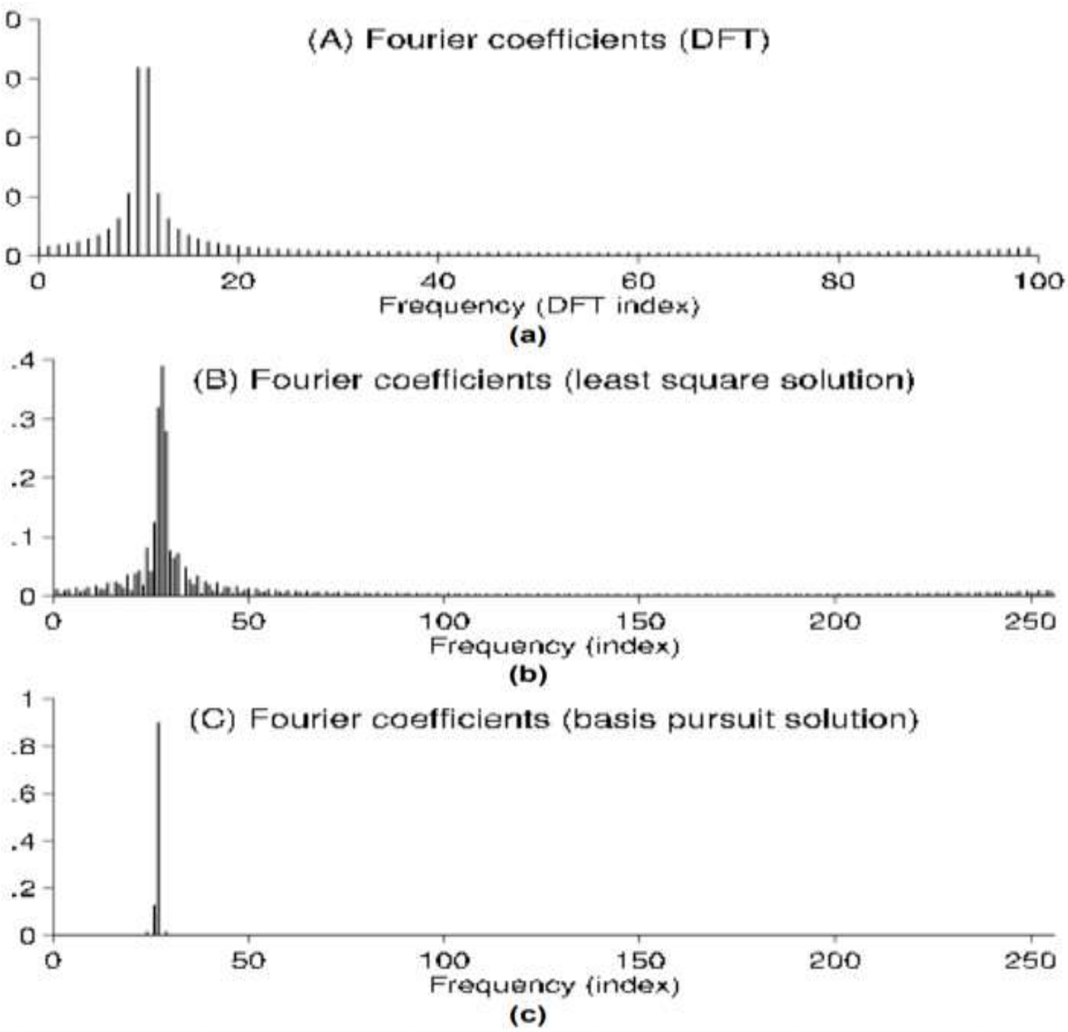
In case of noisy signal, putting in the Lagrangian form :

$$\min_{\alpha \in \mathbb{R}^p} \lambda \|\alpha\|_0 + \frac{1}{2} \|x - D\alpha\|_2^2,$$

The problems are NP-Hard in general, but can be approximated using pursuit algorithms. More specifically, changing the l_0 to an l_1 **- norm**, the following equation is obtained, which represents **Basis Pursuit Denoising.**

$$\min_{\alpha \in \mathbb{R}^p} \lambda \|\alpha\|_1 + \frac{1}{2} \|x - D\alpha\|_2^2,$$

Several algorithms are popular for solving the Basis Pursuit Denoising such as in-crowd algorithm, homotopy continuation, fixed point continuation and spectral projected gradient for L1 minimization. Looking at the comparison results available across different research papers; It is evident that BPD provides better sparse coefficients than the least square method. So, it is having been used and analysed throughout the report.



WHY BPD OVER LTI ?

Digital LTI Filters are commonly used for noise reduction (denoising). If the noise and signal occupy separate frequency bands, and if these frequency bands are known, then an appropriate digital LTI filter can be readily designed so as to remove the noise very effectively.

But when the noise and signal overlap in the frequency domain, or if the respective frequency bands are unknown, then it is more difficult to do noise filtering using LTI filters. However, if it is known that the signal to be estimated has sparse (or relatively sparse) Fourier coefficients, then sparsity methods can be used as an alternative to LTI filters for noise reduction.

The **BPD** solution performs denoising as desired but it is only effective when the coefficients of the signal are sparse or approximately sparse or else the BPD solution will not be effective.

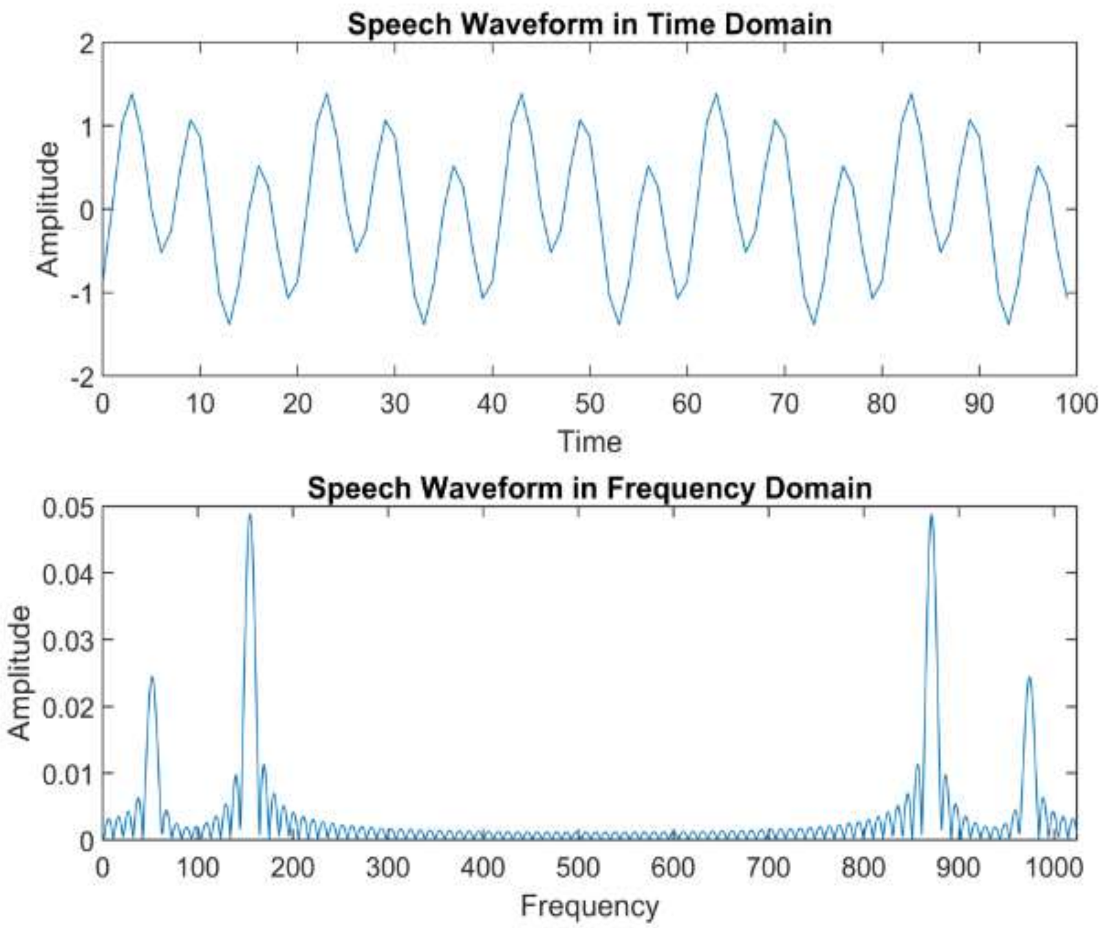
BASIS PURSUIT DENOISING :

Basis Pursuit Denoising is a mathematical optimization technique used extensively in Signal Processing. The below sections emphasize and explain the same using **MATLAB** Software, with the help of two examples, one of a locally generated signal which contains two sinusoidal waves and the other is a speech signal for practical analysis.

Sum of Two Sine Waves :

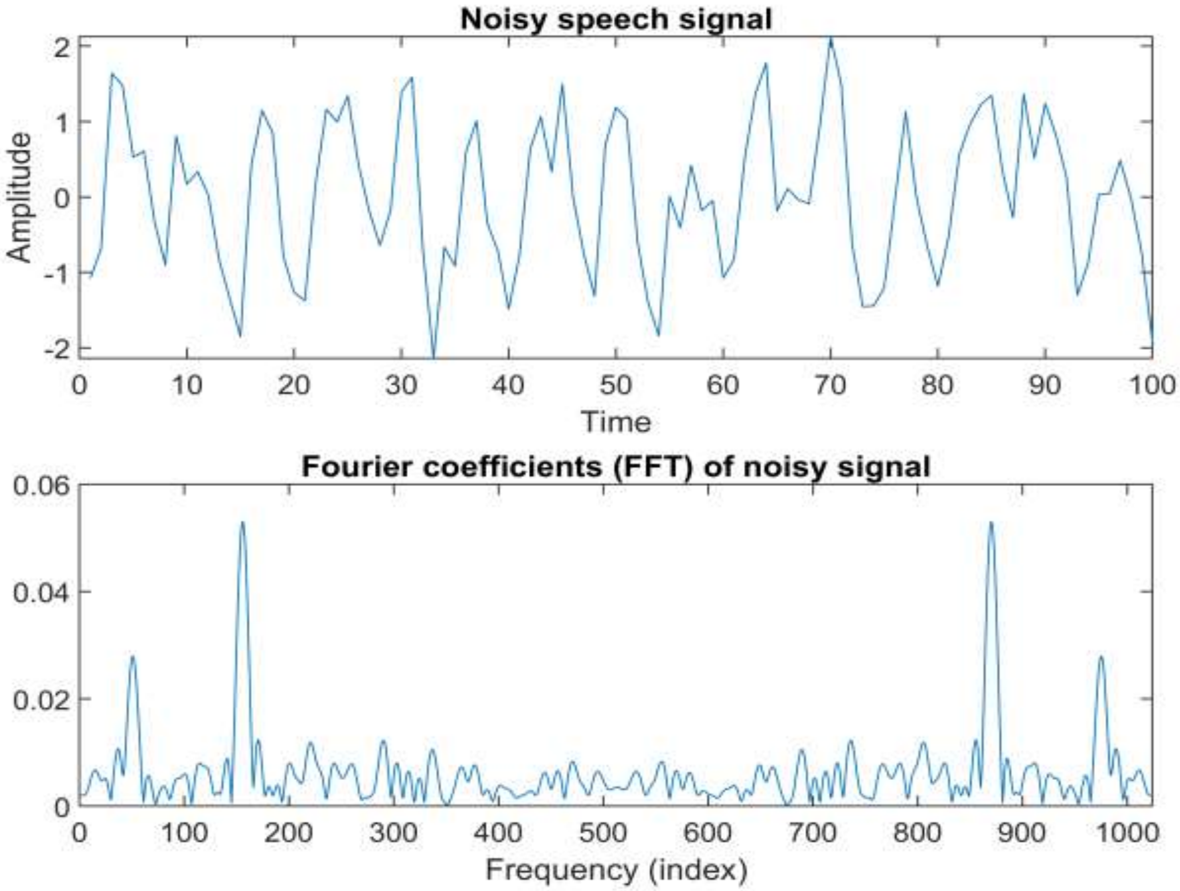
A sum of two sine waves has been considered as the original signal waveform. The frequencies are **0.5 Hz** and the frequency of **0.15 Hz**. Fourier transform of the signal shows the Fourier coefficients of the original noiseless signal. **1024-point DFT** has been used and the plots are obtained using ***plot*** and ***fft*** MATLAB commands.

SIGNAL	***s = 0.5*sin(2*pi*0.05*n) + sin(2*pi*0.15*n-pi/3);***
Time Domain	***plot(n, s);***
Freq Domain	***Y = (1/1024)*fft(s, 1024);***



Gaussian Noise of amplitude 0.5 is now added using the ***randn*** MATLAB command. The noise array side should be same as the signal array size. The noisy signal is plotted in time and frequency domain.

Y(m) = s(m) + w(m); where m = 100 (signal length), s : Original Signal, w - Noise



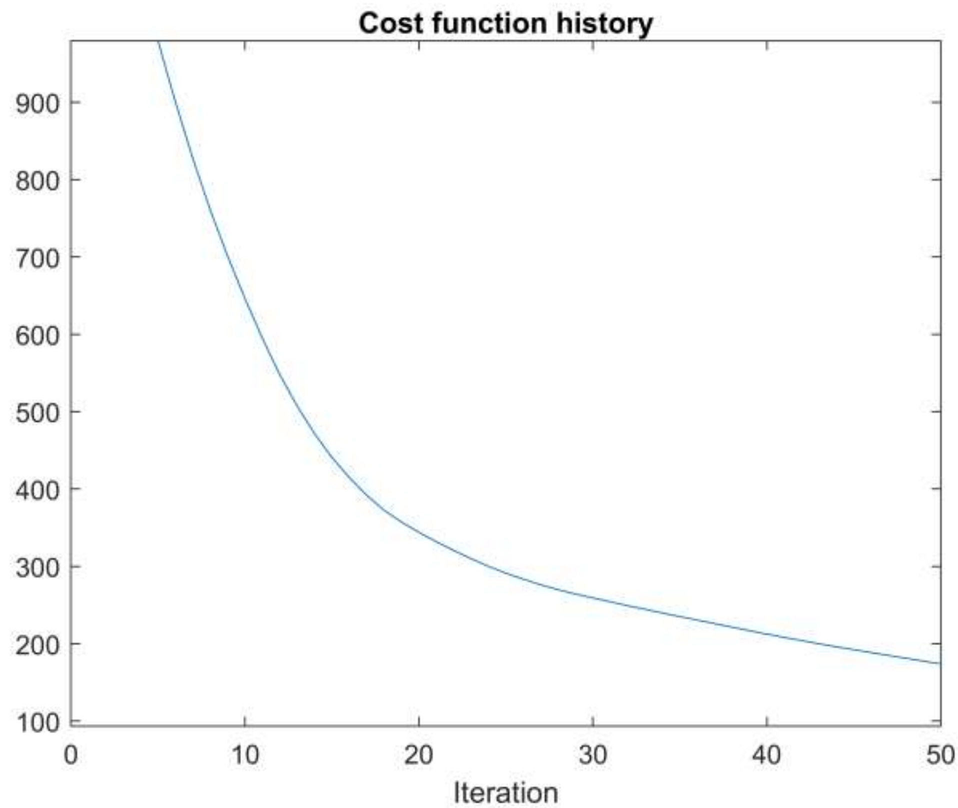
Two functions ***MakeTransform*** and ***BPD*** have been created. Former takes the type of transform, signal length and number of points as it's input argument and returns the function handles of the transform viz. A and AH as output.

MakeTransform	***[A, AH] = Make Transforms('DFT', 100, 2^10);***

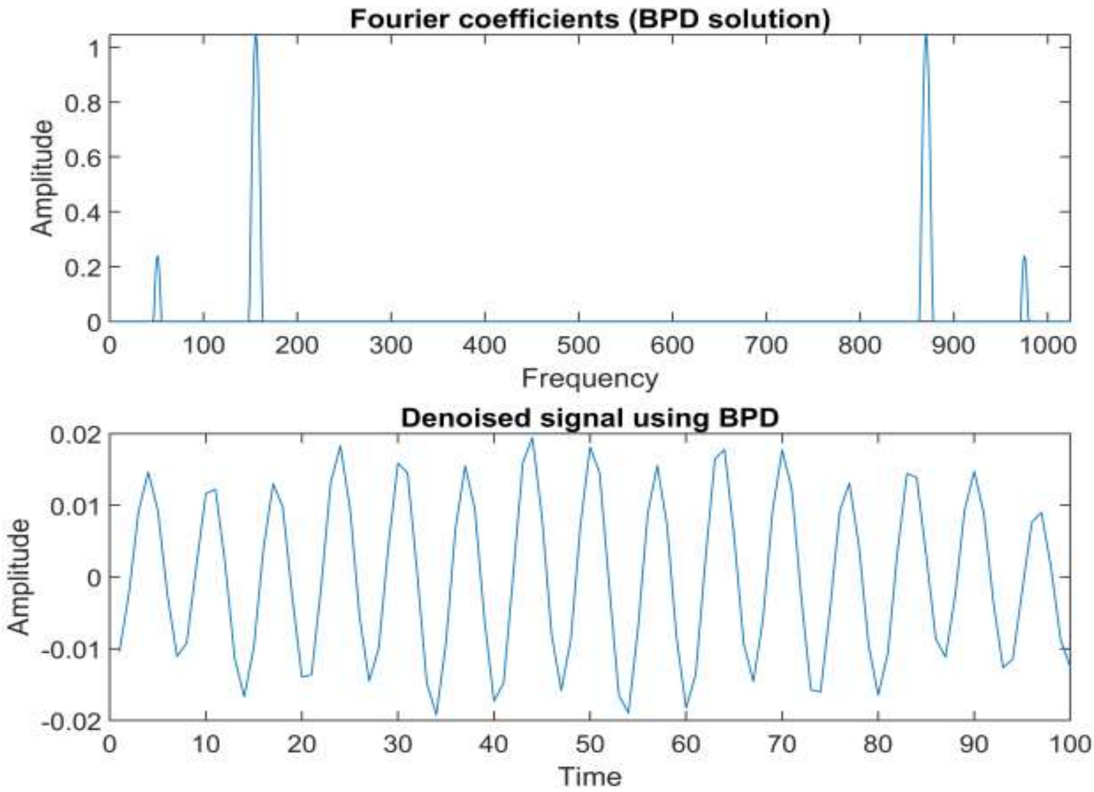
The outputs A and AH are now given as an input argument to the BPD function along with ***Augmented Lagrangian parameter (mu), Nit - Number of iterations*** and the signal Y. The output of BPD algorithm returns the minimizing vector (c) and cost function per iteration (cost).

Regularization Parameter	***lambda = 7;***
Number of Iterations	***Nit = 50;***
ADMM Parameter	***mu = 500;***
BPD	***[c, cost] = BPD(y, A, AH, lambda, mu, Nit);***

To obtain the solution to the basis pursuit problem Equation, 100 iterations of ***SALSA iterative optimization algorithm*** have been used here. It is informative to look at the cost function history of an iterative algorithm in order to evaluate it’s convergence behaviour.



The matrix c contains the denoised signal, and it can be clearly observed that the effects of noise have been effectively nullified here.



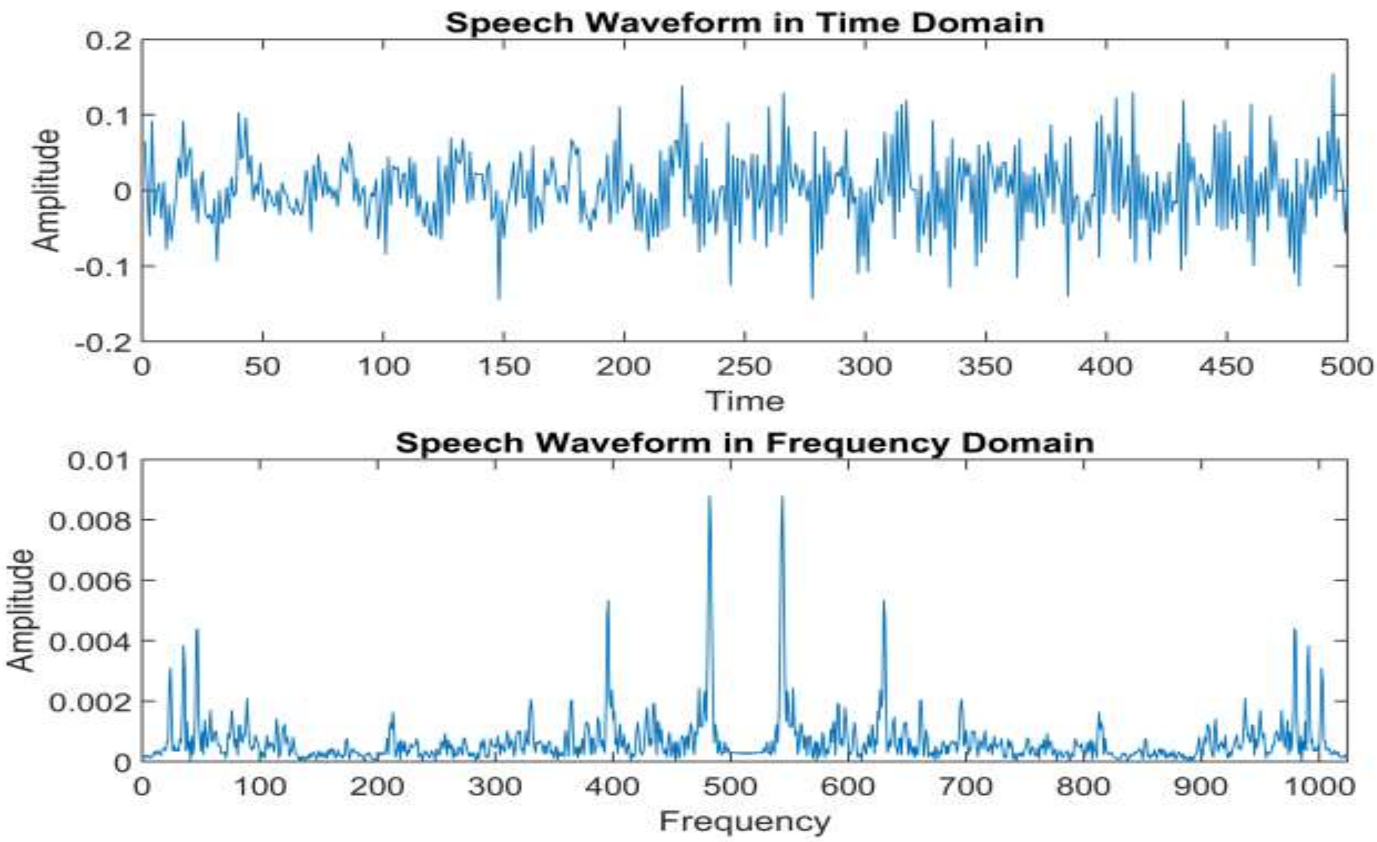
BPD on Noisy Speech Signal :

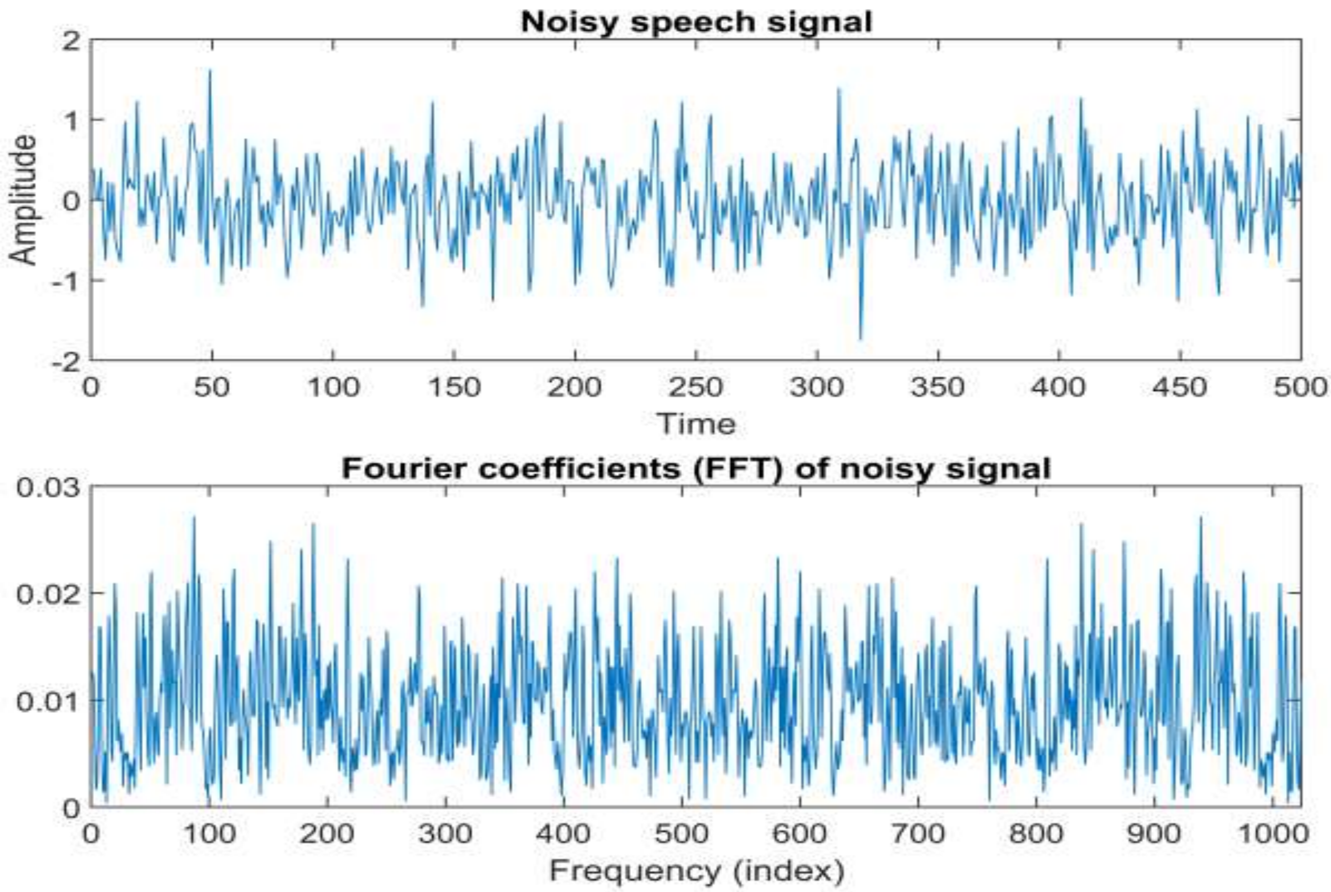
Here, a speech waveform is taken as the input signal and the signal is denoised using the BPD algorithm. The signal length is taken as 500 i.e., only 500 samples are considered from the total number of samples available in the selected speech signal.

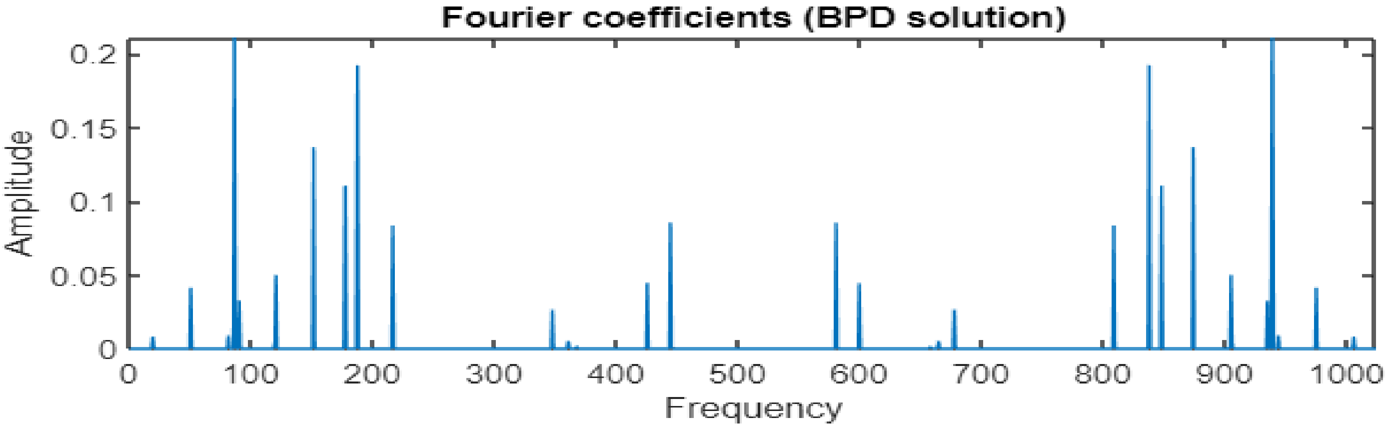
The .wav file is imported using the ***audioread*** command available in MATLAB as follows, and has been processed in the similar manner as illustrated in the previous example.

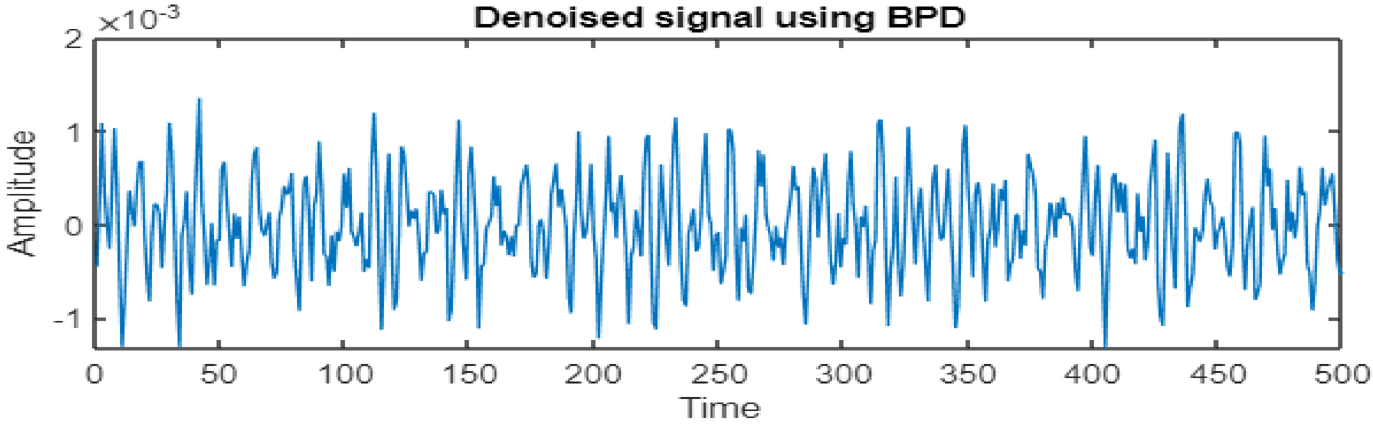
Importing Audio File	***[sp1, fs] = audioread('D:\SEM 5\DSP LAB\SpecialAssignment\file_example_WAV_1MG.wav');***

Outputs :


Speech Waveform in Time Domain
Amplitude
Time
0.2
0.1
0
-0.1
-0.2
0 50 100 150 200 250 300 350 400 450 500
Speech Waveform in Frequency Domain
Amplitude
Frequency
0.01
0.008
0.006
0.004
0.002
0
0 100 200 300 400 500 600 700 800 900 1000


Noisy speech signal
Amplitude
Time
2
1
0
-1
-2
0 50 100 150 200 250 300 350 400 450 500
Fourier coefficients (FFT) of noisy signal
Frequency (index)
0.03
0.02
0.01
0
0 100 200 300 400 500 600 700 800 900 1000





Hence, after the BPD Algorithm is applied on the noisy speech signal, noise is suppressed in the same and a clean speech signal is obtained.

Results and Conclusion :

Different signals were taken and noise was added to them. The signals were denoised and reconstructed using the BPD solution. The accuracy of reconstructed signal observed in the plot proves the effectiveness of BPD Algorithm. Thus, whenever the signal and noise overlap in the frequency domain and the noise can't be filtered using a LTI filter, BPD solution becomes the best alternative for signal reconstruction provided the original signal has relatively sparse coefficients.

Alternatively, the method of least squares can be used in place of BPD algorithm. However, comparison shows that the basis pursuit solution is sparser than the least square solution. The basis pursuit solution does not have the leakage phenomenon unlike least square method making it comparatively more effective.