H64DSP Coursework Report

Name: Hongbin Zhou   Student ID: 2125957

(1) Introduction

The aims of the project:

By using some different methods to process the data in corrupt1718.m4a and recover the data as much as possible to get the clean signal according to the knowledge from the module and out of lectures.

Objectives of the project:

Good understand for the theoretical methods introduced in the lecture.

Be able to distinguish the valid signal and noise and know what methods should be used.

Be able to design a variety of filters to remove the noise signal and interferences by using MATLAB.

Be able to implement a complete signal processing.

Methods used in project:

Median Filter
IIR Filter
FIR Filter with Window Functions
Adaptive Filter
Kalman Filter
Wiener Filter

Recommended methods:

Median Filter
FIR Filter with Window Functions
Adaptive Filter
Kalman Filter
Wiener Filter

Reasons for this recommendation:

Median filter is able to deal with some type of noise that other filters cannot do.

FIR filter with Window Functions is very common in digital signal processing and it is always stable and linearly.

Kalman filter is very similar to adaptive filter, but only need to know the Q and R which can estimate the best results.

Wiener filter is a kind of spectral filtering which is very useful when get the accurate spectrum of the underlying and noise signal.

(2) Methods and Results

The signal methods used in project, the implementations in MATALAB, the reasons for selecting each methods and the justification for choice of input parameters as follow part:

Median Filter:

The MATLAB command is

```
y = medfilt1(x,n);
```

Where n is window lengths, and the x is input signal. The output y is same length with x.

In project, it is used in line 14:

```
Signal = medfilt1(corrputsignal,5);
```

Reasons:

Median filter is an effective method to process some type of noise, for instance the impulsive noise. By observing the corrupt signal, it is obviously there exists a lot impulsive noise.

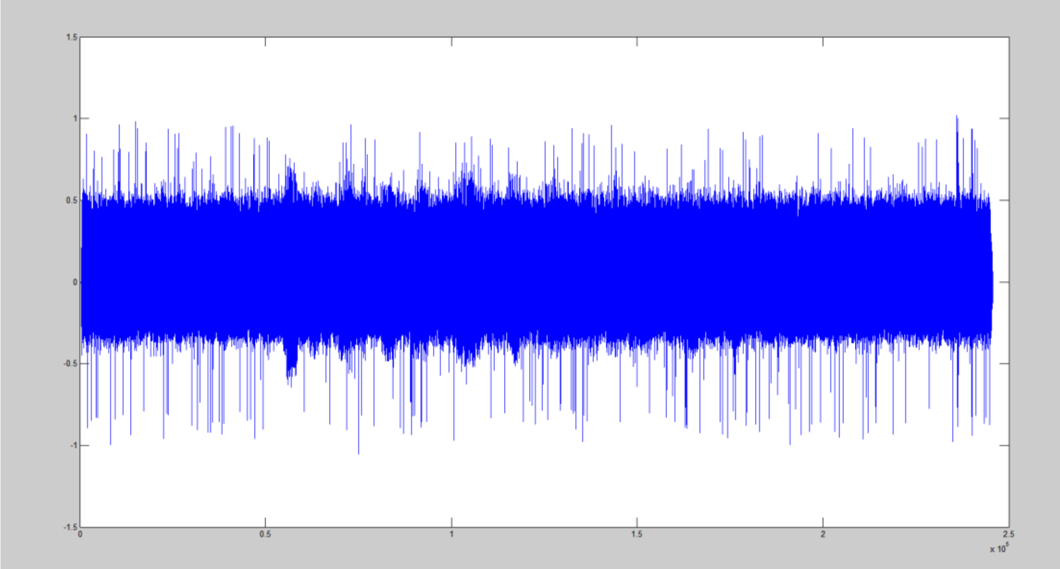
Justification for choice of input parameters:

Where corrputsignal is the input signal; 5 is the window length and Signal is output signal.

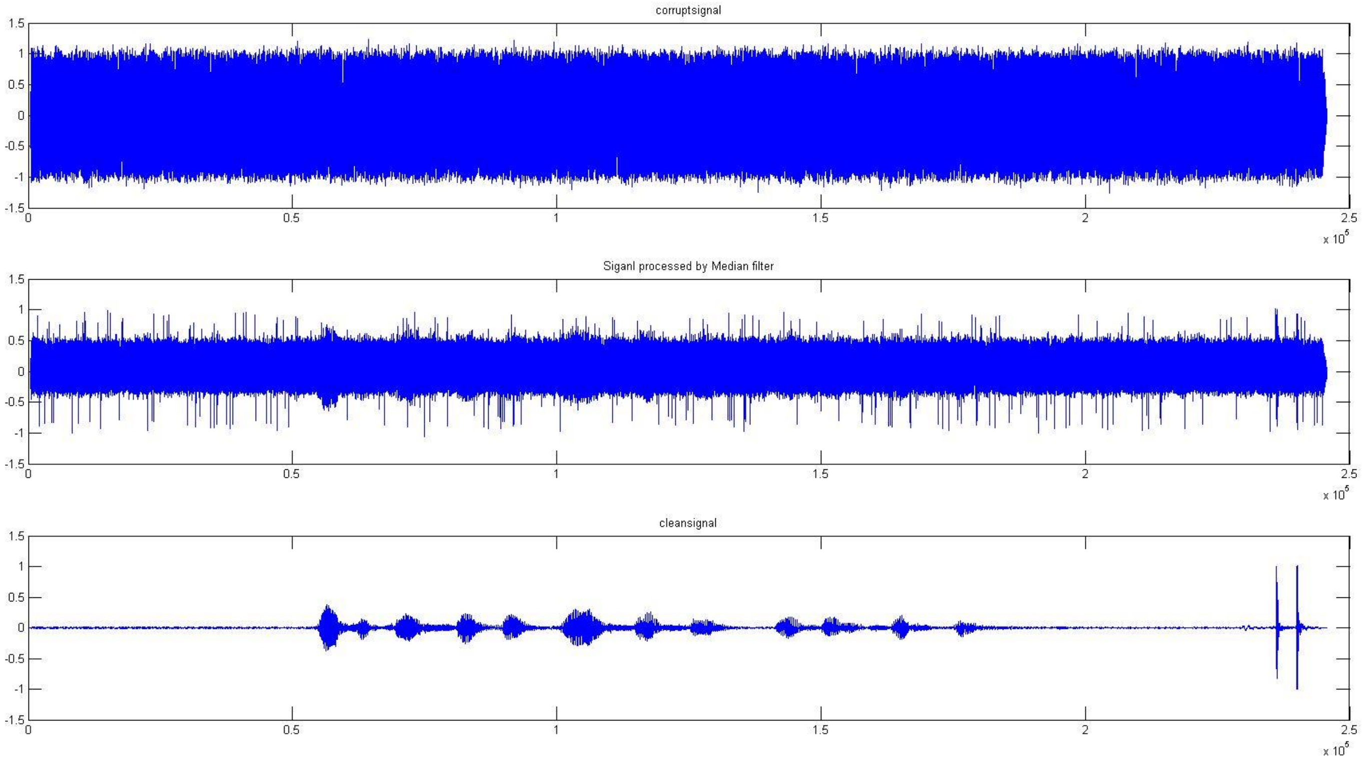
When n is odd, y(k) is the median of x(k-(n-1)/2:k+(n-1)/2).

When n is even, y(k) is the median of x(k-n/2:k+(n/2)-1). In this case, medfilt1 sorts the numbers and takes the average of the two middle elements of the sorted list. So choose n an odd is much better. If n is to large it may miss the valid signal. The default of n is 3; I choose 5 here.

Results shown and comparison of processed and clean data:



The comparison of corrupt signal, processed signal and clean signal:


corruptsignal

Siganl processed by Median filter

cleansignal

FIR Filter with Window Functions:

The MATLAB command is

```
B = fir1(n, Wn, ‘ftype’);
```

Where n is order of the filter, the Wn = fd/(1/2*fs) and ftype refers to the type of filter. B is the coefficients in the transfer function.

```
Y = filtfilt(b, a, x);
```

Where b and a are coefficients in the transfer function; x is input data and Y is output.

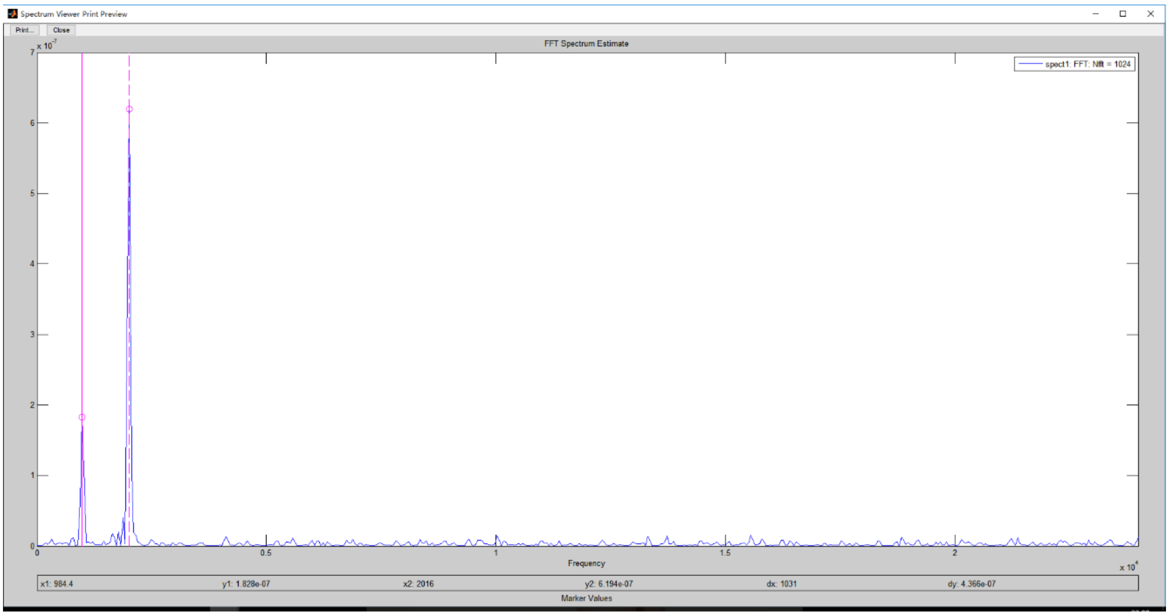
In project, it is used from line 23 to 35 and line 50 to 54:

```
% FIR filter
% FIR bandstop1 filter
order1 = 3000;
Wn1 = [roundn((980*2)/48000,-5),roundn((1020*2)/48000,-5)];
h1 = fir1( order1, Wn1,'stop');
processed1 = filtfilt(h1, 1, processed);

% FIR bandstop2 filter
order2 = 3000;
Wn2 = [roundn((1980*2)/48000,-5),roundn((2020*2)/48000,-5)];
h2 = fir1( order2, Wn2,'stop');
processed2 = filtfilt(h2, 1, processed1);

%FIR Low pass filter,
order3 = 50;
Wn3 = roundn((2000*2)/48000,-5);
h3 = fir1( order3, Wn3,'low');
processed3 = filtfilt(h3, 1, processed2);
```

There are 2 FIR bandstop filters and one FIR low pass filter.

Reasons:

FIR filter and IIR filter are both widely used in digital signal processing. Compared with IIR filter, FIR filter has exactly linear phase and they are always stable. The design methods are generally linear. FIR filter also have finite duration. Although there would a delay after filtered, the function filtfilt() in MATLAB would eliminate the delay. So FIR filter fits it good, I recommend to use FIR filter in this project.



x1: 984.4 y1: 1.828e-07 x2: 2016 y2: 6.194e-07 dx: 1031 dy: 4.366e-07

By using sptool in MATLAB, it is easily to find the signal seriously corrupted by the noise nearby 1khz and 2khz. The best way to use 2 bandstop FIR filters to correct the interferences. It also can be found that there are various noise exists at high frequency band. So it need another FIR low pass filter.

Justification for choice of input parameters:

Where processed is the input signal for first bandstop filter; processed1 is the output signal for first bandstop filter and the input signal for second bandstop filter.

processed2 is also the output for second bandstop filter and the input for low pass filter. processed3 is output for low pass filter.

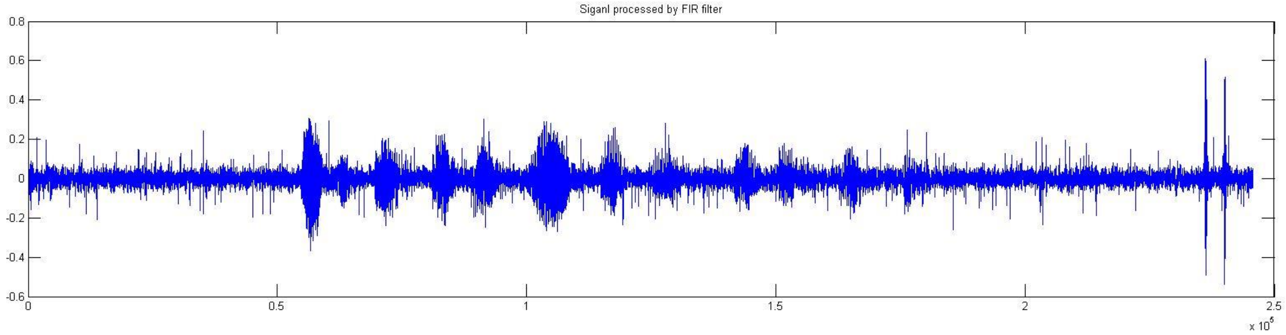
FIR filter always required high orders, so it actually needs 3000 orders for bandstop filter. For Low pass filter, 50 orders is enough.

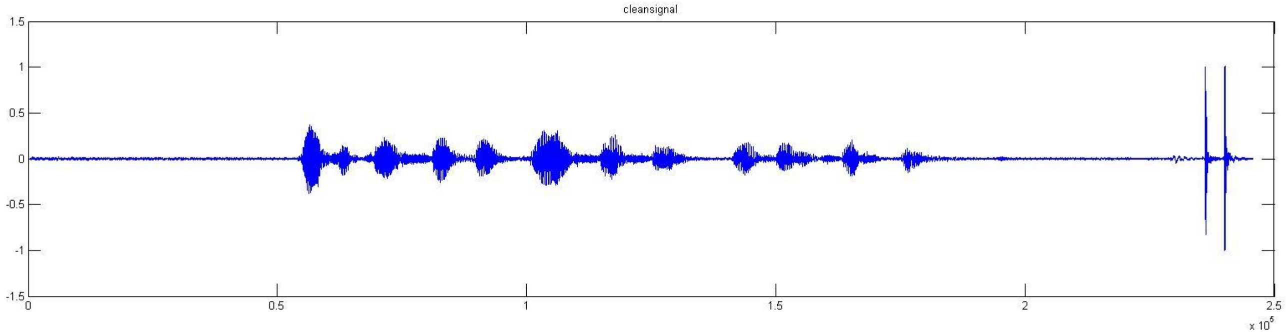
For first bandstop filter, the stop band is from 980Hz to 1020Hz. For second bandstop filter, the stop band is from 1980Hz to 2020Hz. It also can see that there are high frequency noise from 2kHz.

According to the formula Wn = fd/(1/2*fs), we can get the Wn as follows:

```
Wn1 = [roundn((980*2)/48000,-5),roundn((1020*2)/48000,-5)];
Wn2 = [roundn((1980*2)/48000,-5),roundn((2020*2)/48000,-5)];
Wn3 = roundn((2000*2)/48000,-5);
```

Results shown and comparison of processed and clean data:





IIR Filter:

The MATLAB command is

[B, A] = butter(n, Wn, ‘ftype’);

Where n is order of the filter, the Wn = fd/(1/2*fs) and ftype refers to the type of filter. B and A are the coefficients in the transfer fuction.

Y = filter (b, a, x);

Where b and a are coefficients in the transfer function; x is input data and Y is output.

In project, it is used from line 37 to 48:

```
Wn2 = [roundn((980*2)/48000,-3),roundn((1020*2)/48000,-3)];
[B2, A2] = butter(3,Wn2,'stop');
processed2 = filter(B2, A2, processed1);

Wn3 = [roundn((1980*2)/48000,-3),roundn((2020*2)/48000,-3)];
[B3, A3] = butter(3,Wn3,'stop');
processed3 = filter(B3, A3, processed2);
```

There are 2 IIR bandstop filters; the stop band is separately 980hz to 1020hz, 1980hz to 2020hz

Reasons:

Compared with FIR filter, IIR filter only needs a much lower order and it has a lot general purpose filters to meet a set of specifications. It is also a good choice to digital signal processing.

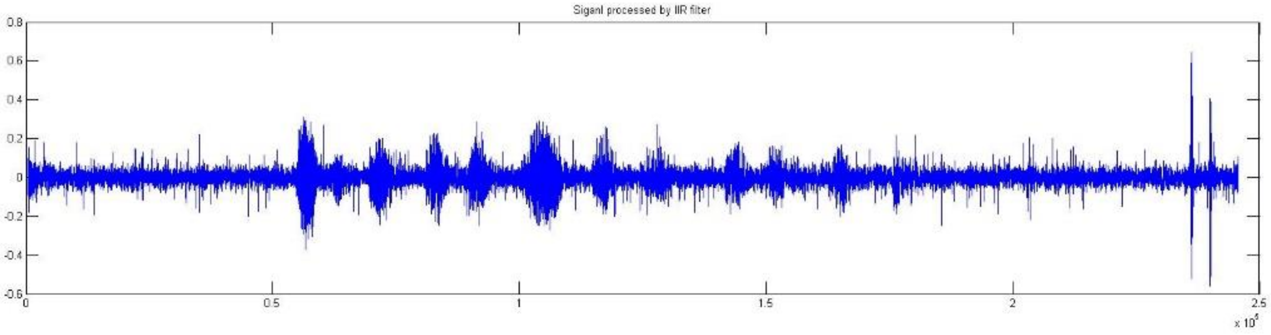
However, I do not recommend to use IIR filter here because it is not a linear phase which may cause distort to processed signal.

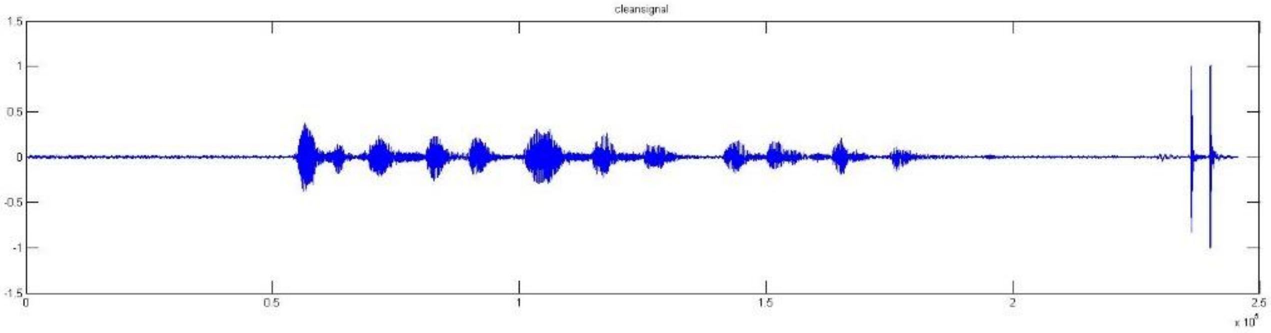
Justification for choice of input parameters:

It is just same as FIR filters but for order. The order is only 3 for both bandstop IIR filter.

Results shown and comparison of processed and clean data:

Kalman Filter:





In project, it is used from line 63 to 83:

```
% Kalman Filter
x_last =0;
p_last = 0;
R    = 0.09;
Q    = 0.005;
kg = 0;
for k = 1:N
    x_mid = x_last;
    p_mid = p_last + Q;
    kg = p_mid/(p_mid+R);
    x_now = x_mid+ kg*(processed3(k)- x_mid);
    p_now = (1-kg)*p_mid;
    x_last = x_now;
    X(k,:) = x_now;
end
```

Where X is estimated signal; Q is covariance of the process noise and R is covariance of the observation noise.

Reasons:

Kalman filter is very similar to adaptive filter, but there are also some differences between them. Kalman filter is use a series of measurements and predictions to obtain the best estimated results. Kalman filter can have a good effect on the signal with gauss noise.

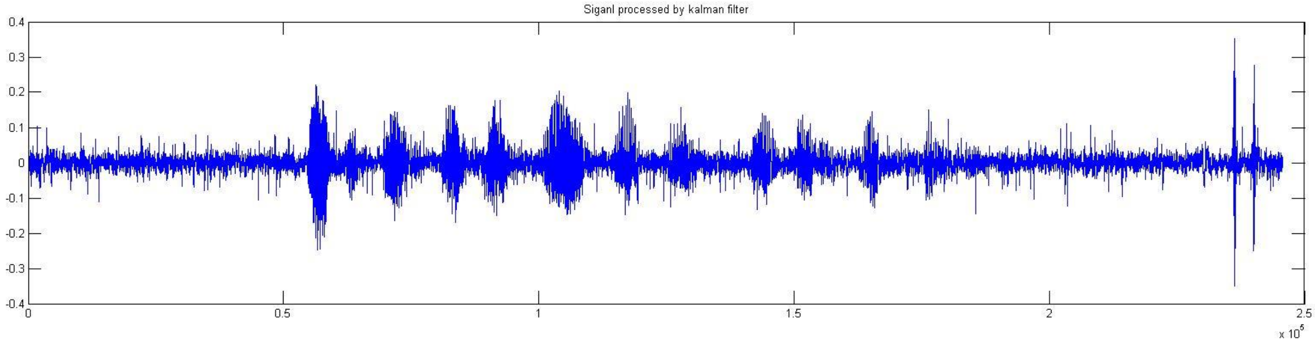
The most significant parameters is the Q(covariance of the process noise) and R(covariance of the observation noise). It determines the performance of your designed filter.

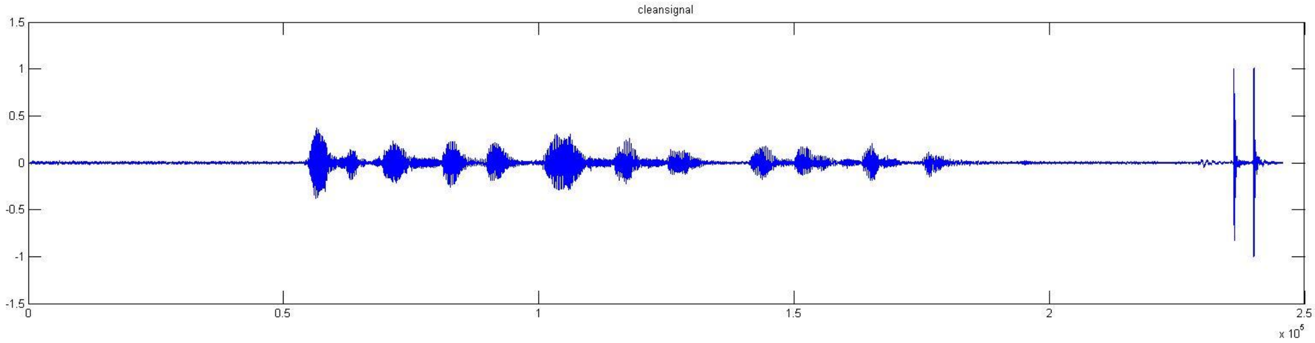
Justification for choice of input parameters:

After the signal processed by FIR filter, it still have some noises in processed signal. Suppose that noises are all gauss noise and use Kalman filter. Actually, what I can do is to try a lot different Q and R and get the best value.

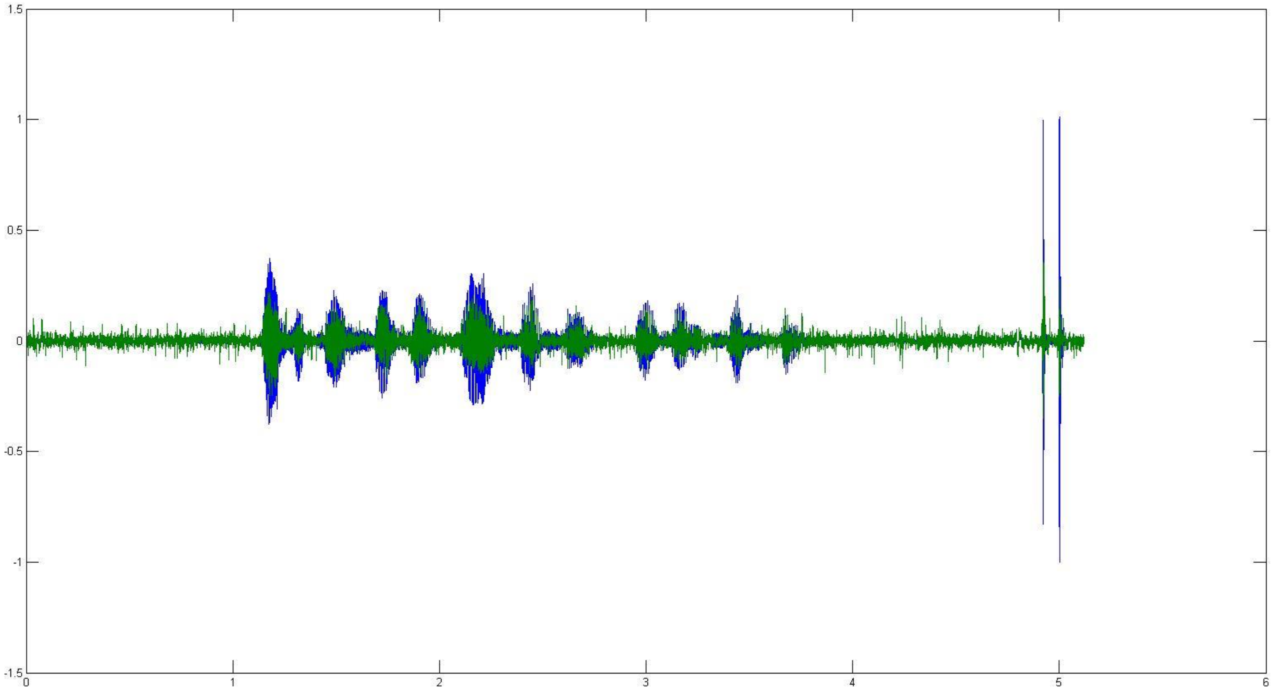
Results shown and comparison of processed and clean data:

Processed signal and clean data:





The processed signal and clean signal were shown in one figure. Green represent the processed signal and blue is clean signal.



Wiener Filter:

In project, it is used from line 85 to 104:

```
    % Wiener Filter
    % Estimate the noise signal
    noisesignal = X(1:53000);
    for i = 1:3
    noisesignal = [noisesignal ; zeros(40000,1)];
    end
    noisesignal = [noisesignal ;zeros(12000,1); X(185001:235000);
X(1:10759)];

    nosiefft = abs(fft(noisesignal));
    = X - noisesignal;
    sfft = abs(fft(UnderlySignal));
    Hf = sfft.^2 ./ (sfft.^2 + nosiefft.^2);
    processed4 = ifft(Hf.* fft(X));
```

Where DesireSignal is underlying signal; noisesignal is noise signal and Hf is frequency response.

Reasons:

Wiener filter can filter the signal with noise when we actually do not know the exact amplitude spectrum about the clean signal. By assuming that have known some information about the amplitude spectrum of clean signal, it may figure out the best H(f) to get the best results about desired signal.

Justification for choice of input parameters:

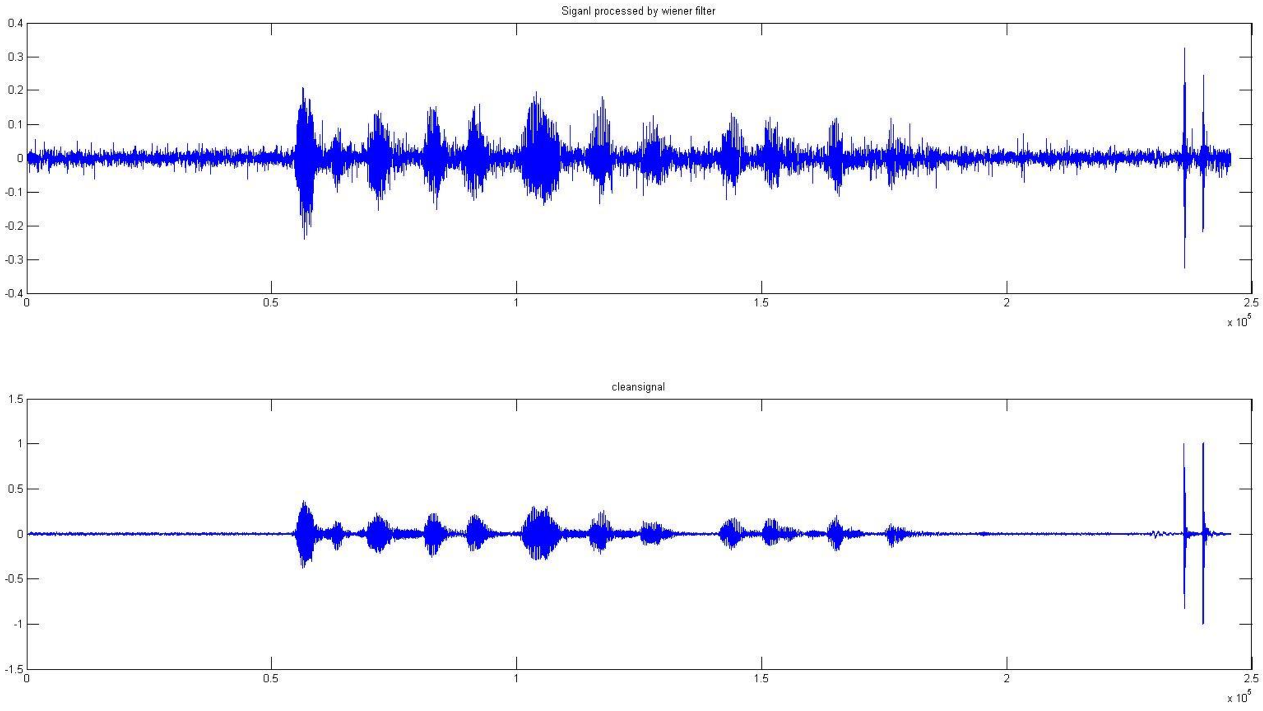
As we known, the data in the same parts of the corrupt signal is noise. I would estimate this part data as a part of noise signal. There were also some noise overlapped with underlying signal. I supposed this part noise were zero and combine all these parts to be a complete noise signal.

Then I got the noise signal and the underlying signal is the subtraction between corrupt signal and noise signal.

X is input corrupt signal and noisesignal is noise signal. The output is processed4.

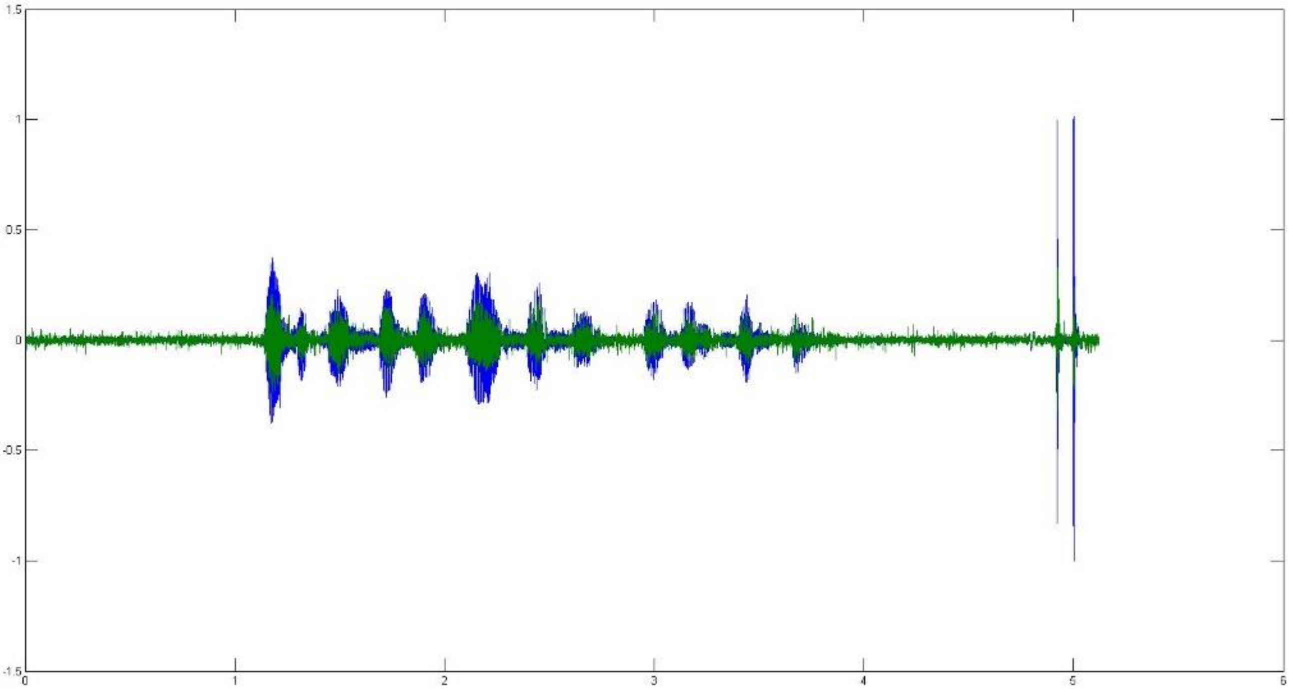
Results shown and comparison of processed and clean data:

Processed signal and clean data:



The processed signal and clean signal were shown in one figure to comparison.

Green represent the processed signal and blue is clean signal.



Adaptive Filter:

In project, it is used from line 106 to 130:

```
% Adaptive Filter
% Estimate the noise signal
noisesignal2 = processed3(1:53000);
for i = 1:3
noisesignal2 = [noisesignal2 ; zeros(40000,1)];
end
noisesignal2 =[noisesignal2 ;zeros(12000,1); process5(185001:235000); process5(1:10759)];
x = processed3 - noisesignal2;

mu = 0.0015;
ordr = 4;
W=zeros(1,ordr); % initialisation of weights to zero
e=zeros(1,N);
u = mean(x.^2);
for k=ordr:N
    Xk=x(k-ordr+1:k);
    yk(k) = W*Xk;
    e(k)=processed3(k)-yk(k);
    W=W+2*mu*e(k)*Xk';
 end
```

Where W is weight; yk is expect signal and x is reference signal.

Reasons:

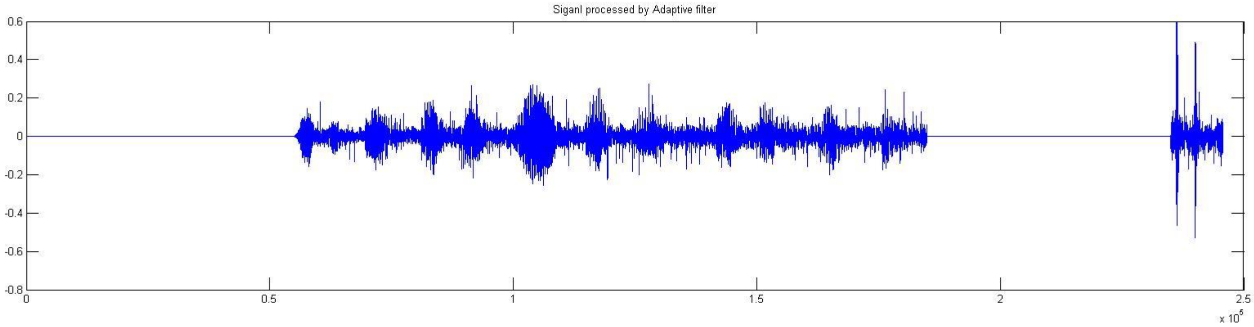
The adaptive filter is very similar to kalman filter, but adaptive filter do not know Q and R. The weights are varied with time as the signal, or noise changes. Adaptive filter has feedback to update its weights information. It is also a good method to process the signal.

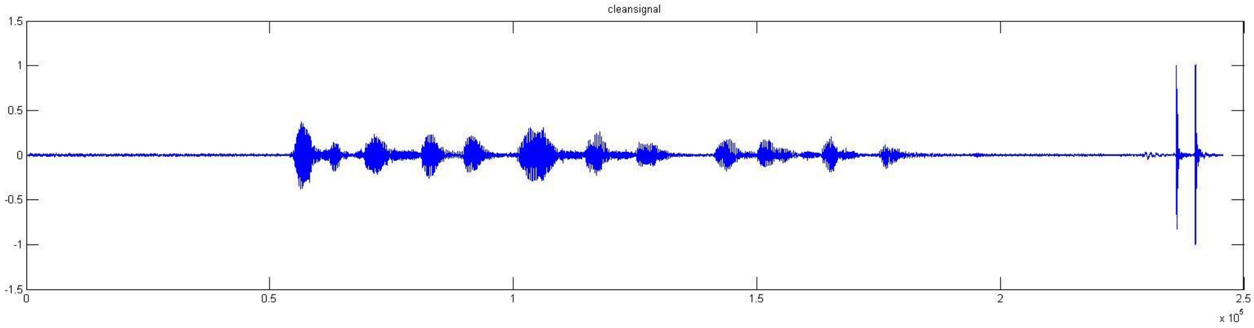
Justification for choice of input parameters:

Compared with Kalman filter, Adaptive filter need to know the reference signal. To get the reference, I use the same way of the wiener filter. Get the noise signal first and then get the reference signal by subtraction between input signal and noise signal.

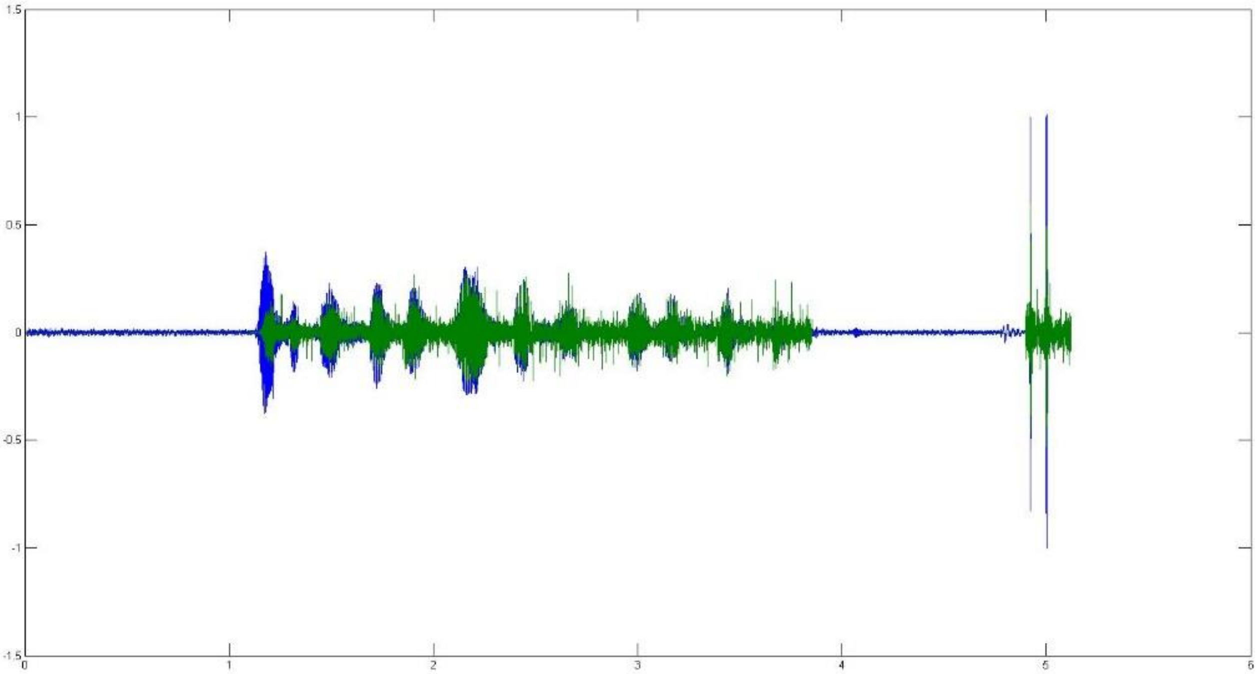
Results shown and comparison of processed and clean data:

Processed signal and clean data:





The processed signal and clean signal were shown in one figure to comparison.
Green represent the processed signal and blue is clean signal.



(3) Discussion of Results

Reasons:
By view the corrupt signal, we can know that the signal has been seriously distorted by noise where existed a lot impulsive noise.

Median filter is able to deal with this case and suits it well.

Using the median filter firstly to eliminate some terrible impulsive noise by choosing a proper window length which can ensure it would not missing the underlying data.

After this, using FFT and observing the amplitude spectrum, and it could be found the noise nearby 1kHz and 2kHz.

In this case, it needs 2 band stop filter. Using FIR filter with a good order and stop band would remove that specific noise.

There also has some high frequency noise from 2kHz, so it need to design a FIR low pass filter with cut off frequency at 2kHz to remove high frequency noise. So the FIR filter works well.

Then, we may find that some low frequency noise overlapped with desired signal. The low pass, high pass or bandstop filter would not work well if want to use these filters to process the overlap. So it actually may be useful for use some advanced filter like kalman filter and wiener filter.

Kalman filter is absolutely an excellent method to process the signal with gauss noise. Here, we suppose that the noise overlapped with desired signal is gauss noise. Next step is to choose the Q (covariance of the process noise) and R (covariance of the observation noise) which is very important to this signal. I tried a lot various Q and R with comparing the processed signal and clean signal, and eventually to get the best value Q and R. The output signal is usually called best estimated value which is combined by the observation, prediction and last time status. So the kalman filter works well.

Finally, get some obvious noise data and make up a complete signal. Then we can get the amplitude spectrum of noise signal and underlying signal. Use the formula to figure out the frequency response H(f). And then we can get the processed signal by use H(f) and FFT. This method would correct the signal to some extent.

(4) Conclusions

The best combination of signal processing method:

Use median filter firstly, then FIR filters including two bandstop filters and one low pass filter, Kalman filter and finally Wiener filter.

This combinations of signal processing methods worked well.

In my project, I totally used 2 combinations of signal processing methods to correct the corrupt signal.

Both of them have the same few methods. The difference is that I choose FIR not IIR filter, kalman filter not the adaptive filter.

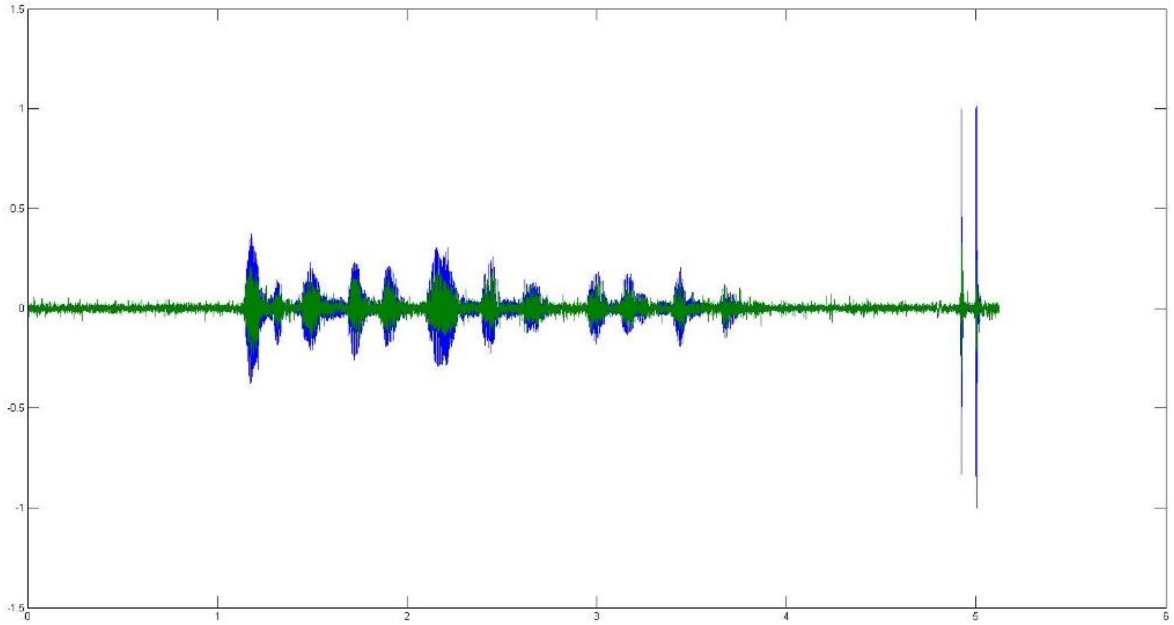
The reason why to use FIR filter is that FIR filter has linear phase and always stable. I can also use filtfilt() function in MATLAB to avoid the delay which caused by phase response and this is not the real-time processing either. So FIR is better than IIR filter.

Compared with adaptive filter, kalman filter only need to the get Q (covariance of the process noise) and R(covariance of the observation noise). We can try use different Q and R to get the best results. But for adaptive filter, you must to get the reference signal. In this project, we cannot know the reference signal and we cannot try if we know nothing.

Optimal input parameters are all mentioned on PART II.

The final processed signal is almost coincide with clean signal. The figure can be seen at next page.

Green represent the processed signal and blue is clean signal.


1.5
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5
6