Spring 5-19-2018

Linear Phase Multi-Frequency Notch Filter via Quadratic Programming

YUERAN MA
yueranm@smu.edu

LINEAR PHASE MULTI-FREQUENCY NOTCH FILTERS VIA QUADRATIC PROGRAMMING

Approved by:

____
Prof. Carlos Davila
Associate Professor of Electrical Engineering

____
Prof. Scott Douglas
Professor of Electrical Engineering

____
Prof. Panos Papamichalis
Professor of Electrical Engineering

LINEAR PHASE MULTI-FREQUENCY NOTCH FILTERS VIA QUDRATIC PROGRAMMING

A Thesis Presented to the Graduate Faculty of

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Master of Science

with a

Major in Electrical Engineering

by

Yueran Ma

B.E., Electrical Engineering, Beijing Jiaotong University, Beijing

May 19, 2018

ACKNOWLEDGMENTS

This work could not have been accomplished without my advisor, Prof. Davila, along with his many colleagues in the Department of Electrical Engineering here at SMU. I’m grateful to my advisor for our discussion and patiently explaining some details to me. I'm also forever grateful to my family for being so patient with me and to my friends for reminding me to take breaks and being with me during my spare time.

Yueran Ma B.E., Electrical Engineering, Beijing Jiaotong University, Beijing, 2016

Linear Phase Multi-Frequency Notch Filters
via Quadratic Programming

Advisor: Associate Professor Carlos Davila

Master of Electrical Engineering conferred May 19, 2018

Thesis completed April 17, 2018

Discrete-time notch filters can be divided into infinite impulse response (IIR) and finite impulse response (FIR) notch filters. Infinite impulse response notch filters are easy to design and implement but suffer from nonlinear phase characteristics and unacceptable startup transients. Finite impulse response filters on the other hand can be designed to have linear phase but require more coefficients to achieve narrow notch widths. Multiple frequency FIR notch filters that can effectively reject several selected spectral regions while providing high transmission at frequencies outside the rejected regions have applications in communication systems, radar systems and biomedical signal processing.

A new multi-frequency notch filter is introduced that is based on quadratic programming. This notch filter has linear phase and has notch widths which can be made arbitrarily narrow. We compare the performance of our notch filter with three existing multi-frequency notch filters: the iteratively reweighted OMP scheme and the multiple exchange algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Introduction

1.1 Background on Filters

Filtering technology plays an important role in signal processing. The filter can be divided into four kinds from the perspective of frequency characteristics: low-pass, high-pass, band-pass and band-rejection. It can be divided into two kinds according to signal processing methods: analog filter and digital filter. The main functions of the filter include: 1) Limit the signal in a specific frequency range, such as low-pass filter and other three kinds of filters; 2) Divide the signal into two or more sub-band signals, such as filter bank, graphic equalizer , sub-band encoders and frequency multiplexers, etc.; 3) Modify the spectral information of the signal, such as the equalizer in the telephone channel and the equalizer in audio; 4) Model the input and output relations of the system, such as telecommunication channels, human audio channels, and music synthesizer and so on. The filter has a wide range of applications, specifically applied to channel equalization, noise reduction, radar system and voice signal processing, etc. For example, a band-pass filter in a radio receiver is used to extract related signals from its channel. In a digital filter system, the discrete signals are already sampled, thereby reducing or increasing the information of some parts of the signals. In the audio graphic equalizer, the input signal is filtered into a plurality of sub-band signals so that the audio can be changed by manually adjusting a set of control signals. In the Dolby system, pre-filtering and post-filtering are used to minimize the effects of noise [1] . In Hi-Fi audio, the compensation filter in the pre-amplifier is used to compensate for the non-ideal frequency response characteristics of the speaker. Filters are also used for music, film, or broadcast rooms to create audiovisual effects [1] . As for the analog filter and digital filter, due to their respective characteristics, they are applied in different fields. Analog filters are cheap and have large dynamic range in both time and frequency domain. In contrast, digital filters have some advantages in attainable performance. From the impulse response point of view, digital filters can be divided into two types: finite impulse response (FIR) filters and infinite impulse response (IIR) filters. The IIR filter has a more mature design method but the designed filter order is lower, and it cannot achieve a strict linear phase. However, the phase of the FIR filter can get linear characteristics, which is very important in some applications with relatively demanding phase requirements, such as image processing [1] .

1.2 Background on Notch Filters

The notch filter is a band-stop filter whose stopband is very narrow, and ideally there is only one single frequency rejection. This kind of stop-band filter is mainly used to eliminate a certain frequency and should have relatively little effect on the amplitude of the remaining frequencies. Because of the special characteristics of the stop band, the notch filter has a very wide range of applications. Digital notch filters are used in communications, control, instrumentation, and biomedical engineering to eliminate-noise [2] [3] . Digital notch filters in medical equipment can eliminate 60Hz interference from the electrocardiogram (ECG) signal and electroencephalograph (EEG) signal. In the field of automatic control, digital notch filters are used in industrial robots to suppress the vibration generated by resonance phenomena due to high-speed operation of robots [3] . In laser system applications, a notch filter with a length of four wavelengths is designed to be placed in the transmitter to suppress undesired local oscillation signals; it can also be used in the coding field. For example, in PSK coding, the receiver can use a single-frequency notch filter to eliminate the carrier frequency [4] ; and in the process of digital signal processing, it can be used to detect the attenuation part or frequency component of the signal, for example, the Dual Tone Multi-Frequency (DTMF) [5] .

1.3 Background on Multi-Frequency Notch Filters

The multi-frequency notch filter includes a plurality of notch frequencies, generally having a non-uniform interval between notch frequencies. When its notch frequencies are evenly spaced, it will be a comb filter [6] . A two-dimensional multi-frequency notch filter can filter out periodic textures of digital images [7] . The multi-frequency notch filter can also be applied in fields such as communications, radar, sonar, biological signal processing and machinery [8] . For example, high-precision wireless positioning by using spread spectrum communication signals, eliminating resonant frequencies in mechanical systems, predicting protein coding regions and predicting protein hotspots [9] . Considering that in actual situations, the order of the FIR filter is usually several hundred or more, the study of low-order and sparse filters has been paid more and more attention. A sparse FIR filter leads to hardware implementation requiring a smaller number of adders and multipliers thereby reducing its computational complexity, improving the accuracy of operation and reducing its energy consumption. The design of low-complexity, high-precision, low-order linear phase FIR multi-frequency notch filters has important practical significance in the above-mentioned fields.

II. Research Status

2.1 Single-Frequency Notch Filter

For one-dimensional linear phase FIR single-frequency notch filter algorithms, researchers have proposed many solutions. Following are some representative methods: the Maximally Flat (MF) Algorithm [11] , the Multiple Exchange (ME) [19] , the Optimal Equiripple (OE) [12] , the Precise Equiripple (PE) [13] and the algorithm based on Iteratively Reweighted Orthogonal Matching Pursuit (IROMP) [8] . The notch frequency in the Maximally Flat (MF) algorithm and the Optimal Equiripple (OE) algorithm is determined by the Zolotarev polynomial, but since the calculation of the order in the Zolotarev polynomial involves a rounding operation, this will cause a shift in the notch frequency. However, the Precise Equiripple (PE) algorithm solves this problem, but it sacrifices the performance of the notch filter. For example, the width of the stopband is enhanced and the ripple in the passband is increased [13] . The linear phase FIR notch filters implemented by these methods are all sparse which means many coefficients are zero, and when the given design requirements (such as passband ripple, stopband bandwidth, etc.) are more demanding, the FIR notch filter order is usually up to several hundred. The algorithm based on Iteratively Reweighted Orthogonal Matching Pursuit (IROMP) has good performance compared with the other filter design algorithms mentioned above, and the number of non-zero taps of the implemented FIR single-frequency notch filter can be reduced by more than 10% when it satisfies the same requirements [8] . At the same time, the linear phase FIR notch filter implemented by this algorithm also overcomes the problem of shifting of the notch frequency.

2.2 Multi-Frequency Notch Filter

Linear phase FIR multi-frequency notch filters, whose notch frequencies are evenly spaced are called comb filters. Comb filters can be designed using Chebyshev polynomials [14] , but these methods are less applicable and do not work when the interval between the notch frequencies is non-uniform [15] [16] [17] [18] . When the interval between the notch frequencies is non-uniform, a number of approaches can be used. Among these are cascading multiple single notch FIR filters (CSSNF), the Multiple Exchange (ME) [19] , the Iterative reweighed l_1 Design Schemes (IRL1) [20] , the Iterative Second-Order Cone Programming (ISOCP) [21] and the algorithm based on Iteratively Reweighted Orthogonal Matching Pursuit (IROMP) [8] . The multi-frequency notch filter obtained by cascading single-frequency notch filters will increase the ripple of the filter passband, and at the same time will decrease its attenuation at the notch frequency and greatly reduces the overall performance of the multi-frequency notch filter. As for the Multiple Exchange (ME) algorithm, it requires a large filter order to ensure the convergence and stability of the algorithm when the interval between notch frequencies is small. At the same time, under the same design criteria, when the notch frequencies change, the Multiple Exchange (ME) algorithm needs to recalculate the entire filter once again. This is rather troublesome which means it is difficult to adapt the notch frequencies. The Iterative reweighed l_1 Design Schemes (IRL1) which was proposed by Cristian Rusu is an algorithm that combines the minimum of the reweighted l_1 and greedy iterations. The algorithm gives the filter more non-zero coefficients after minimizing the reweighed l_1, and the next greedy iteration is used to eliminate several redundant coefficients. The Iterative Second-Order Cone Programming (ISOCP) proposed by Aimin Jiang finds a potential sparse model first, and then uses this sparse model to solve a convex optimization problem to get the final result [21] . However, neither of these two algorithms can achieve frequency adjustability. As for the algorithm based on Iteratively Reweighted Orthogonal Matching Pursuit (IROMP), It is a better algorithm for implementing the multi-frequency notch filter.

3.1 FIR Notch Filters

The notch filter is a type of band-stop filter, ideally with only one or more rejection frequencies. Notch filters can generally be divided into two forms, the first of which is more common. And those two can be written as the same form: $H(\omega) = e^{-\frac{jN}{2}\omega}|H_0(\omega)|$.

The amplitude response of the first type of notch filter is shown in Figure 1:

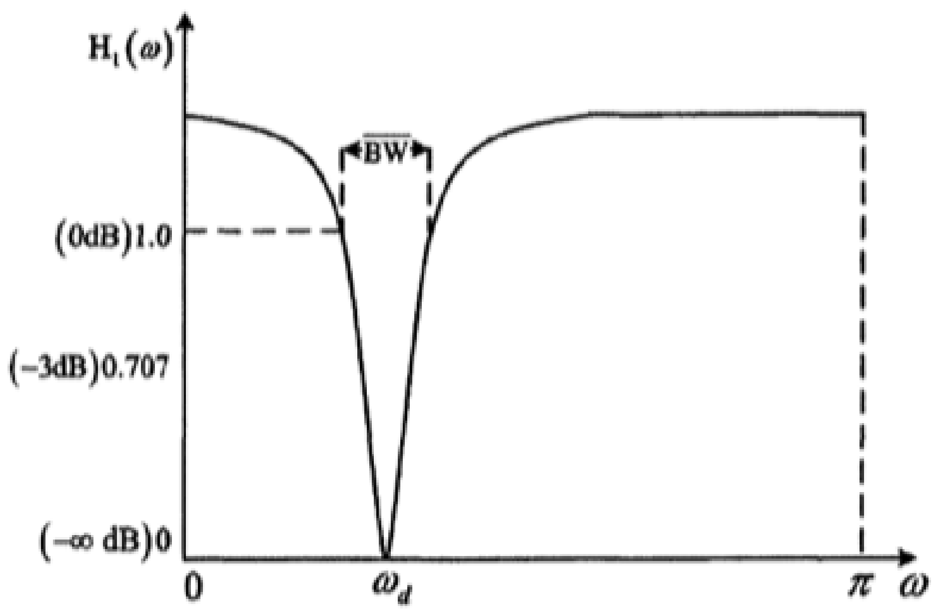

Figure 1 First Type FIR Notch Filter

It is characterized by a notch frequency ω_d and a stopband width $\overline{BW}$. The stopband width for an ideal notch filter should be zero, and the attenuation at the notch frequency location should be infinite, and the passband ripple should be zero.

The amplitude response of the second type of notch filter is shown in Figure 2:

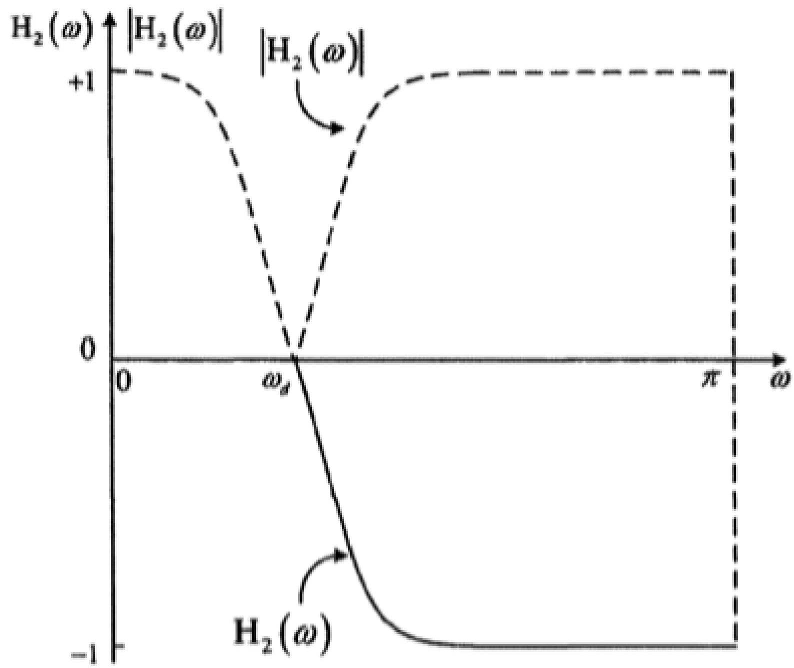

Figure 2 Second Type FIR Notch Filter

The difference from the first type is that it has a 180° inversion at the notch frequency. For the $|H_2(\omega)|$, it has the same amplitude response with the first type.

3.2 Multiple Exchange Algorithm

Compared to single exchange algorithms such as linear programming [23] , Multiple Exchange algorithms (ME) are faster. The Multiple Exchange algorithm needs to find a new set of alternation frequencies to replace the original set of alternation frequencies in each iteration. This type of multiple exchange algorithm can not only complete the design of a single-frequency notch filter, but also the design of a multi-frequency notch filter. The specific implementation of this algorithm will be described below.

Firstly, the multiple exchange method for a single-frequency notch filter is described. To facilitate the elaboration of the method, the mathematical expression of the FIR notch filter is first elaborated, and some symbols to be used in this method are firstly defined and described.

The mathematical expression of the *n*-order FIR filter with causality is:

$$H(z) = \sum_{n=0}^{N} h(n) z^{-n} \tag{3.1}$$

And its amplitude response can be expressed as:

$$A(\omega) = \sum_{n=1}^{M} a_n trig(\omega, n) \tag{3.2}$$

The specific form of the function $trig(\omega, n)$ is determined by the parity of the filter order N and the symmetry of the time domain impulse response. The relationship between the time domain impulse response $h(n)$, the order *N* and the parameter *M*, the coefficients a_n and the function $trig(\omega, n)$ is summarized in Table 1:

$h(n)$	N	Relationship
Symmetric	Even	$M = N/2$ $a_1 = h(N/2)$ $a_n = 2h(N/2 + 1 - n), n = 2, \dots, M$ $trig(\omega, n) = \cos((n-1)\omega)$
Symmetric	Odd	$M = (N+1)/2$ $a_n = 2h((N+1)/2 - n), n = 1, \dots, M$ $trig(\omega, n) = \cos((n-1/2)\omega)$
Antisymmetric	Even	$M = N/2$ $a_n = 2h(N/2 - n), n = 1, \dots, M$ $trig(\omega, n) = \sin(n\omega)$
Antisymmetric	Odd	$M = (N+1)/2$ $a_n = 2h((N+1)/2 - n), n = 1, \dots, M$ $trig(\omega, n) = \sin((n-1/2)\omega)$

Table 1 Relationship between h(n) and Parameters

Define the column vector:

$$a = [a_1, a_2, \dots, a_M]^T \tag{3.3}$$

$$c(\omega) = [trig(\omega, 1), trig(\omega, 2), \dots, trig(\omega, M)]^T \tag{3.4}$$

Then (3.2) can be simplified as:

$$A(\omega) = a^T c(\omega) = c^T(\omega) a \tag{3.5}$$

In this way, the filter design problem is mathematically transformed into finding a coefficient vector a to satisfy the constraint:

$$D(\omega) = \begin{cases} 0, & \omega = \pm\omega_N \\ 1, & otherwise \end{cases} \tag{3.6}$$

where ω_N represents the notch frequency.

From the amplitude response function $A(\omega)$ of the euquiripple FIR notch filter, the amplitude at the alternation frequencies ω_{1l} and ω_{1r} is $1 + \delta$, and the amplitude at the alternation frequencies ω_{il} and ω_{ir} is $1 - (-1)^i \delta$. Assume that there are N_l alternation in the left pass band $[0, \omega_N - \varepsilon]$ and there are N_r alternation in the right pass band $[\omega_N + \varepsilon, \pi]$, in which ε is a specified, smaller, positive number. Because $A(\omega_N) = 0$ is imposed on the amplitude response, there is no iterative theorem that can specify the number of extrema to guarantee optimality. However, Multiple Exchange algorithms can design this type of filter by selecting alternation points. The specific choice is: in the left pass band, the alternation frequencies selection rule is:

$$\omega_{1l} > \omega_{2l} > \cdots > \omega_{N_l l} \tag{3.7}$$

In this way, the original design problem becomes satisfied by finding the filter coefficients having the following conditions:

$$A(\omega_{il}) = 1 - (-1)^i \delta, i = 1, \dots, N_l \tag{3.8}$$

In the right pass band, the alternation frequency selection rule is:

$$\omega_{1r} < \omega_{2r} < \cdots < \omega_{N_r r} \tag{3.9}$$

Like the left pass band alternation selection method, the original design problem is satisfied after finding the filter coefficients under the following conditions:

$$A(\omega_{ir}) = 1 - (-1)^i \delta, i = 1, \dots, N_r \tag{3.10}$$

From the above description, equation (3.2) is substituted into equations (3.8) and (3.10) to obtain the matrix:

$$\begin{bmatrix} trig(\omega_{N_l l}, 1) \dots trig(\omega_{N_l l}, M) \\ \vdots \quad \cdots \quad \vdots \\ trig(\omega_{2l}, 1) \dots trig(\omega_{2l}, M) \\ trig(\omega_{1l}, 1) \dots trig(\omega_{1l}, M) \\ trig(\omega_{N}, 1) \dots trig(\omega_{N}, M) \\ trig(\omega_{1r}, 1) \dots trig(\omega_{1r}, M) \\ trig(\omega_{2r}, 1) \dots trig(\omega_{2r}, M) \\ \vdots \quad \cdots \quad \vdots \\ trig(\omega_{N_r r}, 1) \dots trig(\omega_{N_r r}, M) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} 1 - (-1)^{N_l}\delta \\ \vdots \\ 1 - \delta \\ 1 + \delta \\ 0 \\ 1 + \delta \\ 1 - \delta \\ \vdots \\ 1 - (-1)^{N_r}\delta \end{bmatrix} \tag{3.11}$$

Using Φ to represent the left matrix and b to represent the right vector in (3.11), then the equation (3.11) can be simplified to:

$$\Phi \mathrm{a} = \mathrm{b} \tag{3.12}$$

If $N_l + N_r + 1$ is not less than M, then we can see that the above equation is a set of positively definite equations, and the least squares solution of these equations is:

$$a = (\Phi^T \Phi)^{-1} \Phi^T b \tag{3.13}$$

Based on the above description, specific design steps for designing equiripple filter notch filters using the Multiple Exchange algorithms are:

Step 1: Determine the order N and ω_N based on design criteria;

Step 2: Set the initial value, and find the alternation frequencies $\Omega_{il}(i = 1,2, \dots, N_l)$ and $\Omega_{ir}(i = 1,2, \dots, N_r)$;

Step 3: Set $\omega_{il} = \Omega_{il}(i = 1,2, \dots, N_l)$ and $\omega_{ir} = \Omega_{ir}(i = 1,2, \dots, N_r)$;

Step 4: Calculate the average peak error:

$$\delta = \frac{1}{N_l + N_r}\left(\sum_{i=1}^{N_l} |A(\omega_{il}) - 1| + \sum_{i=1}^{N_r} |A(\omega_{ir}) - 1|\right) \tag{3.14}$$

Step 5: Calculate the new filter coefficients vector a by solution of (3.13);

Step 6: Find the alternation frequencies $\Omega_{il}$and Ω_{ir} in the passband and return to the third step to repeat the loop until the following conditions are satisfied:

$$|\Omega_{il} - \omega_{il}| \leq \varepsilon_1, i = 1, \dots, N_l \tag{3.15}$$

$$|\Omega_{ir} - \omega_{ir}| \leq \varepsilon_1, i = 1, \dots, N_r \tag{3.16}$$

The above describes in detail the application of the Multiple Exchange algorithms for single-frequency notch filters. The following describes its application to multi-frequency notch filters. Take two notch frequencies ω_{1N} , ω_{2N} as an example:

The ideal frequency-domain amplitude of the double notch frequency filter is expressed as:

$$D(\omega) = \begin{cases} 0, & \omega = \pm\omega_{1N}\ or\ \omega = \pm\omega_{2N} \\ 1, & otherwise \end{cases} \tag{3.17}$$

there are N_l alternation in the left pass band $[0, \omega_{1N} - \varepsilon]$, there are N_m alternation in the middle pass band $[\omega_{1N} - \varepsilon, \omega_{2N} + \varepsilon]$ and there are N_r alternation in the right pass band $[\omega_{2N} + \varepsilon, \pi]$, and the frequency specification rules for the three pass band alternation frequencies:

$$\omega_{1l} > \omega_{2l} > \cdots > \omega_{N_l l} \tag{3.18}$$

$$\omega_{1m} > \omega_{2m} > \cdots > \omega_{N_m m} \tag{3.19}$$

$$\omega_{1r} < \omega_{2r} < \cdots < \omega_{N_r r} \tag{3.20}$$

At the same time, the following restrictions must be satisfied when looking for its coefficients:

$$A(\omega_{il}) = 1 - (-1)^i \delta, i = 1, \dots, N_l \tag{3.21}$$

$$A(\omega_{im}) = 1 - (-1)^i \delta, i = 1, \dots, N_m \tag{3.22}$$

$$A(\omega_{ir}) = 1 - (-1)^i \delta, i = 1, \dots, N_r \tag{3.23}$$

Based on the above description, equation (3.11) can be expressed as a matrix:

$$\begin{bmatrix} trig(\omega_{N_l l}, 1) \dots trig(\omega_{N_l l}, M) \\ \vdots \quad \cdots \quad \vdots \\ trig(\omega_{2l}, 1) \dots trig(\omega_{2l}, M) \\ trig(\omega_{1l}, 1) \dots trig(\omega_{1l}, M) \\ trig(\omega_{1N}, 1) \dots trig(\omega_{1N}, M) \\ trig(\omega_{N_m m}, 1) \dots trig(\omega_{N_m m}, M) \\ \vdots \quad \cdots \quad \vdots \\ trig(\omega_{2m}, 1) \dots trig(\omega_{2m}, M) \\ trig(\omega_{1m}, 1) \dots trig(\omega_{1m}, M) \\ trig(\omega_{2N}, 1) \dots trig(\omega_{2N}, M) \\ trig(\omega_{1r}, 1) \dots trig(\omega_{1r}, M) \\ trig(\omega_{2r}, 1) \dots trig(\omega_{2r}, M) \\ \vdots \quad \cdots \quad \vdots \\ trig(\omega_{N_r r}, 1) \dots trig(\omega_{N_r r}, M) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} 1 - (-1)^{N_l} \delta \\ \vdots \\ 1 - \delta \\ 1 + \delta \\ 0 \\ 1 + \delta \\ \vdots \\ 1 - \delta \\ 1 + \delta \\ 0 \\ 1 + \delta \\ 1 - \delta \\ \vdots \\ 1 - (-1)^{N_r} \delta \end{bmatrix} \tag{3.24}$$

And then (3.25) can be simplified as:

$$\Psi a = d \tag{3.25}$$

If $N_l + N_m + N_r$ +2 is not less than M, then we know that (3.25) is a set of positively definite equations, and the least squares solution of these equations is:

$$a = (\Psi^T \Psi)^{-1} \Psi^T d \tag{3.26}$$

Like the previous section, the calculation steps for the multi-frequency notch filter coefficients are summarized as:

Step 1: Determine the order N, ω_{1N} and ω_{1N} based on the design criteria;

Step 2: Use the Lagrange multiplier method to guess the initial value of the filter coefficients and find the alternation frequencies $\Omega_{il}(i = 1,2, \dots, N_l)$, $\Omega_{im}(i = 1,2, \dots, N_m)$, $\Omega_{ir}(i = 1,2, \dots, N_r)$;

Step 3: Set $\omega_{il} = \Omega_{il}(i = 1,2, \dots, N_l)$, $\omega_{im} = \Omega_{im}(i = 1,2, \dots, N_m)$, $\omega_{ir} = \Omega_{ir}(i = 1,2, \dots, N_r)$;

Step 4: Calculate the average peak error:

$$\delta = \frac{1}{N_l + N_m + N_r} \left(\sum_{i=1}^{N_l} |A(\omega_{il}) - 1| + \sum_{i=1}^{N_m} |A(\omega_{im}) - 1| + \sum_{i=1}^{N_r} |A(\omega_{ir}) - 1| \right) \tag{3.27}$$

Step 5: Calculate the new filter coefficients vector a by using the solution (3.26);

Step 6: Find the alternation frequencies Ω_{il}, Ω_{im} and Ω_{ir} in the passband and return to the third step and repeat until the conditions of (3.28), (3.29), and (3.30) are satisfied:

$$|\Omega_{il} - \omega_{il}| \leq \varepsilon_1, i = 1, \dots, N_l \tag{3.28}$$

$$|\Omega_{im} - \omega_{im}| \leq \varepsilon_1, i = 1, \dots, N_m \tag{3.29}$$

$$|\Omega_{ir} - \omega_{ir}| \leq \varepsilon_1, i = 1, \dots, N_r \tag{3.30}$$

3.3 Optimal Equiripple comb FIR Notch Filter

The Multiple Exchange Algorithm introduced in Section 3.2 can only be used for single-frequency or non-uniform multi-frequency notch filters. But for a uniform multi-frequency notch filter (comb filter), the following is based on Chebyshev polynomials ideal equiripple analysis to implement a linear phase comb filter [22] .

The generating polynomial of the comb FIR filter can be expressed as:

$$F(\omega) = T_n[\lambda T_r(\omega)] = \sum_{m=0}^{nr} B(m)\,\omega_m = \sum_{m=0}^{nr} A(m) T_m(\omega) \tag{3.31}$$

The function $T_m(\omega)$ is a Chebyshev type-I polynomial. The real parameter $\lambda = \frac{1}{\kappa} > 1$ affects the magnitude of the passband ripple, and the order r of the internal Chebyshev polynomial $T_r(\omega)$ determines a total of r narrow bands. The specific positions of the narrow bands are consistent with the pole positions of the internal Chebyshev polynomial:

$$\omega_{mi} = cos\frac{i\pi}{r}, i = 0,1, \dots, r \tag{3.32}$$

In the frequency domain, these narrow bands are evenly distributed in $[0, \pi]$, specifically:

$$\omega_{mi} T = \frac{i\pi}{r}, i = 0,1, \dots, r \tag{3.33}$$

The external Chebyshev polynomial's order n determines the number of poles between the narrow bands is $n - 1$. The generating polynomial of the equiripple comb filter $F(\omega)$ satisfies the differential equations:

$$\begin{aligned} &U_{r-1}(\omega)\left(\kappa^2 - T_r^2(\omega)\right)\left[(1-\omega^2)\frac{d^2F(\omega)}{d\omega^2} - \omega\frac{dF(\omega)}{d\omega}\right] \\ &-r(1-\kappa^2)T_r(\omega)\frac{dF(\omega)}{d\omega} + n^2r^2U_{r-1}(\omega)\left(1 - T_r^2(\omega)\right)F(\omega) = 0 \end{aligned} \tag{3.34}$$

The zero-phase transmission equation of a comb FIR filter obtained from a standardized generator polynomial is:

$$Q(\omega) = 1 - \frac{1 + F(\omega)}{C} = 1 - \frac{T_n[\lambda T_r(\omega)]}{C} = \sum_{m=0}^{nr} b(m)\omega^m = \sum_{m=0}^{nr} a(m)T_m(\omega) \tag{3.35}$$

Under the condition $Q(\omega)|_{\omega=1} = 0$, the constant C is:

$$C = 1 + \cosh[nacosh(\lambda)] = 1 + T_n(\lambda) \tag{3.36}$$

The value of (4.36) is independent of the order r. The formula for evaluating the order of the external $T_n(\omega)$ is:

$$n = \frac{\text{acosh}(k)}{\text{acosh}(\lambda)} = \frac{\ln\left(k + \sqrt{k^2 - 1}\right)}{\ln\left(\lambda + \sqrt{\lambda^2 - 1}\right)} \tag{3.37}$$

where the parameters λ and k are:

$$\lambda = \frac{1}{cos\left(r\frac{\Delta\omega T}{2}\right)}, k = \frac{1 + 10^{0.05a}}{1 - 10^{0.05a}} \tag{3.38}$$

In an actual filter design, the real parameter *n* is rounded up to an even value. Such an approach is to ensure that the specific notch number and width in the design criteria are met. The impulse response $h(m)$ contains $2nr$ coefficients, of which $n + 1$ are non-zero.

Based on the above, an algebraic iterative algorithm for calculating the impulse response can be obtained. The detailed calculation of the impulse response is summarized in Table 2:

Given n (even integer), r (integer), $\lambda > 1$ (real)

Initialization $k = \frac{1}{\lambda}$

$$a(n) = \lambda^n$$

$$a(n+2) = a(n+4) = a(n+6) = 0$$

Body (for $\mu = 1 - n/2$) $a(n-2\mu) = \{a(n-2(\mu-1)) \times [(1-k^2)(n-(2\mu-1))(n-(2\mu-2)) + 3(\mu-1)(n-(\mu-1))] - a(n-2(\mu-2)) \times [(1-k^2)(n-(2\mu-4))(n-(2\mu-6)) + 3(\mu-2)(n-(\mu-2))] + a(n-2(\mu-3))(\mu-3)(n-(\mu-3))\}/\mu(n-\mu)$

(end loop on μ)

$$a(0) = \frac{a(0)}{2}$$

Coefficients $A(m)$ of the

generating polynomial $F(\omega)$

Body ($\mu = 0 - n/2$)

$$A(nr - 2\mu r) = a(n-2\mu)$$

(end loop on μ)

Coefficients $a(m)$ of the zero

phase transfer function $Q(\omega)$

$$C = 1 + T_n(\lambda)$$

$$a(0) = 1 - \frac{1 + A(0)}{C}$$

Body (for $\mu = 1, \dots, nr$)

$$a(\mu) = -\frac{A(\mu)}{C}$$

(end loop on μ)

Coefficients of the

impulse response $h(m)$

$$h(nr) = a(0)$$

Body (for $\mu = 1, \dots, nr$)

$$h(nr \pm \mu) = \frac{a(\mu)}{2}$$

(end loop on μ)

Table 2 Recursive Algorithm for Evaluation of Coefficients h(m) of Impulse Response

The specific implementation steps of the Optimal Equiripple comb FIR filter are:

Step 1: Determine the number of notches r according to the design criteria, the maximum attenuation $\Delta\omega T$ at the passband and the bandwidth $\Delta\omega T$ at the notch frequencies;

Step 2: Determine the internal order r and the internal $T_r(\omega)$ from the first step;

Step 3; Calculate additional parameters λ and k;

Step 4: Use the formula (3.37) to calculate the parameter n;

Step 5: Round up the parameter n to get its next even value;

Step 6: Use the algebraic recursive algorithm shown in Table 3 to calculate the $2nr + 1$ parameters of the impulse response;

Step 7: Calculate the size of the passband ripple:

$$a_{act} = 20log\left(1 - \frac{2}{1+T_n(\lambda)}\right) \tag{3.39}$$

The Optimal Equiripple comb FIR filter Method provides a simple and robust implementation tool for the implementation of an ideal equiripple comb FIR notch filter. However, compared with the same design requirements, the proposed algorithm has a faster algorithm and a smaller passband ripple.

3.4 Iterative Reweighted OMP Method

This method is based on the Orthogonal Matching Pursuit(OMP) algorithm.

Given the notch frequencies $\{\overline{w_i}\}_{i=1}^{K}$, the rejection bandwidths $\{\overline{BW_i}\}_{i=1}^{K}$, and the passband ripple δ, the process of the method begins with the estimation of the initial order $\widehat{N}$ of the filter:

$$\widehat{N} = \max_{i\in\{1,\ldots,K\}} N_i \tag{3.40}$$

From (3.40), the N_i is computed as [24] :

$$N_i = max\{\widehat{N}(\omega_{p1i}, \Delta F_i, \delta_p, \delta_s), \widehat{N}(\omega_{p2i}, \Delta F_i, \delta_p, \delta_s)\} \tag{3.41}$$

where the parameters from (3.41) are expressed as:

$$\Delta F_i = \frac{BW_i}{2} \tag{3.42}$$

$$\omega_{p1i} = \frac{\overline{\omega}_i - \Delta F_i}{2} \tag{3.43}$$

$$\omega_{p2i} = \frac{1 - \overline{\omega}_i - \Delta F_i}{2} \tag{3.44}$$

$$\delta_p = \delta_s = \frac{1 - 10^{\frac{\delta}{20}}}{1 + 10^{\frac{\delta}{20}}} \tag{3.45}$$

Define the coefficients matrix:

$$A = \begin{bmatrix} 1 \cos(\omega_0) & \cdots & \cos(m\omega_0) & \cdots & \cos(M\omega_0) \\ 1 \cos(\omega_1) & \cdots & \cos(m\omega_1) & \cdots & \cos(M\omega_1) \\ \vdots \quad \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 \cos(\omega_l) & \cdots & \cos(m\omega_l) & \cdots & \cos(M\omega_l) \\ \vdots \quad \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 \cos(\omega_L) & \cdots & \cos(m\omega_L) & \cdots & \cos(M\omega_L) \end{bmatrix} \tag{3.46}$$

where $\omega_l = \frac{\pi l}{L}, 0 \leq l \leq L.$

Define the initial $1 \times L$ weight vector:

$$w^{(0)} = (1, \dots, 1) \tag{3.47}$$

And the index set:

$$\Omega = \{0, 1, \dots, M\} \tag{3.48}$$

And the amplitude response vector:

$$f = [H_0(\omega_0), H_0(\omega_1), \dots, H_0(\omega_l), \dots, H_0(\omega_L)]^T \tag{3.49}$$

where the zero-phase $H_0(\omega)$ is defined as:

$$H_0(\omega) = h_M + 2 \sum_{m=1}^{M} h_{M-m} \cos(m\omega) \tag{3.50}$$

So, the impulse response vector is defined as:

$$h = [h_M, 2h_{M-1}, \dots, 2h_m, \dots, 2h_0]^T \quad (3.51)$$

And then the specific implementation steps of the Iterative Reweighted OMP Method are:

Step 1: Normalize the column vectors $\{a_m\}_{m=0}^{M}$ of matrix A, and compute:

$$B^{(k)} = AD^{(k)} = \left(b_0^{(k)}, \dots, b_M^{(k)}\right) \quad (3.52)$$

where $D^{(k)} = diag\left(\left(\frac{1}{\|a_0\|_{l\frac{L}{2}\left(w^{(k)}\right)}}\right), \dots, \left(\frac{1}{\|a_M\|_{l\frac{L}{2}\left(w^{(k)}\right)}}\right)\right)$, and k means it is at the kth iteration.

Step 2: Use the OMP algorithm to solve the problem:

$$\min_h \varepsilon$$

$$s.t. \left\|B^{(k)}y^{(k)} - f\right\|_{l\frac{L}{2}(w)} \leq \varepsilon \quad (3.53)$$

where $y^{(k)} = (D^{(k)})^{-1}h$. And the OMP algorithm is shown in Table 3:

Initialize the residual vector $r_0 = f$ and choose the initial index set Z_0 being an empty set. At the tth OMP iteration with $1 \leq t \leq k$, it proceeds as follows.

1) Find the column vector $b_{m_t}^{(k)}$ $(m_t \in \Omega - Z_{t-1})$ from matrix $B^{(k)}$ of (3.52), which maximizes

The inner product $\left|\langle r_{t-1},\ b_{m_t}^{(k)}\rangle_{l_2^L(w^{(k)})}\right|$.

2) Utilize the chosen m_t to compare the set Z_t as:

$$Z_t = Z_{t-1} \cup \{m_t\}$$

And the solution $z_t^{(k)}$:

$$z_t^{(k)} = arg \min_z \left\| f - \phi_t^{(k)} z \right\|_{l_2^L(w^{(k)})}$$

where $\phi_t^{(k)} = \left(b_{m_1}^{(k)}, \dots, b_{m_t}^{(k)}\right)$.

3) Compute the new residual vector r_t as:

$$r_t = f - \phi_t^{(k)} z_t^{(k)}$$

4) If $t \leq k-1$, replace t with $t+1$ and repeat 1) to 3).

If $t = k$, define the residual vector $r^{(k)}$ and index set $Z^{(k)}$ for the kth iteration as:

$$r^{(k)} = r_t, \quad Z^{(k)} = Z_t$$

And proceed to 3)

Table 3 The OMP Algorithm

Step 3: Set $c(\omega) = [1, \cos(\omega), \dots, \cos(m\omega), \dots, \cos(M\omega)]$, and solve the linear programming problem:

$$\min_{h,\mu} \mu$$

$$s.t. |c(\omega_l)h - 1| \leq \frac{1 - 10^{\frac{\delta}{20}}}{1 + 10^{\frac{\delta}{20}}} + \mu, l \in \{l|\omega_l \in \Omega^1\}$$

$$c(\bar{\omega}_i)h = 0, i = 1, \dots, K$$

$$h_{M-m} = 0, m \notin Z^{(k)} \tag{3.54}$$

If $\mu < 0$, then the computed vector h is the linear-phase FIR multi-frequency notch filter that meets the design specification.

Step 4: Compute the new weight vector $w^{(k+1)}$ which is expressed as:

$$w_i^{(k+1)} = \frac{1}{\left(1 + \left(\frac{r_i^{(k)}}{\epsilon}\right)^2\right)^{\frac{1}{4}}}, 0 \leq i \leq L \tag{3.55}$$

where $r_i^{(k)}$ is the ith entry of the residual vector $r^{(k)}$ and $\epsilon = max\left|y^{(k)}\right|/dp$ with dp being a constant, and we set $dp = 100$ here.

Replace $w^{(k)}$ with $w^{(k+1)}$, then repeat from step 1 to step 3.

IV. Quadratic Programming Method

4.1 Definition

We now describe the proposed QP-based notch filter. Suppose that we would like to approximate a noisy signal vector x subject to a finite number of constraints on the discrete-time Fourier transform of x. This will lead to the following approximation problem:

$$\min(x-\hat{x})^T Q(x-\hat{x})$$

$$s.t. \left|\hat{X}(\omega_k)\right| = 0, k = 1, \dots, M \tag{4.1}$$

where $x = [x(0) \dots x(N-1)]^T$ is the noisy signal vector, $\hat{x} = [\hat{x}(0) \dots \hat{x}(N-1)]^T$ is the estimate of x, Q is an $N \times N$-dimentional positive definite matrix, and $\hat{X}(\omega)$ is the discrete-time Fourier transform of $\hat{x}$.

This is a quadradic program with linear equality constraints, which has a closed-form solution.

As $A_n x = 0$, the constraints matrix is expressed as a $2M \times N$ matrix:

$$A_n = \begin{bmatrix} \cos(n\omega_1) & \cos\big((n+1)\omega_1\big) & \cos((n+2)\omega_1) & \cdots \cos((n+N-1)\omega_1) \\ \sin(n\omega_1) & \sin\big((n+1)\omega_1\big) & \sin((n+2)\omega_1) & \cdots \sin((n+N-1)\omega_1) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \cos(n\omega_M) & \cos\big((n+1)\omega_M\big) & \cos((n+2)\omega_M) & \cdots \cos((n+N-1)\omega_M) \\ \sin(n\omega_M) & \sin\big((n+1)\omega_M\big) & \sin((n+2)\omega_M) & \cdots \sin((n+N-1)\omega_M) \end{bmatrix} \tag{4.2}$$

with $n = 0,1,2, \dots$.

Here, we use the first order Karush-Kuhn-Tuncker conditions to solve the approximation problem in (4.1), these can be expressed as:

$$\begin{bmatrix} Q & -A_n^T \\ A_n & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix} \tag{4.3}$$

where λ is a vector containing the Lagrange multipliers for the quadratic program.

So, we get:

$$\hat{x}_n = [Q^{-1} - Q^{-1}A_n^T(A_nQ^{-1}A_n^T)^{-1}A_nQ^{-1}]x_n \quad (4.4)$$

The equation (4.4) shows that $\hat{x}$ is the projection of x onto the subspace spanned by the rows of A_n.

When $M = 1$ and $Q = I$, (4.4) this will represent a notch filter which has zero gain at the frequency ω_1. And then the notch filter can be efficiently implemented by first computing $y_n = A_n x_n$, followed by $\hat{x}_n = A_n^T(A_nA_n^T)^{-1}y_n$.

If the length of the input vector N is low, there will be errors in the vicinity of the notch frequency. This is the result of leakage due to truncating the basis vectors used in the projection matrix. To address this, we zero-pad x as follows:

$$x \leftarrow [\ zeros(1, n_p)\ \ x\ \ zeros(1, n_p)\ ] \quad (4.5)$$

The length of x is now $N' = N + 2n_p$.

Zero padding makes it possible to use arbitrarily long basis vectors in the constraint matrix A_n. This has the effect of reducing the notch bandwidth while reducing the error in the portion of the passband in the immediate area of the notch frequencies. This effect can be seen in Figure 3, where a length N=32 pseudorandom white noise sequence was subjected to a quadratic programming notch filter having three notch frequencies of 60Hz, 180Hz and 300Hz with no zero padding. A considerable difference between the input and output around the notch frequencies can be seen. In Figure 4, a much closer fit between the filtered and unfiltered signal is seen using zero padding 128.

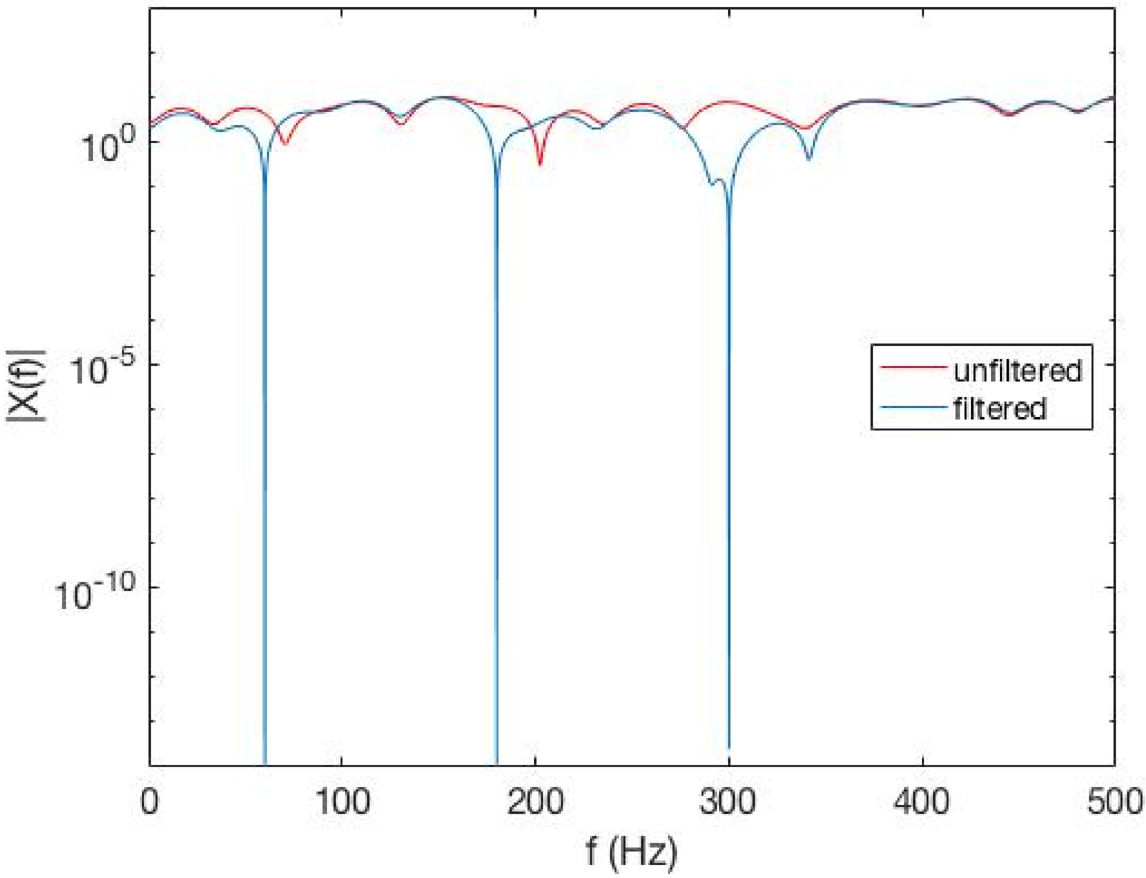

Figure 3 N=N'=32 zero padding (np)=0

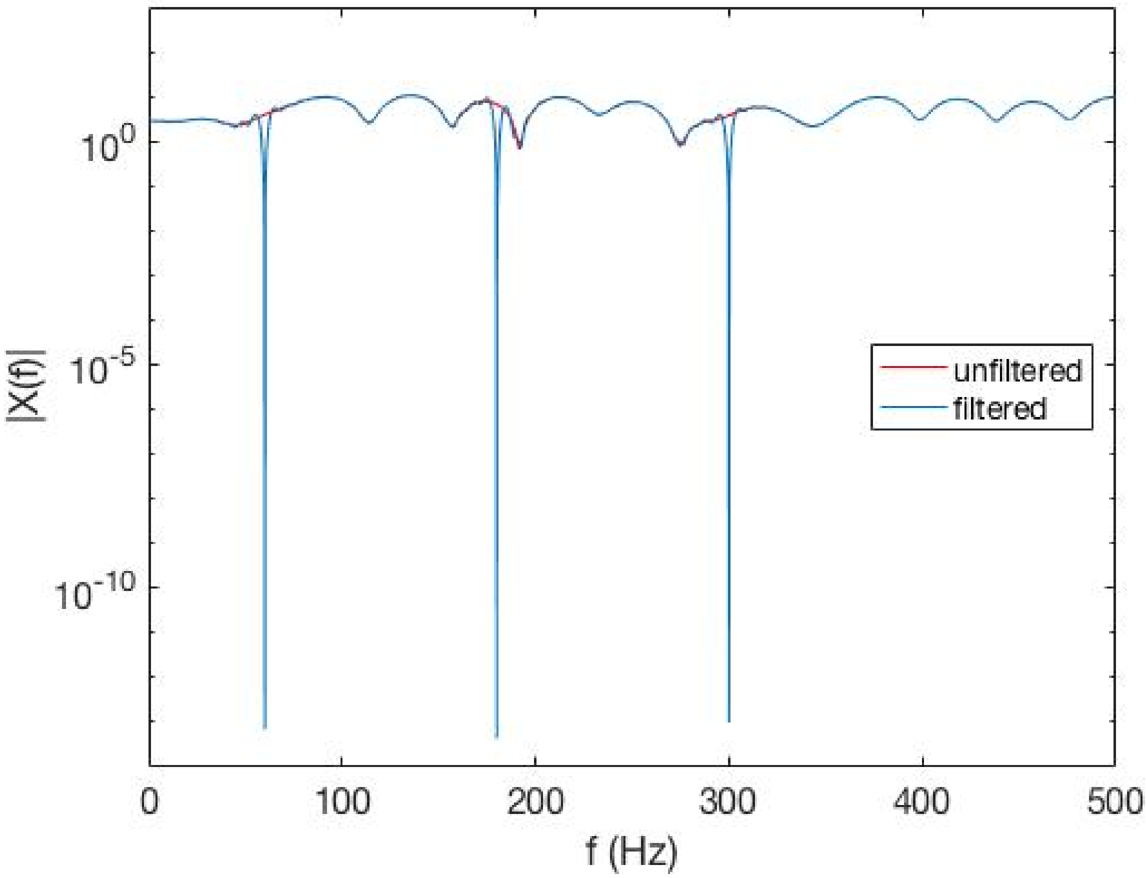

Figure 4 N=32 N'=32+128*2=288 zero padding (np)=128

The comparison of different amounts of zero padding is shown in Figure 5.

As we described above, the projection is defined as:

$$\hat{x}_n = [Q^{-1} - Q^{-1}A_n^T(A_nQ^{-1}A_n^T)^{-1}A_nQ^{-1}]x_n = [I - A_n^T(A_nA_n^T)^{-1}A_n]x_n \quad (4.6)$$

where Q has been replaced by the identity matrix and $x_n = [x(n)\ x(n+1)\ ...\ x(n+N-1)]^T$ is the noisy signal vector.

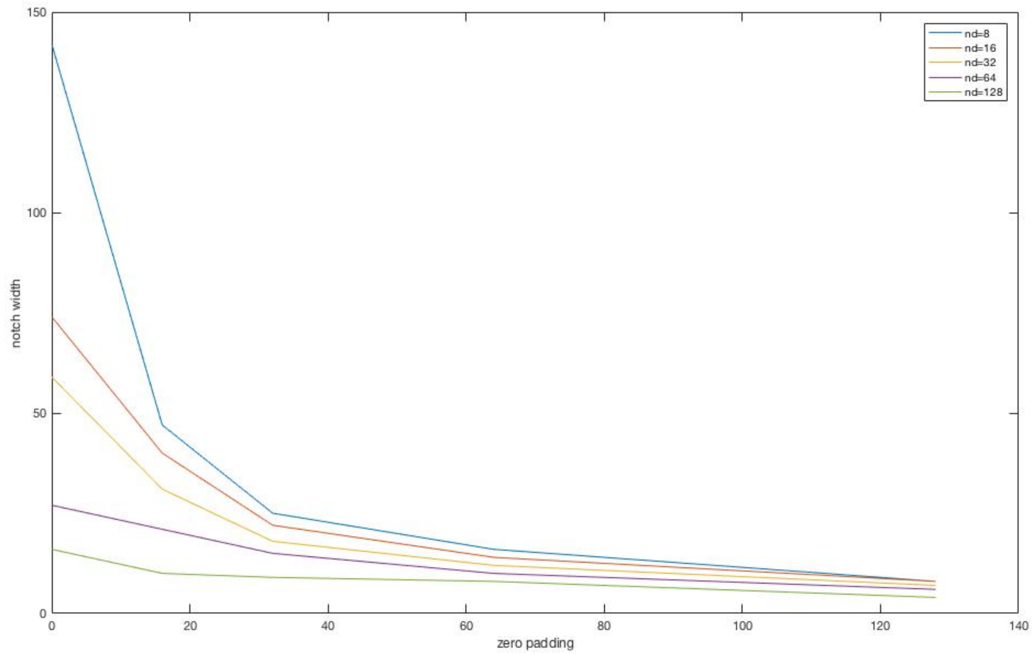

Figure 5 Comparison of Different Zero Padding

5.1 Results for Multiple Exchange Algorithm

The Multiple Exchange algorithm can achieve a single-frequency notch filter with relatively small ripple, while for a multi-frequency notch filter, when the interval between notch frequencies is small, a larger filter order is required. To ensure the convergence of the algorithm, such as the dual-frequency notch filter, when the frequencies are $200Hz$ and $300Hz$, the order should be 200 to ensure the convergence of the algorithm to obtain a better solution.

The result for Multiple Exchange Algorithm of two frequencies 200Hz and 800Hz is shown in Figure 6:

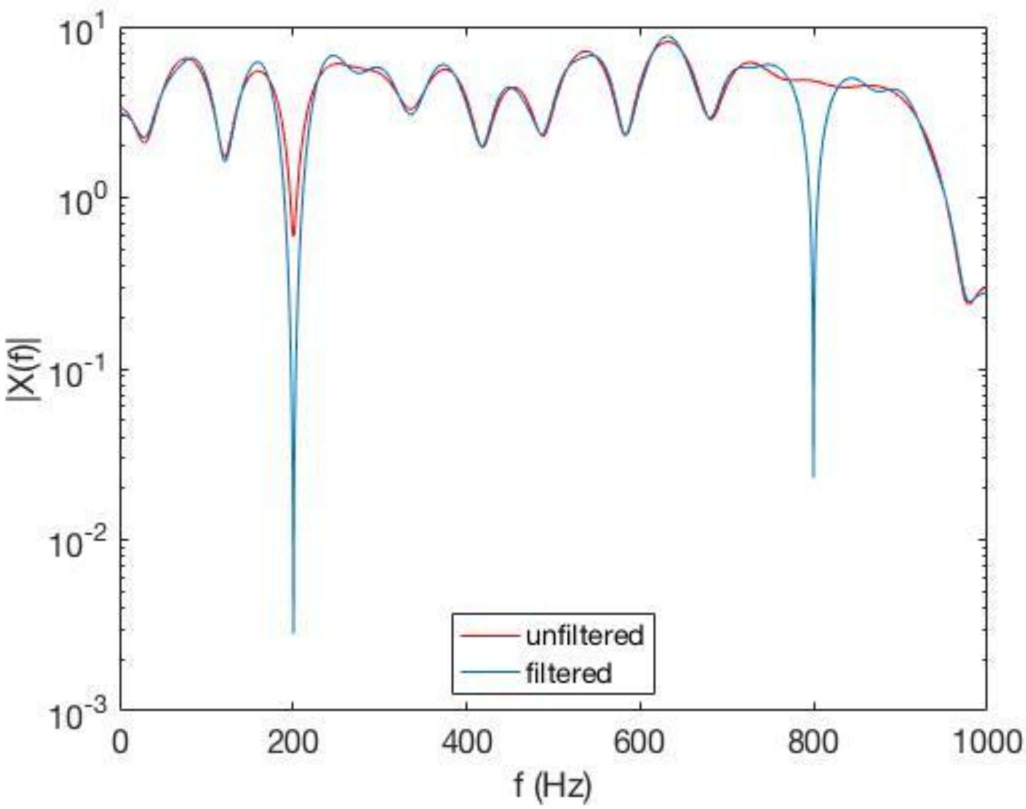

Figure 6 Result for Multiple Exchange Algorithm of 200Hz and 800Hz

And the frequency response of ME is shown in Figure 7:

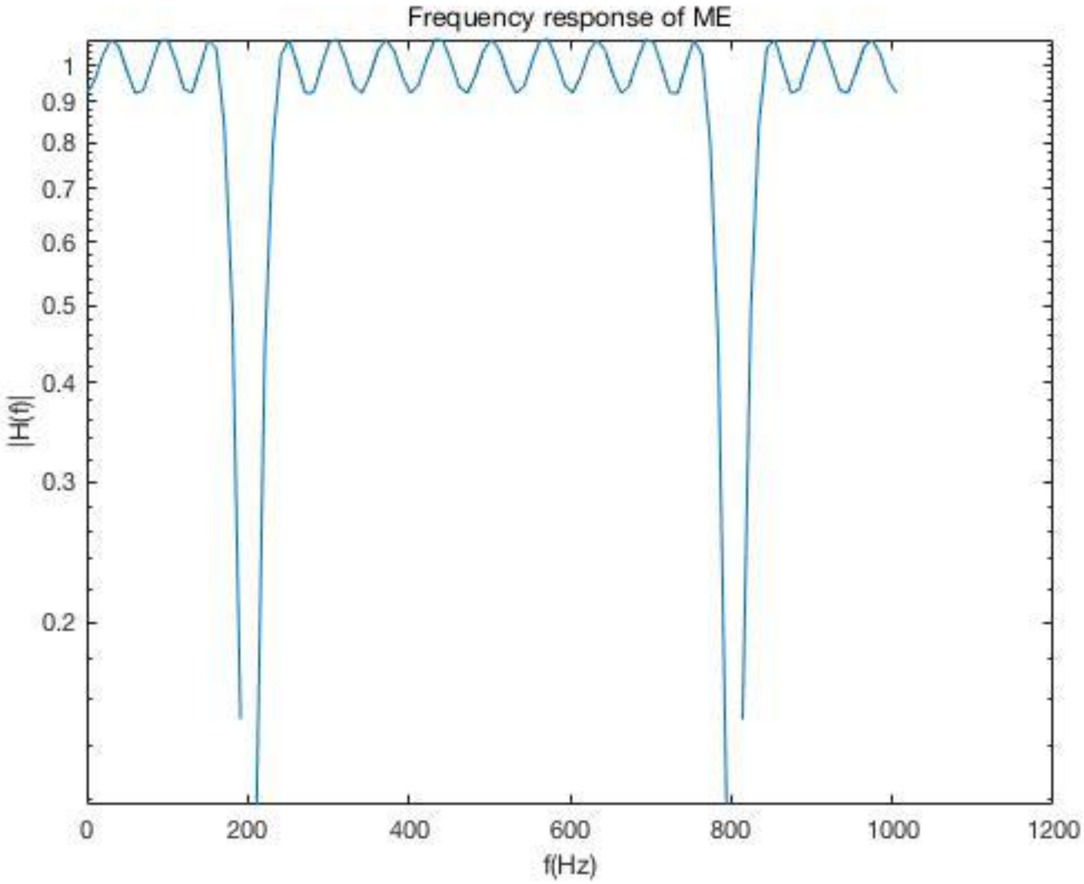

Figure 7 Frequency Response of ME Method

As we can see from Figure 7, at the notch frequencies, the ME method has negative gains which is a disadvantage of the ME algorithm.

5.2 Results for Iterative Reweighted OMP Method

When we set the notch frequencies $100Hz$, $250Hz$ and $760Hz$, their associated rejection bandwidths $\overline{BW_I} = 0.061\pi$ for $i = 1,2,3$ and passband ripple $\delta = -0.95dB$, the result for the Iterative Reweighted OMP Method is shown in Figure 8:

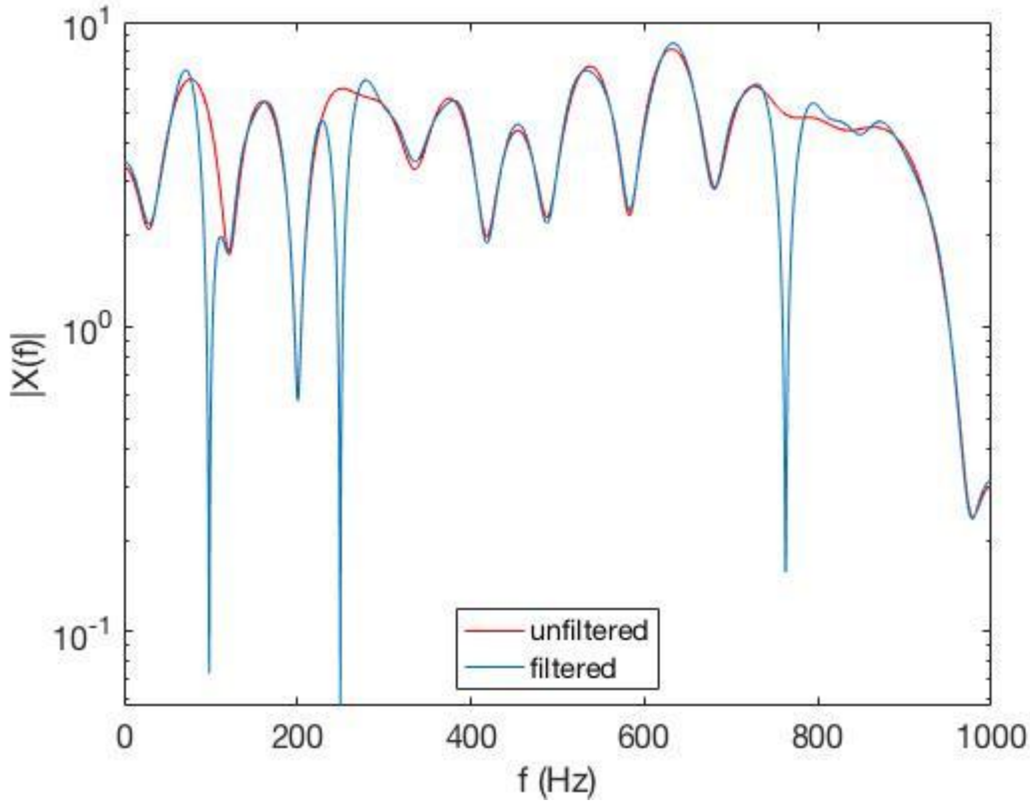

Figure 8 Result for Iterative Reweighted OMP Method

And the frequency response of OMP is shown in Figure 9:

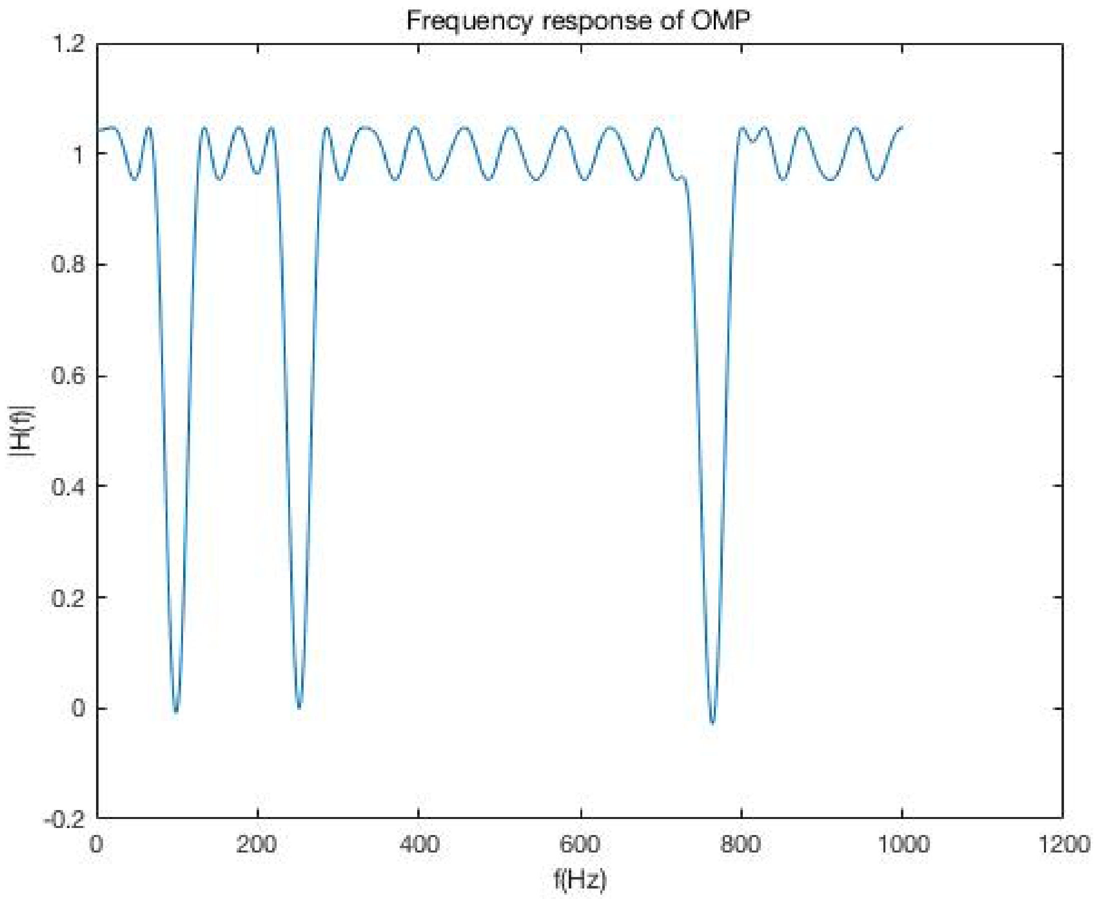

Figure 9 Frequency Response of OMP

As we can see from the result, the Iterative Reweighted OMP Method can achieve the non-uniform multi-frequency notch filter.

5.3 Results for Quadratic Program Method

The result for Multiple Exchange Algorithm of two frequencies 200Hz and 800Hz is shown in Figure 10:

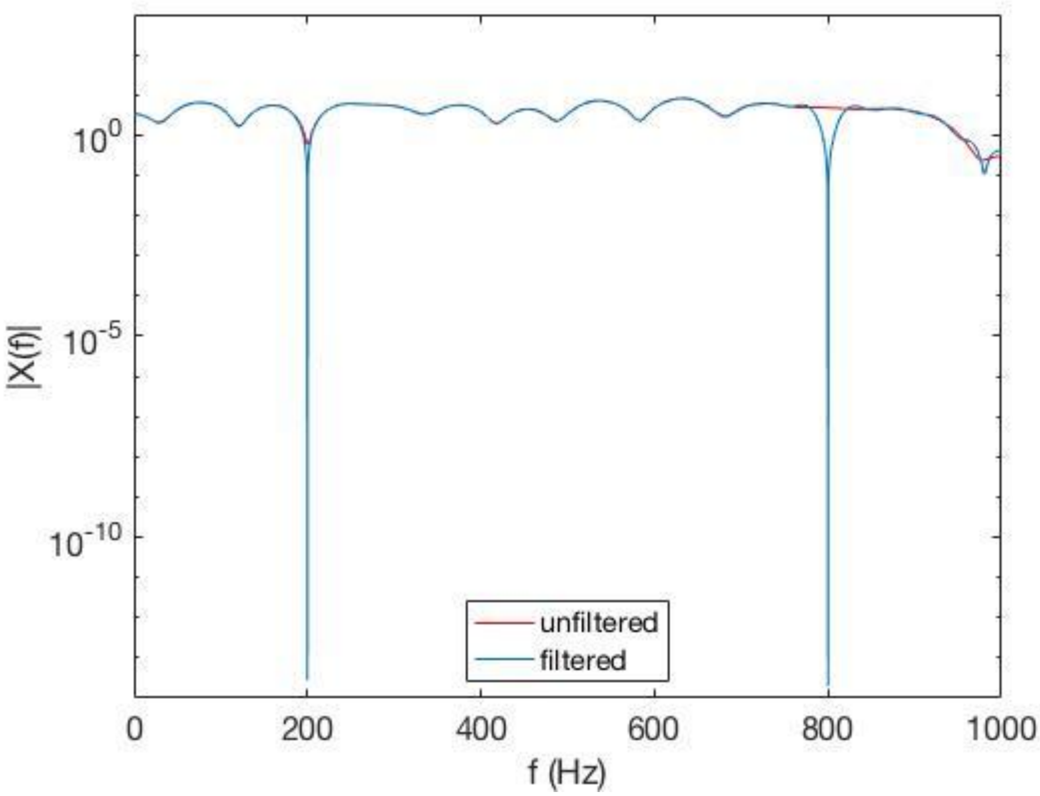

Figure 10 Results for QP Method

And the frequency response of QP is shown in Figure 11:

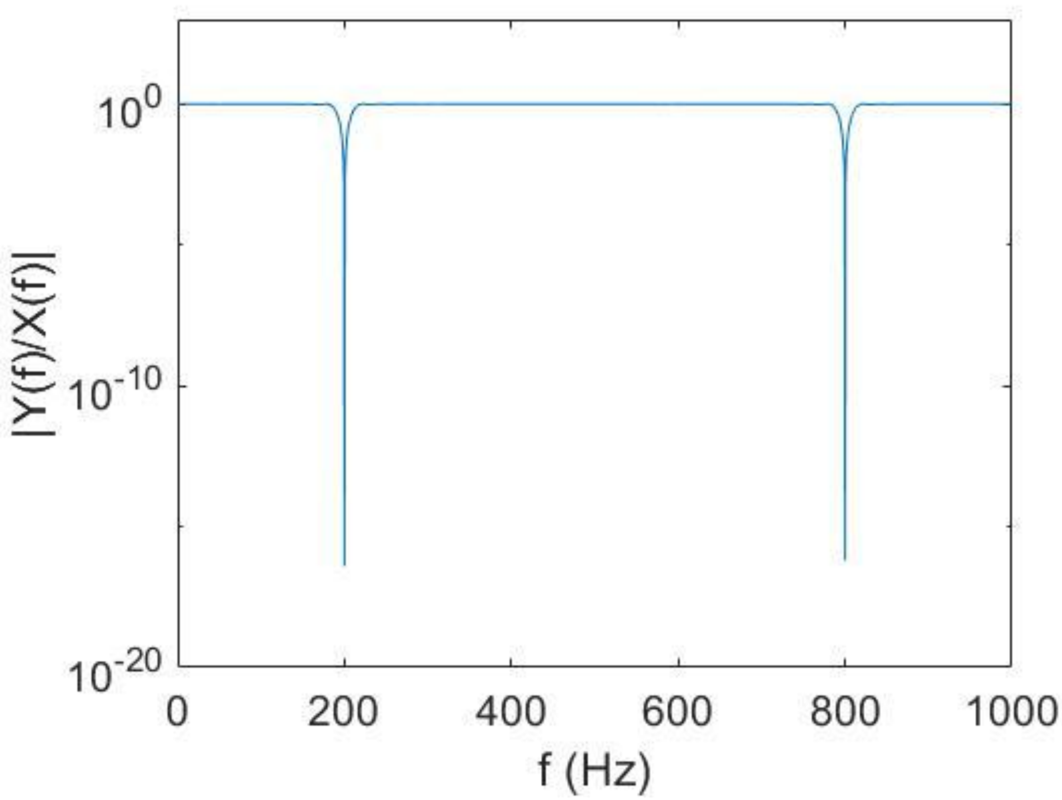

Figure 11 Frequency Response of QP

And the phase is shown in Figure 12:

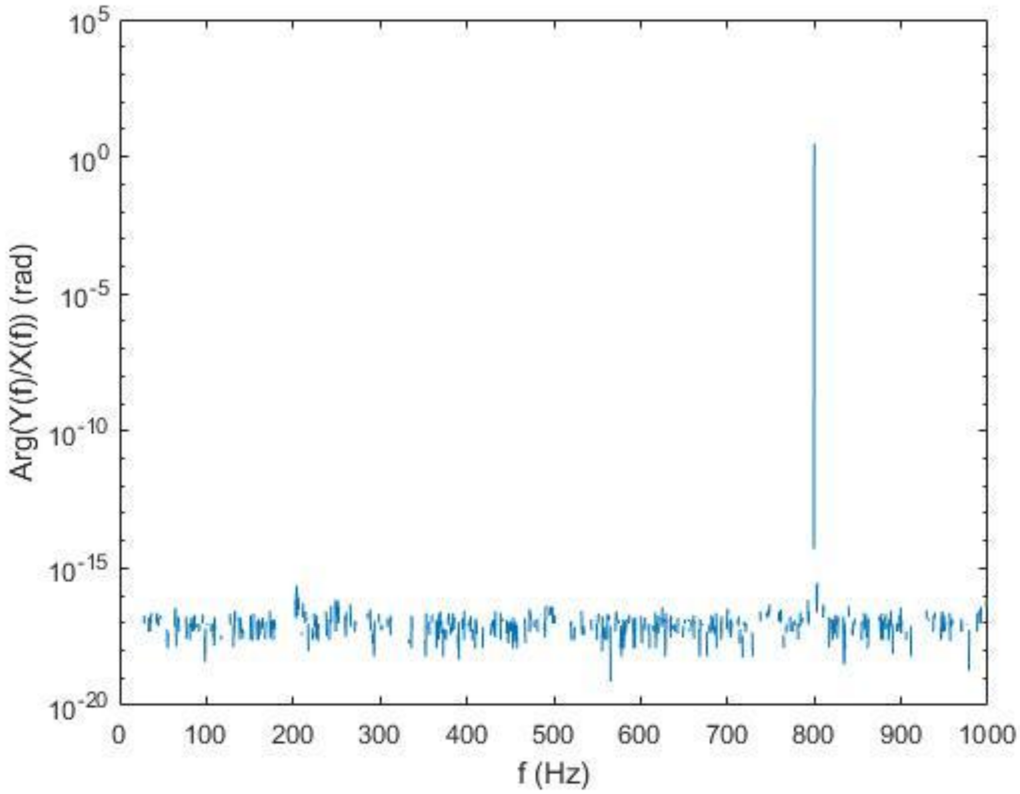

Figure 12 Phase of QP Method

5.4 Comparison between QP Method with Other Methods

(1) QP method compared with the Multiple Exchange Algorithm

When we set the notch frequencies as 200Hz and 800Hz, the comparison is shown in Figure 13:

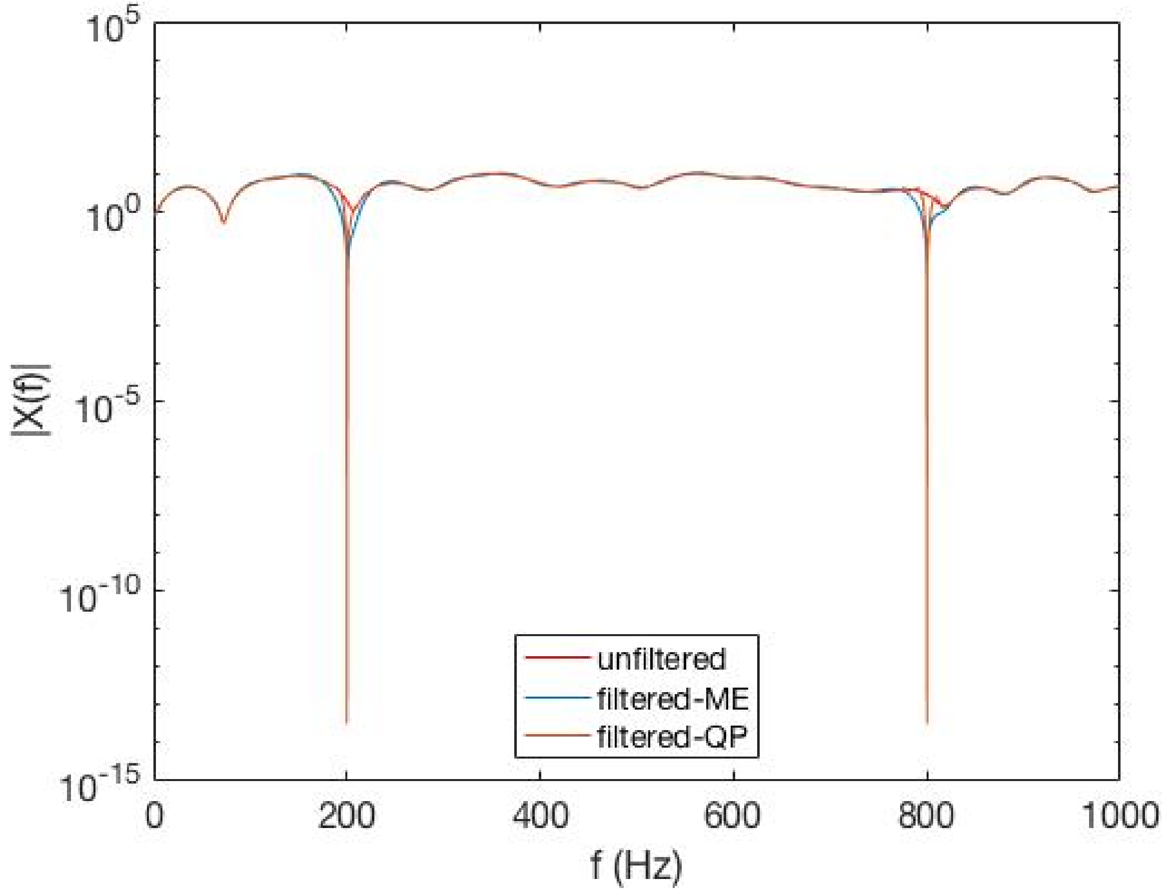

Figure 13 Comparison Between QP Method and ME Method

And when we zoom up, the part of notch is shown in Figure 14:

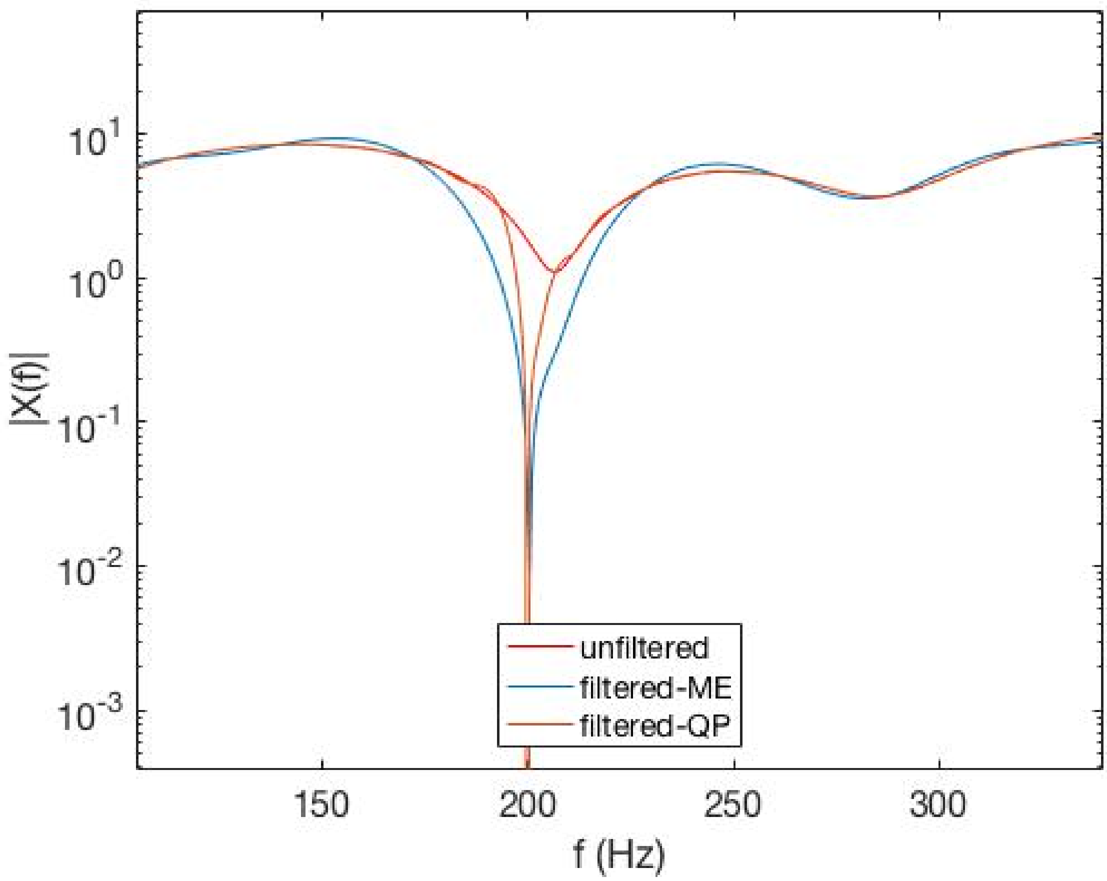

Figure 14 Comparison between QP method and ME method (Zoom up)

Using the freqz function, the magnitude and phase of the ME are shown in Figure 15:

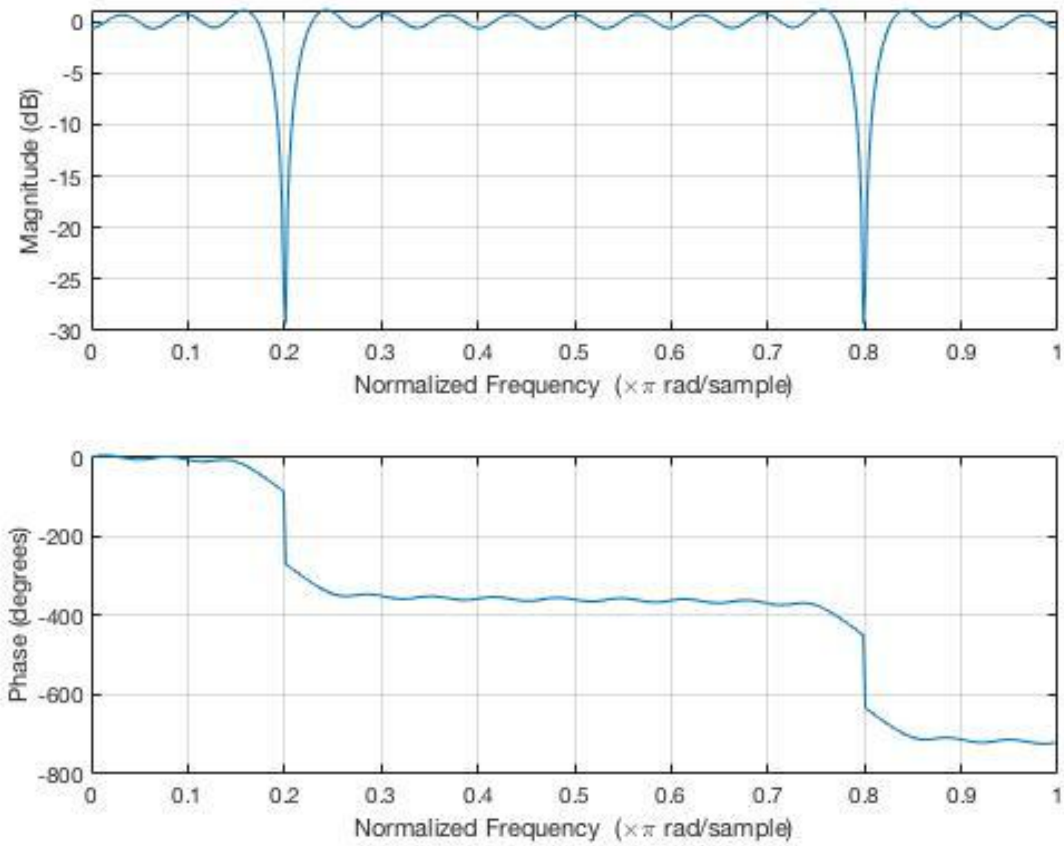

Figure 15 Magnitude and Phase of ME (200Hz and 800Hz)

Also, we can get the frequency response of QP method which is shown in Figure 16:

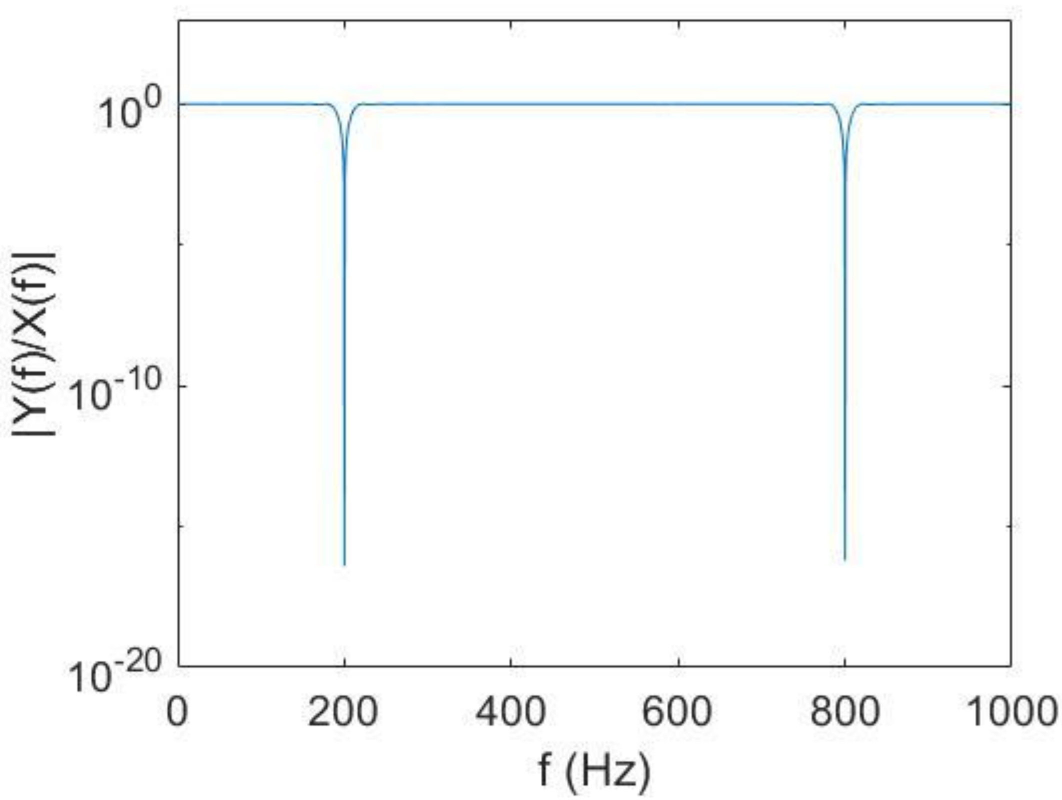

Figure 16 Frequency Response of QP Method (200Hz and 800Hz)

And the phase is shown in Figure 17:

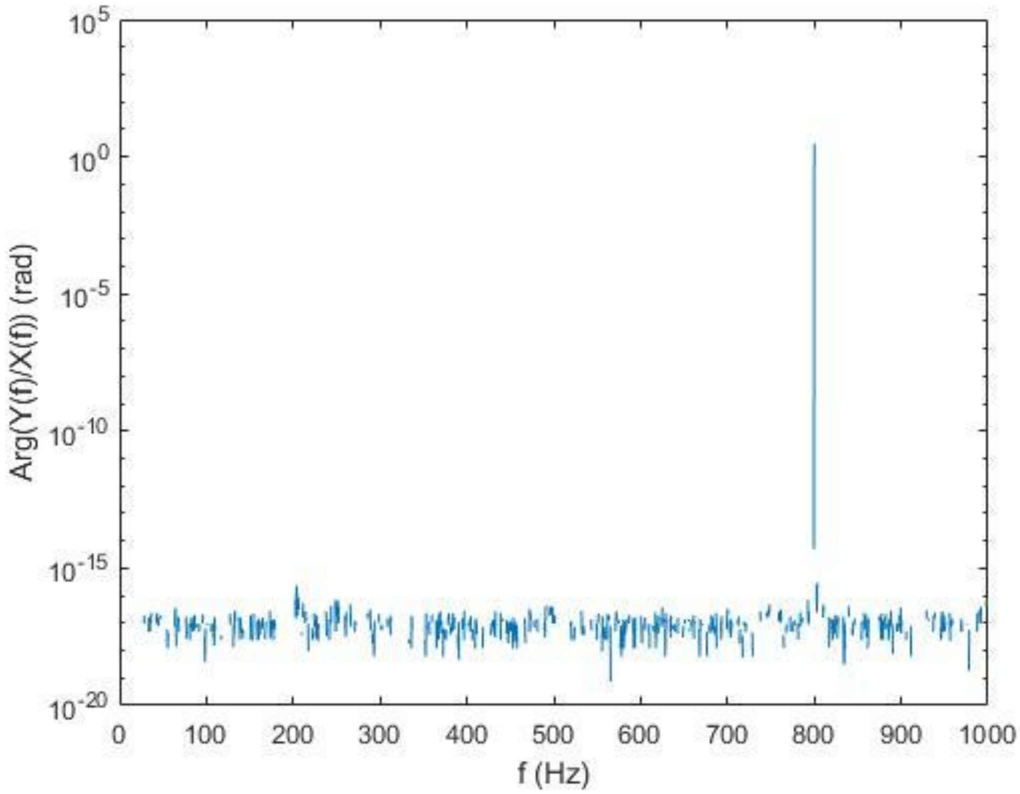

Figure 17 Phase of QP Method

As we can see from Figure 13, at the same situation, the QP Method has narrower stop band than the Multiple Exchange Algorithm. And from Figure 14, at the same situation, the ripple in the ME method is more than the ripple in the QP method. So, the QP method can do better for the ripple problem. And the phase of ME from Figure 15 seems not completely linear.

(2) QP method compared with the Optimal Equiripple Comb Method

The Optimal Equiripple Comb Method can only filter uniform notch frequencies, but the Quadratic Programming Method can efficiently filter non-uniform frequencies. In most actual situations, there are non-uniform frequencies, so the Optimal Equiripple Comb Method will not be available in common situations.

(3) QP method compared with the Iterative Reweighted OMP Method

When we set the notch frequencies as 100Hz, 250Hz and 760Hz, the comparison between the QP Method and the Iterative Reweighted OMP Method is shown in Figure 18:

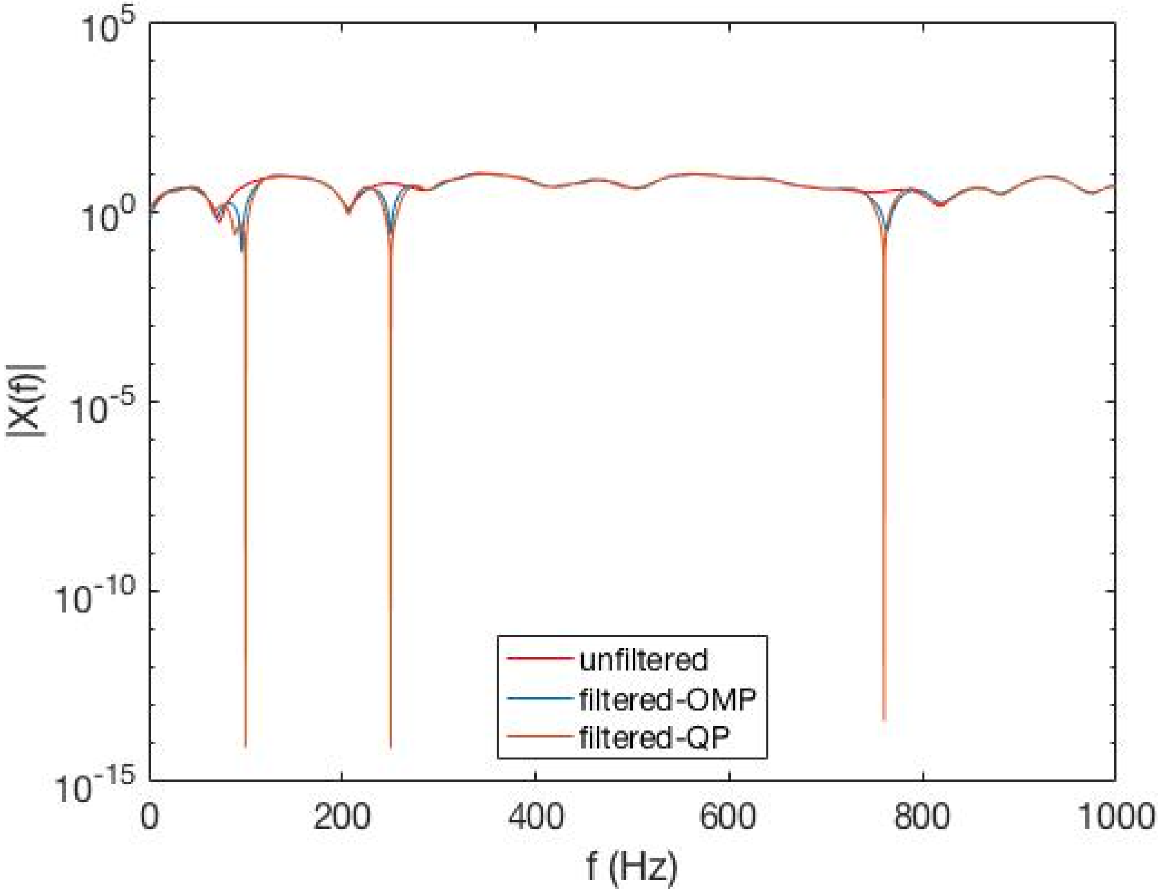

Figure 18 Comparison Between QP Method and OMP Method (1)

And when we zoom up, the part of notch is shown in Figure 19:

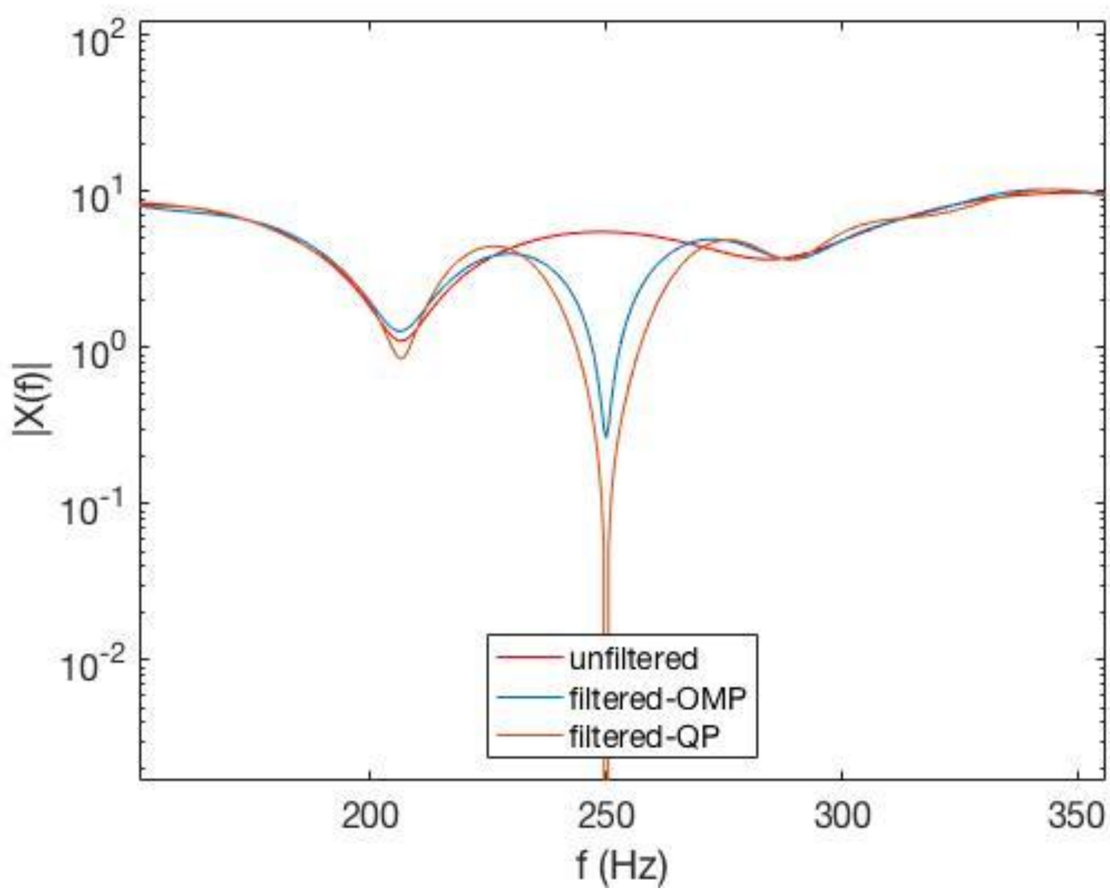

Figure 19 Comparison between OMP method and QP method (zoom up) (1)

Using the freqz function, the magnitude and phase of the OMP are shown in Figure 20:

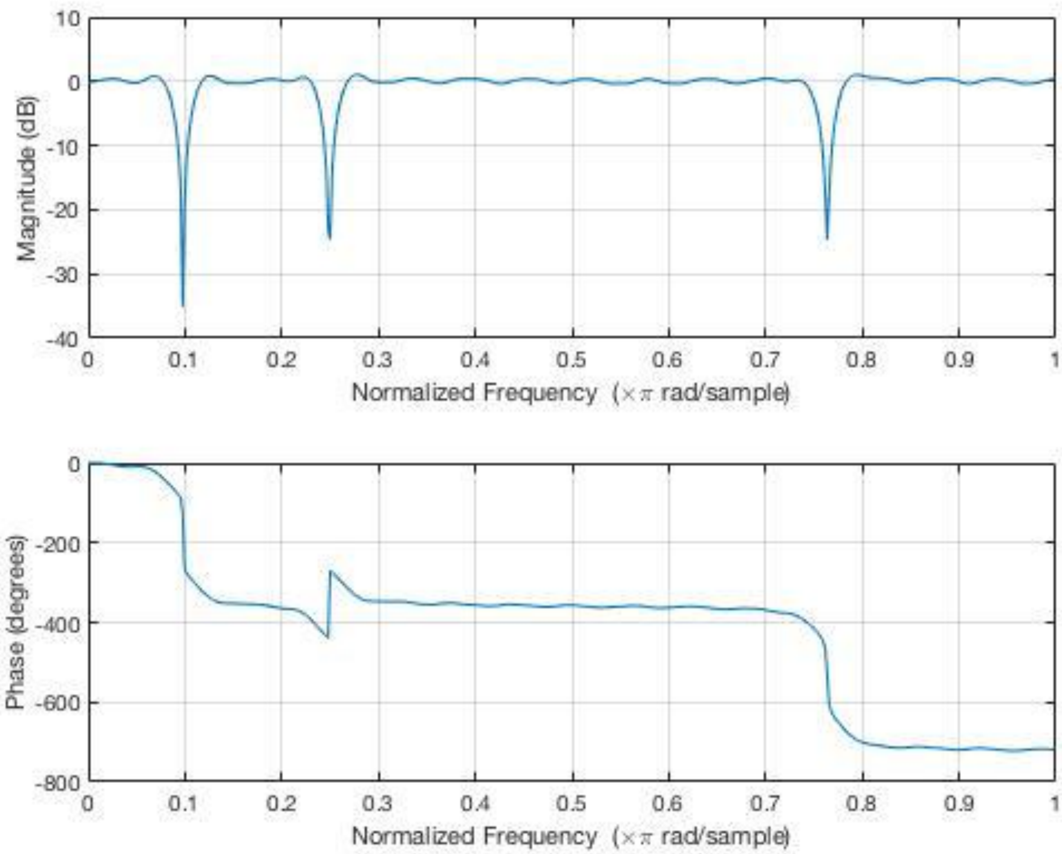

Figure 20 Magnitude and Phase of OMP (100Hz, 250Hz and 760Hz)

Also, we can get the frequency response of QP method which is shown in Figure 21:

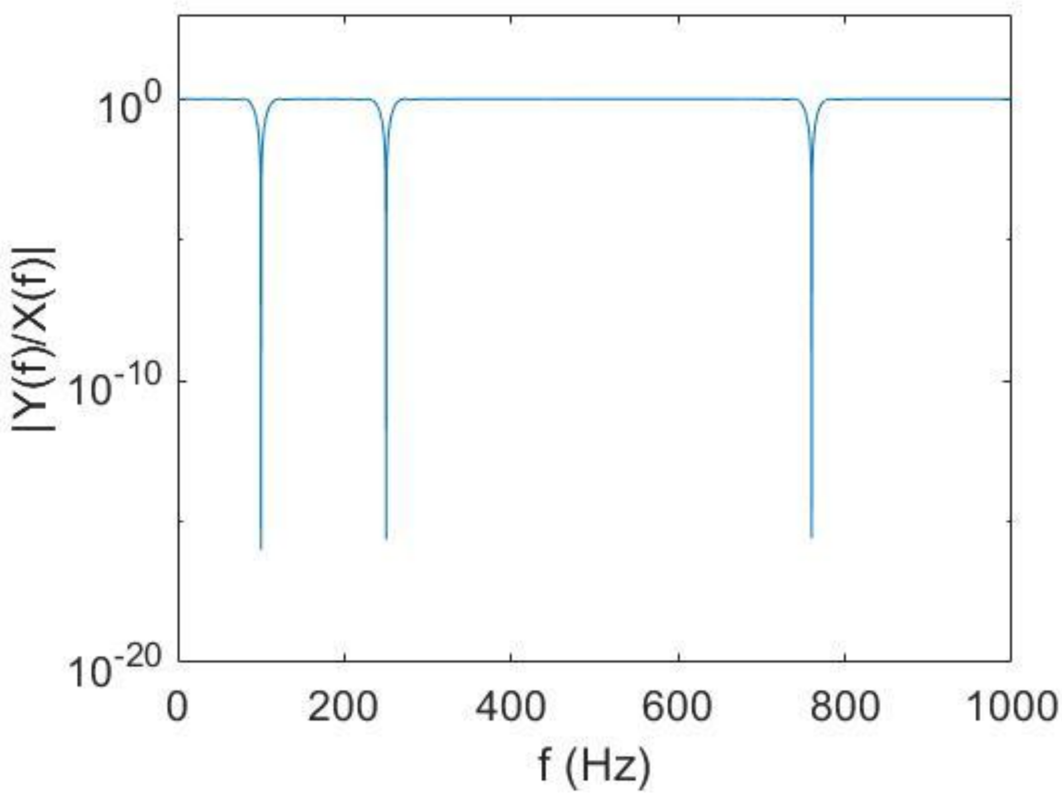

Figure 21 Frequency Response of QP Method (100Hz, 250Hz and 760Hz)

And the phase is shown in Figure 22:

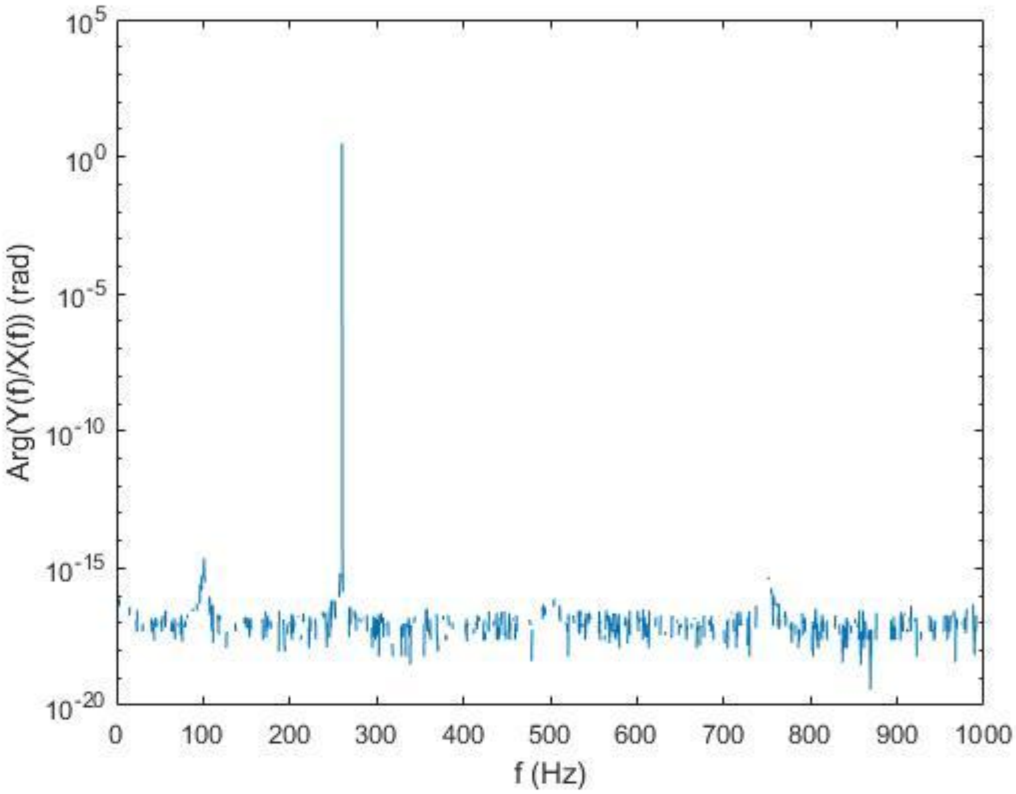

Figure 22 Phase of QP Method

The phase of OMP from Figure 21 seems not completely linear.

And then, we set the frequency are 200Hz and 800Hz, the result is shown in Figure 23:

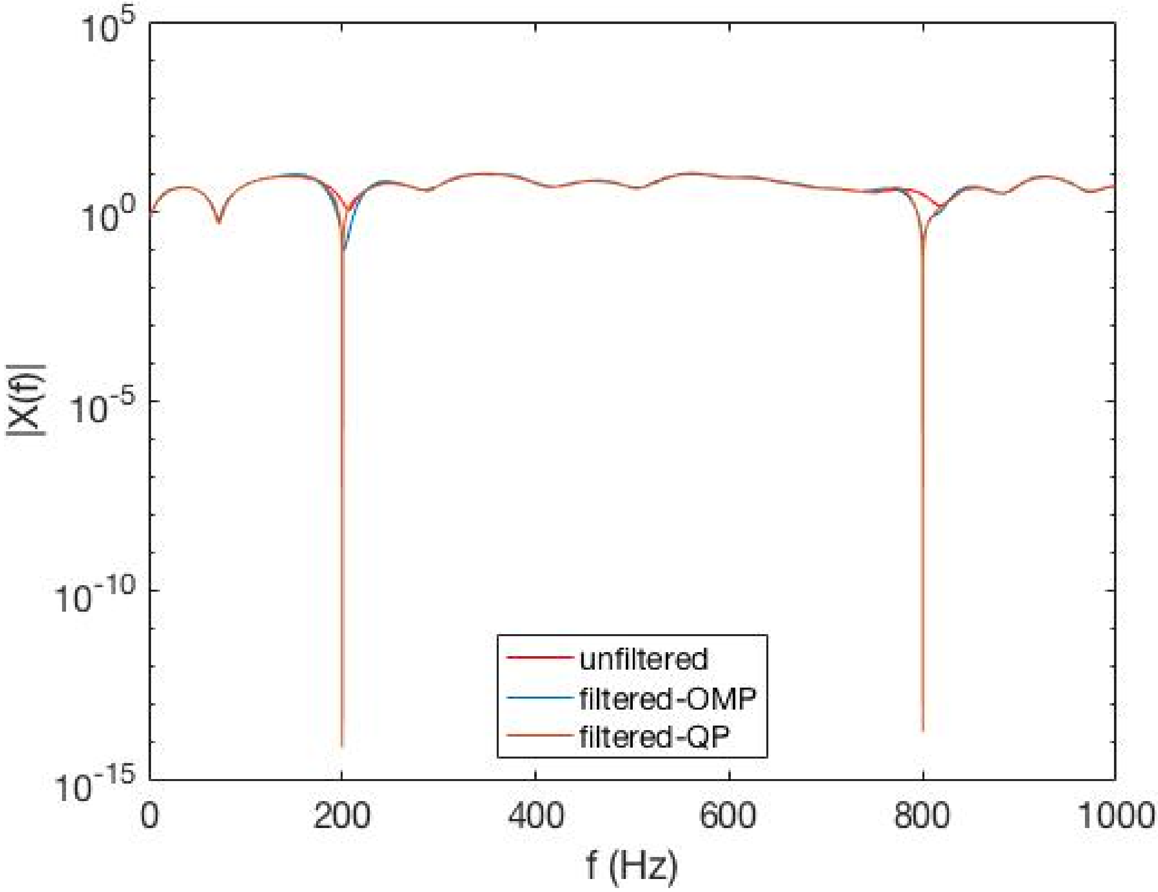

Figure 23 Comparison Between QP Method and OMP Method (2)

And when we zoom up, the part of notch is shown in Figure 24:

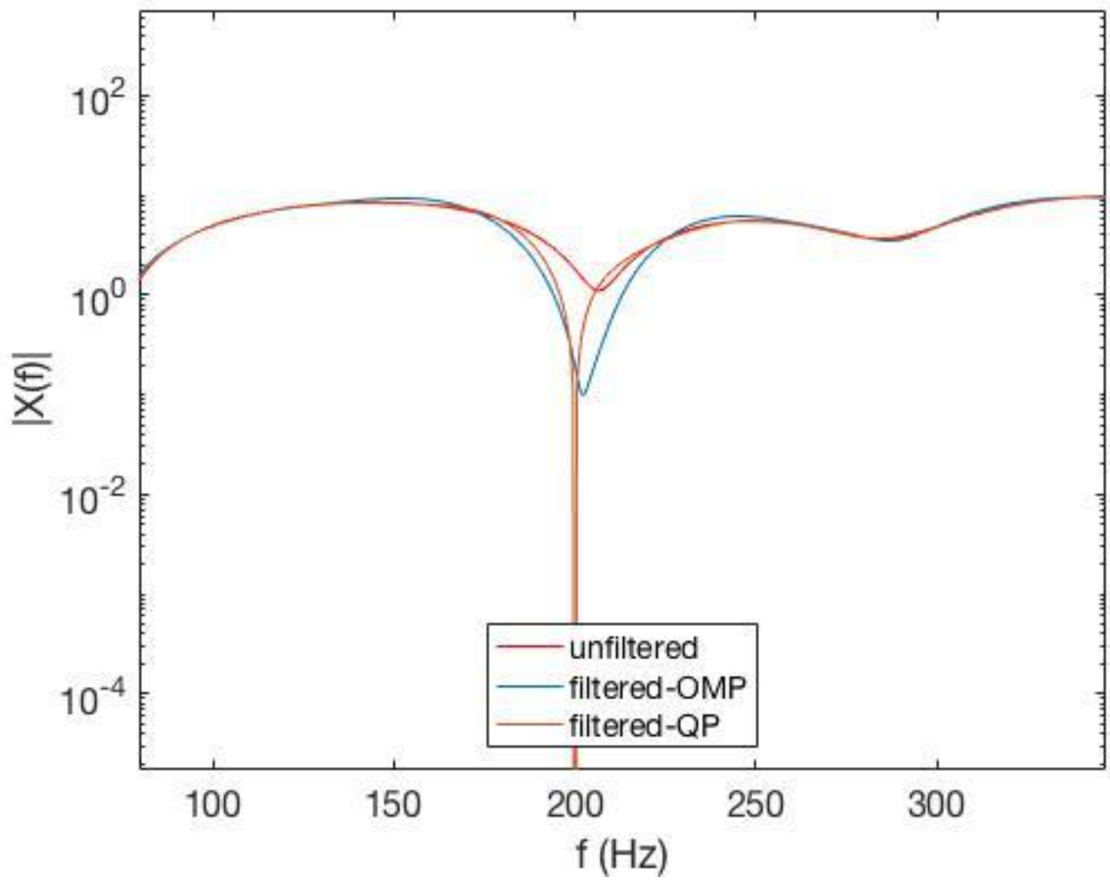

Figure 24 Comparison between OMP method and QP method (zoom up) (2)

Using the freqz function, the magnitude and phase of the OMP are shown in Figure 25:

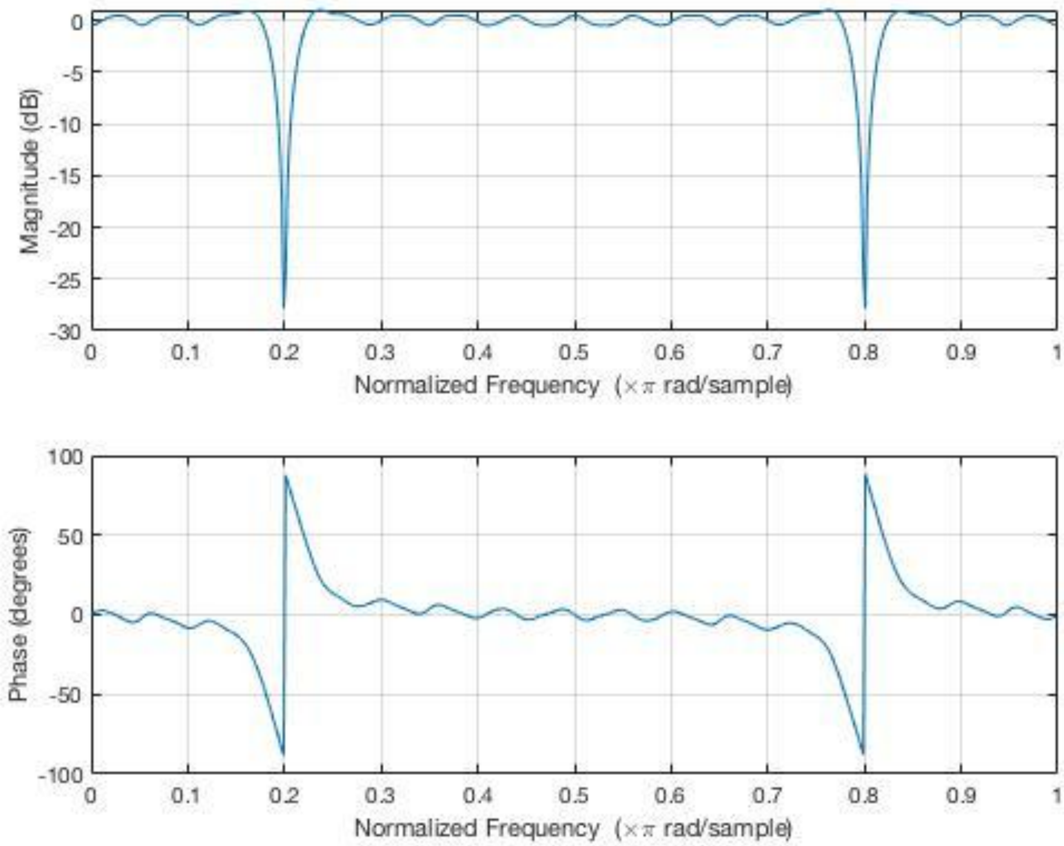

Figure 25 Magnitude and Phase of OMP (200Hz and 800Hz)

Also, we can get the frequency response of QP method which is shown in Figure 26:

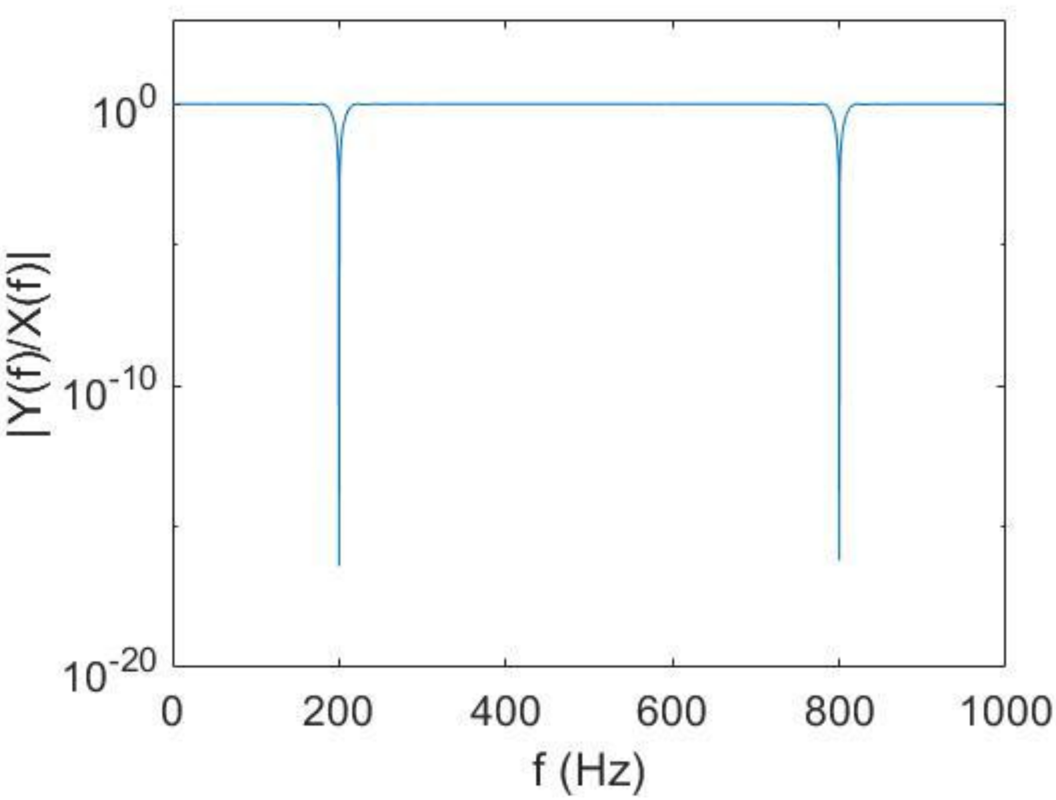

Figure 26 Frequency Response of QP Method (200Hz and 800Hz)

And the phase is shown in Figure 27:

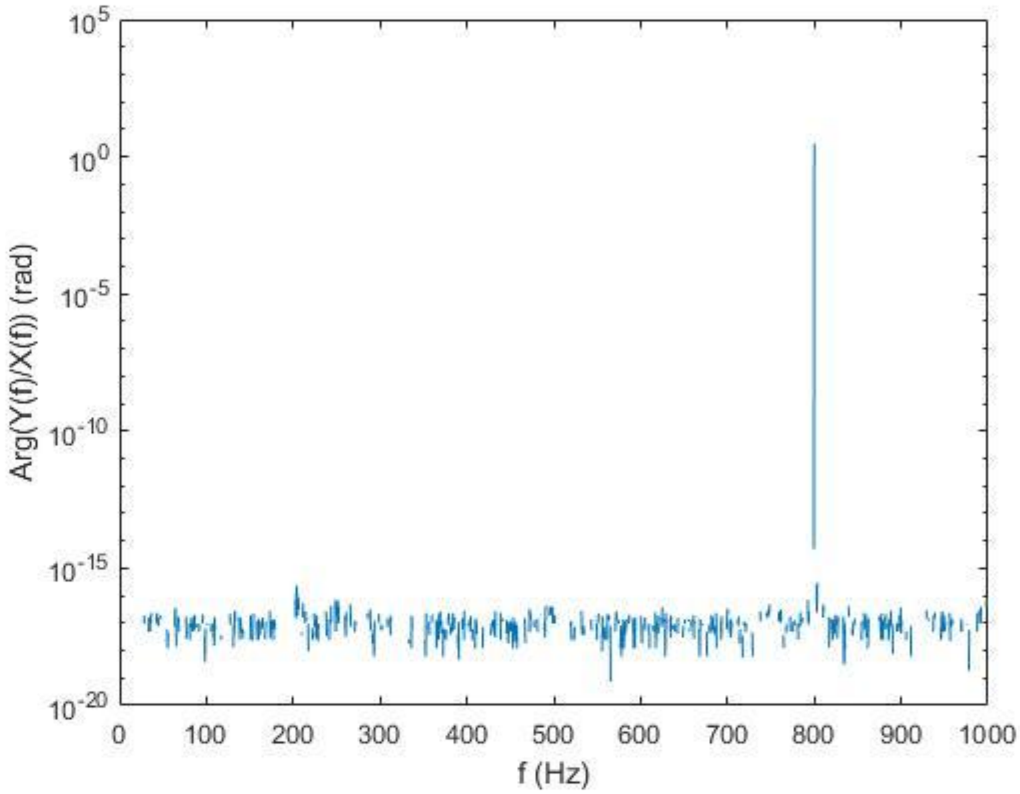

Figure 27 Phase of QP Method

And then, we set 5 notch frequencies at 100Hz, 250Hz, 420Hz, 660Hz and 830Hz, and the result is shown in Figure 28:

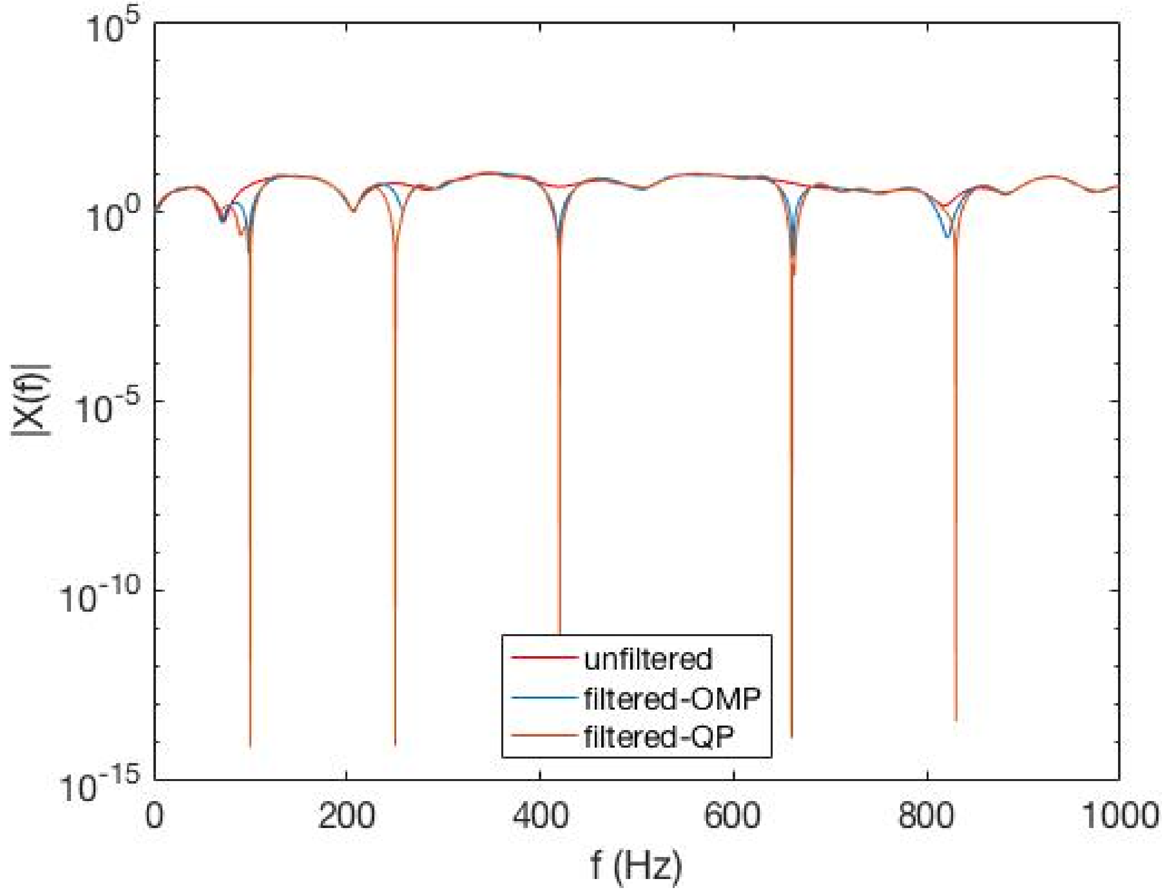

Figure 28 Comparison Between QP Method and OMP Method (3)

And when we zoom up, the part of notch is shown in Figure 29:

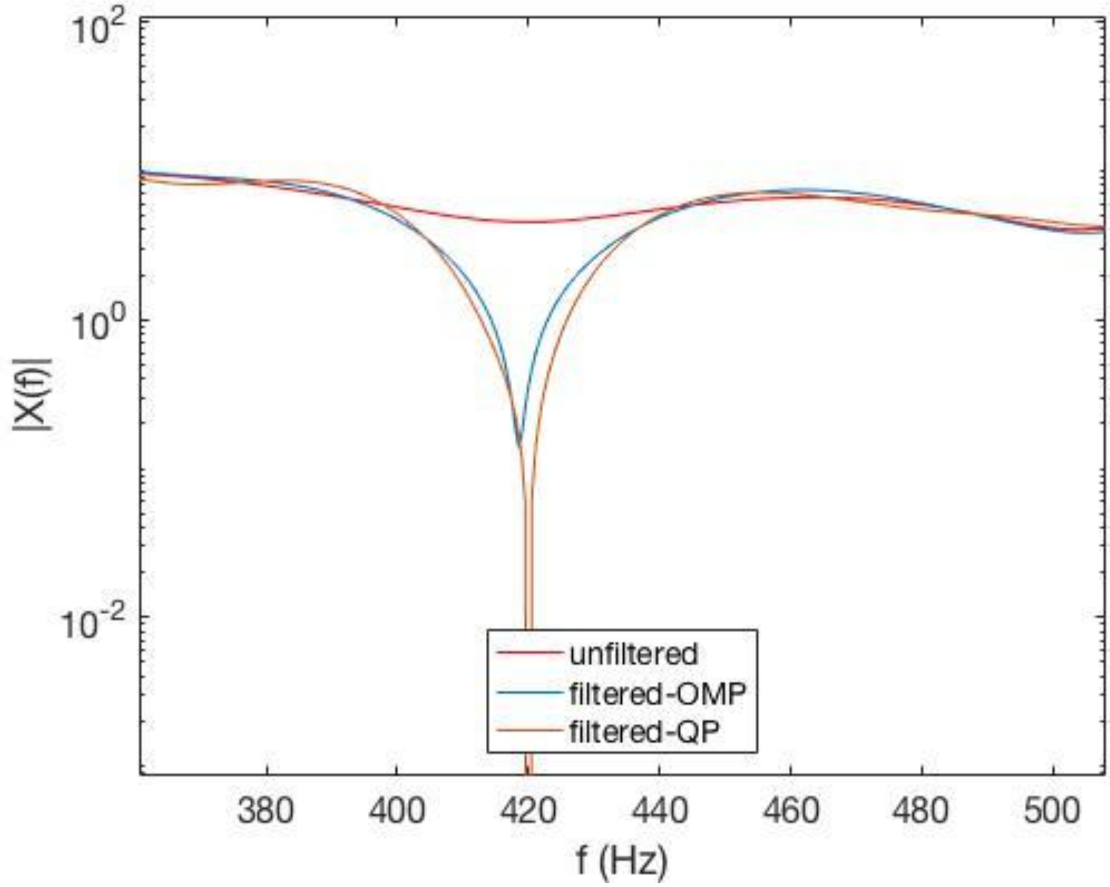

Figure 29 Comparison between OMP method and QP method (zoom up) (3)

Using the freqz function, the magnitude and phase of the OMP are shown in Figure 30:

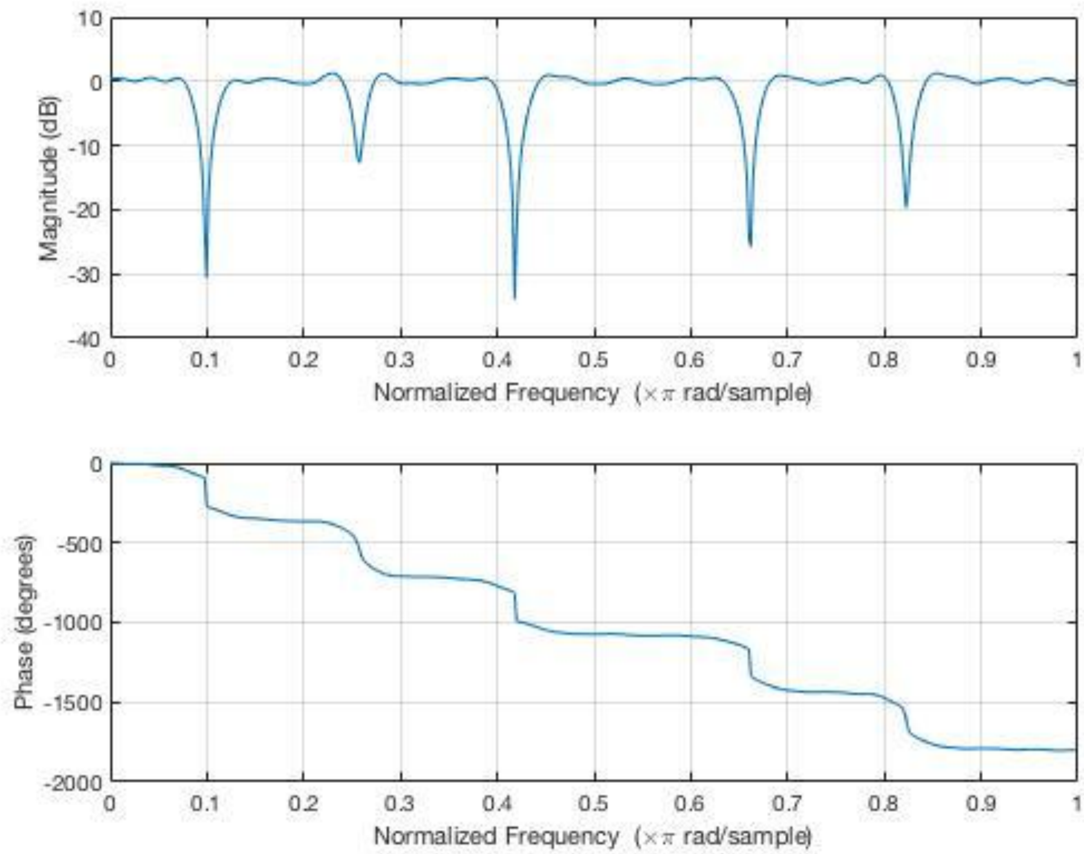

Figure 30 Magnitude and Phase of OMP (100Hz, 250Hz, 420Hz, 660Hz and 830Hz)

Also, we can get the frequency response of QP method which is shown in Figure 31:

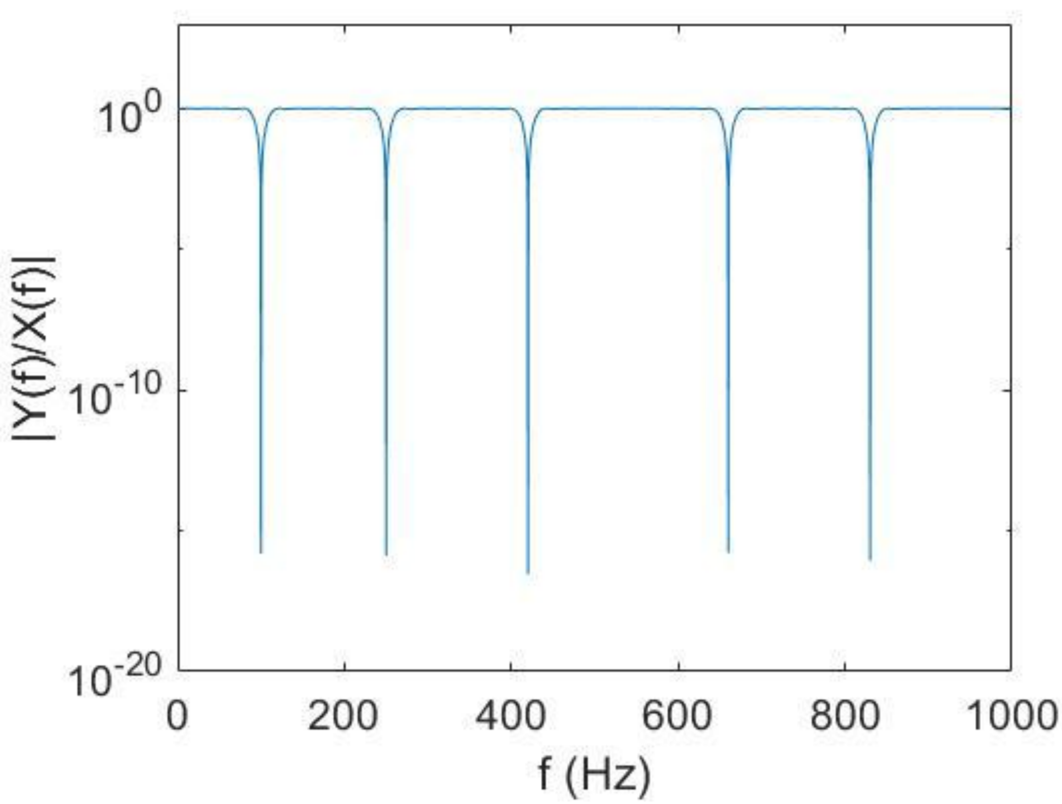

Figure 31 Frequency Response of QP Method (100Hz, 250Hz, 420Hz, 660Hz and 830Hz)

And the phase is shown in Figure 32:

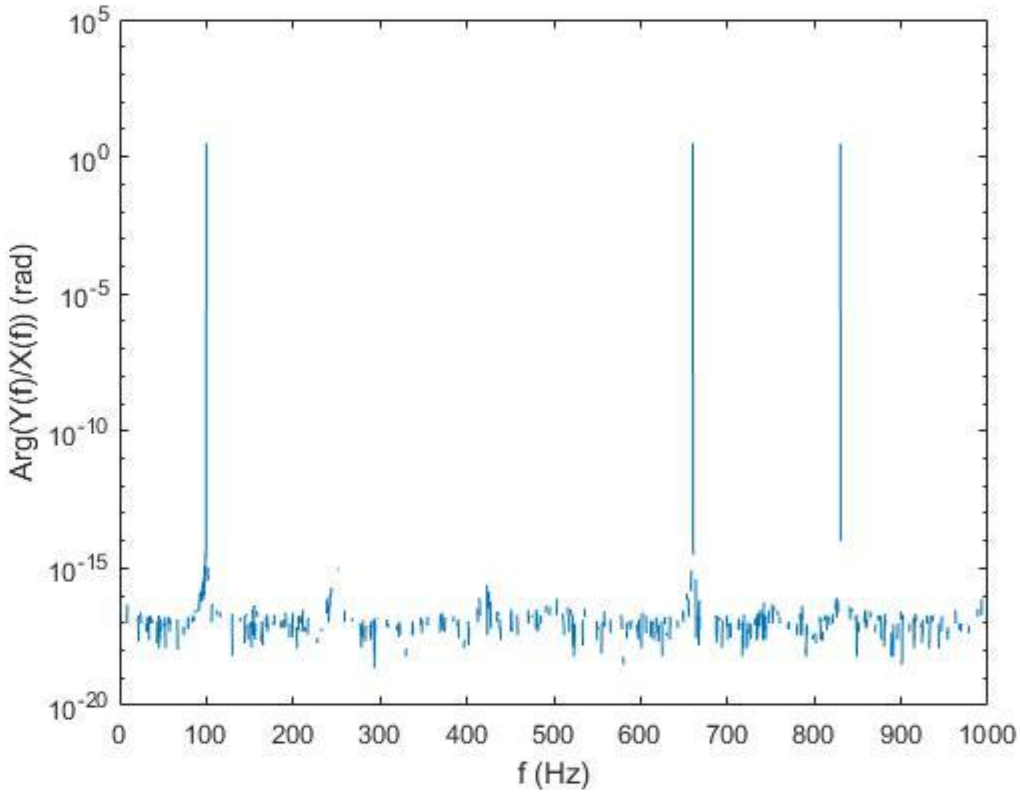

Figure 32 Phase of QP Method

As we can see from Figure 18,19 and 20, the QP Method and the Iterative Reweighted OMP Method have the similar notch locations, but the QP Method still has a narrower notch.

And as we can see from Figure 23, 24 and 25, when we try two notch frequencies, the QP method has less passband ripple and zero phase response, but as we can see from Figure 28, 29 and 30 when we try 5 notch frequencies, the OMP method cannot completely filter some of the 5 frequencies while the QP method does so with no passband ripple and very deep notches.

All in all, the method that we proposed did better results than the traditional methods

VI Conclusion

The QP method has better performance compared with the other methods that were tested both on the width of the stopband and the ripple of the passband. The following figure shows the energy of the signal both without zero-padding and with zero-padding:

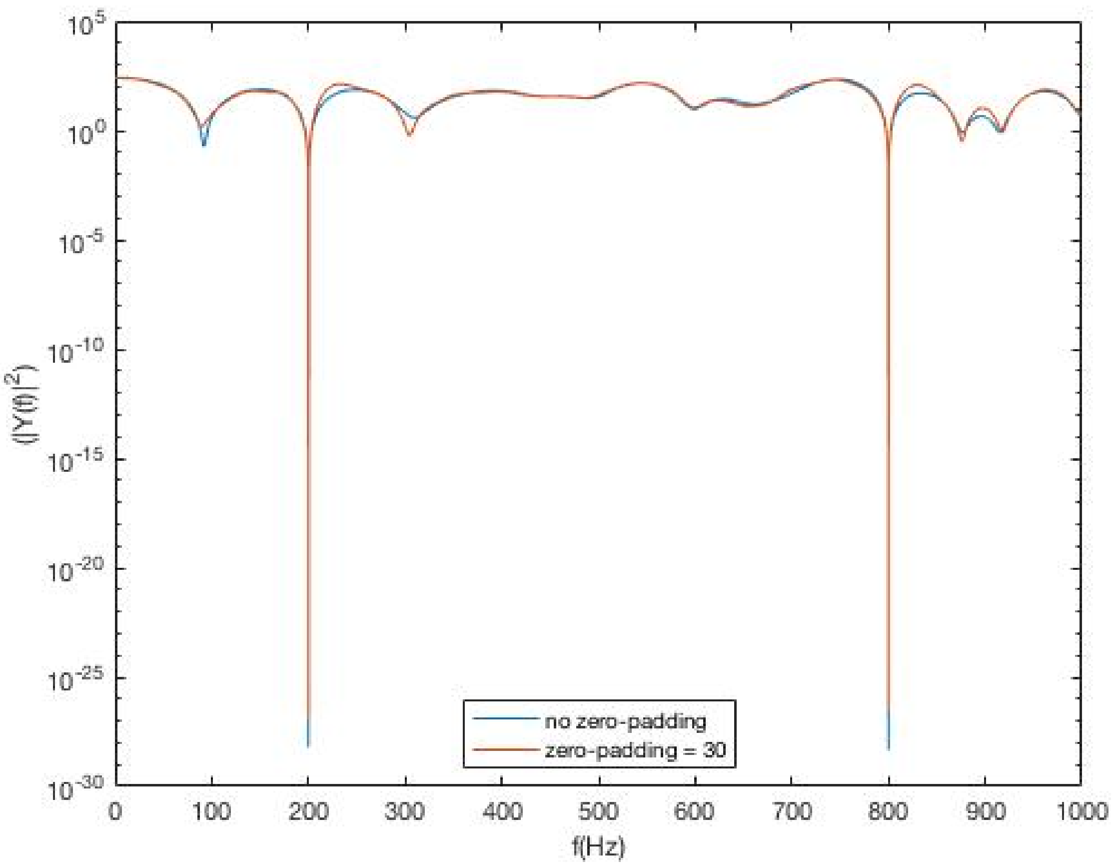

Figure 33 Energy with zero-padding and without zero-padding

From Figure 33, we can see that the zero-padding has less effect on the energy of the signal. The expression here is based on the Parseval's Theorem:

$$E = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|X\left(e^{j\omega}\right)\right|^{2}d\omega \tag{6.1}$$

Then, we summarize an expression to represent the energy which is spread in the zero regions.

First, set up a diagonal matrix as follows:

$$B = \mathrm{diag}([1 \dots 1\ 0 \dots 0\ 1 \dots 1]) \tag{6.2}$$

where the number of 1 in the first part and the last part is N_p and the number of 0 in the middle part is N_d.

Combine (6.2) with (4.4), the expression become:

$$Y = B * [I - A^T(AA^T)^{-1}A] * x \tag{6.3}$$

In (6.3), $A^T\ (AA^T)^{-1}A$ is the projection matrix and we define $P = A^T\ (AA^T)^{-1}A$, the diagonal matrix *B* is used to wipe out the energy in the signal vector and keep the energy in the zero-padding regions, then the energy in the zero-padding regions will be expressed as follows:

$$E = Y^T * Y \tag{6.4}$$

So, (6.4) can be expressed as:

$$\begin{aligned} E &= x^T * (I - P) * B * B * (I - P) * x \\ &= x^T * (I - P) * B * (I - P) * x \end{aligned} \tag{6.5}$$

In the future work, we will update the algorithm in order to save calculations so that we can assure the accuracy of the multi-frequency notch filter because the efficiency of the notch filters is an important issue. And also, we will use our design method into actual signals, such as EEG signals and ECG signals, and see if it works well when operate actual signals. After that, we will update our method again due to the problems we face in the actual signals.

APPENDIX

Works by Yueran Ma

Main code of the OMP Algorithm

```
while mu > 0
    W = diag(w);
    %step 1
    for m = 1:M+1
        %D(m) = 1/norm(A(:,m));
        D(m) = 1/sqrt(A(:,m)'*W*A(:,m));
        B(:,m) = A(:,m)*D(m);
    end
    %step 2
    r = f;
    Z = [];
    for t = 1:k
        [dmax ind] = max(abs(B'*W*r));
        if  t==1;
            Z = [Z ind];
        elseif ~ismember(ind,Z)
            Z = [Z ind];
        end
        Phi(:,t) = B(:,ind);
        zt = inv(Phi'*W*Phi)*Phi'*W*f;
        r = f - Phi*zt;
    end%t
    yk = zt;
    %step 3
    %get indices where Ho = 1
    ind_unity = find((omega < w1-BW/2 | omega > w1+BW/2) & ...
        (omega < w2-BW/2 | omega > w2+BW/2) );
    %create inequality constraint matrix
    Aineq =  cos(omega(ind_unity)*[0:M]);
    [ai_row ai_col] = size(Aineq);
    %create column for mu
    Aineq(:,ai_col+1) = -ones(ai_row,1);
    Aineq1 = -Aineq;
    Aineq1(:,ai_col+1) = -ones(ai_row,1);
    Aineq = [Aineq;Aineq1];
    %create equality constraint matrix
    Aeq = cos(omega_zero*[0:M]);
    Aeq(:,ai_col+1) = zeros(2,1);
    %setup constraint matrix that zeros out elements of h not in Z
    %augment equality constraint matrix with it
    C = eye(M+1);
    C(:,M+2) = zeros(M+1,1);
    Znot = setdiff(Z_all,Z);
    Aeq = [Aeq; C(Znot,:)];
    %set equality bounds
    b3 = zeros(length(Znot),1);
```

```
    beq = [0; 0; b3];
    %set inequality bounds
    b1 = 1 + ones(ai_row,1)*(1-10^(delta/20))/(1+10^(delta/20));
    b2 = -1 + ones(ai_row,1)*(1-10^(delta/20))/(1+10^(delta/20));
    b = [b1;b2];
    c = [zeros(1,M+1) 1]';
    %call linear program
    x = linprog(c,Aineq,b,Aeq,beq);
    h = x(1:M+1);
    mu = x(M+2);
    %plot notch filter amplitude response
    figure(1)
    plot(A*h);
    %semilogy(A*h)
    pause(0.1)
    disp([k x(M+2)])
    %step 4, recompute new weight vector
    eps = max(abs(yk))/dp;
    w = 1./(1 + (r'/eps).^2).^0.25;
    k = k+1;
end
```

BIBLIOGRAPHY

[1] Vaseghi S V. Advanced digital signal processing and noise reduction[M]. John Wiley & Sons, 2008.

[2] Yu T, Mitra S K, Bable H. Design of linear phase FIR notch filter[J]. Sadhana, 1990, 15(3):133-155.

[3] Deshpande R, Kumar B, Jain S B. Highly narrow rejection bandwidth finite impulse response notch filters for communications, IET, 2010, 4(18):2208-2216.

[4] Candes E J, Tao T. Decoding by linear programming[J]. Information Theory, IEEE Transactions on, 2005, 51(12):4203-4215.5.

[5] Tseng C C, Pei S C. Sparse FIR notch filter design and its application[J]. Electronics Letters, 1997, 33(33): 1131-1133.

[6] Vlcek M, Zahradnik P. Digital multiple notch filters performance[C]//Proceedings of the 15th European Conference on Circle Theory and Design ECCTD'01. 2001: 49-52.

[7] Yang J Y, Liu F L, et al. Textured image denoising via signal decomposition and guided filtering[J]. Image Processing, IEEE Transactions on, 2017, 26(7): 3528-3541.

[8] Xu W, Zhao J X, Gu C. Design of Linear-Phase FIR Multiple-Notch Filters via an Iterative Reweighted OMP Scheme[J]. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2014, 61(10): 813-817.

[9] Ma W K, Zhang Y T, Yang F S, A fast recursive-least-square adaptive notch filter and its applications to biomedical signals[J]. Med. Biol. Eng. Comput, 1999, 37(1):99-103.

[10] Eleti A A, Zerek A R. FIR digital filter design by using windows method with

MATLAB[C]//Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2013 14th International Conference in. IEEE, 2013:282-287.

[11] Deshnpande R, Jain S B, Kurmar B. Design of maximally flat linear phase FIR notch filter with controlled null width[J]. Signal Processing, 2008, 88(10): 2584-2592.

[12] Zahradnlk P, Vlcek M. Fast analytical design algorithm for FIR notch filters[J]. Circuits and Systems I: Regular Papers, IEEE Transactions on, 2004, 51(3): 608-623.

[13] Zahradnlk P, Simak B, Vlcek M. Fast Design of Precise Equiripple Notch FIR Filters[C]. //Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCI, 2006. International Conference on. IEEE, 2006:154.

[14] Gu C, Zhao J X, Xu W, et al. Design of linear-phase notch filters based on the OMP scheme and the Chebyshev window[J]. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2012, 59(9): 592-596.

[15] Zahradnlk P, Vlcek M, Unbehauen R. Design of optimal comb FIR filters-speed and robustness[J]. Signal Processing Letters, IEEE, 2009, 16(6): 465-468.

[16] Zahradnlk P, Vlcek M. Analytical design method for optimal equiripple comb FIR filters[J]. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2005, 52(2): 112-115.

[17] Zahradnlk P, Vlcek M. Note on the design of an equiripple DC-notch filter[J]. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2007, 54(2): 196-199.

[18] Zahradnlk P, Vlcek M. An analytical procedure for critical frequency tuning of FIR filters[J]. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2006, 53(1):72-76.

[19] Tseng C C, Pei S C. Design of an equiripple FIR notch filter using a multiple exchange algorithm[J]. Signal Processing, 1999, 75(3): 225-237.

[20] Rusu C, Dumitrescu B. Iterative reweighted l1 design of sparse FIR filters[J]. Signal Processing, 2012, 92(4): 905-911.

[21] Jiang A, Kwan H K, Zhu Y. Peak-error-constrained sparse FIR filter using iterative SOCP[J]. Signal Processing, IEEE Transactions on, 2012, 60(8):4035-4044.

[22] Vlcek M, Unbehauen R. Zolotarev polynomials and optimal FIR filters[J]. Signal Processing, IEEE Transactions on, 1999, 47(3): 717-730.

[23] Baran T, Wei D, Oppenheim A V. Linear programming algorithm for sparse filter design[J]. Signal Processing, IEEE Transactions on, 2010,58(3): 1605-1617.

[24] K. Ichige, M. Iwaki, and R. Ishii. Accurate estimation of minimum filter length for optimum FIR digital filters[J]. Circuits and Systems II: Analog Digital Signal Processing, IEEE Transactions on, 2000, 47(10): 1008-1016.