Digital Signal Processing

**Project 1**

Audio Equalizer

horizontal line

# **Team members**

Fatma Mohammed (43)

Salma Samir (31)

Sohayla Mohammed (32)

# **Contents**

[**Team members**](#_vydniszftb1n) **1**

[**Contents**](#_lni7dpgtz8ox) **1**

[**FIR**](#_htxwzoqyyzts) **4**

[Introduction](#_la5jp5tnimjw) 4

[Equiripple](#_5e4rbmiv86d0) 5

[Introduction](#_ygac2rqqb0mh) 5

[Algorithm and Implementation](#_arljn5tn5fw0) 6

[Analysis](#_unwbhxi5zpeh) 6

[Specifications](#_q58wztnamphe) 6

[Low pass](#_4wu0tkzera9d) 6

[Frequency](#_isblnrs7212c) 6

[Magnitude](#_jmpjpr6isxz8) 6

[Band pass](#_bqp2iocupohh) 7

[Frequency](#_89coigb5n3g) 7

[Magnitude](#_d3039fqil0e1) 7

[Calculating](#_xcwpp5hib6qx) 7

[Predefined Filter Length](#_4vln736yef7k) 8

[Predefined Specification](#_dbgotivnq55e) 8

[Error](#_xcwpp5hib6qx) 9

[Minimax or Chebyshev Criterion](#_81094bor4yzj) 9

[Bands plots](#_jyh2ftl4sjld) 9

[Impulse response](#_il8gx19gfdtt) 10

[Low pass](#_vn2oskt11t7c) 10

[Band pass](#_nfny5tvuofb7) 10

[frequency response](#_hxf3oycurhm9) 11

[Low pass](#_22433oo1eyq) 11

[Band pass](#_ooffme4vg5vm) 12

[Group delay](#_il8gx19gfdtt) 13

[Low pass](#_fx5nnqc9czbk) 13

[Band pass](#_3gkkasfe2jak) 13

[pole/zero](#_il8gx19gfdtt) 14

[Low pass](#_hfpjgfkfowt0) 14

[Band pass](#_id2tmat0kbqw) 14

[Least Square](#_kr30qg7ildt0) 14

[Introduction](#_k0zjd5atyef) 14

[Algorithm and Implementation](#_l7hjc717q5te) 16

[Analysis](#_horbsfhvo3qj) 16

[Specifications](#_l1ij8raljrj6) 17

[Low pass](#_alwl78fzbv21) 17

[Frequency](#_9kcmoerg88tl) 17

[Magnitude](#_5gv4r59bghm9) 17

[Band pass](#_56q7uq9xs8ht) 17

[Frequency](#_74y3sql1i9rf) 18

[Magnitude](#_mnblvww5rnl) 18

[Error](#_kq10o8k44xts) 18

[Bands plots](#_bgviy8qwfkwi) 18

[Impulse response](#_il8gx19gfdtt) 19

[Low pass](#_rljkf7uguroo) 19

[Band pass](#_z0f6zhjsaahu) 19

[frequency response](#_2dj7ti4n420x) 20

[Low pass](#_64jyhs3rwynd) 20

[Band pass](#_ylx4uc3bawn2) 20

[Group delay](#_il8gx19gfdtt) 22

[Low pass](#_w0whci8c1xus) 22

[Band pass](#_7ctio2q1atgu) 22

[pole/zero](#_fczcxv2cl053) 23

[Low pass](#_4ece0mjvqitx) 23

[Band pass](#_pimmwugkp9ao) 23

[Alternatives](#_puyst1gtj9va) 23

[Difference between Equiripple and Least square in error](#_yufp4vdbakcj) 23

[Window method](#_pnynnaiomt2c) 24

[**IIR**](#_g4zo1f45mr00) **24**

[Introduction](#_ug8p06rccwqk) 24

[Butterworth](#_gxqz9sayrumx) 24

[Introduction](#_4kuqrg2kvvk0) 24

[Algorithm and Implementation](#_kd269jpnu5wc) 24

[Analysis](#_3ysmngrelx77) 24

[Specifications](#_r50a0esw2h6p) 24

[Error](#_r9tyj0ttj90l) 24

[Bands plots](#_a6ww1571ubsb) 24

[Impulse response](#_c9xe9j5luyhy) 25

[frequency response](#_mvtsf2dgd8k9) 25

[Gain](#_yw267xn6cv6) 25

[Phase](#_9qb7i1uixn8l) 25

[Group delay](#_dxqu7s9sq2f0) 25

[pole/zero](#_l14z7dax5ink) 25

[Chebyshev type 1](#_l74yk6elhwop) 25

[Introduction](#_nsmd0lbd0eaj) 25

[Algorithm and Implementation](#_fe6e8u15idsj) 25

[Analysis](#_hr5ud7clarkm) 25

[Specifications](#_3d1f61c28tdj) 25

[Error](#_z2vpfgm02gaz) 25

[Bands plots](#_4l3470yjoexb) 25

[Impulse response](#_40mlmjkruknl) 25

[frequency response](#_mw4oudiyxm0w) 25

[Gain](#_etzms9nzaxlp) 25

[Phase](#_w8vhmqi98lat) 25

[Group delay](#_mjvrvx1xdqfu) 25

[pole/zero](#_qfyt2p3rg3go) 25

[Alternatives](#_1kume7jmn2a8) 25

[**FIR vs IIR**](#_t3m6uxhsp930) **26**

[Computational and Speed](#_x6xl10opj7jf) 26

[Delay](#_708urah5mazc) 26

[**GUI**](#_nqxrfkf82xc0) **27**

[User Guide](#_j2wwhaynkwfb) 27

[**Sample Runs**](#_i0uhe8ckiyxz) **27**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

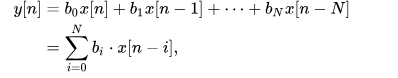
# 

# 

# 

# **FIR**

## **Introduction**

* In [signal processing](https://en.wikipedia.org/wiki/Signal_processing), a finite impulse response (FIR) filter is a [filter](https://en.wikipedia.org/wiki/Filter_(signal_processing)) whose [impulse response](https://en.wikipedia.org/wiki/Impulse_response) (or response to any finite length input) is of finite duration, because it settles to zero in finite time. This is in contrast to [infinite impulse response](https://en.wikipedia.org/wiki/Infinite_impulse_response) (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying).
* For a [causal](https://en.wikipedia.org/wiki/Causal_filter) [discrete-time](https://en.wikipedia.org/wiki/Discrete-time) FIR filter of order N, each value of the output sequence is a weighted sum of the most recent input values:
* Properties
  + Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler.
  + Are inherently [stable](https://en.wikipedia.org/wiki/BIBO_stability), since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than times the largest value appearing in the input.
  + Can easily be designed to be linear phase by making the coefficient sequence symmetric. This property is sometimes desired for phase-sensitive applications, for example data communications, [seismology](https://en.wikipedia.org/wiki/Seismology), [crossover filters](https://en.wikipedia.org/wiki/Audio_crossover), and [mastering](https://en.wikipedia.org/wiki/Audio_mastering).
* In this project we implemented two types of FIR
  + Equiripple.
  + Least Square.

## **Equiripple**

### **Introduction**

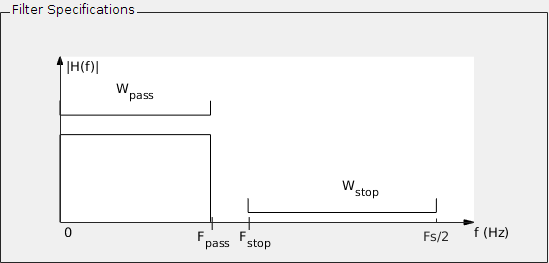
* The FIR Equiripple method, which is also known as Remez Exchange method, Parks-McClellan, always provides the FIR filter with minimum effort. The Parks-McClellan method provides the solution for the FIR filter using the Remez Exchange algorithm in an iteration process. Since filters can be designed using this approximation method with a constant ripple both in the passband and stopband, the filter is also known as the equiripple filter.
* The Remez Exchange algorithm is an optimizing algorithm based on Chebyshev polynomials. An error function is formed for the desired filter from a linear combination from cosine functions and is minimized by an efficient optimization process.

### **Algorithm and Implementation**

### **Analysis**

#### Specifications

##### Low pass



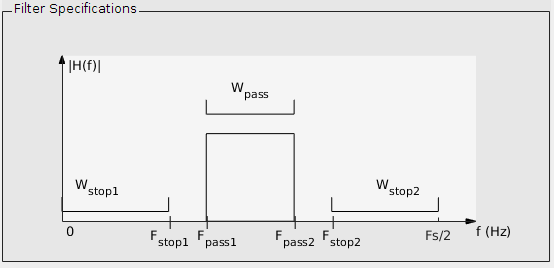
###### Frequency

* Fs … Sampling Frequency
* Fpass … Passband Frequency
* Fstop … Stopband Frequency

###### Magnitude

* Wpass … weight for pass band
* Wstop … weight for stop band
* Each of them are set to their default value of 1.

##### Band pass



###### Frequency

* Fs … Sampling Frequency
* Fpass1 … first Passband Frequency
* Fstop1 … first Stopband Frequency
* Fpass2 … second Passband Frequency
* Fstop2 … second Stopband Frequency

###### Magnitude

* Wpass … weight for pass bands
* Wstop1 … weight for first stop band
* Wstop2 … weight for second stop band

Each of them are set to their default value of 1.

#### Calculating

There are two options for calculating an FIR filter. Either you can specify the filter length or you can pre-define a specification and calculate the length of the filter from this.

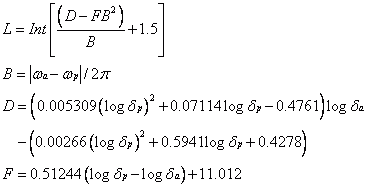
##### Predefined Filter Length

If the filter length is specified, only the ratio of errors of individual bands can be defined, since there is no assurance as to whether the desired specification (absolute error limits) can be preserved using the specified filter length.

The predefined filter length corresponds to the length of the impulse response. In this case, the weighting of the errors of the individual bands is specified. The weighting factors are in relation to one another. For a multiband filter with three bands, the weighting information {1, 10, 1} means, for instance, that the second band has one ripple that is ten times lower than the other two bands. This range is thus smoother. If the weighting is not specified, all bands will have the same weighting ratio.

##### Predefined Specification

The filter length is not specified in this case. Instead, all desired error limits are specified. Using this information, the filter length L is empirically estimated using the following formula:



This formula applies only to low pass and high pass filters. For multiband filters, the filter length is calculated for each transition from one band to the next. The largest value is taken as the starting value. Next, the filter length is increased or reduced until the optimal filter has been determined for the desired specification.

The procedure can only be solved numerically and therefore it is a relatively intensive procedure for higher orders. However, it is an optimal solution as opposed to the window method. It is possible, however, that the algorithm may find no solution due to problems with convergence.

#### Error

##### Minimax or Chebyshev Criterion

The Chebyshev or Minimax Criterion is used for this design method. For this method, a search is performed in fixed frequency intervals for a frequency response A(ejω) that minimizes the maximum weighted approximation error of the error function. The following error function is generated for this:

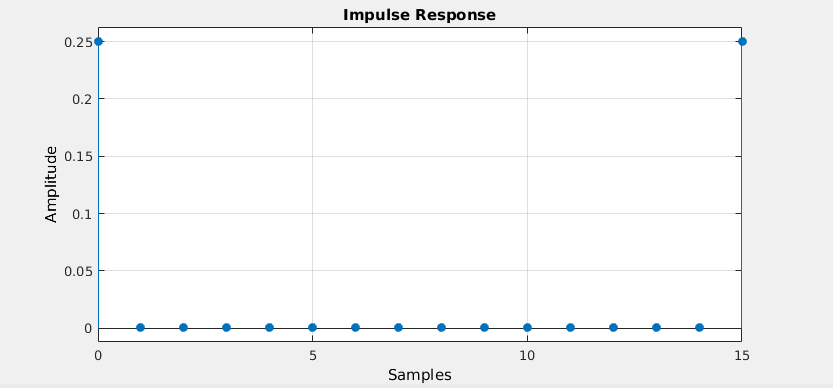


* E(ω) … error function
* W(ω) … weighting function
* H(ejω) … desired frequency response
* A(ejω) … approximation function

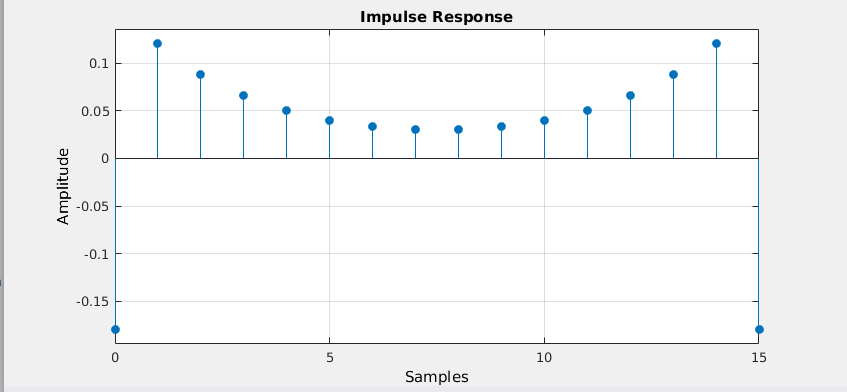
### **Bands plots**

#### Impulse response

##### Low pass

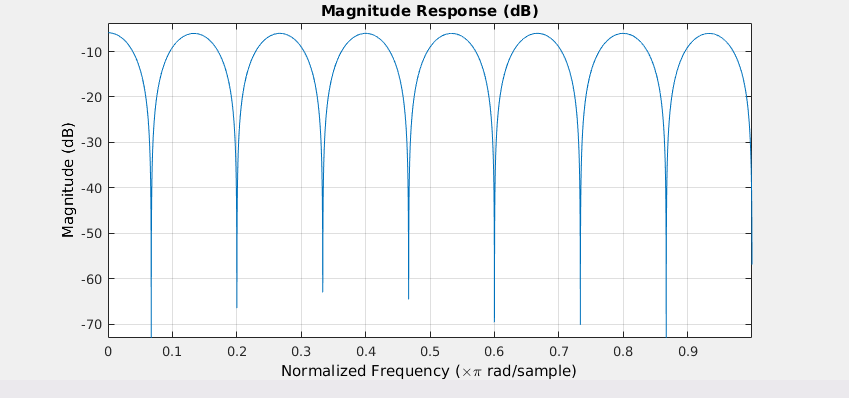


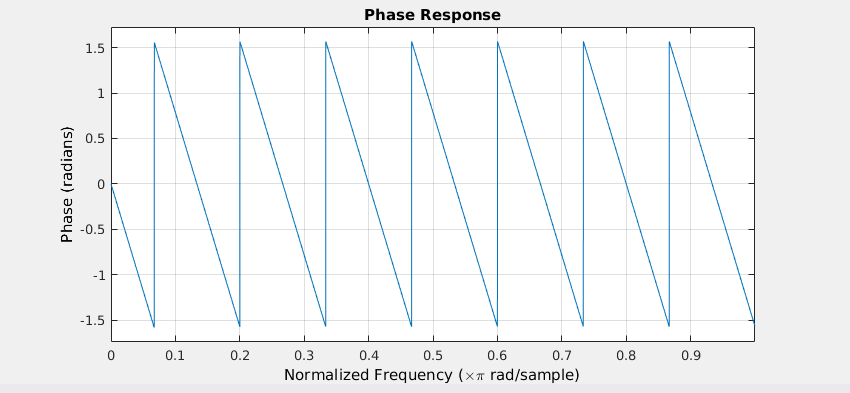
##### Band pass



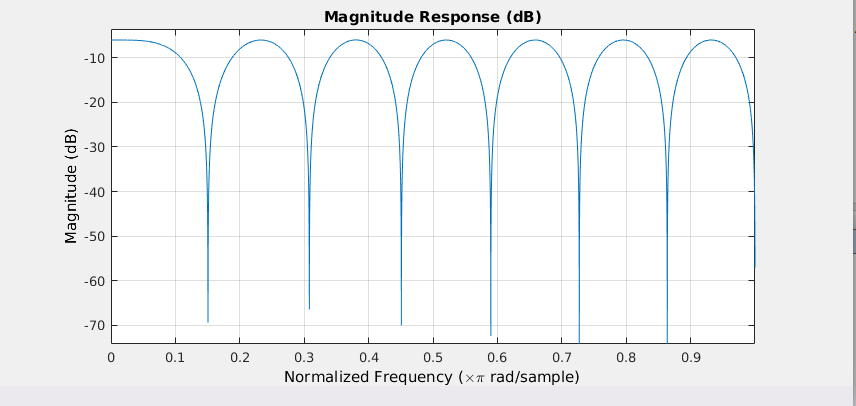
#### frequency response

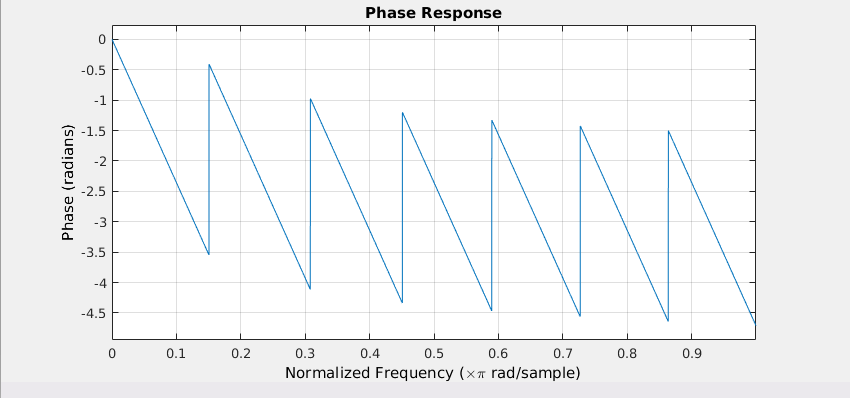
##### Low pass





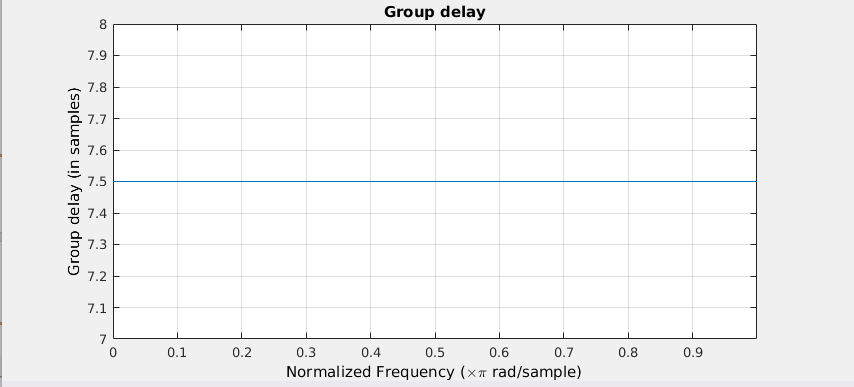
##### Band pass



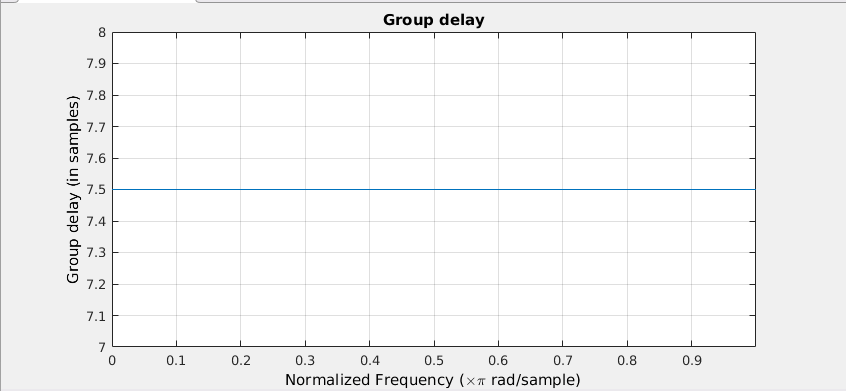


#### Group delay

##### Low pass

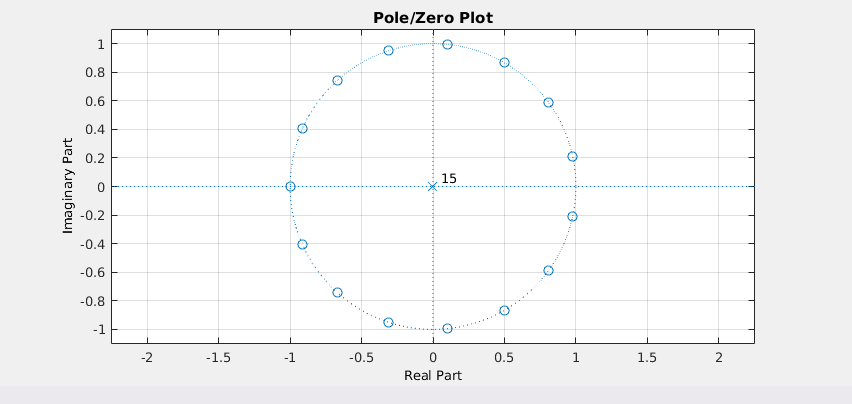


##### Band pass

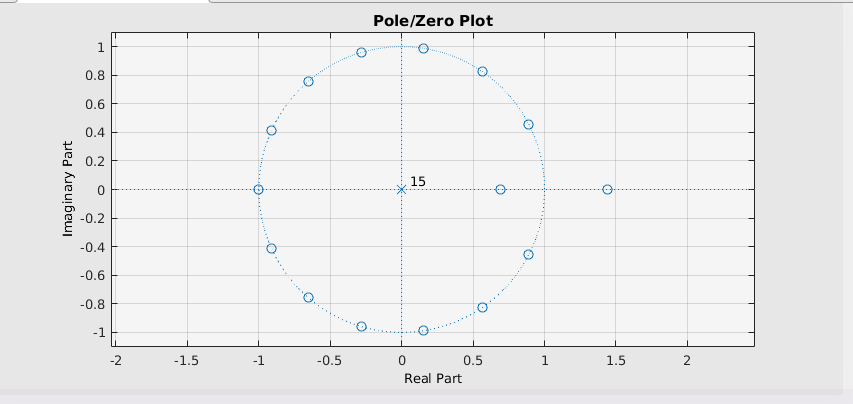


#### pole/zero

##### Low pass

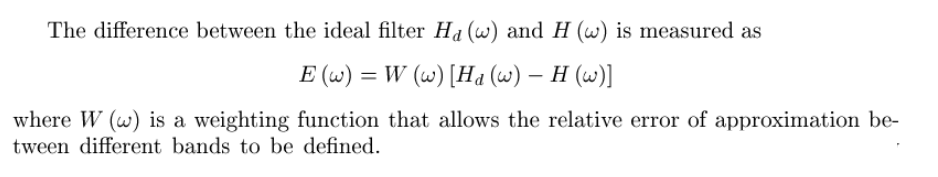


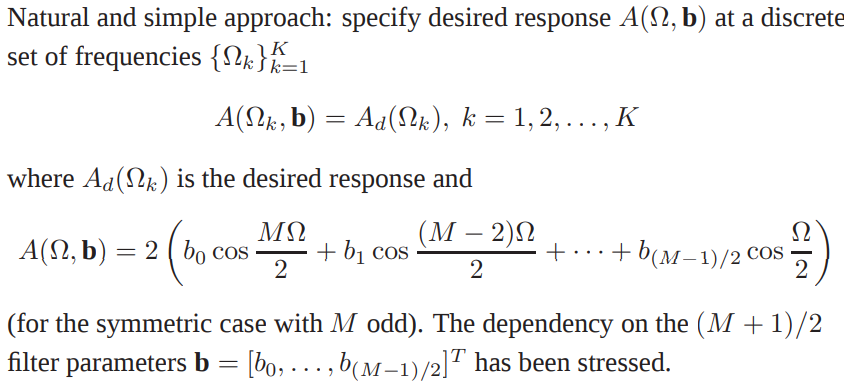
##### Band pass

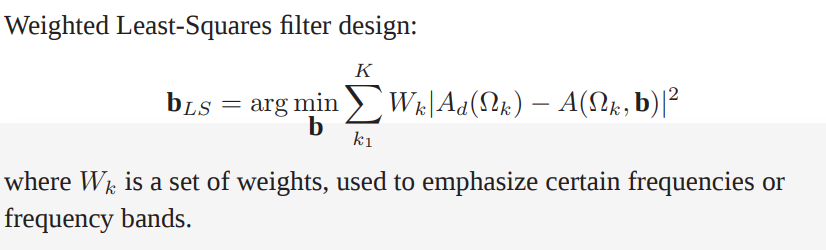


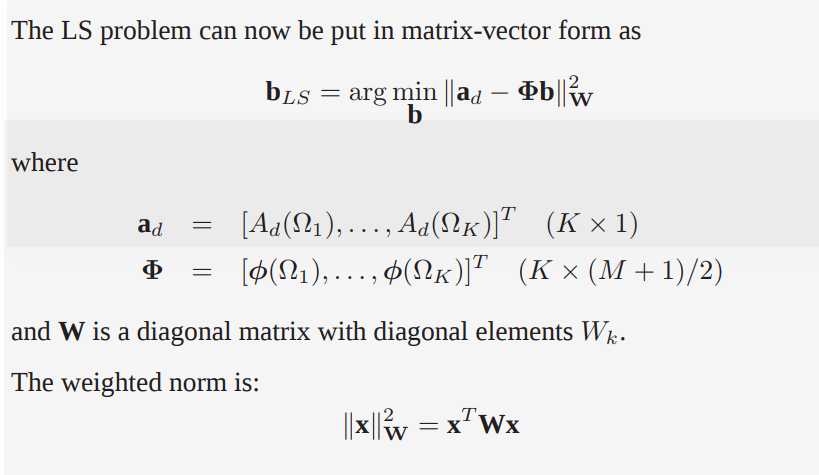
## **Least Square**

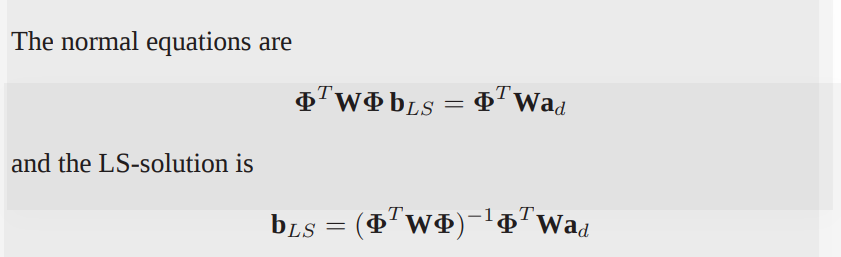
### **Introduction**

* A method that approximates the desired frequency response by a linear-phase FIR amplitude function according to the following optimality criterion
* An alternative definition, an FIR filter with (anti-)symmetric coefficients that has "perfect" (linear) phase. We need only worry about the amplitude!







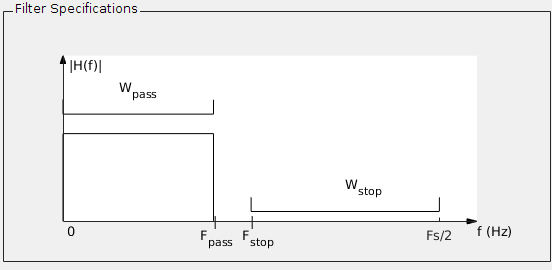


### **Algorithm and Implementation**

### **Analysis**

#### Specifications

##### Low pass



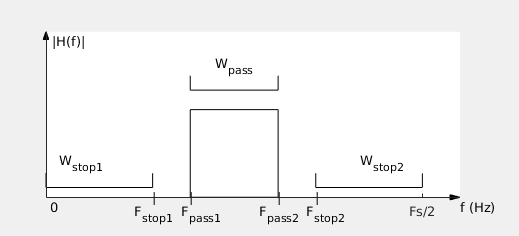
###### Frequency

* Fs … Sampling Frequency
* Fpass … Passband Frequency
* Fstop … Stopband Frequency

###### Magnitude

* Wpass … weight for pass band
* Wstop … weight for stop band
* Each of them are set to their default value of 1.

##### Band pass



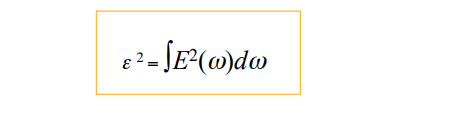
###### Frequency

* Fs … Sampling Frequency
* Fpass1 … first Passband Frequency
* Fstop1 … first Stopband Frequency
* Fpass2 … second Passband Frequency
* Fstop2 … second Stopband Frequency

###### Magnitude

* Wpass … weight for pass bands
* Wstop1 … weight for first stop band
* Wstop2 … weight for second stop band

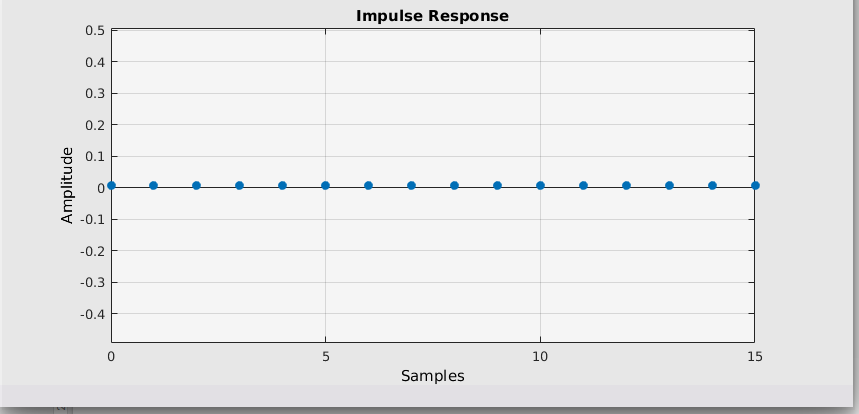
#### Error

* The integral of the weighted square frequency-domain error is given by
* Designing a FIR filter now reduces to determining the coefficients that would minimise ε 2
* The least squares design minimizes the error energy but its maximum error is relatively large.

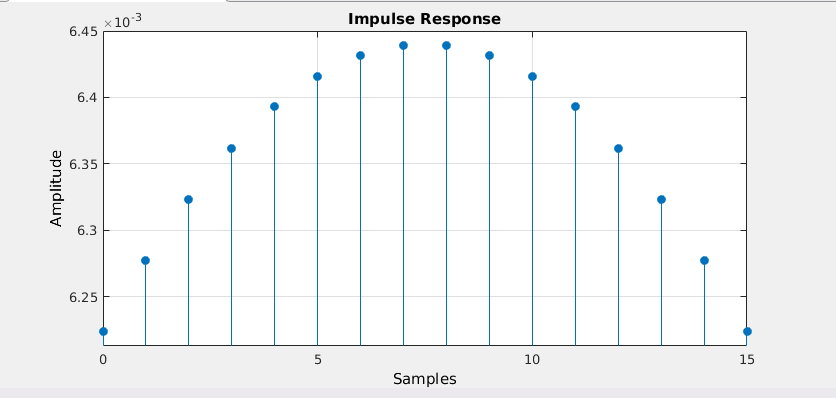
### **Bands plots**

#### Impulse response

##### Low pass

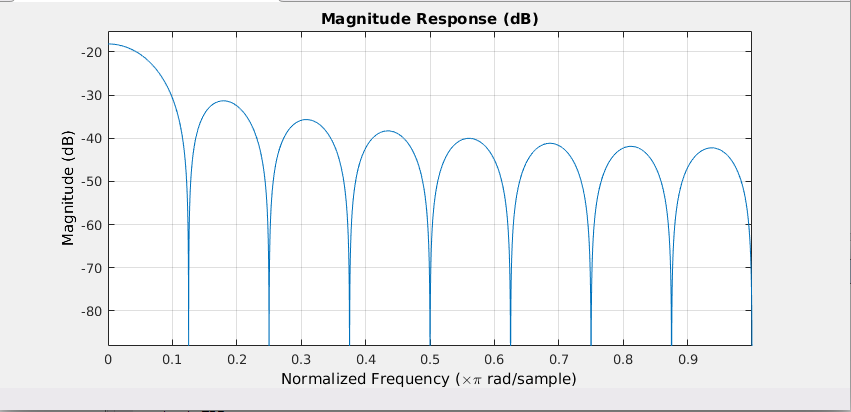


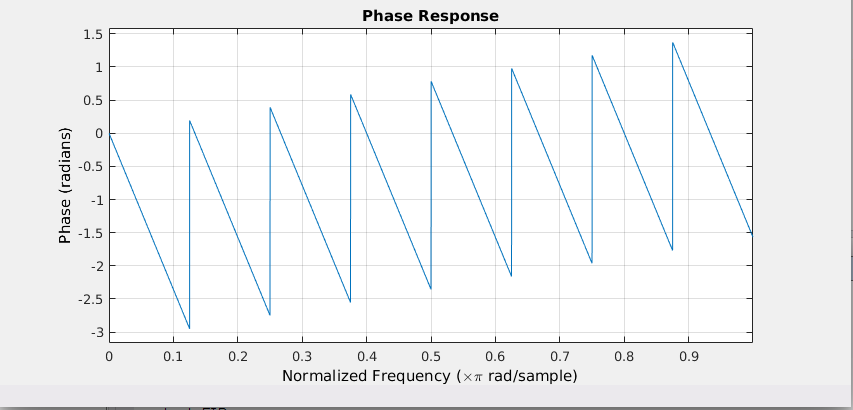
##### Band pass



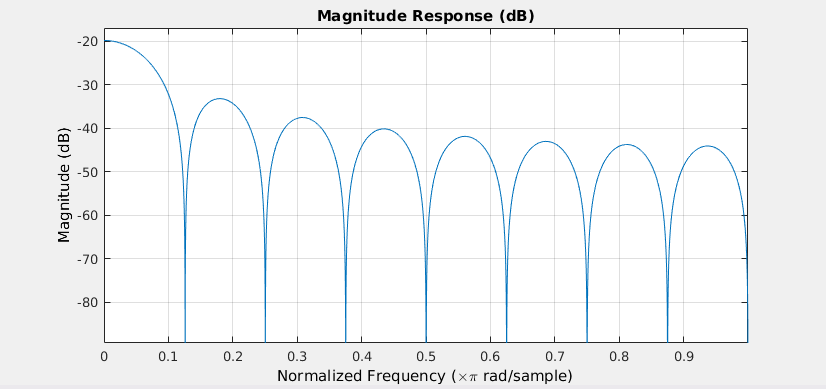
#### frequency response

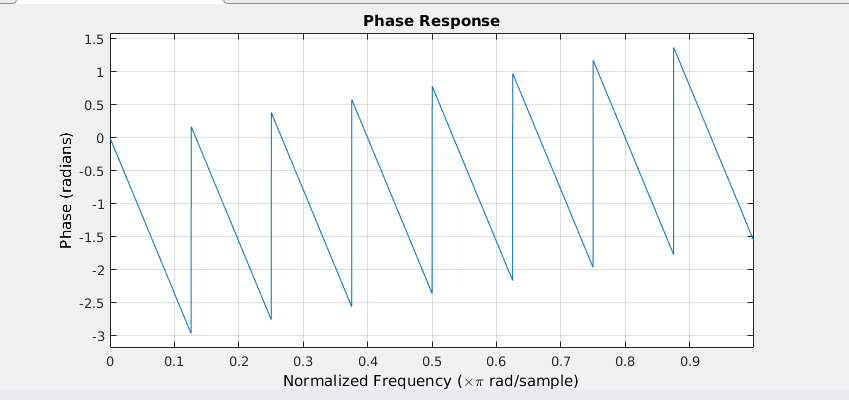
##### Low pass





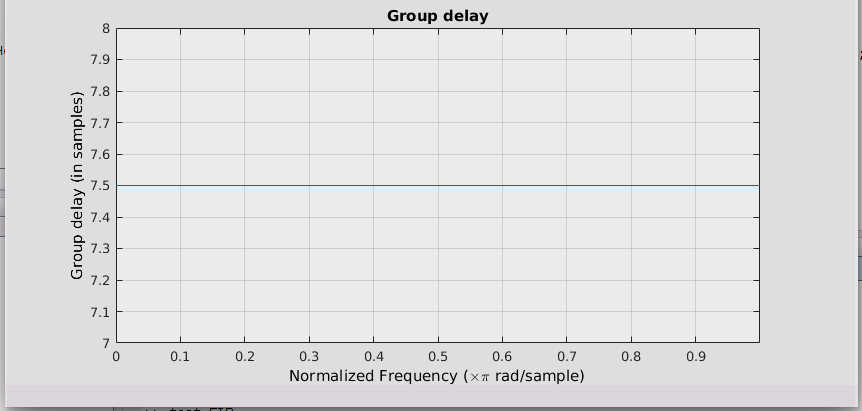
##### Band pass



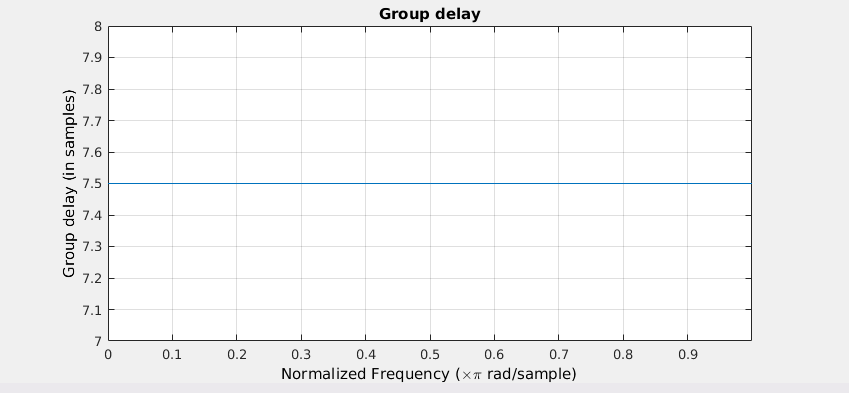


#### Group delay

##### Low pass

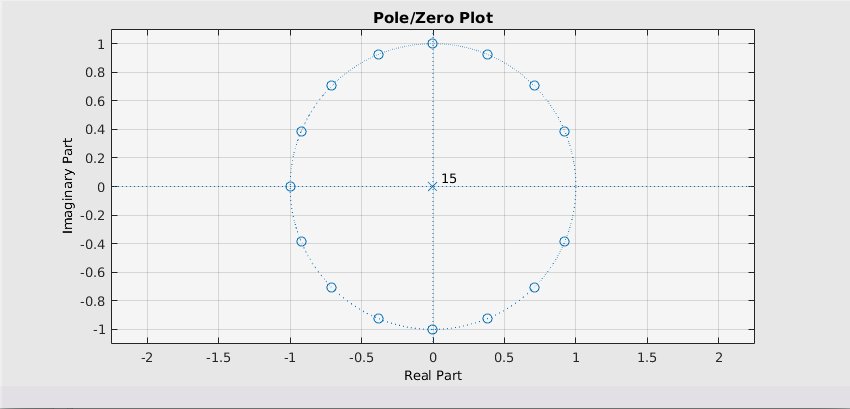


##### Band pass

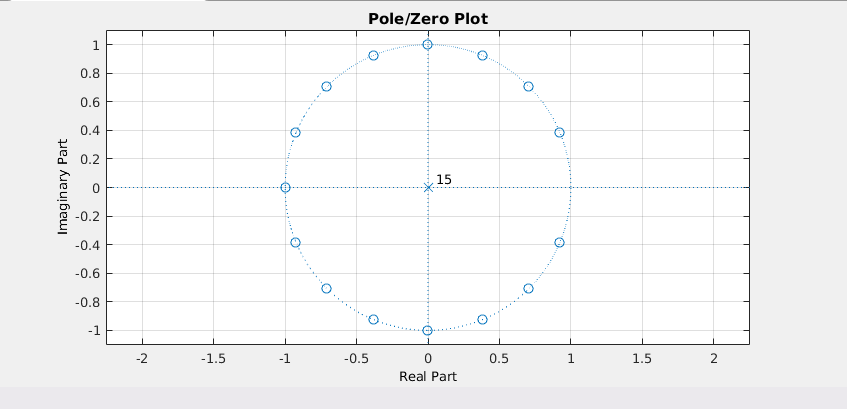


#### pole/zero

##### Low pass



##### Band pass



## **Alternatives**

### **Difference between Equiripple and Least square in error**

One main difference is the cost function used in the two design methods:

* Equiripple filters seek to minimize the maximum error between the desired filter response and the designed approximation.
* Least-squares filters seek to minimize the total squared error betwen the desired filter response and the designed approximation.

### **Window method**

The disadvantage to designing FIR filters using the window method is that the approximation error cannot be influenced in different frequency ranges. It is therefore often better when designing filters to implement the minimax strategy (minimize the maximum error) or an error criterion with weighting of the frequency. This results in the "best" filter that can be achieved for a defined specification.

# **IIR**

## **Introduction**

## 

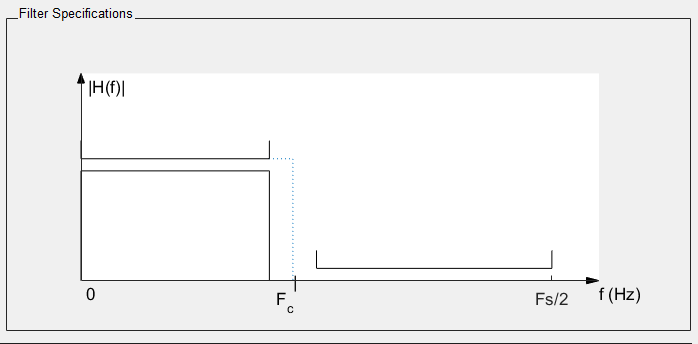
## **Butterworth**

### **Introduction**

### **Algorithm and Implementation**

Specifications

Low pass



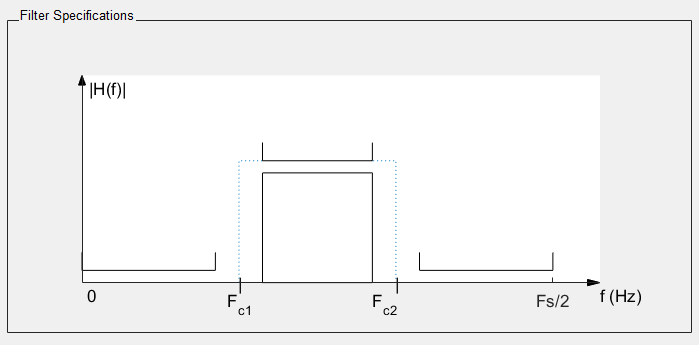
*frequency*

* Fc ... cut-off frequency
* Fs … Sampling frequency

*Magnitude*

* No magnitude specifications

Band pass



*frequency*

* Fc1 ... first cut-off frequency
* Fc2 … second cut-off frequency
* Fs … Sampling frequency

*Magnitude*

* No magnitude specifications

### **Analysis**

#### Specifications

#### Error

### **Bands plots**

#### Impulse response

#### frequency response

##### Gain

##### Phase

#### Group delay

#### pole/zero

## **Chebyshev type 1**

### **Introduction**

### **Algorithm and Implementation**

### **Analysis**

#### Specifications

#### Error

### **Bands plots**

#### Impulse response

#### frequency response

##### Gain

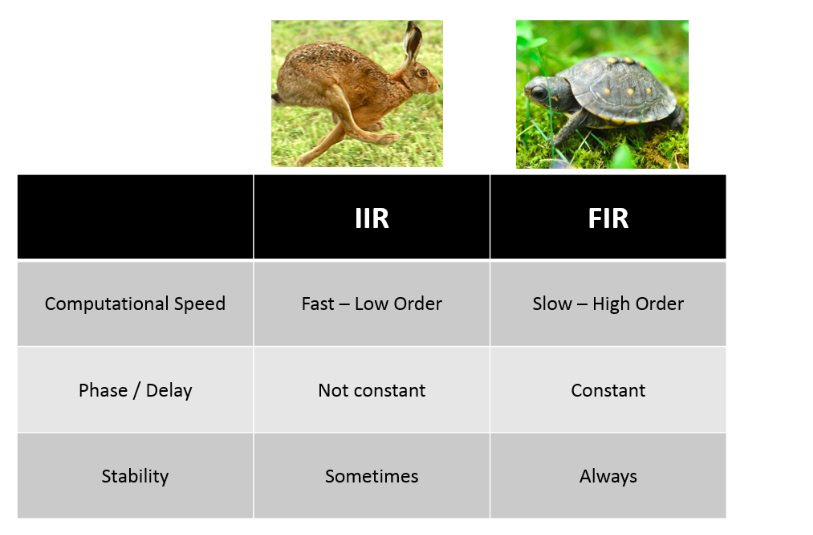
##### Phase

#### Group delay

#### pole/zero

## **Alternatives**

# **FIR vs IIR**



### **Computational and Speed**

* For the same order, IIR filter has a sharper roll of than a FIR filter.
* But IIR filter has a set of drawbacks:
  + Delays - An IIR filter has an unequal delay at different frequencies, while an FIR has a consistent delay at every frequency.
  + Stability – Due to its construction, an IIR filter can sometimes be unstable and not be able to be computed or applied to the data. The FIR filter formulation is always stable.

### 

### 

### 

### **Delay**

* A FIR filter has equal time delay at all frequencies, while the IIR filter time delay varies with frequency. Usually the biggest time delay in the IIR filter is at the cut off frequency of the filter.

# **GUI**

## **User Guide**

### **Introduction**

Audio Equalizer is an application program which uses a signal wave as an input and its output is a signal wave also which is the input signal with some changes according to the user preferences.

### **Steps**

1. The user has to enter the address of the file which is the input signal by using the “Browse” button and choose it.
2. The user can choose the gains for each band from the nine bands (0-170 Hz, 170-310 Hz, 310-600 Hz, 600-1000 Hz, 1-3 KHz , 3-6 KHz, 6-12 KHz, 12-14 KHz, 14-16 KHz), by default each gain equals to zero and the gain of each band appears below the band itself.
3. In Settings, The user chooses the filter type (FIR Equir, FIR LeastSq, IIR Butter, IIR Cheby) and enters the sample rate value or chooses an fixed sample rate (normal, half, double).
4. In Plot, If the user clicks on “Input” Button, a figure of the plotting of the input signal appears, if the user clicks on “Output” Button , a figure of the plotting of the output signal appears and if the user chooses one of the bands, the plotting of the input signal after the band before multiplying the gain appears.
5. When the user clicks on “Save” Button, the output signal plot will be saved.

# **Sample Runs**

FIR-Equiripple

FIR-Least Square

IIR-Butterworth

IIR-Chebyshev