UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES

# MINIMIZATION OF THE FINITE WORD LENGTH NOISE IN THE IIR DIGITAL FILTERS USING TRUNCATION AND ROUNDATION

M.Sc.THESIS
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

BY

TURGUT DEVECİ

MAY 2013

# Minimization Of The Finite Word Length Noise In The IIR Digital Filters Using Truncation and Roundation

**M.Sc.Thesis**
**in**
**Electrical and Electronics Engineering**
**University of Gaziantep**

**Supervisor**

**Prof. Dr. Arif NACAROĞLU**

**by**

**Turgut DEVECİ**

**May 2013**

REPUBLIC OF TURKEY

UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELECTRICAL – ELECTRONICS ENGINEERING

Name of the thesis: Minimization of the Finite Word Length Noise in the IIR Digital Filters Using Roundation and Truncation

Name of the student: Turgut DEVECİ

Exam date: 22.05.2013

Approval of the Graduate School of Natural and Applied Sciences

Doç. Dr. Metin BEDİR
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Celal KORAŞLI
Head of Department

This is to certify that we have read this thesis and that in our consensus opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Arif NACAROĞLU
Supervisor

Examining Committee Members

Doç. Dr. Ahmet İhsan KUTLAR (Chairman) ..........................

Prof. Dr. Arif NACAROĞLU ..........................

Prof. Dr. Rauf MİRZABABAYEV ..........................

Asst. Prof. Dr. Nurdal WATSUJİ ..........................

Asst. Prof. Dr. Tolgay KARA ..........................

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

**Turgut Deveci**

# ABSTRACT

## MINIMIZATION OF THE FINITE WORD LENGTH NOISE IN THE IIR DIGITAL FILTERS USING TRUNCATION AND ROUNDATION

DEVECİ, Turgut

Master Science Thesis, Department of Electrical and Electronics Engineering

Supervisor: Prof.Dr.Arif NACAROĞLU

May 2013

38 Pages

In this thesis, effects of finite word length on Infinite Impulse Response digital filters (IIR Digital Filters) are studied and it is tried to approach ideal filter characteristics with the aim of using minimum bit number(s) and getting minimum error. The coefficients of the filter are changed by using truncation and/or roundation approximation to obtain the acceptable gain characteristic with shorter binary coefficients.

**Key Words**: Infinite Impulse Response, IIR, Digital Filter, Finite Word Length Noise, Truncation, Roundation

# ÖZ

## KESME VE YUVARLAMA İŞLEMLERİNİ KULLANARAK REKÜRSİF SAYISAL FİLTRELERDE OLUŞAN SONLU BİT UZUNLUĞU GÜRÜLTÜSÜNÜ MİNİMİZE ETME

DEVECİ, Turgut

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof.Dr.Arif NACAROĞLU

Mayıs 2013

38 Sayfa

Bu tezde, sonlu bit uzunluğu gürültüsünün sonsuz darbe cevaplı (rekürsif - IIR {geribeslemeli}) sayısal filtreler üzerindeki etkileri incelenmiş ve pratikte elde edilen alçak geçiren ve bant geçiren filtre özelliklerine, en az bit sayısı ve en az hata hedefiyle yaklaşılmaya çalışılmıştır. Ideal filtrede elde edilen katsayılar kesme (truncation) ve yuvarlama (round-off) işlemine ayrı ayrı tâbi tutulmuş, verilen filtre özelliklerini sağlayan en kısa ikilik katsayılar elde edilmeye çalışılmıştır.

**Anahtar Kelimeler**: Rekürsif Dijital Filtre, IIR, Sonlu Bit Uzunluğu Gürültüsü, Truncation, Roundation

# ACKNOWLEDGEMENTS

I express sincere appreciation to Prof. Dr. Arif NACAROGLU, for his guidance, encouragement, and support in every stage of this thesis. It has been a unique privilege to study with him. Especially his kindness and positive energy affected me so much.

I would like to thank my friends Hasan TOKSÖZ, İlker DÖNER, Nasır CEYLAN, Barış GÖREN, Mehmet DEMİR, Ali Osman ARSLAN and especially Seydi KAÇMAZ for their grand help, support and encouragement.

I also would like to thank my 15 year-old friends Cüneyt AKYÜZ, Emin BAKIRCI, Mehmet ÇİFTÇİ, Fethi GÜLLÜ, Haluk ÖZBEK, Hüseyin İŞLEK, Mehmet Ali MELİK and Muhammet BÜYÜKKIRCALI for their support.

Lastly, I wish to special thank to my father, mother and brother.

I hope this thesis will be useful source studying in approximation methods.

## Table Of Contents

## List of Figures

## List of Tables

## List of Symbols

x(n) : Input Sequence of the Digital Filter

y(n) : Output Sequence / Excitation of the Filter

$a_k$ : Coefficients of Denominator in Transfer Function

$b_k$ : Coefficients of Nominator in Transfer Function

h(n) : Length-N Impulse Response of the Filter

x(z) , y(z) & h(z) : z-transform of x(n), y(n) & h(n) respectively

A : Gain of the Frequency Response

$\emptyset$ : Phase of the Frequency Response

D : Floating-point Number

s : Sign Bit

m : Mantissa

c : Characteristic or Exponent

f : Unsigned Fraction

$Q_r(D)$ : Rounded Value

$Q_t(D)$ : Truncated Value

$Q_{mt}(D)$ : Magnitude Truncated

$\varepsilon_r$ : Rounding Error

$\varepsilon_t$ : Truncating Error

$\varepsilon_{rel}$ : Relative Error

# CHAPTER 1

# INTRODUCTION

Filters are one of basic component of all signal processing and telecommunication systems. The primary functions of a filter are one or more of the followings:

(a) to restrict a signal into a demanded frequency band or channel; as an example as in a radio/tv channel selector or anti-aliasing filter,

(b) to seperate a signal into two or more sub-band signals for subband signal processing, as an example in music coding,

(c) to alter the frequency spectrum of a signal, for example in audio graphic equalizers,

(d) to model the input-output relation of a system such as a voice production, musical instruments, mobile communication channel, room acoustics and telephone line echo.

Through computational algorithm, a digital filter performs on input signal to produce and output signal. A digital filter can be modelled with digital hardware or can be simulated on a general-purpose computer These filters have found important applications in an increasing number of fields in science and engineering, and design techniques have been developed to achieve desired filter characteristics.

The price of a digital filter, if carried out as a special-purpose computer, contingent on the word length of the coefficients in a heavy manner. In many examples of real-time applications, the infinite precision coefficients of digital filters derived from their design have to be replaced by finite word length equivalents.

Mainly, the word length should be reduced as much as possible. This reduction can be done in two ways: Truncation and/or Roundation

Truncation means deleting digits of a number. In next two examples, numbers are truncated to 2 decimal digits.

12.28392 ~ 12.28

23.45672 ~ 23.45

-6.39812 ~ -6.39

Rounding means dropping digits from a number and modifying the remained digits corresponding to some rule. The rule generally is to make the rounded value as near to the original value as possible. In next two examples, numbers are rounded-off to 2 decimal digits.

12.28392 ~ 12.28

23.45672 ~ 23.46

-6.39812 ~ -6.40

Our main aim in this thesis is to minimize the finite word length noise in IIR filters. This noise occurs due to truncation or roundation. The shorter word length results with more error and more noise in the system. Therefore, without losing the requirements of the filter gain, it is aimed to use the shortest binary coefficients in IIR filter design.

# CHAPTER 2

# DIGITAL FILTERS

## 2.1 Introduction

Filters are widely employed in signal processing and communication systems in applications such as channel equalization, noise reduction, radar, audio processing, video processing, biomedical signal processing, and analysis of economic and financial data [1,2,3].

Owing to the structure of the signal, the filters are designed as analog or digital filters.

- ✓ Analog filters use the analog components such as resistors, capacitors, inductors and/or active components.
- ✓ Digital filters operate in discrete logic and flow of the signal through these networks is discrete with the speed of the clock signal. Digital filters employ the logic components like adder, multiplier and parallel shift registers.

## 2.2 The Digital Filter as a System

A digital filter can be represented by the block diagram of Figure 2.1.

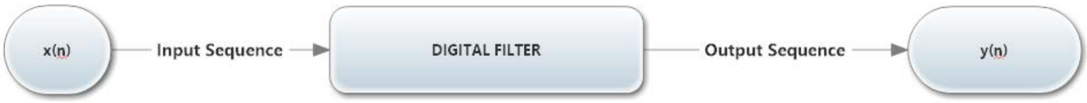

Figure 2. 1 Digital Filter Structure

Input *x(nT)* and output *y(nT)* are the excitation and response of the filter, respectively. The response is related to the excitation by some rule of correspondence. We can indicate this fact notationally as

$$y(nT) = \mathcal{R} * x(nT) \tag{2.1}$$

where $\mathcal{R}$ is an operator.

## 2.3 Linear-Time Invariant (LTI) Digital Filters

A digital filter is linear if and only if it satisfies the conditions

$$\mathcal{R}\alpha x(nT) = \alpha\mathcal{R}x(nT) \tag{2.2}$$

$$\mathcal{R}[x_1(nT) + x_2(nT)] = \mathcal{R}x_1(nT) + \mathcal{R}x_2(nT) \tag{2.3}$$

On the other hand, a digital filter is said to be time-invariant if its response to an arbitrary excitation does not depend on the time of application of the excitation. As in other types of system, the response of a digital filter depends on a number of internal system parameters. In a time-invariant digital filter, these parameters do not change with time.

When a digital filter satisfies both the linearity and time invariance conditions it is called linear-time invariant (LTI) digital filter [4]. The linear time-invariant digital filter can then be described by the linear difference equation:

$$y_n = -a_1 y_{n-1} - a_2 y_{n-2} - \cdots - a_N y_{n-N} + b_0 x_n + \cdots + b_M x_{n-M} \tag{2.4}$$

$$= -\sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k x_{n-k} \tag{2.5}$$

where $a_k$and $b_k$ are real. The order of the filter is $N \geq M$.

## 2.4 Digital Filter Types

Digital filters can be classified in several different groups, depending on what criteria are used for classification. The two major types of digital filters are finite impulse response digital filters (FIR Filters) and infinite impulse response digital filters (IIR Filters).

### 2.4.1 Finite Impulse Response Filters (FIR Filters)

In signal processing, a finite impulse response (FIR) filter is a filter whose 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 (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying).

The impulse response of an Nth-order discrete-time FIR filter (i.e., with a Kronecker delta impulse input) lasts for N + 1 samples, and then settles to zero. The figure of a discrete-time FIR filter of order N is shown in Figure 2.2.

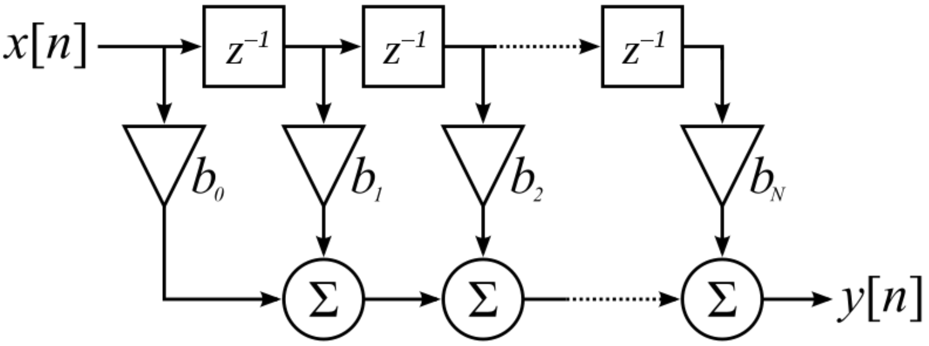

Figure 2. 2 A discrete-time FIR filter of order N. The top part is an N-stage delay line with N + 1 taps. Each unit delay is a z−1 operator in Z-transform notation.

FIR filters can be discrete-time or continuous-time, and digital or analog.

FIR filters have characteristics that make them useful in many applications:

- FIR filters can achieve an exactly linear phase frequency response
- FIR filters cannot be unstable.
- FIR filters are generally less sensitive to coefficient round-off and finite-precision arithmetic than IIR filters.
- FIR filters design methods are generally linear.

The duration or sequence length of the impulse response of these filters is by definition finite; therefore, the output can be written as a finite convolution sum by

$$y(n) = \sum_{m=0}^{N-1} h(m)x(n-m) \tag{2.6}$$

where $n$ and $m$ are are integers, perhaps representing samples in time, and where $x(n)$ is the input sequence, $y(n)$ is the output sequence and $h(n)$ is the length-N impulse response of the filter. With a change of index variables, this can also be written as

$$y(n) = \sum_{m=n}^{n-N+1} h(n-m)x(m) \tag{2.7}$$

### 2.4.2 Infinite Impulse Response Filters (IIR Filters)

A digital filter with impulse response having infinite length (i.e., its values outside any finite interval cannot all be zero) is termed infinite impulse response (IIR) filter. The most important class of IIR filters can be described by the difference equation

$$y(n) = b_0 * x(n) + b_1 * x(n-1) + \cdots + b_M * x(n-M)$$

$$-a_1 * y(n-1) - a_2 * y(n-2) - \cdots - a_N * y(n-N) \tag{2.8}$$

where $x(n)$ is the input, $y(n)$ is the output of the filter, $\{a_1, a_2, \dots, a_N\}$ and $\{b_1, b_2, \dots, b_M\}$ are the filter coefficients. We assume that $a_N \neq 0$. The impulse response is the output of the system when it is driven by a unit impulse at n = 0, with the system being initially at rest, i.e., the output being zero prior to applying the input. We denote the impulse response by *h(n)*. With *x(0)* = 1, *x(n)* = 0, for $n \neq 0$, and *y(n)* = 0 for $n < 0$, we can compute *h(n)*, $n \leq 0$, from above equation in a recursive manner. Taking the z-transform of the Equation 2.8, we obtain the system function

$$\mathrm{H(z)} = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 * z^{-1} + \cdots + b_M * z^{-M}}{1 + a_1 * z^{-1} + a_2 * z^{-2} + \cdots + a_N * z^{-N}} \tag{2.9}$$

N is the order of the filter. The system function and the impulse response are related through the z-transform and its inverse, i.e.,

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) * z^{-n} \qquad h(n) = \frac{1}{2\pi j}\oint_C H(z) * z^{n-1} * dz \qquad (2.10 \,\&\, 2.11)$$

where C is a closed counterclockwise contour in the region of convergence.

For ease of implementation, it is desirable that the coefficients $\{a_1, a_2, \ldots, a_N\}$ and $\{b_1, b_2, \ldots, b_M\}$ be real numbers (as opposed to complex numbers), which is another assumption that we make, unless it is specified otherwise.

A realization of an IIR filter, is shown in Figure 2.3(a), which is called Direct Form I. By rearranging the structure, we can obtain Direct Form II, as shown in Figure 2.3(b). Through transposition, we can obtain Transposed Direct Form I and Transposed Direct Form II as shown in Figure 2.3(c) and (d).

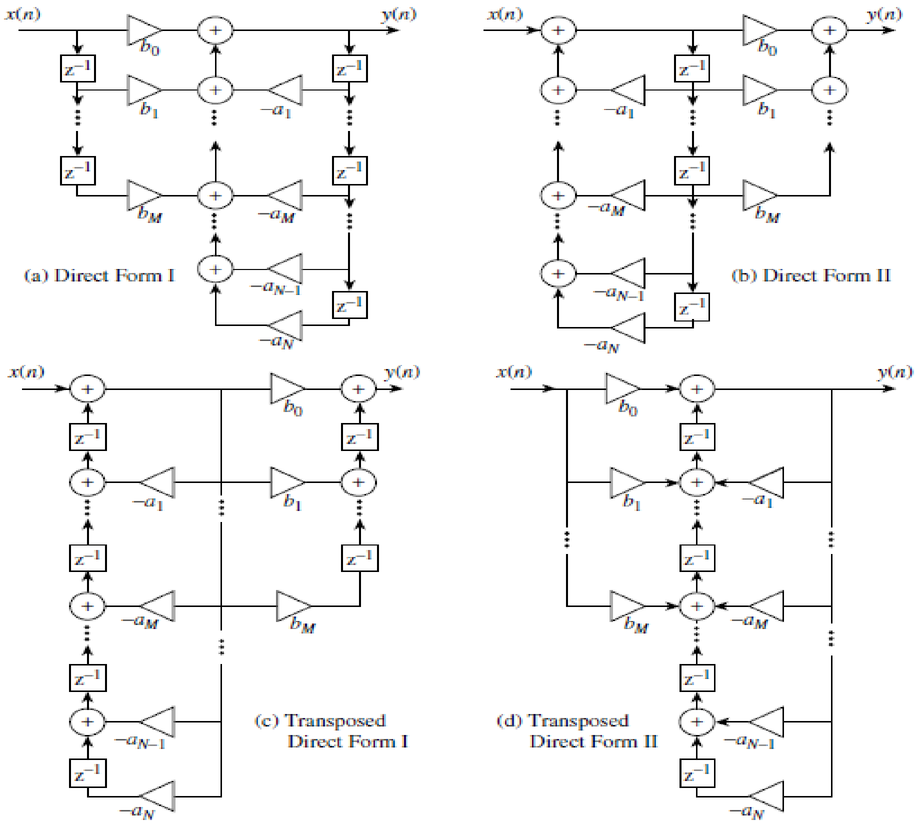

Figure 2. 3 Direct form realizations of IIR filters

## 2.5 The Pulse Transfer Function

The pulse transfer function is the ratio of $z$-transform of output to $z$-transform of input.

Let the impulse response, for example of an FIR filter, be $a_0$ at $t = 0$, $a_1$ at $t = T, \ldots, a_i$ at $t = iT$ with $i = 0, \ldots, N$.

Let $G(z)$ be the z-transform of this sequence:

$$G(z) = a_0 + a_1 * z^{-1} + a_2 * z^{-2} + \cdots + a_i * z^{-i} + \cdots + a_N * z^{-N} \qquad (2.12)$$

Let $X(z)$ be an input:

$$X(z) = x[0] + x[1] * z^{-1} + x[2] * z^{-2} + \cdots + x[k] * z^{-k} \qquad (2.13)$$

The product $G(z)X(z)$ is:

$$G(z)X(z) = \left(a_0 + a_1 * z^{-1} + a_2 * z^{-2} + \cdots + a_i * z^{-i} + \cdots + a_N * z^{-N}\right) * \\ (x[0] + x[1] * z^{-1} + x[2] * z^{-2} + \cdots + x[k] * z^{-k}) \qquad (2.14)$$

in which the coefficient of $z^{-k}$ is:

$$a_0 * x[k] + a_1 * x[k-1] + \cdots + a_i * x[k-i] + \cdots + a_N * x[k-N] \qquad (2.15)$$

This is nothing else than the value of the output sample at $t = kT$. Hence the whole sequence is the z-transform of the output, say $Y(z)$, where $Y(z) = G(z)X(z)$. Hence the pulse transfer function, $G(z)$, is the z-transform of the impulse response.

For non-recursive filters (FIR Filters):

$$G(z) = \sum_{i=0}^{N} a_i * z^{-i} \qquad (2.16)$$

For recursive filters (IIR Filters)

$$Y(z) = \sum_{i=0}^{N} a_i * z^{-i} * X(z) + \sum_{i=1}^{M} b_i * z^{-i} * Y(z) \qquad (2.17)$$

$$G(z) = \frac{Y(z)}{X(z)} = \frac{\sum a_i * z^{-i}}{1 - \sum b_i * z^{-i}} \qquad (2.18)$$

## 2.6 Frequency Response of a Digital Filter

Frequency response of a digital filter can be obtained by evaluating the (pulse) transfer function on the unit circle ($i.e. z = e^{jwT}$).

*Proof*

Consider the general filter difference equation is [5]

$$y[k] = \sum_{i=0}^{\infty} a_i * x[k-i] \qquad (2.19)$$

**NB**: A recursive type can always be expressed as an infinite sum by dividing out:

e.g., for

$$G(z) = \frac{a_0}{1 - b_1 * z^{-1}} \tag{2.20}$$

we have

$$y[k] = \sum_{i=0}^{\infty} a_0 * b_1^i * x[k-i] \tag{2.21}$$

Let input before sampling be $\cos(wt + \theta)$, sampled at $t = 0, T, \ldots, kT$. Therefore

$$x[k] = \cos(wkt + \theta) = \frac{1}{2}\left\{e^{j(wkT+\theta)} + e^{-j(wkT+\theta)}\right\} \tag{2.22}$$

i.e.

$$\frac{1}{2}\sum_{i=0}^{\infty} a_i * e^{j\{w[k-i]T+\theta\}} + \sum_{i=0}^{\infty} a_i * e^{-j\{w[k-i]T+\theta\}} \tag{2.23}$$

$$= \frac{1}{2}e^{j(wkT+\theta)}\sum_{i=0}^{\infty} a_i * e^{-jwiT} + \frac{1}{2}e^{-j(wkT+\theta)}\sum_{i=0}^{\infty} a_i * e^{jwiT} \tag{2.24}$$

Now

$$\sum_{i=0}^{\infty} a_i * e^{-jwiT} = \sum_{i=0}^{\infty} a_i * (e^{jwT})^{-i} \tag{2.25}$$

But G(z) for this filter is $\sum_{i=0}^{\infty} a_i * z^{-i}$

and so

$$\sum_{i=0}^{\infty} a_i * e^{-jwiT} = G(z)_z = e^{jwT} \tag{2.26}$$

Let $G(z)_z = e^{jwT} = Ae^{j\emptyset}$, then

$$\sum_{i=0}^{\infty} a_i * e^{jwiT} = Ae^{-j\emptyset} \qquad (complex\ conjugate) \tag{2.27}$$

Hence

$$y[k] = \frac{1}{2}e^{j(wkT+\theta)}Ae^{j\emptyset} + \frac{1}{2}e^{-j(wkT+\theta)}Ae^{-j\emptyset} \tag{2.28}$$

or

$$y[k] = Acos(wkT + \theta + \emptyset) \qquad when\ x[k] = \cos(wkT + \theta) \tag{2.29}$$

Thus $A\ and\ \emptyset$ represent the gain and phase of the frequency response. i.e. the frequency response (as a complex quantity) is

$$G(z)|_{z=e^{jwT}} = G\left(e^{jwT}\right) = A(wT)e^{j\emptyset T} \qquad (2.30)$$

In design process of the digital filters, in general, the gain is given (or phase) and the correct coefficients for FIR and/or IIR structures are found [6].

# CHAPTER 3
# FINITE WORD LENGTH

## 3.1 Introduction

Practical digital filters must be executed with limited length and arithmetic. Thus, the coefficients of the filter and the filter signals in input and output are in discrete form. This guides to 4 types of finite wordlength effects [8,9,10,11,12,13].

Filter coefficients discretization (quantization) has the effect of disordering the filter poles and zeroes' locations. Thus, the real response of filter differs delicately from the ideal response of the filter. This *decisive* frequency response error is applied to as **coefficient quantization error** [14].

Using limited length arithmetic makes it require to make discrete filter calculations by rounding or truncation. The error seen in filter output which results because of rounding or truncating inwith the filter is called as **Roundoff noise** [15].

Quantization (the process of approximating a continuous signal by a set of discrete symbols or integer values) of the filter calculations also cause the filter to become delicately nonlinear. For huge signals this nonlinearity can be excluded and roundoff noise is the main interest. Nevertheless, for infinite impulse response filters (recursive filters) with a zero or constant input, this nonlinearity can set off fake oscillations called as **limit cycles** [16].

It is possible for filter calculations to exceed with fixed-point arithmetic. A high-level oscillation which may be in a different stable filter because of the nonlinearity join with the overflow of internal filter calculations is called as **overflow oscillation** which is sometimes also called as **adder overflow limit cycle** [17].

## 3.2 MATLAB "yulewalk" Subroutine

*yulewalk* lays out recursive infinite impulse response digital filters using a least-squares fit to a indicated frequency response. [b,a] = yulewalk(n,f,m) returns row vectors b and a containing the n+1 coefficients of the order n infinite impulse response filter whose frequency-magnitude characteristics approximately match those given in vectors f and m [7]:

- f: vector of frequency points, indicated in the range between 0 and 1, where 1 corresponds to the Nyquist frequency. The first point of f must be 0 and the last point 1, with all intermediate points in increasing order.
- m is a vector including the wanted magnitude response at the points indicated in f.

The output filter coefficients are ordered in descending powers of *z*.

$$\frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \cdots + b(n+1)z^{-m}}{a(1) + a(2)z^{-1} + \cdots + a(n+1)z^{-n}} \tag{3.16}$$

*yulewalk* performs a least-squares fit in the time domain. It computes the denominator coefficients using modified Yule-Walker equations, with correlation coefficients computed by inverse Fourier transformation of the specified frequency response.

### 3.3 Truncation & Roundation

*Truncation* is the term for limiting the number of digits right of the decimal point, by discarding the least significant ones.

For example, consider the real numbers:

1.25987416657955441

-15.233333333333

are the outcome of yulewalk representing the coefficients of the IIR digital filter.

To truncate these numbers to 4 decimal digits, we only consider the 4 digits to the right of the decimal point.

The result would be:

1.2598

-15.2333

With these new values the gain of the filter may change. If this change remains in acceptable ranges, we prefer to use the shorter numbers.

*Rounding* a numerical value means replacing it by another value that is approximately equal but has a shorter, simpler, or more explicit representation.

We will take the same examples that we have defined for truncation with a different process. To round these numbers to 4 decimal digits, this time we consider 5 digits to the right of the decimal point. If the 5th digit is equal or greater than 5, we will add 1 to 4th digit. If not it will be same as truncation. By this manner the result would be:

1.2599

-15.2333

In both cases, there will be errors between the new number and the original one. Sometimes truncation sometimes roundation will give less error. The error here is the deviation from the ideal characteristics and the new gain may remain in tolerances.

# CHAPTER 4

# APPLICATIONS

In this chapter, the method of roundation and truncation to limit the wordlengths of the coefficients of the IIR filter will be presented as the program written for MATLAB. The program will be run for different types of filters such as Low Pass Filter and Band Pass Filter.

## 4.1 Program

The program is written in MATLAB environment. MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

## 4.2 Designs

In this section, the main part of the program which applies truncation and roundation are given for both Low Pass Filter and Band Pass Filter applications. The program is run for different bit lengths.

### 4.2.1 Design for Low Pass Filter

### 4.2.1.1 Main Program

This part is the main graphical user interface of the program. It calls other subroutine programs and it asks for an input either the user wants to apply truncation or roundation.

```
close all
clear all
clc

selection=input('Enter 0 if you want to make Truncation, enter 1 if you want to make
Roundation: ');
```

```
if selection==0
    [f,m,h,h1,y,z,a,anew,b,bnew]= Truncation();
    elseif selection==1
     [f,m,h,h1,y,z,a,anew,b,bnew]= Roundation();
end
```

### 4.2.1.2 Program of Truncation Part

The theoretical explanation of “yulewalk” command is given in Section 3.2. The theoretical explanation of  the truncation and the roundation process is given in section 3.3. Here the source code is given, which finds the N-th order recursive filter coefficients b and a, convert the coefficients to binary, then make truncation process and convert them back to decimal. Finally it draws the figures which are ideal, yulewalk found and with the coefficients that are changed by truncation.

```
%Truncation

function [f,m,h,h1,y,z,a,anew,b,bnew]= Truncation()

Kp=input('Enter attenuation value in pass band region:');
Ks=input('Enter attenuation value in stop band region:');

X=1/(10^(Kp/20));
X1=1/(10^(Ks/20));

Wp=input('Enter cut-off frequency value:');
Wsf=input('Enter sampling frequency value:');

f=[linspace(0,(Wp/Wsf),64) linspace((Wp/Wsf),1,64)];
m=[ones(1,64)*((X+1)/2) ones(1,64)*(X1/2)];

n=5; %order of filter
nd=input('Enter the desired length of binary number: '); %length of binary number

[b,a]=yulewalk(n,f,m)

signb=sign(b);
signa=sign(a);

[h,w] = freqz(b,a,128);
plot(f,m,w/pi,abs(h),'--')
hold on
title('Comparison of Frequency Response Magnitudes')

%%
for kk=1:n+1
    aa=abs(b(kk))
```

```
for k=1:nd;
    aa=aa*2;
    if (aa<=1) y(kk,k)=0
    else
        y(kk,k)=1
        aa=aa-1;
    end
end
end
%%
b=zeros(1,n+1);
for kk=1:n+1
sum=0
for k=1:nd;
    sum=sum+y(kk,k)*2^(-k);
end
b(kk)=sum
end
bnew=b.*signb
%%
for kk=1:n+1
    aa1=abs(a(kk))
for k=1:nd;
    aa1=aa1*2;
    if (aa1<=1) z(kk,k)=0
    else
        z(kk,k)=1
        aa1=aa1-1;
    end
end
end

%%

a=zeros(1,n+1)
for kk=1:n+1
sum=0
for k=1:nd;
    sum=sum+z(kk,k)*2^(-k);
end
a(kk)=sum
end
anew=a.*signa

%%

[h1,w] = freqz(bnew,anew,128);
plot(f,m,w/pi,abs(h1),'*r')
legend('Ideal','yulewalk Designed','inputs changed')
title('Comparison of Frequency Response Magnitudes')
```

### 4.2.1.3 Program of Roundation Part

Here the source code which finds the N-th order recursive filter coefficients b and a, convert the coefficients to binary, then make roundation process and convert them back to decimal is given. It draws the figures which are ideal, yulewalk found and with the coefficients that are changed by roundation.

```
%Roundation

function [f,m,h,h1,y,z,a,anew,b,bnew]= Roundation()

Kp=input('Enter attenuation value in pass band region:');
Ks=input('Enter attenuation value in stop band region:');
X=1/(10^(Kp/20));
X1=1/(10^(Ks/20));

Wp=input('Enter cut-off frequency value:');
Wsf=input('Enter sampling frequency value:');

f=[linspace(0,(Wp/Wsf),64) linspace((Wp/Wsf),1,64)];
m=[ones(1,64)*((X+1)/2) ones(1,64)*(X1/2)];

n=5; %order of filter
nd=input('Enter the desired length of binary number: '); %length of binary number

[b,a]=yulewalk(n,f,m)

signb=sign(b);
signa=sign(a);

[h,w] = freqz(b,a,128);
plot(f,m,w/pi,abs(h),'--')
hold on
title('Comparison of Frequency Response Magnitudes')

%%

for kk=1:n+1
   aa=abs(b(kk))
for k=1:nd+1;
   aa=aa*2;
   if (aa<=1) y(kk,k)=0
   else
      y(kk,k)=1
      aa=aa-1;
   end
end
end
```

```
%%

b=zeros(1,n+1);
for kk=1:n+1
sum=0
for k=1:nd;
    sum=sum+y(kk,k)*2^(-k);
end
b(kk)=sum
end

%Roundation part
for i=1:n+1
    if y(i,nd+1)==1
        b(i)=b(i)+2^(-(nd+1))
    else
    end
end
bnew=b.*signb
%%
for kk=1:n+1
    aa1=abs(a(kk))
for k=1:nd+1;
    aa1=aa1*2;
    if (aa1<=1) z(kk,k)=0
    else
        z(kk,k)=1
        aa1=aa1-1;
    end
end
end
%%
a=zeros(1,n+1)
for kk=1:n+1
sum=0
for k=1:nd;
    sum=sum+z(kk,k)*2^(-k);
end
a(kk)=sum
end

%Roundation Part
for i=1:n+1
    if z(i,nd+1)==1
        a(i)=a(i)+2^(-(nd+1))
    else
    end
end
anew=a.*signa
```

```
%%
[h1,w] = freqz(bnew,anew,128);
plot(f,m,w/pi,abs(h1),'*r')
legend('Ideal','yulewalk Designed','inputs changed')
title('Comparison of Frequency Response Magnitudes')
```

#### 4.2.1.4 Computation of Whole Error

```
%Error Computation

error=abs(h)-m';
errorsq=error.^2;

errornew=sum(errorsq)  %Error Of Yulewalk

errorr=abs(h1)-m';

errorrsq=errorr.^2;
errorneww=sum(errorrsq) %Error Of The Inputs We Changed
```

#### 4.2.1.5 Computation of Error of Passband-Stopband Region

```
%Error Computation of PassBand&Stopband
error=abs(h)-m';

errorsq=error.^2;

error1=sum(errorsq(1:64))  %Error Of Yulewalk PassBand

error2=sum(errorsq(65:128)) %Error Of Yulewalk StopBand

errorr=abs(h1)-m';

errorrsq=errorr.^2;

errorneww1=sum(errorrsq(1:64)) %Error Of The Inputs We Changed PassBand

errorneww2=sum(errorrsq(65:128)) %Error Of The Inputs We Changed StopBand
```

### 4.3 Results for Low Pass Filter

#### 4.3.1 Truncation Applications for Low Pass Filter

As a first example, the design of the low pass filter with 7.66 dB maximum attenuation in pass-band region, 13.9794 dB minimum attenuation in stop-band region, pass-band cut off frequency of 40 rad/sec. and sampling frequency of 100 rad/sec. is studied. Ideal gain, the gain with yulewalk (MATLAB subroutine) and the gain after coefficients are truncated are sketched.

- **For 3-bits:**

The low pass filter gain characteristic is obtained for 3-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.1.

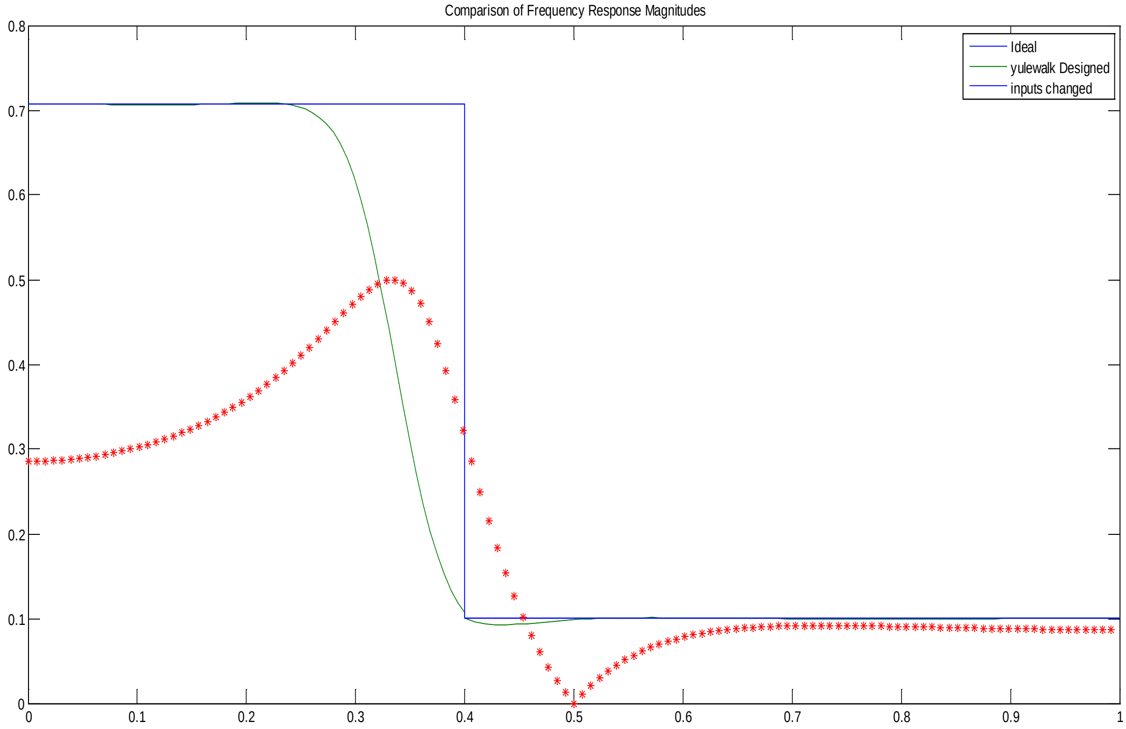

Figure 4. 1 Comparison of Frequency Response Magnitudes (3-bits) x-axis:Frequency y-axis:Gain

From the datas taken from MATLAB;

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

Table 1 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 10.3390 | 10.2872 | 0.0518 |

Table 2 Errors of Ideal vs. Input Parameters are Truncated (3-bits)

➢ **For 5-bits:**

The low pass filter gain characteristic is obtained for 5-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.2.

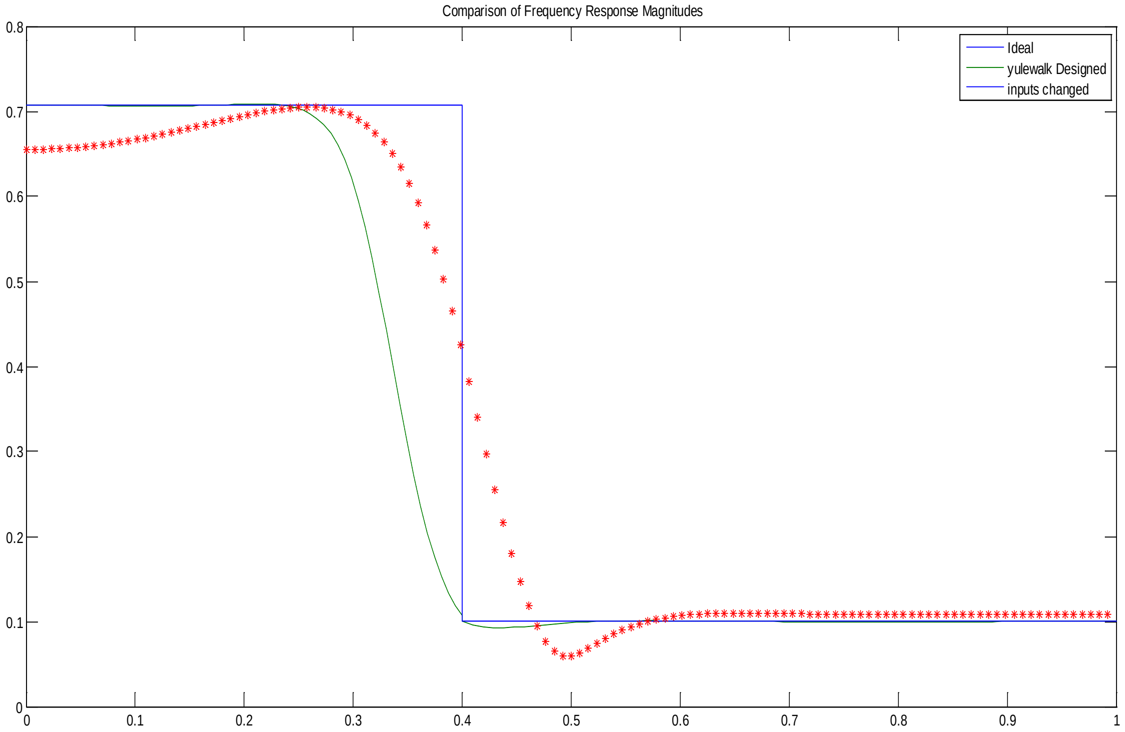

**Figure 4. 2 Comparison of Frequency Response Magnitudes (5-bits) x-axis:Frequency y-axis:Gain**

From the datas taken from MATLAB;

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

**Table 3 Errors of Ideal vs. Yulewalk Subroutine Designed**

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 3.7075 | 3.6983 | 0.0093 |

**Table 4 Errors of Ideal vs. Input Parameters are Truncated (5-bits)**

- **For 8-bits:**

The low pass filter gain characteristic is obtained for 8-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.3.

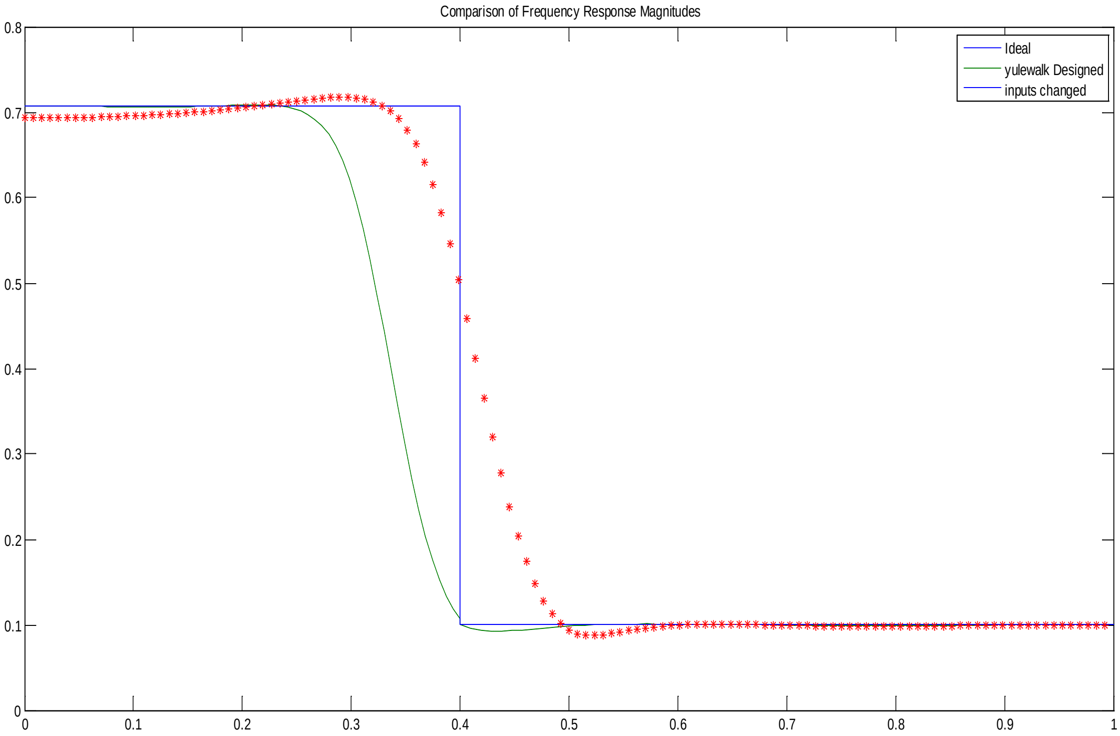

Figure 4. 3 Comparison of Frequency Response Magnitudes (8-bits) x-axis:Frequency y-axis:Gain

From the datas taken from MATLAB:

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

Table 5 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8257 | 2.8248 | 8.9811e-004 |

Table 6 Comparison of Frequency Response Magnitudes (8-bits)

- **For 16-bits:**

The low pass filter gain characteristic is obtained for 16-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.4.

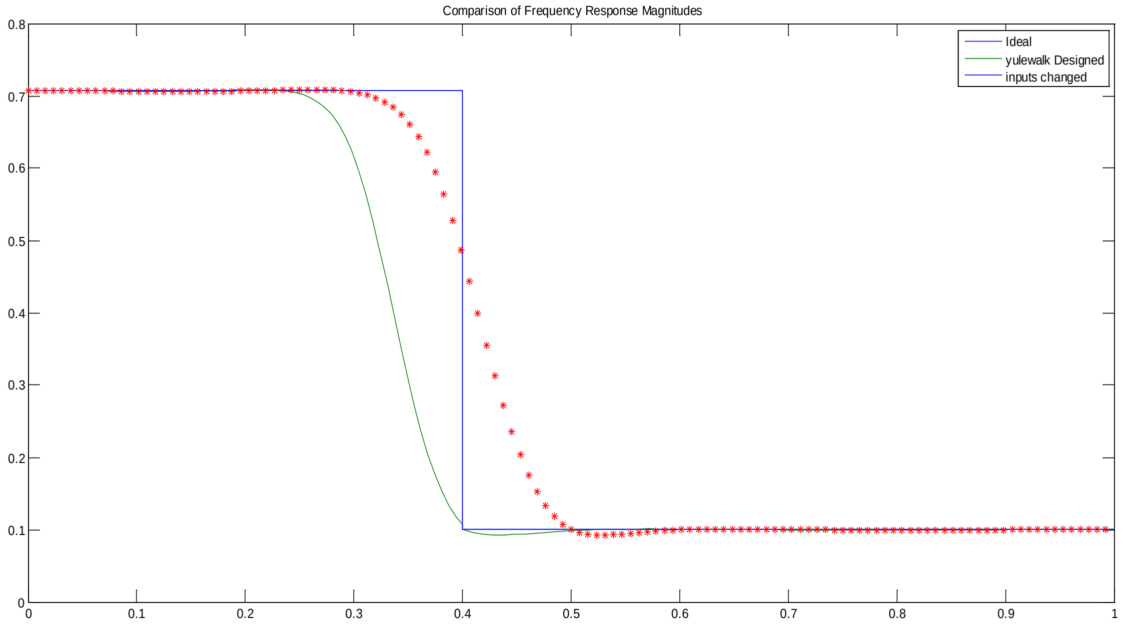

Figure 4. 4 Comparison of Frequency Response Magnitudes (16-bits) x-axis:Frequency y-axis:Gain

From the datas taken from MATLAB:

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

Table 7 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8596 | 2.8593 | 3.1118e-004 |

Table 8 Errors of Ideal vs. Input Parameters are Truncated (16-bits)

- If we compare the figures and datas taken from MATLAB, it can be said that that the errors taken from between 5-8 bits are acceptable. Specifically for our datas 8-bit is more acceptable for truncation process for Low Pass Filter.

### 4.3.2 Roundation Applications for Low Pass Filter

As the second example, the design of the low pass filter with 7.66 dB maximum attenuation in pass-band region, 13.9794 dB minimum attenuation in stop-band region, pass-band cut off frequency of 40 rad/sec. and sampling frequency of 100 rad/sec. is studied. Ideal gain, the gain with yulewalk (MATLAB subroutine) and the gain after coefficients are rounded-off are sketched.

- **For 3-bits:**

The low pass filter gain characteristic is obtained for 3-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.5.

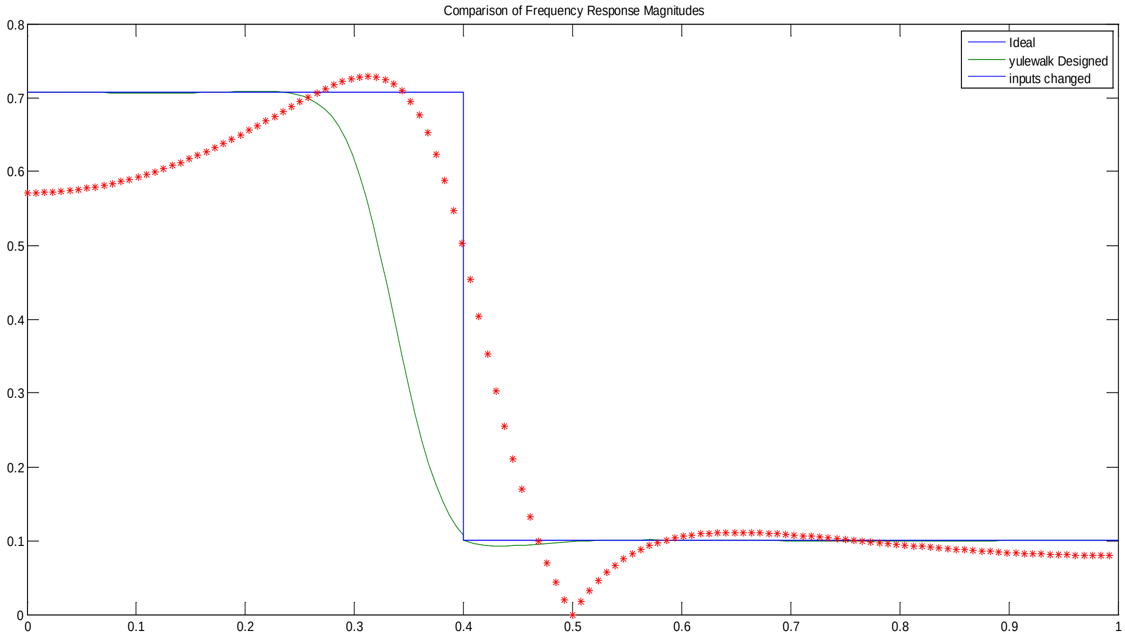

Figure 4. 5 Comparison of Frequency Response Magnitudes (3-bits) x-axis:Frequency y-axis:Gain

From the datas taken from MATLAB:

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

Table 9 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 3.6622 | 3.6265 | 0.0358 |

Table 10 Errors of Ideal vs. Input Parameters are Rounded (3-bits)

- **For 5-bits:**

The low pass filter gain characteristic is obtained for 5-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.6.

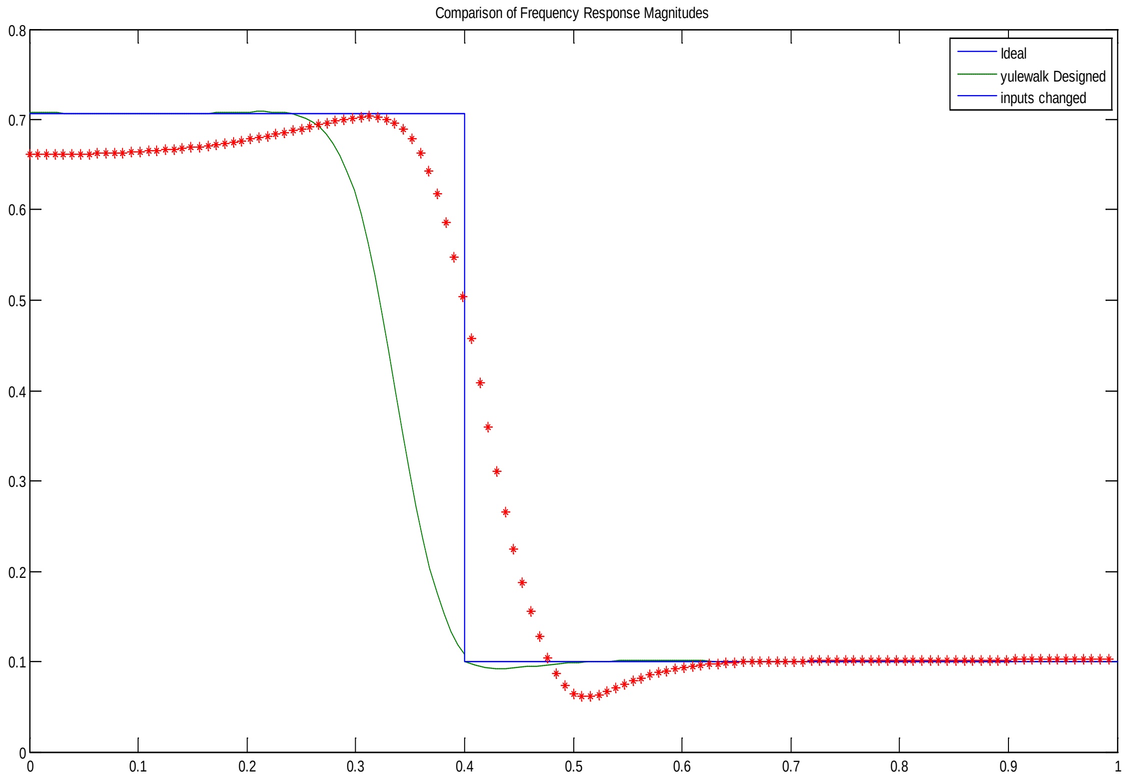

**Figure 4. 6 Comparison of Frequency Response Magnitudes (5-bits) x-axis:Frequency y-axis:Gain**

From the datas taken from MATLAB:

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

**Table 11 Errors of Ideal vs. Yulewalk Subroutine Designed**

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

**Table 12 Errors of Ideal vs. Input Parameters are Rounded (5-bits)**

- **For 8-bits:**

The low pass filter gain characteristic is obtained for 8-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.7.

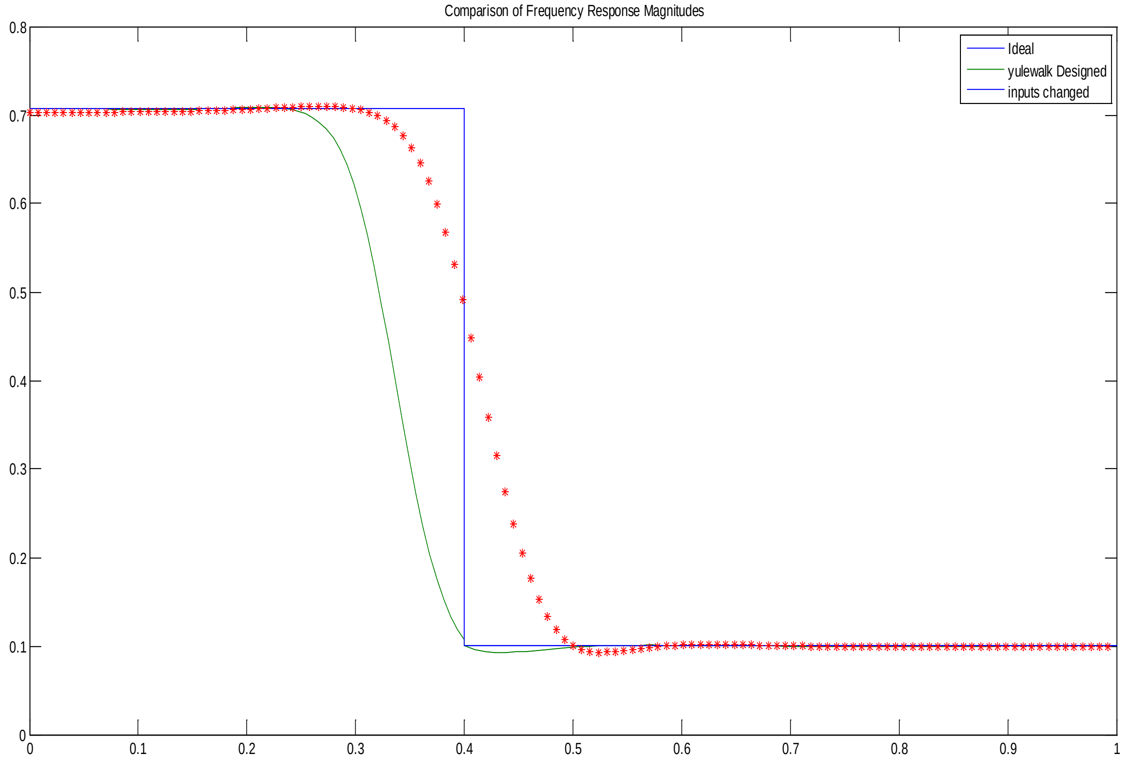

Figure 4. 7 Comparison of Frequency Response Magnitudes (8-bits) x-axis:Frequency y-axis:Gain

From the datas taken from MATLAB:

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

Table 13 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8393 | 2.8390 | 2.6430e-004 |

Table 14 Errors of Ideal vs. Input Parameters are Rounded (8-bits)

- **For 16-bits:**

The low pass filter gain characteristic is obtained for 16-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.8.

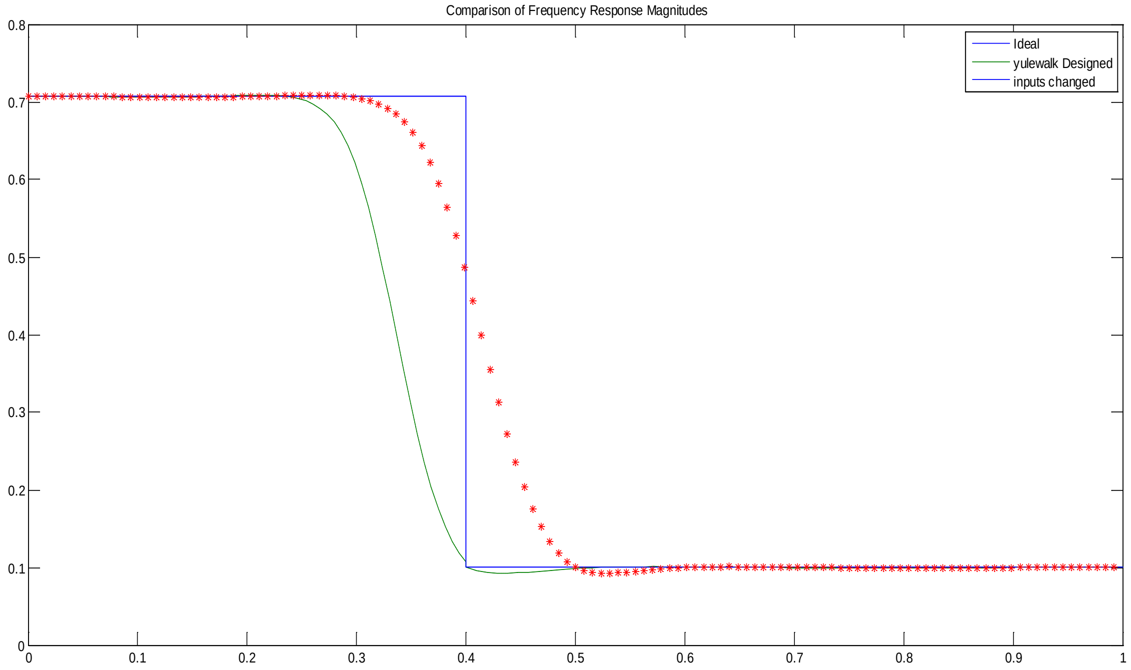

Figure 4. 8 Comparison of Frequency Response Magnitudes (16-bits) x-axis:Frequency y-axis:Gain

From the datas taken from MATLAB:

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8595 | 2.8592 | 3.0993e-004 |

Table 15 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.8598 | 2.8595 | 3.0997e-004 |

Table 16 Errors of Ideal vs. Input Parameters are Rounded (16-bits)

- If we compare the figures and datas taken from MATLAB, it can be said that the errors taken from between 3-5 bits are acceptable. Specifically for our datas 5-bit is more acceptable for roundation process for Low Pass Filter.

## 4.4 Results for Band Pass Filter

### 4.4.1 Truncation Applications for Band Pass Filter

As a first example for Band Pass Filter, the design of the band pass filter with 7.66 dB maximum attenuation in pass-band region, 13.9794 dB minimum attenuation in stop-band region, pass-band cut off frequency of 30 rad/sec. and sampling frequency of 70 rad/sec. is studied. Ideal gain, the gain with yulewalk (MATLAB subroutine) and the gain after coefficients are truncated are sketched.

- **For 3-bits:**

The band pass filter gain characteristic is obtained for 3-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.9.

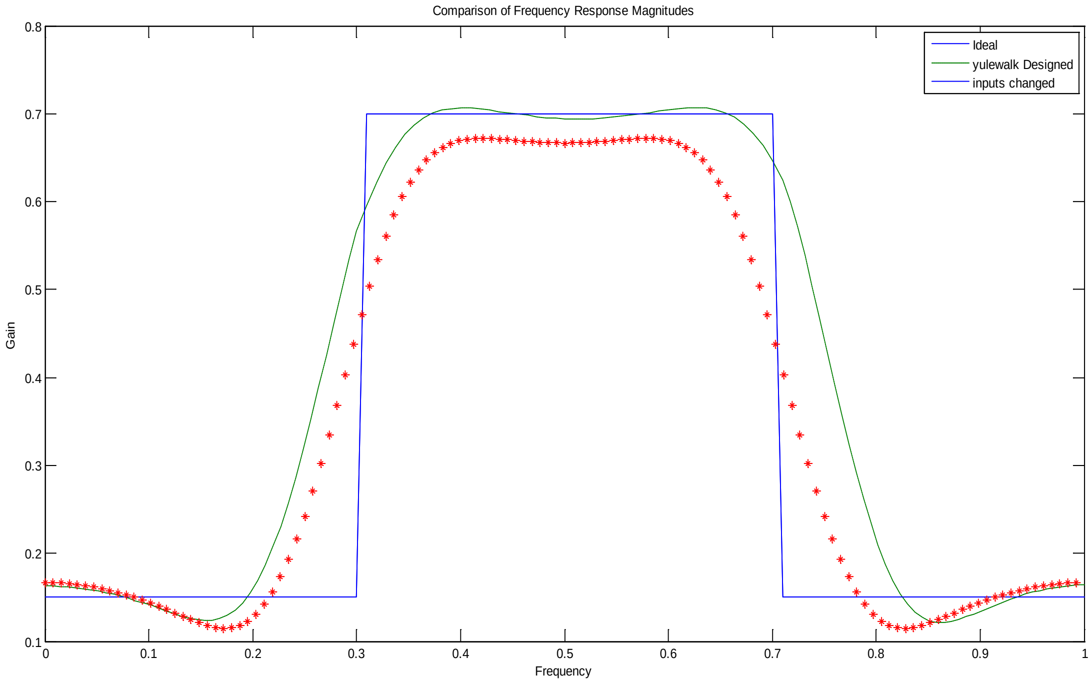

Figure 4. 9 Comparison of Frequency Response Magnitudes (3-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 17 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 1.6300 | 0.1235 | 1.5065 |

Table 18 Errors of Ideal vs. Input Parameters are Truncated (3-bits)

- **For 5-bits:**

The band pass filter gain characteristic is obtained for 5-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.10.

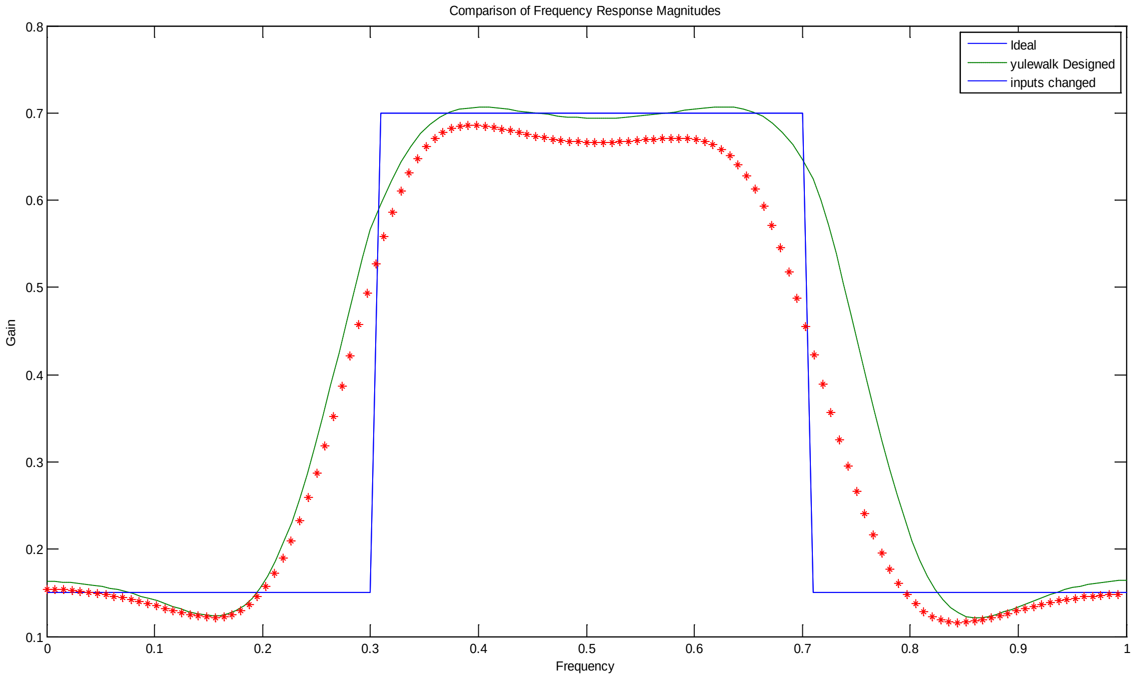

Figure 4. 10 Comparison of Frequency Response Magnitudes (5-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 19 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 1.9785 | 0.0753 | 1.9032 |

Table 20 Errors of Ideal vs. Input Parameters are Truncated (5-bits)

- **For 8-bits:**

The band pass filter gain characteristic is obtained for 8-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.11.

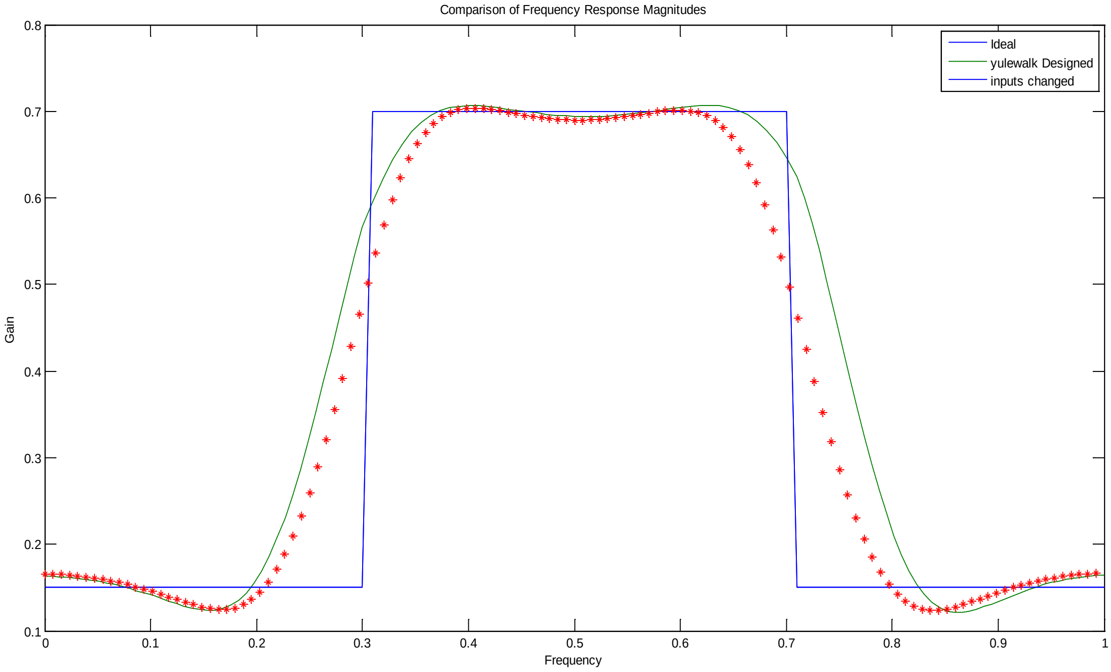

Figure 4. 11 Comparison of Frequency Response Magnitudes (8-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 21 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0320 | 0.0296 | 2.0024 |

Table 22 Errors of Ideal vs. Input Parameters are Truncated (8-bits)

- **For 16-bits:**

The band pass filter gain characteristic is obtained for 16-bit truncated coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with truncated coefficients are sketched in Figure 4.12.

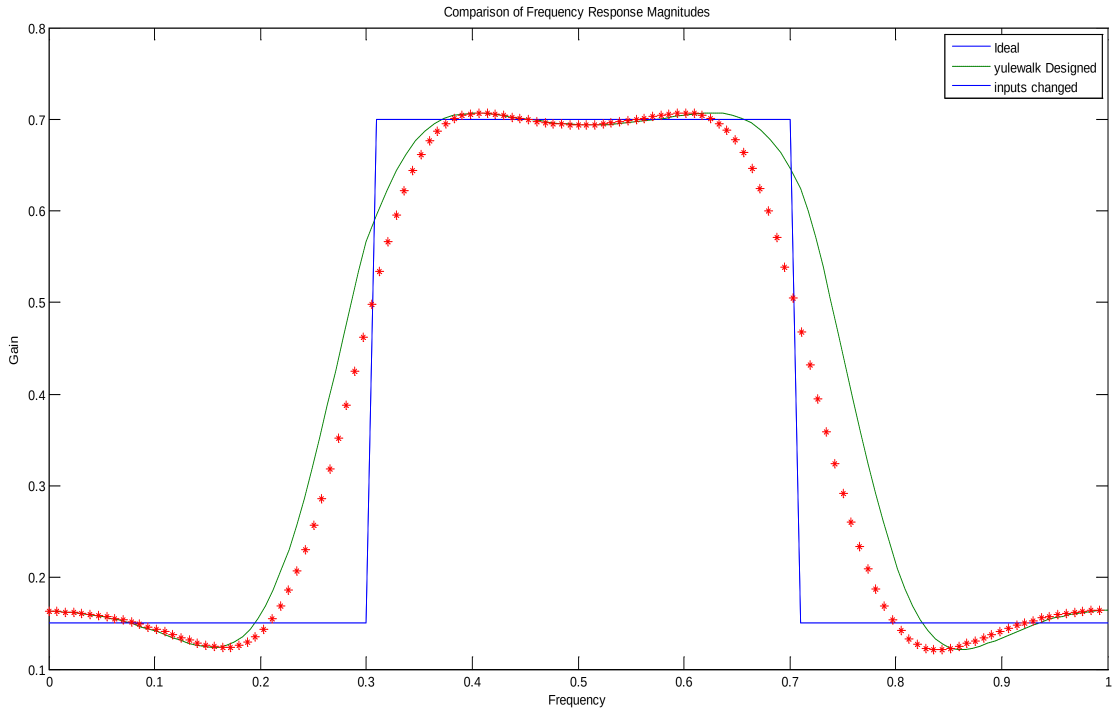

Figure 4. 12 Comparison of Frequency Response Magnitudes (16-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 23 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0632 | 0.0279 | 2.0353 |

Table 24 Errors of Ideal vs. Input Parameters are Truncated (16-bits)

- If we compare the figures and datas taken from MATLAB, it can be said that all stop-band errors which are taken after coefficients are truncated are less than the yulewalk subroutine found. But on the other hand, comparing the band-pass errors; we see that 5-8 bit is acceptable.

### 4.4.2 Roundation Applications for Band Pass Filter

As the second example for Band Pass Filter, the design of the band pass filter with 7.66 dB maximum attenuation in pass-band region, 13.9794 dB minimum attenuation in stop-band region, pass-band cut off frequency of 30 rad/sec. and sampling frequency of 70 rad/sec. is studied. Ideal gain, the gain with yulewalk (MATLAB subroutine) and the gain after coefficients are rounded are sketched.

- **For 3-bits:**

The band pass filter gain characteristic is obtained for 3-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.13.

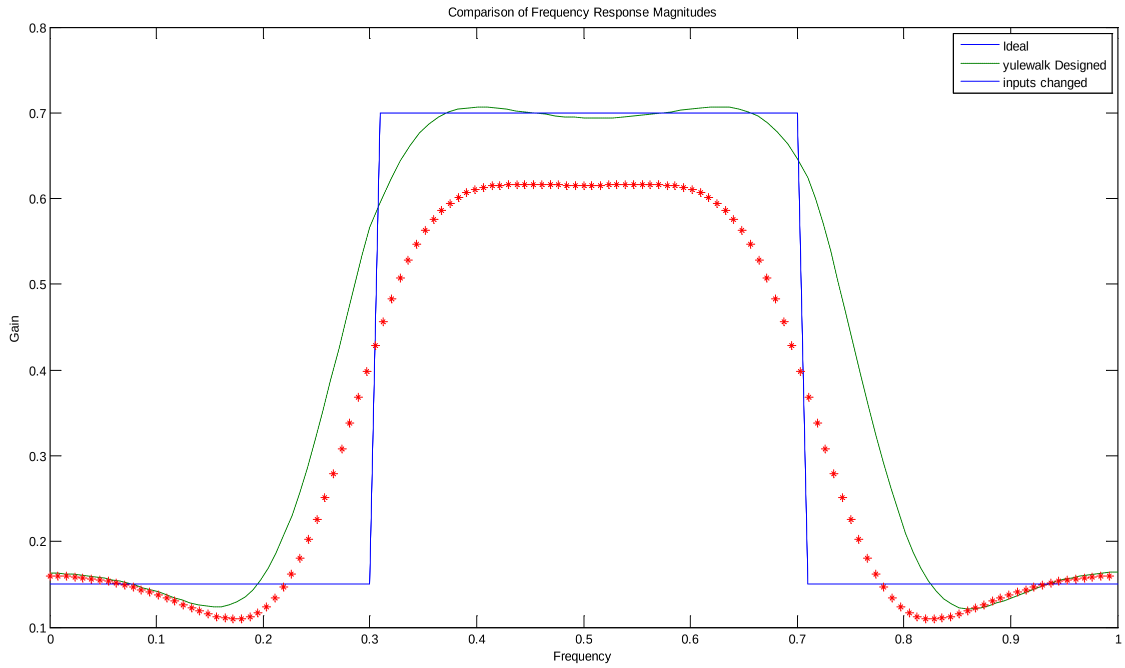

Figure 4. 13 Comparison of Frequency Response Magnitudes (3-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 25 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 1.6250 | 0.4908 | 1.1342 |

Table 26 Errors of Ideal vs. Input Parameters are Rounded (3-bits)

- **For 5-bits:**

The band pass filter gain characteristic is obtained for 5-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.14.

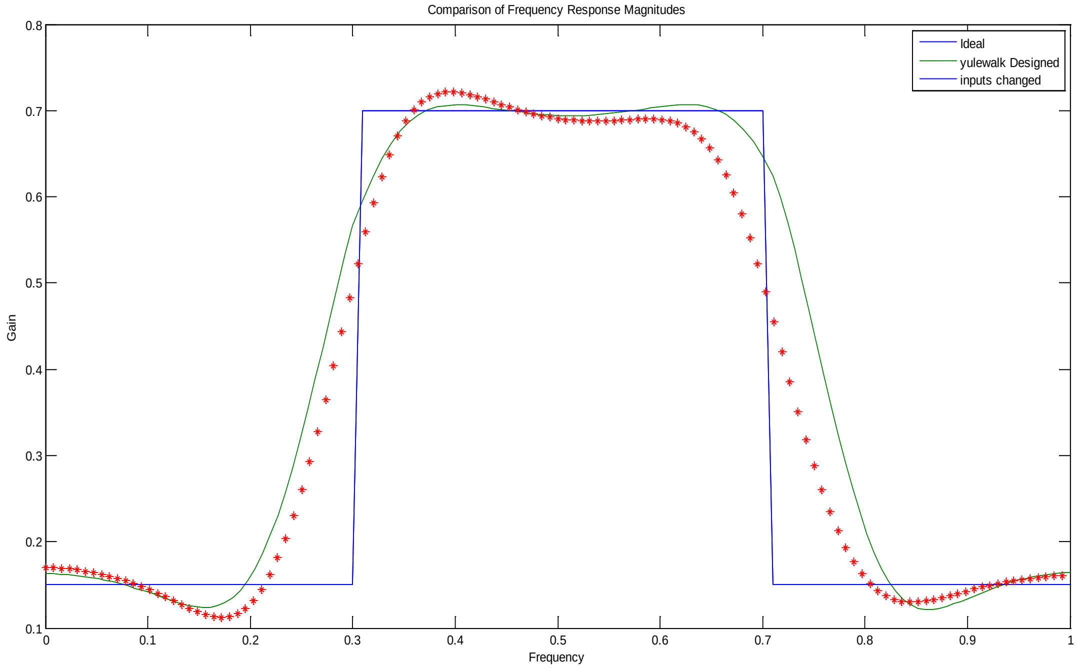

Figure 4. 14 Comparison of Frequency Response Magnitudes (5-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 27  Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0732 | 0.0271 | 2.0461 |

Table 28 Errors of Ideal vs. Input Parameters are Rounded (5-bits)

- **For 8-bits:**

The band pass filter gain characteristic is obtained for 8-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.15.

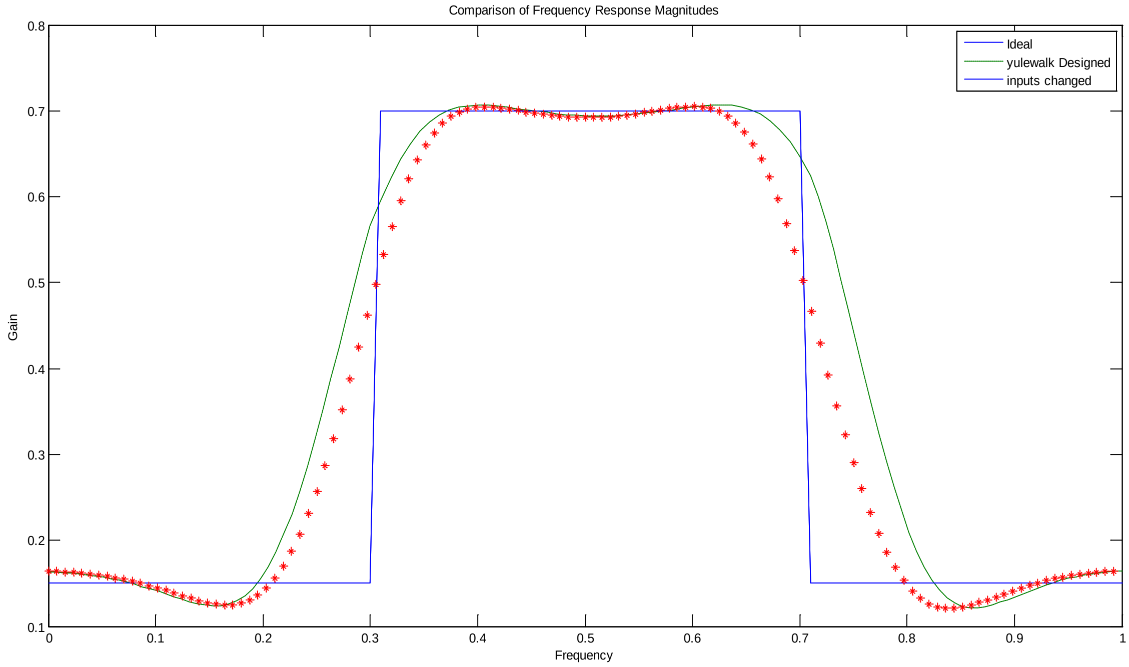

Figure 4. 15 Comparison of Frequency Response Magnitudes (8-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 29 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0526 | 0.0290 | 2.0236 |

Table 30 Errors of Ideal vs. Input Parameters are Rounded (8-bits)

- **For 16-bits:**

The band pass filter gain characteristic is obtained for 16-bit rounded coefficients and the figure with ideal filter characteristic, the figure with yulewalk subroutine and the figure with rounded coefficients are sketched in Figure 4.16.

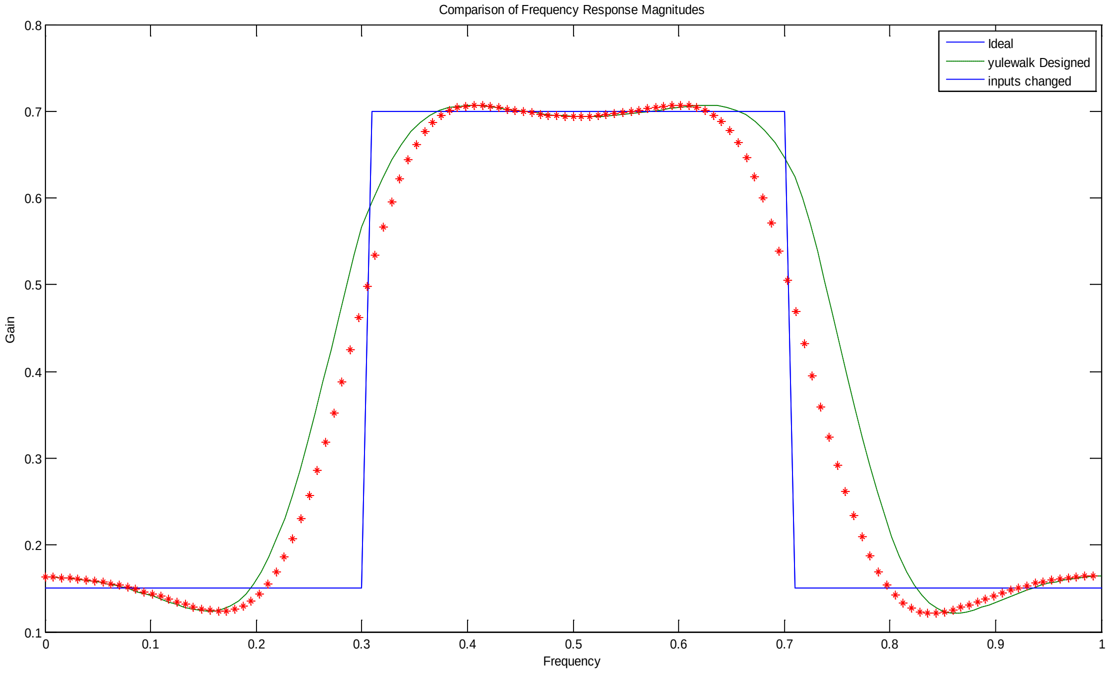

Figure 4. 16 Comparison of Frequency Response Magnitudes (16-bits)

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0635 | 0.0278 | 2.0356 |

Table 31 Errors of Ideal vs. Yulewalk Subroutine Designed

| Total Error | Pass-Band Error | Stop-Band Error |
|---|---|---|
| 2.0526 | 0.0290 | 2.0236 |

Table 32 Errors of Ideal vs. Input Parameters are Rounded (16-bits)

If we compare the figures and datas taken from MATLAB, it can be said that all stop-band errors which are taken after coefficient changed are less than the yulewalk subroutine found. But on the other hand, comparing the band-pass errors; we see that 5-bit is acceptable.

# CHAPTER 5

# RESULTS AND CONCLUSION

The correctness of an IIR digital filter is restricted by finite word length in its applications. In real life or simulated on a computer we want accurate, fast and cheap work. So we need to determine the minimum word length needed for a specified performance.

In this thesis, we have analysed the effects of limited word length on infinite impulse response (IIR) filters and tried to determine the best word length for designed filter in order to prevent data losses. Problem have been investigated for different bits for low pass filter and band pass filter respectively. Particularly, statistical mean-squared errors (total, pass-band side & stop-band side) are calculated at the output vectors.

Studies show that when the bit length is increased the error in the output decreases and vice versa.

In this study;

- Phase is not considered. Phase may affect the accuracy so in future work it can be add on this work.
- Only low pass and band pass filters are considered. There may be different results for high pass and band stop filters.
- Order of the filter is chosen “5”. It may give different results for low and/or high order filters.

# REFERENCES

[1] Ke-Lin Du & M. N. S. Swamy (April 15, 2010). Wireless Communication Systems, *Cambridge University Press*, pp. 158 – 172.

[2] Saeed V. Vaseghi (2000). Advanced Digital Signal Proccesing and Noise Reduction, *John Wiley & Sons Ltd.*

[3] Hsun-Hsien Chang & José M. F. Moura (2010). Biomedical Signal Processing, ed. Myer Kutz in Biomedical Engineering and Design Handbook, 2nd Edition, volume 1, *McGraw Hill,* pp.559-579.

[4] Alan V. Oppenhaim, Ronald W. Schafer & John R. Buck (1998). Discrete Time Signal Processing, *Prentice Hall Inc., p. 18.*

[5] Andreas Antoniou (1993). Digital Filters: Analysis, Design, and Applications, *McGraw-Hill.*

[6] Clyde Herrick (1996). Basic Electronics Math, *Newnes,* p. 33.

[7] MATLAB Help Files. © 1984-2011 The MathWorks, Inc.

[8] Nicholas, Henry T., III & Samueli, Henry & Kim, Bruce C. (1-3 Jun 1988). The optimization of direct digital frequency synthesizer performance in the presence of finite word length effects, *Frequency Control Symposium*, 1988., Proceedings of the 42nd Annual.

[9] Liu, Bede (November 1971). Effect of finite word length on the accuracy of digital filters--a review, *Circuit Theory, IEEE Transactions.*

[10] Chang, Kyung & Bliss, William G. (August 1994). Finite word-length effects of pipelined recursive digital filters, *Signal Processing, IEEE Transactions.*

[11] Tzafestas, Spyros G. & Kanellakis, A. J. & Theodorou, Nicolas J. (September 1992) Two-dimensional digital filters without overflow oscillations and instability due to finite word length, *Signal Processing, IEEE Transactions.*

[12] Oppenheim, Alan V. & Weinstein, Clifford J. (August 1972). Effects of finite register length in digital filtering and the fast Fourier transform, *Proceedings of the IEEE.*

[13] Swamy, M.N.S. & Roytman, L.M. & Delansky, J.F. (October 1981). Finite word length effect and stability of multidimensional digital filters, *Proceedings of the IEEE.*

[14] Rao, D.V.B. (February 1986). Analysis of coefficient quantization errors in state-space digital filters, Acoustics, *Speech and Signal Processing*, *IEEE Transactions on* (Volume:34 , Issue: 1 ), pp. 131 – 139.

[15] Jackson, L.B. (June 1970). Roundoff-noise analysis for fixed-point digital filters realized in cascade or parallel form, *Audio and Electroacoustics*, *IEEE Transactions on* (Volume:18 , Issue: 2 ), pp. 107 – 122.

[16] Bomar, B.W. (February 1994). Low-roundoff-noise limit-cycle-free implementation of recursive transfer functions on a fixed-point digital signal processor, *Industrial Electronics*, *IEEE Transactions on* (Volume:41 , Issue: 1 ), pp.70 – 78.

[17] Willson, A.N., Jr. (July 1972). Limit cycles due to adder overflow in digital filters, *Circuit Theory, IEEE Transactions on* (Volume:19 , Issue: 4 ), pp.342-346.

[18] Turgut DEVECİ, Prof.Dr.Arif NACAROĞLU (April 2013). Minimization Of The Finite Word Length Noise In The IIR Digital Filters Using Optimization Algorithms, *EMO Bilimsel Dergi* (Sent for publishing).