

Design and comparison of butterworth and chebyshev type-1 low pass filter using Matlab

Tushar Malica¹, Singdha Shekhar², Zakir Ali³

Institute of Engineering & Technology, Bundelkhand University, Jhansi, India
tmalica27@gmail.com, snigdha2090@gmail.com, zakirali008@gmail.com

Abstract: *In the communication system, signals are analogous to carriers of information, be it useful or unwanted. Hence it becomes essential to extract or enhance the useful information and remove any redundancy from a mix of conflicting information. This is the simplest aim of signal processing. The objective of this paper is to know characteristics of each filter for a particular frequency specification- Butterworth low-pass and Chebyshev type-1 low-pass filter and to know how these filters compare. The main concentration being on the magnitude response. An ideal low pass filter has a gain of one in the pass band, zero outside that region. In this paper we compare a Butterworth low pass filter and a Chebyshev type-I low pass filter designed using MATLAB codes which gives the magnitude and phase responses of each of the filters.*

I. Introduction

Signal processing is an operation designed for extracting, enhancing, storing and transmitting useful information. The distinction between useful and unwanted information is situation and need dependant. Hence signal processing is application dependent.

An important element in digital signal processing is the filter. In circuit theory, a filter is an electrical circuit that manipulates the amplitude and/or phase characteristics of a signal with respect to frequency. In signal

processing, the function of a filter is to remove unwanted parts of the signal, such as noise, or selective extraction of the signal such as the components in a certain frequency range. Hence filter has a gain which depends on signal frequency. Our study will concentrate on two types of filters only-

- Butterworth low-pass filter
- Chebyshev type-1 low pass filter

Some of the important terms used here are :-

Transfer function

The frequency-domain behaviour of a filter is described mathematically in terms of its transfer function or network function. This is the ratio of the Laplace transforms of its output to its input signals. The voltage transfer function $H(s)$ of a filter can therefore be written as :-

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} \quad (i)$$

where $V_{in}(s)$ and $V_{out}(s)$ are the input and output signal voltages respectively and s is the complex frequency variable.

The transfer function defines the filter's response to any arbitrary input signal. The magnitude of the transfer function as a function of frequency, indicates the effect of the filter on the amplitudes of sinusoidal signals at various frequencies. Knowing the transfer function magnitude (or gain) at each frequency allows us to determine how well the filter can distinguish between signals at different frequencies. By replacing the variable s in (i) with $j\omega$, where ω is the radian frequency ($2\pi f$), we can find the filter's effect on the magnitude and phase of the input signal.

The magnitude is found by taking the absolute value of (i):-

$$|H(j\omega)| = \left| \frac{V_{out}(j\omega)}{V_{in}(j\omega)} \right|$$

And the phase is :-

$$\arg H(j\omega) = \arg \frac{V_{out}(j\omega)}{V_{in}(j\omega)}$$

Order of a filter

The order of a filter is the highest power of the variable s in its transfer function. With every increase of order n , the attenuation at higher frequencies increases by n 20 dB. It is directly related to the number of components in the filter, and therefore to its cost, its physical size, and the complexity of the design task. Therefore, higher-order filters are more expensive, take up more space, and are more difficult to design.

The primary advantage of a higher-order filter is that it will have a steeper roll-off slope than a similar lower order filter.

In general, a transfer function for an n th-order network, (one with “ n ” capacitors and inductors), can be written as below:-

$$H(s) = H_0 \frac{s^n + b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \dots + b_1 + b_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \dots + a_1 + a_0}$$

Poles and zeroes

The factored form of a network function can be depicted graphically in a pole-zero diagrams.

A pole anywhere to the right of the imaginary axis indicates instability. If the pole is located on the positive real axis, the network output will be an increasing exponential function. A positive pole not located on the real axis will give an exponentially increasing sinusoidal output

Stable networks will have their poles located on or to the left of the imaginary axis. Poles on the imaginary axis indicate an undamped sinusoidal output (in other words, a sine-wave oscillator), while poles on the left real axis indicate damped exponential response, and complex poles in the negative half plane indicate damped sinusoidal response.

Basic elements in digital filter designing

To completely describe digital filters, three basic elements (or building blocks) are needed: an adder, a multiplier, and a delay element. The adder has two inputs and one output, and it simply adds the two inputs together. The multiplier is a gain element, and it multiplies the input signal by a constant. The delay element delays the incoming signal by one sample. Digital filters can be implemented using either a block diagram or a signal flow graph.

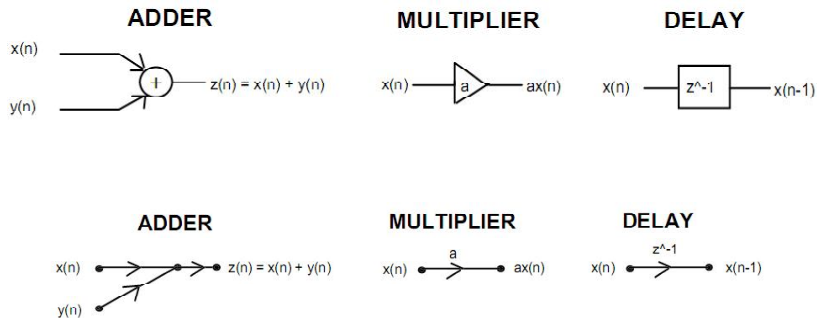


Figure 1. Depiction of the basic designing elements

With the basic building blocks, the two different filter structures depending on the form of the system's response to a unit pulse input can easily be implemented. These two structures are Infinite Impulse Response (IIR) and Finite Impulse Response (FIR). The former is commonly implemented using a feedback (recursive) structure, while the latter usually require no feedback (non-recursive).

The system response and the difference equation for IIR filter is as follows:-

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}}$$

$$Y(n) = \sum_{m=0}^M b_m x(n-m) - \sum_{m=1}^N a_m y(n-m)$$

Summation being from zero to M and N respectively.

The system response and the difference equation for a FIR filter is as follows:

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_M z^{1-M} = \sum_{k=0}^M b_k z^{-k}$$

$$Y(n) = \sum_{k=0}^M b_k x(n-k)$$

Summation being from zero to m-1.

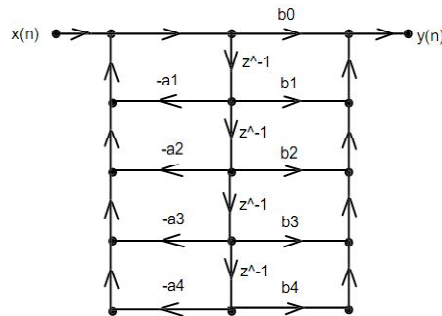


Figure 2. Signal Flow Graph of IIR Filter

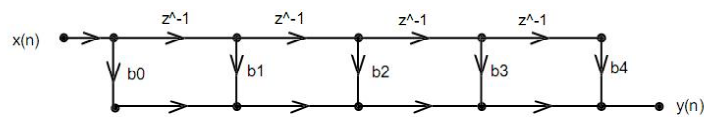


Figure 3. Signal flow graph of FIR filter

II. Filter Design

MATLAB has several design algorithms that can be used to create and analyze both IIR and FIR digital filters. The IIR filters that can be created in MATLAB are Butterworth, Chebyshev type 1 and 2, and Elliptic. The FIR filter algorithms in Matlab are equiripple, least squares, and Kaiser window. Among the most widely used filters are the Butterworth low pass filter and Chebyshev low pass filter.

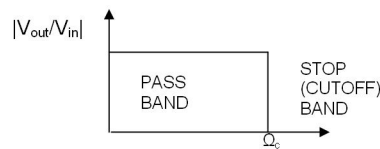


Figure 4. An ideal low pass filter

Butterworth low pass filter

The first and probably best-known filter approximation is the Butterworth. The rolloff is smooth and monotonic, with a low-pass or high-pass roll-off

rate of 20 dB/decade(6 dB/octave) for every pole. The general equation for a Butterworth filter's amplitude response when $\varepsilon = 1$ is:

$$\begin{aligned} |H_a(j\Omega)|^2 &= \frac{1}{1+(\Omega/\Omega_c)^{2N}} & \Omega < \Omega_c & \quad \text{(ii)} \\ |H_a(j\Omega)|^2 &= \frac{1}{(\Omega/\Omega_c)^{2N}} & \Omega > \Omega_c & \end{aligned}$$

where N is the order of the filter, and can be any positive whole number (1, 2, 3, ...), and Ω_c is the cut-off frequency in rad/sec.

The code used displays the magnitude and phase responses of a butterworth low-pass filter of order ranging from 1-10 and normalized value of 0.6 (i.e. 300/500).

The results are shown in appendix- I.

Chebyshev type-I low-pass filter

Another approximation to the ideal filter is the Chebyshev or equal ripple response. The addition of pass band ripple as a parameter makes the specification process for a Chebyshev filter a bit more complicated than for a Butterworth filter, but also increases flexibility. The nominal gain of the filter is equal to the filter's maximum pass band gain. An odd-order Chebyshev will have a dc gain (in the low-pass case) equal to the nominal gain, with "dips" in the amplitude response curve equal to the ripple value. An even-order Chebyshev low-pass will have its dc gain equal to the nominal filter gain minus the ripple value; the nominal gain for an even-order Chebyshev occurs at the peaks of the pass band ripple. The general equation for a Chebyshev filter's amplitude response when ε (epsilon) =1 is:-

$$\begin{aligned} |H_a(j\Omega)|^2 &= \frac{1}{1+\cos^2\left[N\cos^{-1}\left(\frac{\Omega}{\Omega_c}\right)\right]} & \Omega < \Omega_c \\ |H_a(j\Omega)|^2 &= \frac{1}{1+\cosh^2\left[N\cosh^{-1}\left(\frac{\Omega}{\Omega_c}\right)\right]} & \Omega > \Omega_c \end{aligned}$$

where N is the order of the filter, and can be any positive whole number (1, 2, 3, ...), and ω_c is the cut-off frequency in rad/sec.

The code used displays the magnitude and phase responses of a Chebyshev type-I low-pass filter of order ranging from 1-10 and normalized value of 0.6 (i.e. 300/500). The pass band ripple is calculated by :

$$R = -10 \log_{10} |H(j\Omega)|^2$$

$$\text{As } R = -10 \log_{10} \frac{1}{[1 + \varepsilon^2]}$$

and $|H(j\Omega)|^2$ has its maximum values as $1/[1 + \varepsilon^2]$ for $\Omega < \omega_c$.

The results are shown in appendix- IV.

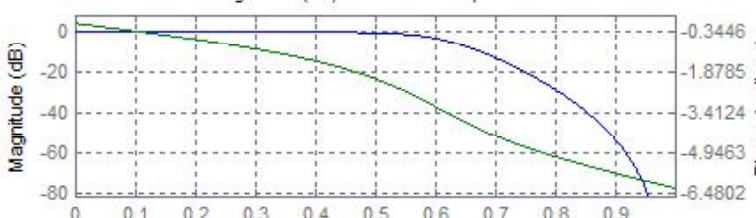
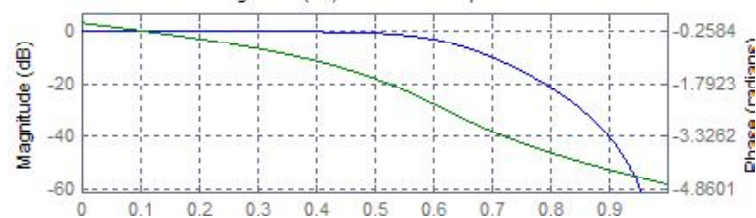
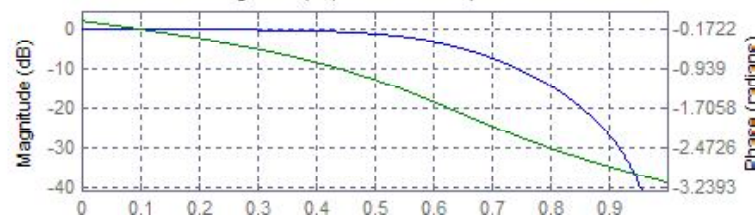
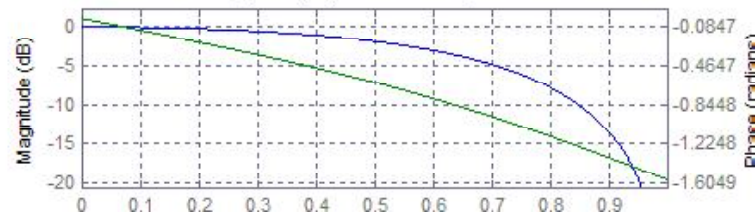
Results and Discussion

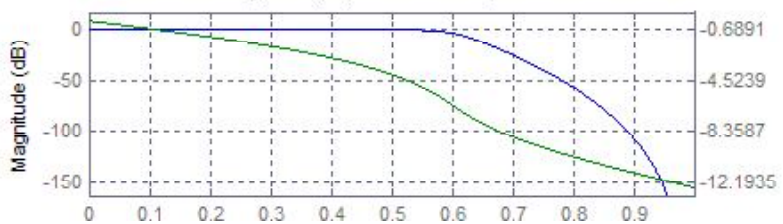
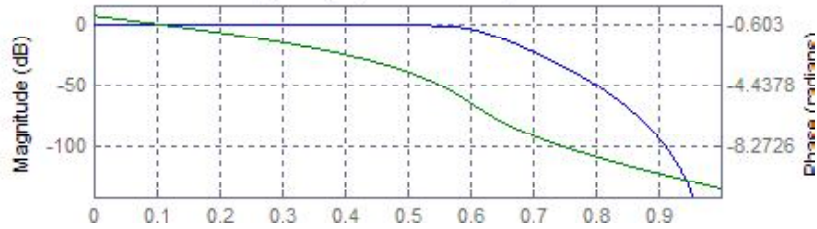
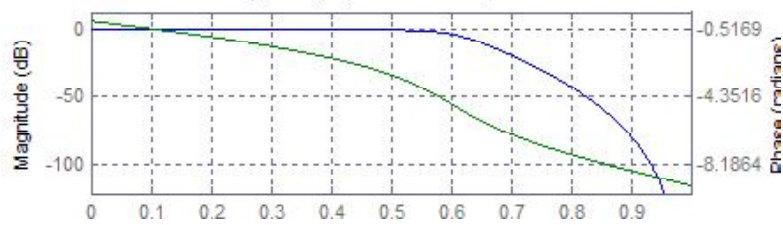
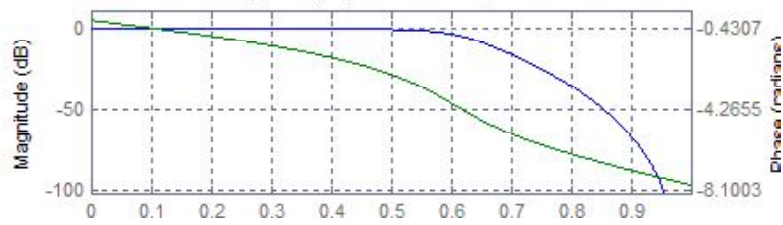
- Hardware required for similar order filter is same irrespective of the type. The increase in order indicates increase in hardware too. This increases the complexity of the design and the production cost too. For instance, number of adders can be given by $2N$, where N is the order of the filter. [refer appendix- III]
- The Chebyshev characteristic has a steeper rolloff near the cutoff frequency when compared to the Butterworth. Though the monotonicity in the pass band is compromised. [refer appendix-VI]. This means that the Chebyshev filter for the same order attenuates or rejects frequencies at the stop band in a better way than the Butterworth filter. Hence, a same order Chebyshev low-pass filter will work more effectively than a Butterworth low-pass filter in disposing unwanted frequencies but, if the Butterworth filters maybe the better choice when a ripple-less and maximally flat response is desired. While, the Butterworth filter is maximally flat at $\Omega=0$, the Chebyshev filter may have ripple in the pass band of amplitude response. This can be further proved by putting $\Omega=0$ in (ii). The gain comes out to be 1 which is the maximum Butterworth magnitude value. It exhibits a nearly flat pass band with no ripple and smooth roll-off. [refer appendix- I]. On the other hand, The amount of pass band ripple is one of the parameters used in specifying a Chebyshev filter. [refer appendix-IV]

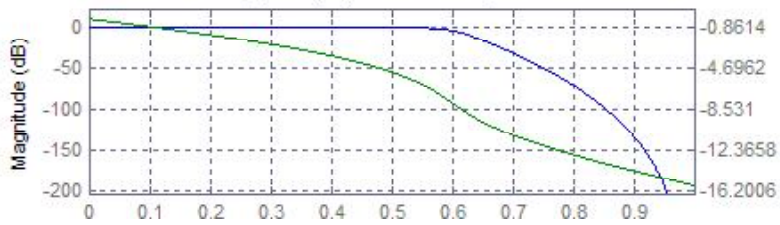
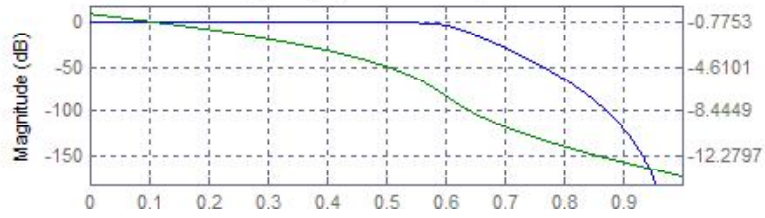
- Also the Chebyshev filter has a bigger decrease in magnitude with the increase in frequency than Butterworth. So, the derivative of Chebyshev's gain will be more negative than Butterworth filter.
- The phase response is linear at first but some ripple or non-linearity is observed as the order of the Butterworth filter increases.[refer appendix-I]. Phase response maybe linear piece-wise in Chebyshev filter.[refer appendix-IV]

APPENDIX-I

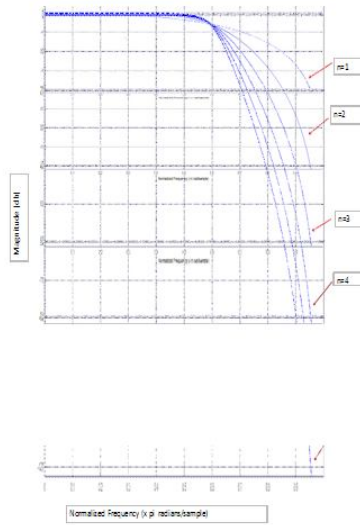
The output of Butterworth low-pass filter with uniformly increasing order of the filter is shown below. The blue line represents the magnitude response and the green line traces the phase response.







APPENDIX-II



Butterworth filters' magnitude responses of different order compared.

APPENDIX-III

Butterworth filter structure: direct form II transposed, second order sections

Chebyshev filter structure: direct form II transposed, second order sections

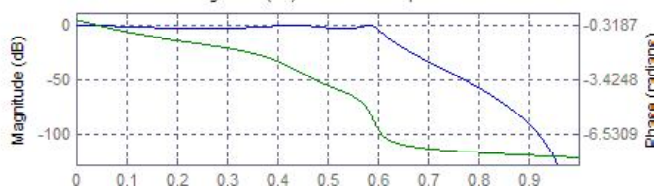
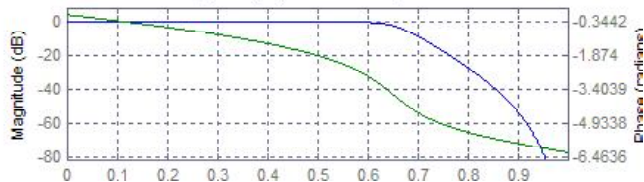
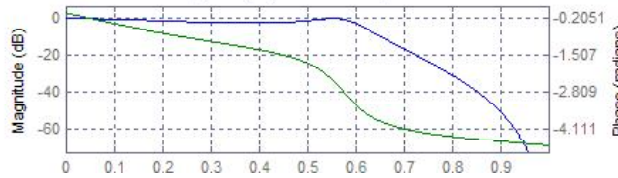
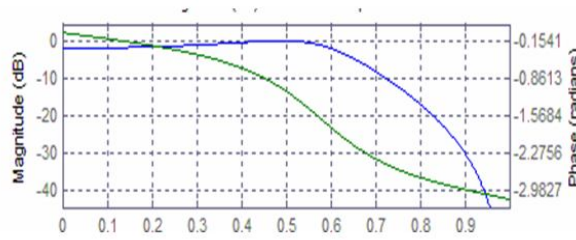
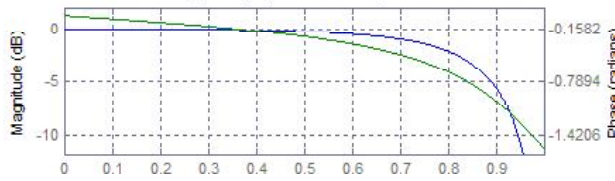
Order	Number of sections	Stable	Linear phase	Number of multipliers	Number of adders
1	1	yes	no	2	2
2	1	yes	no	4	4
3	2	yes	no	5	6
4	2	yes	no	7	8
5	3	yes	no	8	10
6	3	yes	no	10	12
7	4	yes	no	11	14
8	4	yes	no	13	16
9	5	yes	no	14	18
10	5	yes	no	16	20

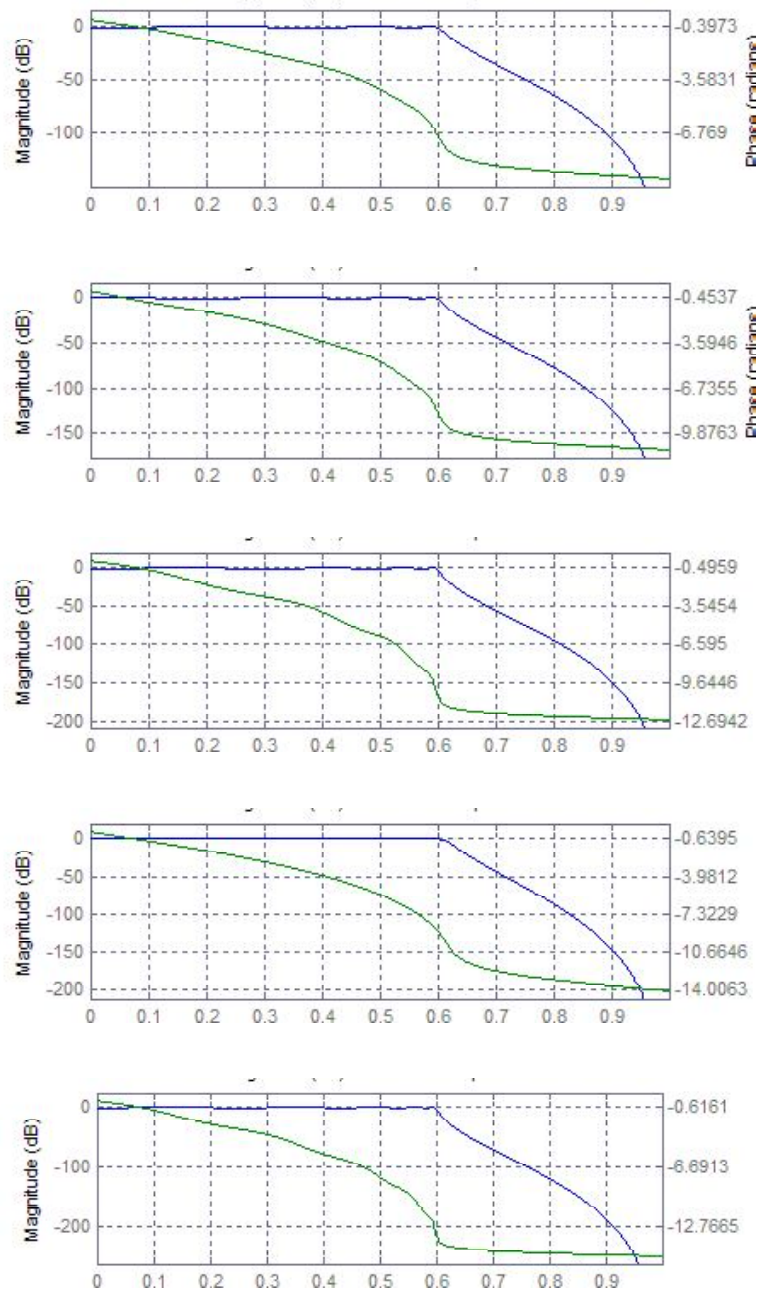
Table 1. Butterworth low-pass filter. Filter information data acquired and compiled

Order	Number of sections	Stable	Linear phase	Number of multipliers	Number of adders
1	1	yes	no	2	2
2	1	yes	no	4	4
3	2	yes	no	5	6
4	2	yes	no	7	8
5	3	yes	no	8	10
6	3	yes	no	10	12
7	4	yes	no	11	14
8	4	yes	no	13	16
9	5	yes	no	14	18
10	5	yes	no	16	20

Table 2. Chebyshev type-I low-pass filter. Filter information data acquired and compiled

APPENDIX-IV





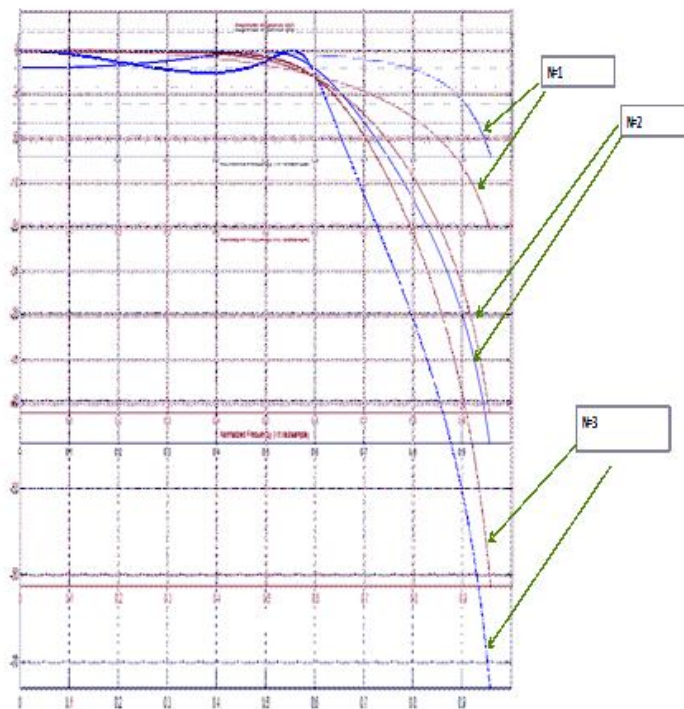
The output of Chebyshev type-I low-pass filter with uniformly increasing order of the filter is shown below. The blue line represents the magnitude response and the green line traces the phase response

APPENDIX-V

Chebyshev filters' magnitude responses of different order compared.

APPENDIX-VI

Comparison of magnitude responses of Butterworth and Chebyshev type-I low pass filters. The red trace indicates the Butterworth responses while the blue lines traces the Chebyshev magnitude responses.



References

- [1] Vinay K. Ingle, John G. Proakis, Digital Signal Processing Using MATLAB V.4
- [2] Digital Filter Design Using Matlab, by Timothy J. Schlichter, EE 4000 Introduction to Digital Filtering, February 1999
- [3] 10 MHz Butterworth Filter Using the Operational Amplifier THS4001 Dirk Gehrke and Andreas Hahn, Application Report SLOA032 - October 1999, Texas Instruments
- [4] Signals and systems with Matlab computing and simulink modellong by Steven T. Karris
- [5] www.mathworks.com