NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

FILTERING BAIS ON MATLAB

Graduation Project
EE - 400

Student: Tarig Mohammed(971382)

Supervisor: Professor Dr. Fakhreddin Mamedov

Lefkosa-2001

ACKNOWLEDGMENTS

First I want to thank God for every thing.

Secondly I want to thank my family, specially my parents. Without their endless support and love for me, I would never achieve my current position. I wish my mother lives happily always, and my father in the heaven be proud of me.

Moreover, thanks to my dearly uncle Dr. Nasir Elddin A.Madani for his love and supporting me kindly to be in a good position.

I also want to thank Prof.Dr FAKHREDDIN MAMEDOV to be my advisor. Under his guidance, I successfully overcome many difficulties and learn a lot about Filtering. In each discussion, he explained my questions patiently, and I felt my quick progress from his advises. He always helps me a lot either in my study or my life. I asked him many questions in Electronics and communication and he always answered my questions quickly and in details.

Special thanks to all the doctor's and teachers staff. Thanks to Faculty of Engineering for having such a good computational environment.

I also want to thanks husam and samir gamal. With their kind help, I could use Matlab successfully to perform computational problems.

Finally I want to thank my Sudanese friends in NEU:samir O.zaidan, Gafar,Mohaiad,M.Khair Waseem and Mr. Abd Elazeem Mustafa. Being with them make my 3 years in NEU full of fun.

ABSTRACT

The importance of filters is well known through the spread of technology and communication. Analog filters use in communication is decreasing and the digital filters are increasing gradually.

The applications of filters are increasing, audio systems, image processing systems, audio and video recording systems, communication systems and data smoothing systems are all some of the wide range of application fields that filters are involved in.

The variations between time domain and frequency domain are highly determining the type of filter and even in the time domain there are the continuous and discrete time frequency selective filters.

The design of filters beginning from the low pass to the high pass and band pass filter by using computer programs is really the field of interest of all engineers and using MATLAB is making things more convenient.

Methods of approximation Chebyshev, Butterworth and Elliptic approximations are all used to distinguish and provide solutions for design problems for filters.

For soft ware development of MATLAB and using it in the design of filtering circuits are commonly dependent on the function derivation of inputs and results obtained in out puts.

MATLAB is a high performance language for technical computing. It integrates computation, by visualization and programming in an easy to use environment where problems and solutions are expressed in familiar mathematical notation. It is an interactive system whose basic data element is an array that does not require dimensioning.

TABLE OF CONTENTS

Introduction

In a variety of important application it is of interest to change the amplitudes of the frequency components in a signal or perhaps eliminate some frequency components entirely, a process referred to as filtering. For linear time invariant system the spectrum of the output is that of the input multiplied by the frequency response of the system, consequently, filtering can be contently accomplish through the use of such system with an appropriately chosen frequency response these represent one of the very important application of linear time invariant system.

One example in which leaner time invariant filtering is encountered Is in audio system. In such systems, a filter is typically included to permit the leister to modify the relative amounts of low frequency energy (bass) and high frequency energy (treble). The filter corresponds to a linear time variant system whose frequency responds is changed by manipulating the tone controls. Also, in high fidelity audio system, a filter is often included in the preamplifier compensate for the frequency response characteristics of the speakers. An example.

In this thesis, Design and implementation of Filters in general is discussed.

The first chapter represents classification, characteristics and applications of filters. Chapter provides comparison of analog and discrete filters.

Chapter tow presents Approximation using Butterworth, Chebyshev, Elliptic and Linear phase approximations Chapter discusses analog filter design using Matlab.

Chapter three provides different kinds of filters including the Fast Fourier Transformed (FFT) and the Discrete Fourier Transform. Chapter also presents Low pass, High pass, Band pass, Narrow Band and IF Amplifier filters.

Chapter four shows various applications of filters of some chosen types of filtering designs as high pass anti-aliasing, low pass anti-aliasing, nonlinear, advanced and discrete filtering methods.

The conclusion presents important results, contribution of the authors and future research areas.

Chapter 1
THEORY OF FILTER1NG

1.1. Introduction

For linear time invariant systems, the spectrum of the output $Y(j\omega)$ is that of the input $U(j\omega)$ multiplied by the frequency response of the system $H(j\omega)$. A filtering is a process of changing a relative amplitudes and frequency of spectrum or completely rejection some frequency components of signal.

1.2. Applications of Filters

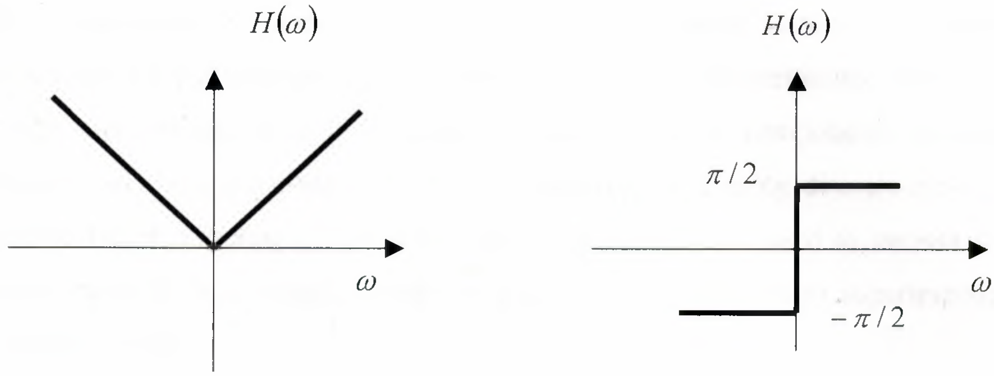
1.2.1. Audio systems

In such systems, a filter is typically included to permit the listener to modify the relative amounts of low frequency (bass) and high frequency (treble) energies. Manipulating the tone controls changes the filter frequency response. In high fidelity audio systems, the equalizing circuits are used to compensate for the frequency response of the speakers and the listening room.

1.2.2. Image Processing Systems

The frequency-response $H(\omega)$ characteristic of differentiating filter are shown in Figure 2.1(a) and (b). Differentiating filter provides rapid transitions and is useful to enhance of edges in picture processing.

$$u(j\omega) \rightarrow H(J\omega) \rightarrow y(j\omega) \quad ;\ldots\ldots\ldots(1.1)$$



(a) (b)

Figure 1.1

1.2.3. Audio And Video Recording Systems

Frequency-selective filters, which pass signals undistorted in one or a set of frequency bands and attenuate or totally eliminate signals in the remaining frequency bands, are another important class of LTI filters. The Use of frequency-selective filters arises in a variety of situations. For example, if surface noise in an audio recording is in a higher frequency band than the music or voice on the recording, it can be removed by frequency-selective filtering.

1.2.4. Communications Systems

Another important application of frequency-selective filters is in communications systems. The basis for amplitude modulation (AM) systems is the transmission of information from many different sources simultaneously by putting the information from each channel into a separate frequency band and extracting the individual channels or bands at the receiver using frequency-selective filters. Frequency-selective filters for separating the individual channels and frequency-shaping filters for adjusting the tone quality form a major part of any home radio and television receiver.

1.2.5. Data Smoothing Systems

Analysis of economic data sequences such as the stock market average commonly utilizes discrete-time filters. Often the long-term variations (which correspond to low frequencies) have a different significance than the short-term variations (which correspond to high frequencies), and it is useful to analyses these components separately. The separation of these components is typically accomplished using discrete-time frequency-selective filters. Filtering of economic data sequences is also used to *smooth* the data to remove random fluctuations (which are generally high frequency) superimposed on the meaningful data.

1.3. Frequency-Domain Characteristics Of Ideal Frequency Selective Filters

The frequency-selective filters were defined in terms of a mathematical modeling. The ideal models represent lowpass, highpass, bandpass, bandstop, and all-pass filters (see Figure 1.2). Their shape represents the steady-state magnitude-frequency response of a filter with a transfer function of

$$H(\Omega)=H(s)\big|_{s=j\Omega} ;\ldots\ldots (1.2)$$

$$\text{Ideal highpass}\ |H(\Omega)| = \begin{cases} 0 & if\Omega \in [-B, B] \\ 1 & otherwise \end{cases} ;\ldots\ldots (1.3)$$

$$\text{Ideal Bandpass}\ |H(\Omega)| = \begin{cases} 1 & if\Omega \in [-B_2, -B_1]or\Omega \in [B_1, B_2] \\ 0 & otherwise \end{cases} ;\ldots\ldots (1.4)$$

Ideal

$$\text{Bandstop}\ |H(\Omega)| \begin{cases} 0 & if\Omega \in [-B_2, -B_1]or\Omega \in [B_1, B_2] \\ 1 & otherwise \end{cases} ;\ldots\ldots (1.5)$$

$$\text{All-pass}\ |H(\Omega)| = 1 forall\Omega \in [-\infty, \infty] ;\ldots\ldots (1.6)$$

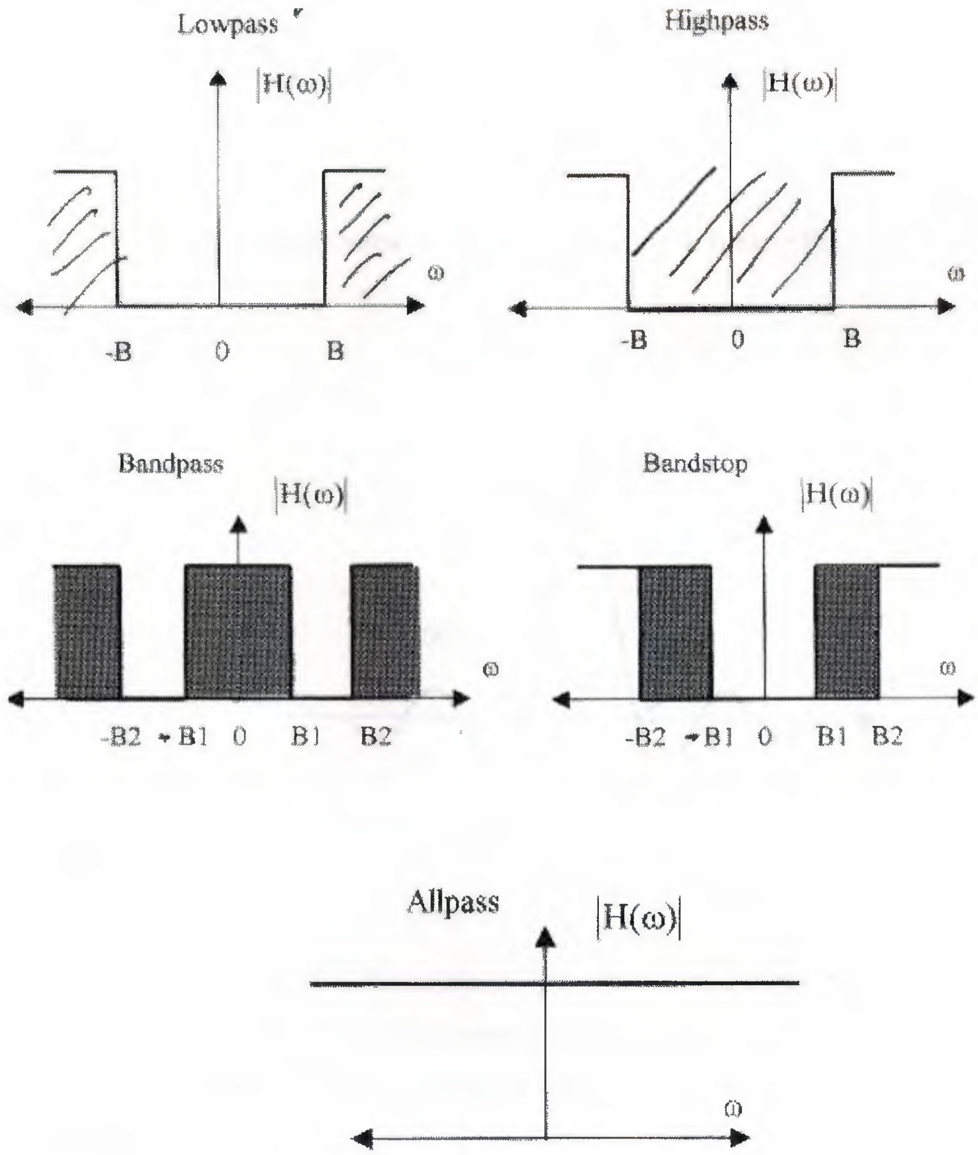


Figure 1.2

1.4. Time-Domain Characteristics of Ideal Frequency Selective Filters

Often in designing and utilizing filters, it is also important to take into account the time-domain characteristics, such as the impulse response and step response.

The impulse response of the ideal low-pass filter corresponds to the inverse Fourier transform of the frequency response of filter. It is given by the equation:

$$h(t) = \frac{\omega_c}{\pi} \sin c\left(\frac{\omega_c t}{\pi}\right); \qquad (1.7)$$

This impulse response is sketched in Figure 1.3. Here the width of the filter pass-band.

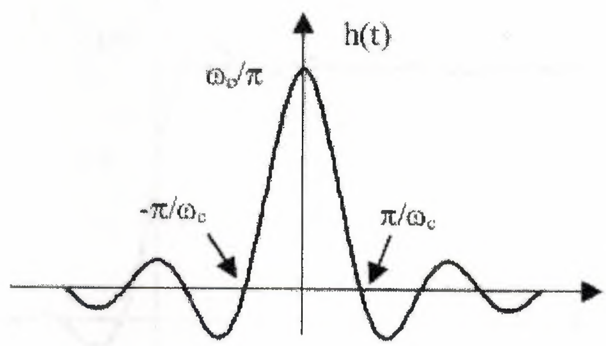


Figure 1.3

Is proportional to ωc and the width of the main lobe of the impulse response is proportional to $1\ /\omega c$. As the filter bandwidth increases, the impulse response bee narrower, and vice versa. This is, of course, consistent with the scaling property for Fourier transforms.

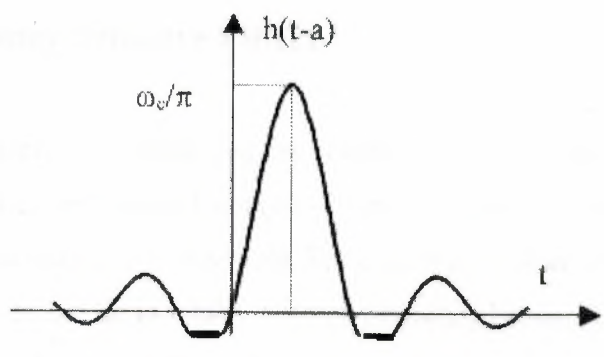


Figure1.4

For the ideal low-pass filter with linear phase corresponding to figure 1.3. The impulse response is simply delayed by a, as indicated in Figure 1.4. The step responses of the ideal low-pass filter are illustrated in Figure 1.5.

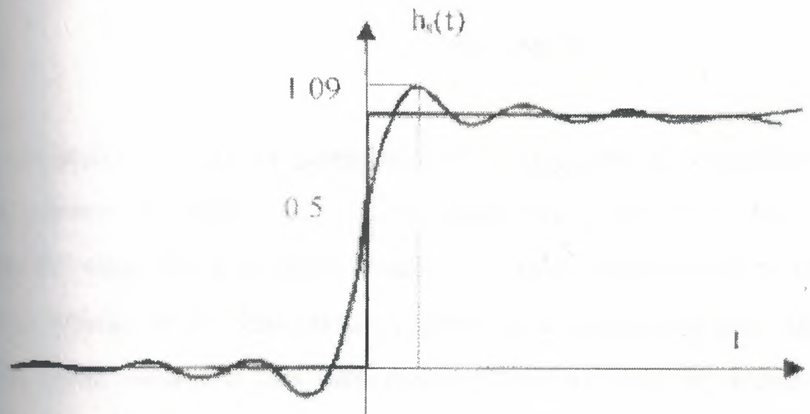


Figure1.5

We see that in the time domain, the response of an ideal low-pass filter exhibits overshoot. In some contexts this time-domain behavior may be undesirable. As we consider in the next section, for this and other reasons it is often of interest to design filters with a more gradual transition from pass-band to stop-band.

1.5 Non-Ideal Frequency Selective Filters

The classes of filters discussed are referred to as ideal filters because they exactly pass one set of frequencies and *completely* reject others. However, this is in practical impossible and not necessarily desirable. For example, in many filtering contexts, the signals to be separated do not lie in totally disjoint frequency bands. A typical situation might be depicted in Figure 1.6 where the spectra of two signals overlap slightly. A filter with a gradual transition from pass-band to stop-band is generally preferable when filtering the sum of signals with overlapping spectra.

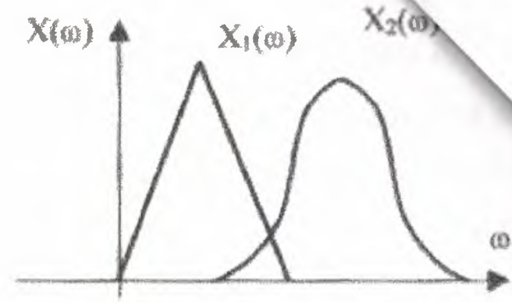


Figure1.6

As we indicated another consideration is suggested by examination of the step response, shown in Figure 1.5, of an ideal low-pass filter. The step response sufficiently far away from the discontinuity is approximately equal to the value of the step. In the vicinity of the discontinuity, however, it overshoots this value and exhibits ringing. In some situations this time-domain behavior may be undesirable. Since in many cases, the characteristics of the "ideal" frequency-selective filter are undesirable it is often preferable to allow some flexibility in the behavior of the filter in the pass-band and in the stop-band as well as to permit a more gradual transition between the pass-band and stop-band as opposed to the abrupt transition characteristic of the "ideal" filters. Thus, specifications for a low-pass filter are often stated to require the magnitude of the filter frequency response to lie in the no shaded area indicated in Figure1.7.

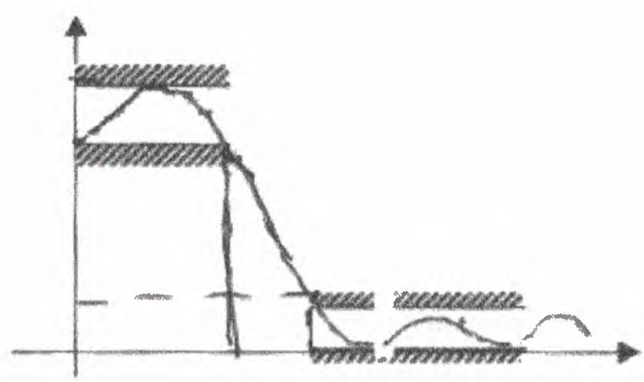


Figure1.7

In this figure a deviation from unity of plus and minus δ_1 is allowed in the pass-band and a deviation of δ_2 zero is allowed in the stop-band. The amount by which the frequency response differs from unity in the pass-band is referred to as the pass-band ripple and the amount by which it deviates from zero in the stop-band is referred to as the *stop-band ripple.* The frequency ω_p, is referred to as the *pass-band edge* and ω_s as the *stop-band edge.* The *transition band* $\Delta\omega = \omega_s - \omega_p$ provides the transition from pass-band to stop-band.

Even in cases when the ideal frequency-selective characteristics are desirable, they may not be attainable. It is evident that the ideal low-pass filter is no causal and consequently must be approximated for real-time filtering by a causal system. It can similarly be shown that the ideal high-pass and band-pass filters are no causal. When filtering is to be carried out in real time, causality is a necessary constraint, and thus a causal approximation to the ideal characteristics would be required A further consideration that motivates providing some flexibility in the filter characteristics is ease of implementation. In general, the more precisely we try to approximate or implement an ideal frequency-selective filter, the more complicated or costly the implementation becomes whether in terms of components such as resistors, capacitors, and operational amplifiers, in continuous time, or in terms of memory registers, multipliers, and adders, in discrete time. In many filtering contexts a precise filter characteristic may not be essential and a simple filter will suffice.

1.6.Examples Of Continuous-Time Frequency Selective Filters

1.6.1. RC Low-Pass and High-Pass Filters

As an example of a simple continuous-time low-pass filter, consider the first-order *RC* circuit in Figure 1.8. The capacitor voltage V_0. Is considered to be the system output and the source voltage the system input the output voltage is related to the input voltage E through the linear constant-coefficient differential equation.

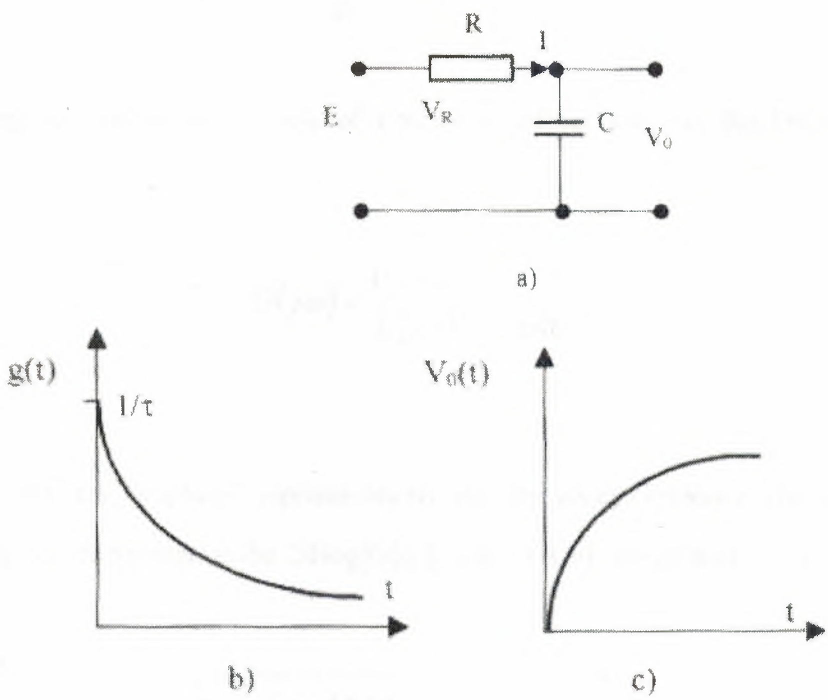


Figure1.8

Applying Kirshhofs voltage law to the system, we obtain the following equations:

$$RI(t)+V_0 = E(t); \quad \text{...(1.8)}$$

$$V_0(t) = \frac{1}{C}\int I(t)dt \;; \quad \text{..(1.9)}$$

$$I(t) = C\frac{dV_0(t)}{dt} \;; \quad \text{...(1.10)}$$

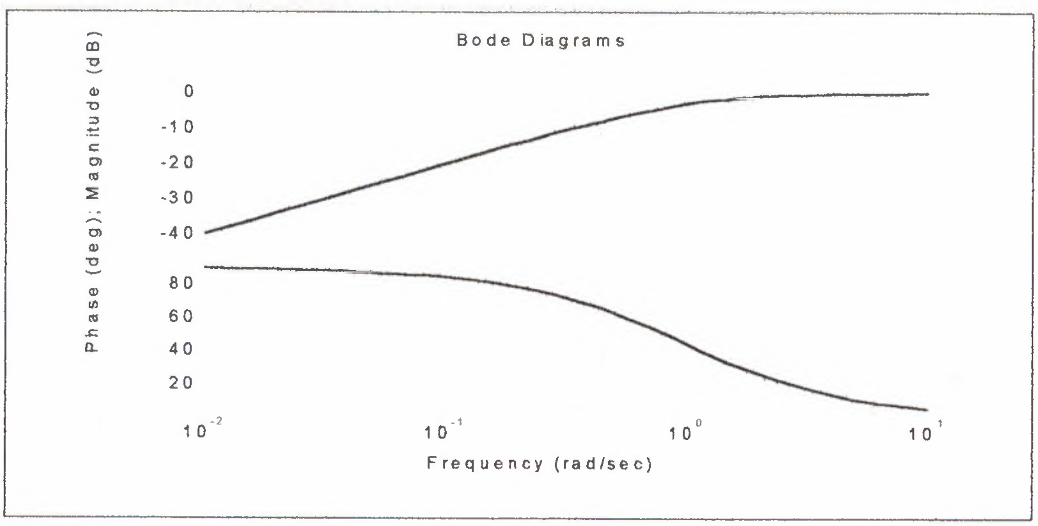
$$V_0(t) + RC\frac{dV_0(t)}{dt} = E(t);\text{..................................}(1.11)$$

Using the derivation property of Fourier transform we find the frequency response of filter

$$G(j\omega) = \frac{V_0(j\omega)}{E(j\omega)} = \frac{1}{j\omega RC + 1};\text{.......................}(1.12)$$

Consider the graphical representations the frequency response characteristic using Bode plot. In this representation the $20\log|G(\omega)|$ and $\angle G(\omega)$ are plotted versus frequency.

```
» num=[1 0];
» den=[1 1];
» bode(tf(num,den))
```



Frequency (rad/sec) Impulse response (see Figure 1.2 (b)) corresponding to the inverse Fourier transform

$$g(t) = V_0(t) = \frac{1}{\tau} e^{-t/\tau}; \qquad (1.13)$$

Where τ = RC is a time constant of circuit.

For step impulse we have

$$E(j\omega) = \frac{1}{j\omega}\ ;\ V_0 = \frac{1}{j\omega(\tau j\omega + 1)} = \frac{1}{j\omega} = \frac{\tau}{\tau j\omega + 1}; \qquad (1.14)$$

Taking the Inverse Fourier transform we define step response shown in Figure 1.7 (c).

$$V_0(t) = 1 - e^{-t/\tau}; \qquad (1.15)$$

From Figure 1.7(a) note that with resistor the voltage taken as the output; the *RC* circuit behaves as an approximation to a high-pass filter

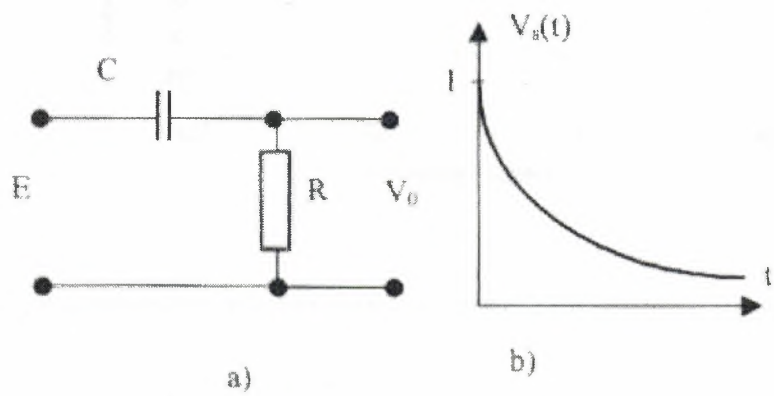


Figure1.10

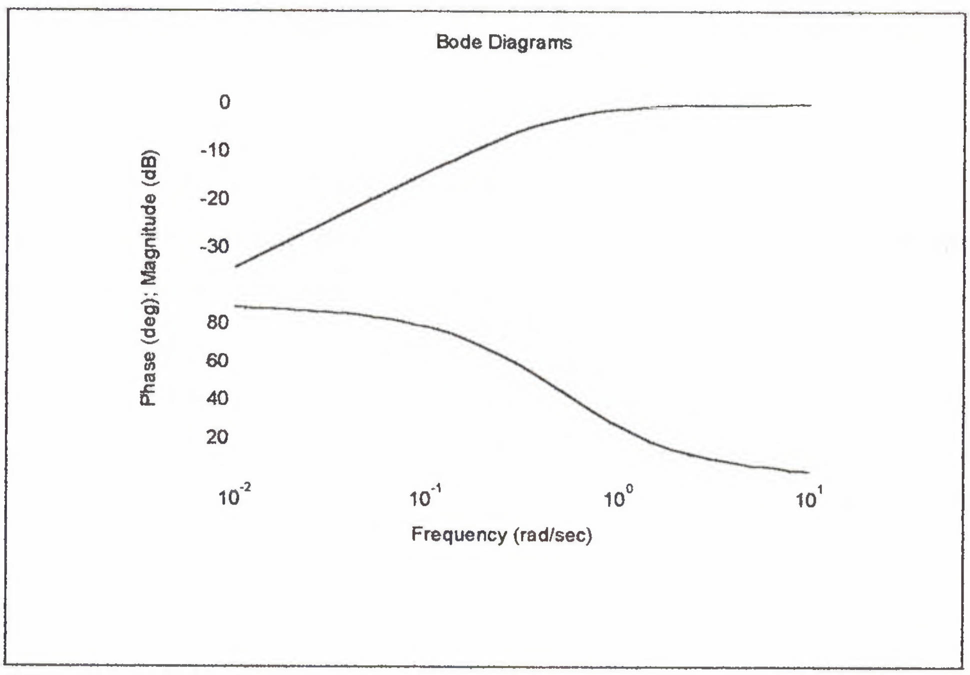
The frequency response $G(\omega)$ is given by

$$G(\omega)=\frac{V_R(\omega)}{V_S(\omega)}=\frac{j\omega RC}{1+j\omega RC};\ldots\ldots(1.16)$$

Step response of filter

$$V_S(j\omega)=\frac{1}{j\omega}\cdot\frac{j\omega RC}{1+j\omega RC}=\frac{RC}{1+j\omega RC}=\frac{1}{\frac{1}{\tau}+j\omega}\ldots\ldots(1.17)$$

In the time domain $V_s(t)=e^{-t/\tau}$ Bode diagram



The step response is shown in Figure1.10(b) and the Bode plot for the frequency response is shown in Figure1.11 As shown a simple *RC* circuit can serve as high-pass or a low-pass filter, depending upon the choice of the physical output variable. These variable. These filters do not have a sharp transition from pass-band to stop-band. If desired, more complex filters with a sharper transition can be implemented by using more energy storage element (capacitances or inductances), leading to higher-order differential equations.

1.7. Introduction Discrete-Time Frequency Selective Filters

Discrete-time systems characterized by linear constant-coefficient difference equations, are conveniently implemented using coefficient multipliers, storage or delay elements registers, and adders. We will introduce the tow basic classes of difference equations:

Non recursive difference equations, for which the impulse response is of finite length, (FIR) and recursive difference equations, for which the impulse response is of infinite length,(FIR) There are specific advantages and disadvantages to recursive and no recursive filters.

For example, it is often desirable for the phase characteristics of a filter to be zero or linear, so that the phase affects the shape of the output signal by at most a time delay. If a filter is to be *causal* and have exactly linear phase, its impulse response must be of finite length, and consequently the difference equation must be no recursive. On the other hand, it is generally true that the same filter specifications require a higher-order equation and consequently more coefficients and delays when implemented using a no recursive difference equation compared with using a recursive difference equation. In the following discussion we consider separately the two classes of discrete-time filters.

1.8. Non-recursive Discrete-Time Filters

As we have emphasized in several discussions, low-pass filtering can be thought of as a smoothing operation. For discrete-time sequences a common smoothing operation is one referred to as a *moving average,* where the smoothed value $y[n]$ for any n, n_0, is an average of values of *x[n]* in the vicinity of n_0. The basic idea is that by averaging values locally, rapid variations from point to point will be averaged out and slow variations will be retained, corresponding to smoothing or low-pass filtering the original sequence. As an example, a three-point moving average of an input *x[n]* is of the form

$$y[n] = \frac{1}{3}(x[n-1] + x[n] + x[n+1]); \quad \text{.............................(1.18)}$$

So that each output y[n] is the average of three consecutive input values. The frequency response associated with equation 1.9

$$H(\Omega) = \frac{1}{3}\{1 + 2\cos\Omega\}; \quad \text{..(1.19)}$$

The magnitude and phase of $H(\Omega)$ are sketched in Figure 1.12 if. We observe that it has the general characteristics of a low-pass filter although, as with the circuit, it does not have a sharp transition from pass-band to stop-band.

A similar approach can also be used to approximate a high-pass filter as well as a low-pass filter. To illustrate this, again with a simple example, consider the difference equation.

$$y[n] = \frac{x[n] - x[n-1]}{2}; \quad \text{...................................(1.20)}$$

M*atlab*

```
» omega=0:2*pi/50:2*pi;
» h=(1/3)*(1+2*cos(omega));
» plot(omega,h);
» omega=0:2*pi/50:2*pi;
» h=(1/3)*(1+2*cos(omega));
» H=abs(h);
» plot(omega,H)
```

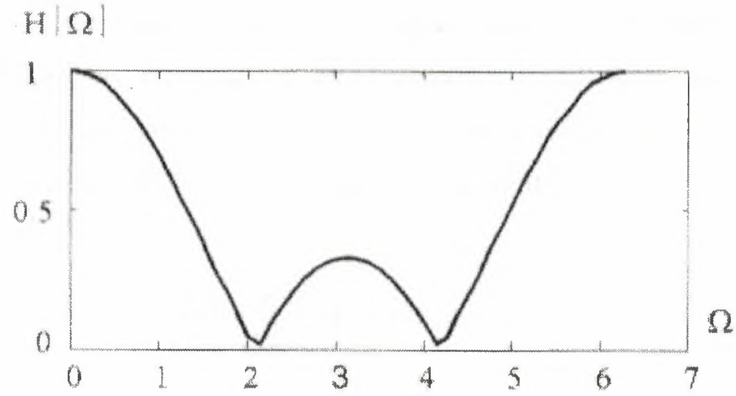


Figure1.12

For input signals that are approximately constant, the value of $y[n]$ is close to zero. For input signals that vary greatly from sample to sample, the values of $y[n]$ can be expected to have large amplitude. We would thus expect this equation to approximate a high-pass filter, since high-frequency components are reflected in large variations between adjacent sequence values. The frequency response associated with Equation. (1.11) Is

$$H(\Omega) = \frac{1}{2}\left[1 - e^{-j\omega}\right] = je^{-j\Omega/2}\sin\left(\frac{\Omega}{2}\right); \qquad (1.21)$$

The three-point moving-average filter has no parameters that can be changed to adjust the effective cut-off frequency. As a generalization of this moving-average filter, we can consider averaging over $N + M — 1$ neighboring points, that is, to use a difference

Equation of the form $y[n] = \frac{1}{N+M+1}\sum_{k=-N}^{M} x[n-k]; \qquad (1.22)$

The corresponding impulse response is a rectangular pulse. If N=0 or is negative the moving average filter is causal.

It is common to apply a moving-average filter to many economic indicators to attenuate the short-term fluctuations in relation to longer-term trends.

A further generalization of the moving-average filter can be made by forming a *weighted* average of *(N + M +* 1) neighboring points, that is, by using a difference equation of the form

$$y[n] = \sum_{k=-N}^{M} b_k x[n-k]; \qquad (1.23)$$

Where the coefficients bk can be selected to achieve the prescribed filter characteristics

There are a variety of techniques available for choosing the coefficients in Eq. (1.13) to meet certain specifications on the filter. In general, the coefficients b_k can be adjusted so that the cut-off is at a desired frequency.

1.9. Recursive Discrete-Time Filters

We considered moving-average or no recursive filters. Another important class of discrete-time filters *are* those described by the class of recursive difference equations. Consider the discrete-time system described by the difference equation

$$y[n] - ay[n-1] = x[n] \text{ or } y[n] = ay[n-1] + x[n]; \qquad (1.24)$$

The corresponding frequency response is

$$H(\Omega) = \frac{1}{1 - ae^{-j\Omega}}; \qquad (1.25)$$

The difference equation (1.15) behaves as a low-pass filter, whereas for a negative, it behaves as a high-pass filter. Just as with differential equations, higher-order recursive difference equations can be used to provide sharper filter characteristics and to provide more flexibility in balancing time-domain and frequency-domain constraints. There are a number of specific classes of continuous- and discrete-time filters for which standard procedures have been developed to determine the coefficients of the associated differential or difference equation.

1.10. Design of Analog Filters

Analog filter design is often based on the use of several well-known models called Butterworth, Chebyshev, and elliptic (Cauer) filters. To standardize the design procedure, a set of normalized analog prototype filter models was agreed upon and reduced to tables, charts, and graphs. These models, called prototypes, were all developed as lowpass systems having a known gain (typically -1 dB or -3 dB passband attenuation) at a known critical cut-off frequency (typically 1 radian/second). The transfer function of an analog prototype filter, denoted $H_p(s)$, would be encapsulated in a standard table as a function of filter type and order. The prototype filter $H_p(s)$ would then be mapped into a final filter H(s) having critical frequencies specified by the designer. The ma ping rules, called frequency - frequency transforms are shown in figure1.13

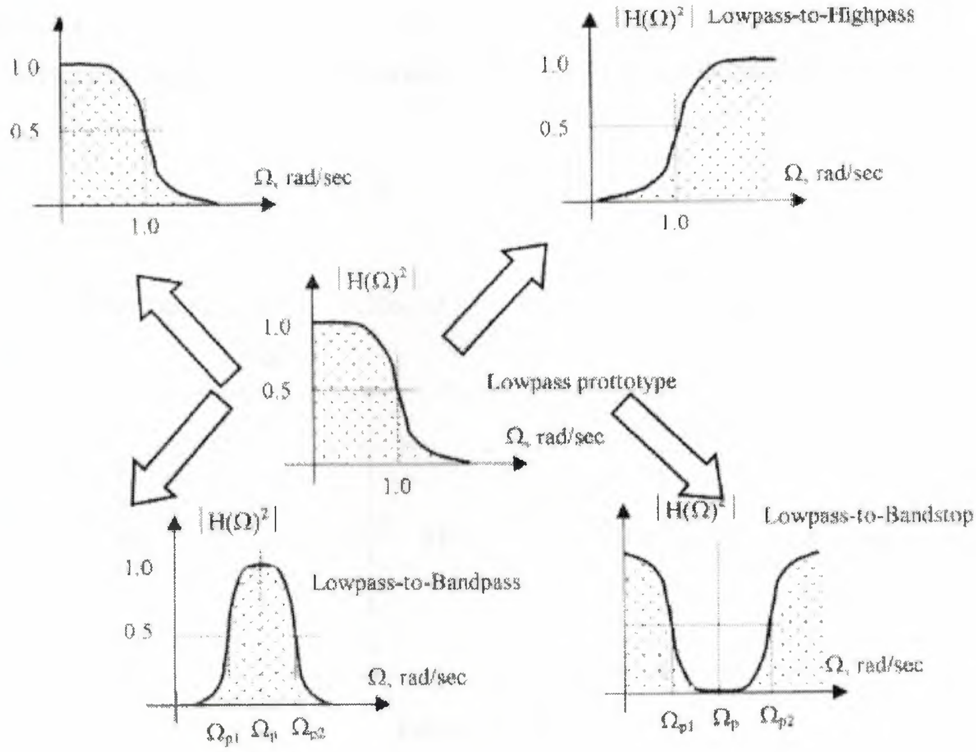


Figure1.13

Frequency-to-frequency scaling to different types filters is shown in table Below

Table 1.1

Nth Order prototype	Frequency-to-frequency Transform	Order
Lowpass to Lowpass	$s \Leftarrow s/\ \Omega_p$	N
Lowpass to Highpass	$s \Leftarrow \Omega_p/s$	N
Lowpass to Bandpass	$s \Leftarrow \left(s^2 + \left(\Omega_{p1}\Omega_{p2}\right)\right)/\left(s\left(\Omega_{p2} - \Omega_{p1}\right)\right)$	2N
Lowpass to Bandstop	$s \Leftarrow \left(s\left(\Omega_{p2} - \Omega_{p1}\right)\right)/\left(s^2 + \left(\Omega_{p1}\Omega_{p2}\right)\right)$	2N

Example1.1

The analog filter implementation in figure (2.12) and defined by

$$H_p(s) = 1/\left(s^3 + 2s^2 + 2s + 1\right); \ldots\ldots\ldots\ldots\ldots\ldots (1.26)$$

The filter has poles located at s=-1.0 and s= 0.5 ±j0.866.

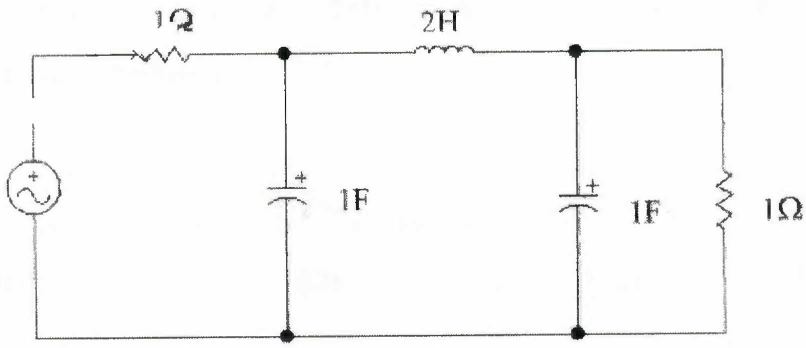


Figure 1.14(a)

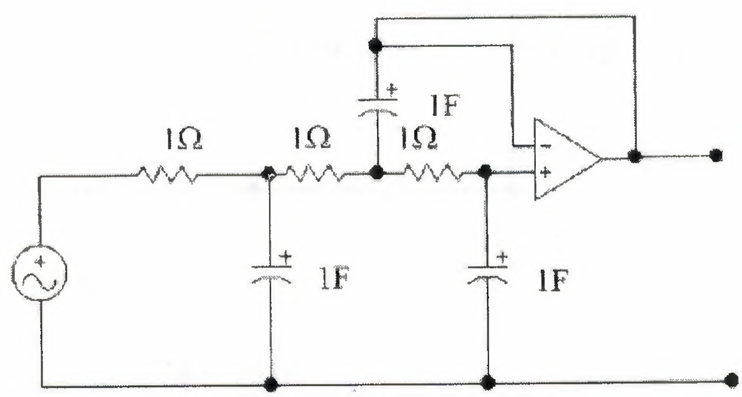


Figure1.14(b)

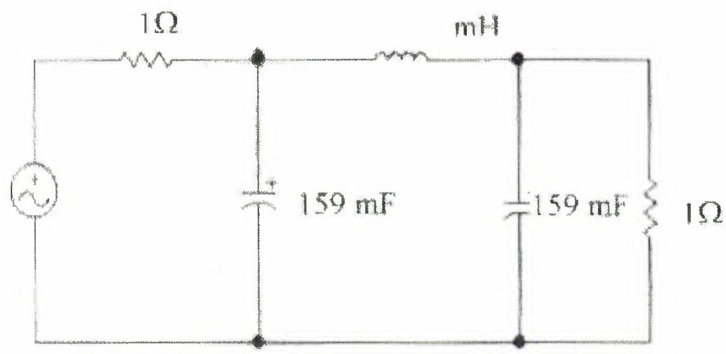
The magnitude-frequency response of the prototype filter has the gain is –3 dB at $\Omega = 1$ rad/sec. Assume that the desired filter has a similar shape, except that the desired –3 dB critical frequency is translated out to 1 kHz. The desired filter can be realized by scaling the prototype model $H_p(\Omega)$, under the appropriate lowpass-to-lowpass frequency-frequency in particular,

$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}\bigg|_{s \to s/\Omega p} ; \qquad (1.27)$$

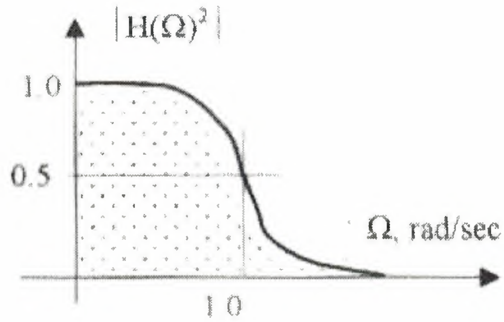
Where Ω_p is given by $\Omega_p = 2\pi 1000(rad/\sec)/1(rad/\sec) = 6283$. The final filter H(s) then becomes,

$$H(s) = \frac{1}{4.03 \times 10^{-12} s^3 + 5.07 \times 10^{-8} s^2 + 3.18 \times 10^{-4} s + 1} ; \qquad (1.28)$$

The new poles are located at s = -6283 and s = -3142 $\pm$j5441. Note also that the poles



(a)



(b)

Figure 1.15

Are scaled by $s = s/\ \Omega_p$. The frequency response of the final filter, and its attendant circuit elements are shown in Figure(1.13)

Chapter 2

2.APPROXIMATION POLYNOMIAL FOR FILTERING

2.1. Approximations

2.1.1. Butterworth Approximation

The magnitude-squared response of an analog lowpass Butterworth filter $H_a(s)$ of N^{th} order is given by,

$$|H_a(j\Omega)|^2 = \frac{1}{1+(\Omega/\Omega_C)^{2N}} \quad \text{.....................................(2.1)}$$

It can be easily shown that the first 2N-1 derivatives of $|H_a(j\Omega)|^2$ at $\Omega = 0$ are equal to zero. and as a result, the Butterworth filter is said to have a maximally-flat magnitude $\Omega = 0$ The gain of the Butterworth filter in dB is given by,

$$G(\Omega) = 10\log_{10}|H_a(j\Omega)|^2 dB \quad \text{.....................................(2.2)}$$

A dc ie , at = 0, the gain in dB is equal to zero, and at c, the gain is,

$$G(\Omega_C) = 10\log_{10}(1/2) = -3.0103 \cong -3dB \quad \text{.....................(2.3)}$$

And therefore, He is often called the *3*-dB cutoff frequencies. Since the derivative of the squared-magnitude response, or equivalently, of the magnitude response is always negative for positive values of Ω, the magnitude response, is monotonically decreasing with increasing Ω for $\Omega >> \Omega_c$ the squared-magnitude function can be approximated by,

$$|H_a(j\Omega)|^2 = \frac{1}{1+(\Omega/\Omega_c)^{2N}} \quad \text{..........................(2.4)}$$

The gain $g(\Omega_2)$ in dB at $\Omega_2 = 2\Omega_1$ with $\Omega_1 >> \Omega_c$ is given by,

$$G(\Omega_c) = -20\log_{10}\left(\frac{\Omega_2}{\Omega_c}\right)^{2N} = G(\Omega_1) - 6N \text{ dB,...(2.5)}$$

Where G(Ω_1) is the gain in dB at Ω_1 As a result, the gain roll-off per octave in the stopband decreases by 6 dB, or equivalently, by 20 dB per decade for an increase of the filter order by one. In other words, the passband and the stopband behaviors of the magnitude response improve with a corresponding decrease in the transition band as the fitter order N increases. A plot of the magnitude response of the normalized Butterworth lowpass filter with $\Omega_c = 1$ for some typical values of N is shown in Figure 2.1

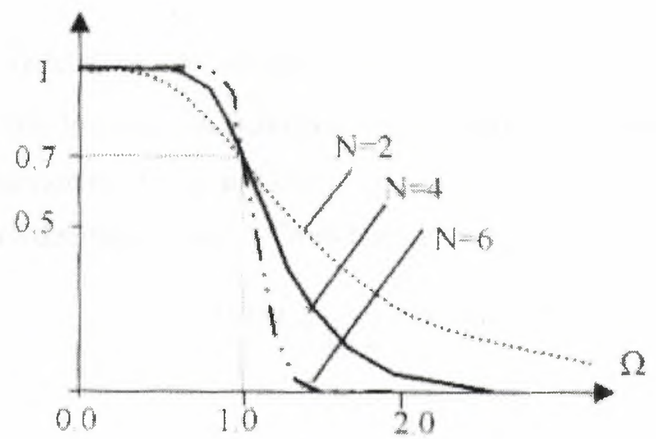


figure2.1

The two parameters completely characterizing a Butterworth filter are therefore the 3-dB cutoff frequency Ω_c and the order N. These are determined from the specified passband edge Ω_p the minimum passband magnitude $1/\sqrt{1+\varepsilon^2}$, the stopband edge Ω_s , and the maximum stopband ripple 1/A we get,

$$|H_a(j\Omega_p)|^2 = \frac{1}{1+(\Omega_p/\Omega_c)^{2N}} = \frac{1}{1+\varepsilon^2} \quad \text{..................(2.6)}$$

$$|H_a(j\Omega_S)|^2 = \frac{1}{1+(\Omega_S/\Omega_c)^{2N}} = \frac{1}{A^2} \quad \text{......................(2.7)}$$

By solving the above we get the expression for the order N as,

$$N = \frac{1}{2}\cdot\frac{1\log_{10}\left[(A^2-1)/\varepsilon^2\right]}{\log_{10}(\Omega_s/\Omega_p)} = \frac{\log_{10}(1/K_1)}{\log_{10}(1/K)} \quad \text{........................(2.8)}$$

Since the order N of the filter must be an integer, the value of N computed using the above expression is rounded up to the next higher integer This value of N can be used next in either Eq.(2.7) or (2.8) to solve for the 3-dB cutoff frequency Ω_c If it is used in Eq.(2.7), the passband specification is met exactly, whereas the stopband specification is exceeded On the Other hand, if it is used in Eq.(2.8), the stopband specification is met exactly, whereas the passband specification is exceeded

The expression for the transfer function of the Butterworth lowpass filter is given by,

Example2.1. Determine the lowest order of a transfer function $HH_a(s)$ having a maximally flat lowpass characteristic with a I-dB cutoff frequency at I kHz and a minimum attenuation of 4O dB at 5 kHz

We first determine and A From Eq( )

$$10\log_{10}\left(\frac{1}{1+\varepsilon^2}\right) = -1, \quad \text{..........................(2.9)}$$

Which yields Likewise,

$$10\log_{10}\left(\frac{1}{A^2}\right) = -40, \quad \text{..........................(2.10)}$$

Which leads to $A^2 = 10{,}000$. Substituting the values in Eq.

$$K_1 = \frac{\varepsilon}{\sqrt{A^2-1}}$$ We get,

$$\frac{1}{K_1} = 196.51334 \quad \text{..............................(2.11)}$$

The inverse transition ratio here is 1/k = 5000/1000= 5 Substituting these values in Eq(2.8) we get

$$N = \frac{\log_{10}(1/K_1)}{\log_{10}(1/K)} = \frac{\log_{10}(196.51334)}{\log_{10}(5)} = 302811022 \quad ...(2.12)$$

Since the order of the transfer function must be an integer, we round the above to the nearest integer N= 4

2.1.2.Chebyshev Approximation

There are two types of Chebyshev transfer functions In the Type 1 approximation, the magnitude characteristic is equiripple in the passband and monotonic in the stopband, whereas in the Type 2 approximation, the magnitude response is monotonic in the passband and equiripple in the stopband.

a)Type 1 Chebyshev Approximation

The type 1 Chebyshev transfer function $H_a(s)$ has a magnitude response given by,

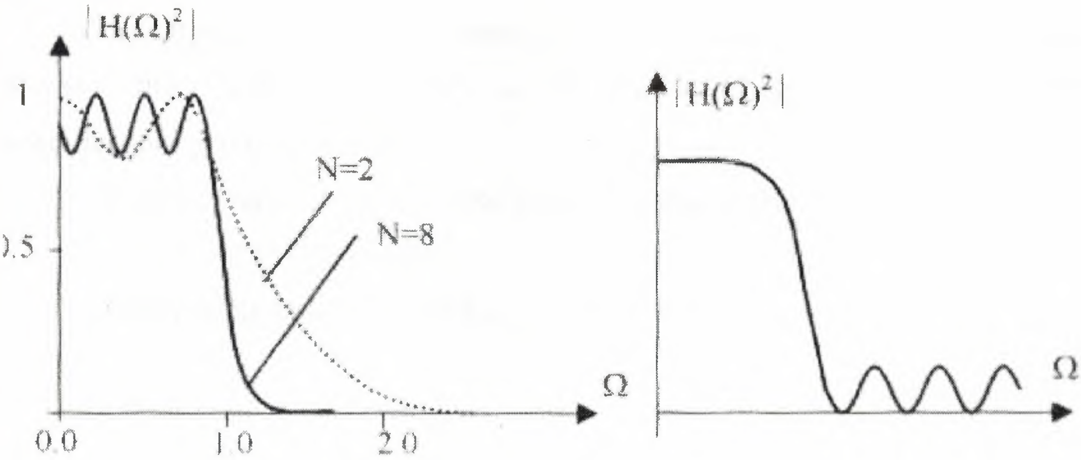
$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 T_N^2(\Omega/\Omega_p)}, \quad ...(2.13)$$

Where is the Chebyshev polynomial of order N

$$T_N(\Omega) = \begin{cases} \cos(N\cos^{-1}\Omega), & |\Omega| \le 1, \\ \cosh(N\cosh^{-1}\Omega), & |\Omega| > 1, \end{cases} \quad ...(2.14)$$

Typical plots of the magnitude responses of the Type 1 Chebyshev Iowpass filter are shown in Figure2.6 for three different values of filter order N with the same passband ripple ε From these plots it is seen that the square-magnitude response is equiripple between $\Omega = 0$ and $\Omega = 1$, and it decreases monotonically for all $\Omega > 1$

The order N of the transfer function is determined from the attenuation specification in the



a) b)

Figure2.2

Stopband at a particular frequency. For example, if at $\Omega = \Omega_s$ the magnitude is equal to 1/A, then from Eqs (2.12) and (2.13),

$$|H_a(j\Omega_s)|^2 = \frac{1}{1+\varepsilon^2 T_N^2(\Omega_S/\Omega_P)} = \frac{1}{1+\varepsilon^2\{\cosh[N\cosh^{-1}(\Omega_s/\Omega_p)]\}^2} = \frac{1}{A^2} \quad \text{............(2.15)}$$

$$N = \frac{\cosh^{-1}\left(\sqrt{A^2-1}/\varepsilon\right)}{\cosh^{-1}(\Omega_s/\Omega_p)} = \frac{\cosh^{-1}(1/K_1)}{\cosh^{-1}(1/K)} \quad \text{............(2.16)}$$

b) Type 2 Chebyshev Approximation

The Type 2 Chebyshev magnitude response, also known as the inverse Chebyshev response, exhibits a monotonic behavior in the passband with a maximally *flat* response at $\Omega = 0$ and an equiripple behavior in the stopband. The square-magnitude response expression here is given by,

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2\left[\frac{T_N(\Omega_S/\Omega_P)}{T_N(\Omega_S/\Omega)}\right]^2} \quad \text{............(2.17)}$$

Typical responses are as shown in Figure2.7. The transfer function of a Type 2 Chebyshev lowpass filter is no longer an all-pole function and has both poles and zeros If we write, The order N of the Type 2 Chebyshev lowpass filter is determined from given ε, Ω_S and A using Eq. (2.15).

Example2.2 We wish to determine the minimum order N required designing a lowpass filter with a Chebyshev or an inverse Chebyshev response with the specifications given in Example2.1.

From Example2.1, we have the following parameters.

$1/k_1 = 19651334$ and $1/k = 5$.

Substituting these values in Eq. (2.16), we arrive at,

N = 2.6059 1..................................(2.18)

Since the order of the filter must be an integer, we choose the next highest integer value 3 for N. Note that the order of the Chebyshev lowpass filter, is lower than that of a Butterworth lowpass filter meeting the same specifications as given by Eq (2.12). This is invariably the case for $N \geq 2$.

2.1.3.Elliptic Approximation

An elliptic filter, also known as a Gauer filter, has an equiripple passband and an equiripple stopband magnitude response, as indicated in Figure1.3 for typical elliptic lowpass filters. The transfer function of an elliptic filter meets a given set of filter specifications, passband edge frequency Ω_P, stopband edge frequency, passband ripple Ω_S, and minimum stopband attenuation A, with the lowest filter order N. The theory of elliptic filter approximation is mathematically quite involved. The square-magnitude response of an elliptic lowpass filter is given by,

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 R_N^2(\Omega/\Omega_P)} \quad \text{...(2.19)}$$

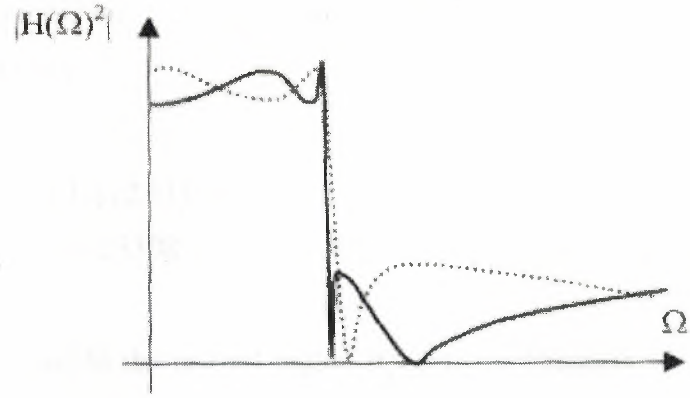


figure2.3

$$H(\Omega)=\frac{1}{N+M+1}e^{j\Omega[(N-M)/2]}\frac{\sin\left[\Omega\left(\frac{M+N+1}{2}\right)\right]}{\sin(\Omega/2)} \quad \text{.........................(2.20)}$$

for most applications, the filter order meeting a given set of specifications of passband edge frequency Ω_P , passband ripple ε, , stopband edge frequency Ω_S and the minimum stopband ripple A can be estimated by using the approximate formula,

$$N\cong\frac{2\log_{10}(4/K_1)}{\log_{10}(1/p)} \quad \text{..(2.21)}$$

Where k_1 is the discrimination parameter and is computed as follow

$$K^{'}=\sqrt{1-K^2} \quad \text{...(2.22(a))}$$

$$\rho_0=\frac{1-\sqrt{K^{'}}}{2\left(1+\sqrt{K^{'}}\right)} \quad \text{...(2.22(b))}$$

$$\rho=\rho_0+2(\rho_0)^5+15(\rho_0)^9+150(\rho_0)^{13} \quad \text{...............................(2.22(c))}$$

Where k is the selective parameter.

Example2.3 We wish to determine the minimum order N required designing an Iowpass elliptic filter with the specifications given in Example 2.1

From Example2.1, we note that

K *1/5* 0 2 and $1/K_1$ = 196.5 134

Substituting these values in Eq (2.22(a), 2.22(c)), We arrive at

$$K' = 0.979796. \qquad \rho_0 = 0.00255135,$$

$$\rho = 0.0025513525 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.22(d))$$

Using the above values in Eq (2.21), we get

$$N=2.23308. \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.22(e))$$

Rounding the above value to the nearest higher integer, we obtain N =3 as the appropriate order for the elliptic lowpass filter.

2.1.4.Linear-Phase Approximation

The previous three approximation techniques are for developing analog lowpass transfer functions meeting specified magnitude or gain response specifications without any concern for their phase responses. In a number of applications it is desirable that the analog lowpass, filter being designed have a linear-phase characteristic in the passband, in addition to approximating the magnitude specifications. One way to achieve this goal is to cascade an analog allpass filter with the filter designed to meet the magnitude specifications, so that the phase response of the overall cascade realization approximates linear-phase response in the passband This approach increases the overall hardware complexity of the analog filter and may not be desirable for designing an analog anti-aliasing filter in some A/D conversion or designing an analog reconstruction filter in D/A conversion applications It is possible to design a lowpass filter that approximates a linear-phase characteristic in the passband but with a poorer magnitude response than that can be achieved by the previous three techniques. Such a filter has an all-pole transfer function of the form,

$$H(s) = \frac{d_0}{B_N(s)} = \frac{d_0}{d_0 + d_1 s + \ldots + d_{N-1} s + s^N} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.23)$$

and provides a maximally flat approximation to the linear-phase characteristic at $\Omega = 0$, i.e., has a maximally flat constant group delay at dc $(\Omega = 0)$ For a normalized group delay of unity at dc, the denominator polynomial $fi\ B_N(s)$ of the transfer function, called the Bessel polynomial, can be derived via the recursion relation,

$$B_N(s) = (2N-1)B_N - 1(s) + s^2 B_{N-2}(s) \qquad (2.24)$$

starting with $B_1(s) = s+1$ and $B_2(s) = s^2 + 3s + 3$. Alternatively, the coefficients of the Bessel polynomial $B_N(s)$ can be found from,

$$d_\ell = \frac{(2N-\ell)!}{2^{N-\ell}\ell!(n-\ell)!}, \qquad \ell = 0,1,\ldots,N-1 \qquad (2.25)$$

These filters are often referred to as Bessel filters.

2.2. Analog Filter Design Using MATLAB

The Signal Processing Toolbox in MATLAB includes **a** number of M-file functions to directly develop analog transfer functions for each one of the above approximation techniques.

2.2.1. Butterworth filter

The Matlab files for the design of analog Butterworth filters are

```
[z, p, k]      =buttap( N ),
[num, deni]    = butter ( N, Wn, 's' ),
[num, den]     = butter ( N, Wn, 'filtertype', s' )
[N,Wn]         =buttord( Wp,Wn,Rp,'s' )
```

The function buttap (N) computes the zeros, poles, and gain factor of the normalized analog Butterworth lowpass filter transfer function of order N with a 3-dB cutoff frequency of I The output files are the length N column vector p providing the locations of the poles, a null vector z for the zero locations, and the gain factor k The form of the transfer function obtained is given by,

$$H_a(s) = \frac{P_a(s)}{D_a(s)} = \frac{K}{(s-p(1))(s-p(2))\ldots(s-p(N))} \qquad (2.26)$$

To determine the numerator and denominator coefficients of the transfer function from the zeros and poles computed, we need to use the M-file function *zp2tf (z, p, k)*.

Alternatively, we can use the function butter *(N, Wn, 's'*) to design an order N lowpass transfer function with a prescribed 3-dB cutoff frequency at *Wn rad/sec,* a nonzero number The *output data of this function are the numerator and the denominator coefficient vectors, num and den,* respectively, in descending powers of s If *Wn is* a two element vector *[W1, W2]* with *WI < W2,* the function generates an order *2N* bandpass transfer function with 3-dB bandedge frequencies at *WI* and *W2* with both being nonzero numbers To design an order *N* highpass or an order *2N* bandstop filter, the function *butter(N,Wn, 'filter type'* , 's') is employed where filter type = high for a highpass filter with a 3-dB cutoff frequency at *Wn* or filter type = stop for a bandstop filter with 3-dB stopband edges given by a two-element vector of nonzero numbers *Wn =[W1, W2]* with *WI < W2.*

The function *buttord (Wp, Ws, Rp, Rs, 's')* computes the lowest order *N* of a Butterworth analog transfer function meeting the specifications given by the filter parameters, *Wp, Ws, Rp. and Rs,* where *Wp* is the passband edge angular frequency in rad/sec, *Ws* is the stopband edge angular frequency in rad/sec, Rp is the maximum passband attenuation in dB, and *Rs* is the minimum stopband attenuation in dB The output data are the filter order *N* and the 3-dB cutoff angular frequency *Wn* in rad/sec This function can also be used to calculate the order *of any one of the four basic types of analog Butterworth filters. For lowpass design, Wp<Ws* whereas for highpass design, *Wp> Ws.* For the other two types, *Wp* and *Ws* are two-element vectors specifying the passband and stopband edge frequencies.

2.2.2. Type 1 Chebyshev Filter

The M-file functions for the design of analog Type 1 Chebyshev filters are as follows.

```
[z, p, k]      = cheblap ( N, Rp )
[num, den]     = chebyl ( N, Rp, Wn, 's' )
```

[num, den] = chebyl (M, Rp, Wn, 'filter-type', 's')

[N, Wn] = cheblord (Wp, Ws, Rp, Rs, 's')

Where Rp - passband ripple in dB;

Rs - the minimum stopband attenuation in dB,

Wn - the passband edge angular frequency in rad/sec

2.2.3. Type 2 Chebyshev Filter

The M-file functions for the design of analog Type 2 Chebyshev filters are,

[s, p, k] =cheb2ap(N,Rs)

[num, den] =cheby2(N,Rs,Wn,'s')

[num, den] =cheby2(N,Rs,Wn,filtertypes,'s')

[N, Wn] = cheb2ord(Wp, Ws ,Rp , Rs, 's')

The function cheb2ap (N, Rs) returns the zeros, poles, and gain factor of a normalized analog Type 2 Chebyshev lowpass filter of order N with a minimum stopband attenuation of Rs in dB. The normalized stopband edge angular frequency is set to 1. The output files are the length-N column vectors s and p. providing the locations the zeros and the poles, respectively, and the gain factor k. If N is odd, z is of length N-1 The form of the transfer function obtained is given by,

$$H_a(s) = \frac{p_a(s)}{D_a(s)} = K\frac{(s-z(1))(s-z(2)).....(s-z(N))}{(s-p(1))(s-p(2))....(s-p(N))} \qquad (2.27)$$

Where Wn is the stopband edge angular frequency in rad/sec.

2.2.4. Elliptic (Gauer) Filter

The M-file functions for the designs of analog elliptic filters are

[z, p, k] =ellipap (N, Rp, Rs)

[num, den] =ellip (N, Rp, Rs, Wn,'s)

[num, den] =ellip (N, Rp, Rs, Wn,'filtertype','s')

[N, Wn] =ellipord (Wp, Ws, Rp, Rs, 's')

The function ellipap (N, Rp, Rs) determines the zeros, poles, and gain factor of a normalized analog elliptic lowpass filter of order N with a passband ripple of Rp dB and a minimum stopband attenuation of Rs dB The normalized passband edge angular frequency is set to 1.

The output files are the length-N column vectors z and p. providing the locations of the zeros and the poles, respectively, and the gain factor k If N is odd, z is of length N-1 The form of the transfer function obtained is as given in Eq (2.27)

The function ellip (N, Rp, Rs, Wn, 's') returns the transfer function of an elliptic analog lowpass filter when Wn is a scalar defining the passband edge angular frequency in rad/sec or a bandpass filter when Wn is a two-element vector defining the passband edge frequencies in rad/sec. The function ellip (N, Rp, Rs, Wn, 'filtertype', 's') is used to determine the transfer function of an elliptic highpass when filtertype = high, and Wn is a scalar defining the stopband edge angular frequency in rad/sec, or a bandstop filter when filtertype stop and Wn is a two-element vector defining the stopband edge angular frequencies in rad/sec. In all cases, the specified passband ripple is Rp dB and the minimum stopband attenuation is Rs in dB the output files are the vectors, num and den, containing the numerator and denominator coefficients in descending powers of s.

2.2.5. Bessel Filter

For the design of a Bessel tiller, the available M-file functions are,

```
[z,p,k]=besselap(N)
[num,den]=besself(N,Wn)
[num, den] = besself ( N, Wn, 'filtertype')
```

The function besselap (N) is employed to compute the zeros, poles, and gain factor.

2.3.Design examples

Examples blow are illustrating the use of some of the above functions in the design of analog filters. In the first three examples, we repeat Examples2.3 through2.4 to determine the order of the transfer function using the respective M-file functions In the remaining examples, we determine the corresponding transfer functions and then compute the frequency response using the M-file function freqs(num, den, w), where num and den are the vectors of the numerator and denominator coefficients in descending powers of s, and w is a set of specified discrete angular frequencies. This function generates a complex vector of frequency response samples from which magnitude and/or phase response samples can be readily computed.

Example2.4. We next determine the order of analog Type 1 and Type 2 Chebyshev lowpass filter meeting the same specifications as above To this end, we employ the commands [N, Wn] = cheblord (Wp, Ws, Rp, Rs, 's') and [N, Wn] = cheblord (Wp , Ws, Rp, Rs, 's') respectively. For the Type 1 Chebyshev filter the computed output data are N = 3 and Wn = 6 2 8 3 18, and for the Type 2 Chebyshev filter, the computed output data are N = 3 and Wn = 23440 97 The computed Wn in the former case, is the 3-dB passband edge angular frequency in rad/sec, whereas in the second case, it is the 40-dB stopband edge angular frequency in rad/sec. The order determined is identical to that derived in Example.

```
» [N,Wn]=buttord(2*pi*(1000),2*pi*(5000),1,40,'s')
N = 4
Wn = 9.9347e+003
» [z,p,k]=buttap(4)
z = []
p =
 -0.3827 + 0.9239i
 -0.3827 - 0.9239i
 -0.9239 + 0.3827i
 -0.9239 - 0.3827i
k =1
» [num,den]=zp2tf(z,p,k)
num = 0    0    0    0    1
den = 1.0000    2.6131    3.4142    2.6131    1.0000
» omega=0:1:5;h=freqs(num,den,omega);
» gain=20*log10(abs(h));
» plot(omega,gain)
» grid
```

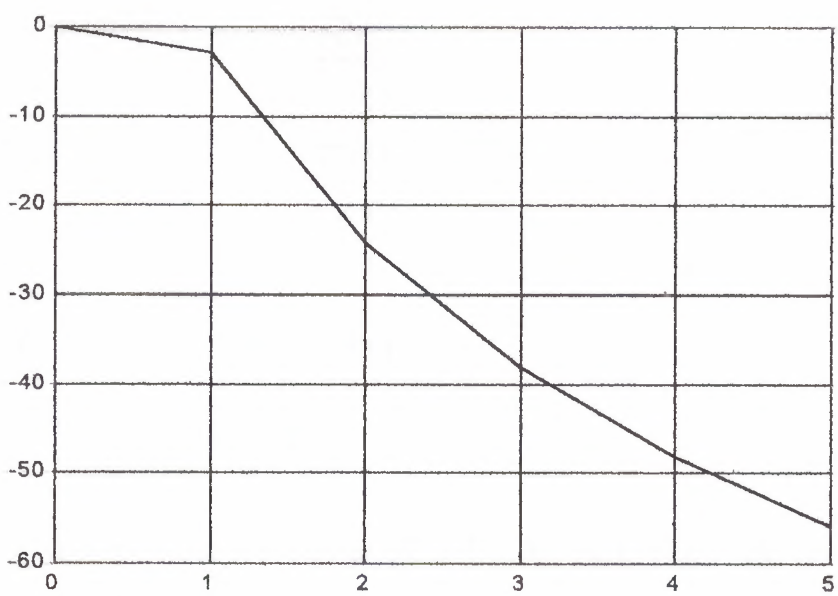


figure2.4

Example2.5. To determine the order of an analog elliptic lowpass filter meeting the above specifications, we use the command [M, Wh] = elliprod (Wp, Ws, Rp, Rs, 's'), resulting in the output data N = 3 and Wn = 6283 .185 and verifying the order obtained in Example2.3. Here Wn is the 1-dB passband edge angular frequency.

```
» Fp=1000;N=4;
» [num,den]=butter(N,Fp,'s')
num =1.0e+011 *
0       0       0       0  10.0000
den =1.0e+011 *
0.0000   0.0000   0.0000   0.0261   10.0000
» omega=0:200:6000;
» h=freqs(num,den,omega);
» gain=20*log10(abs(h));
» plot(omega,gain);
» grid
```

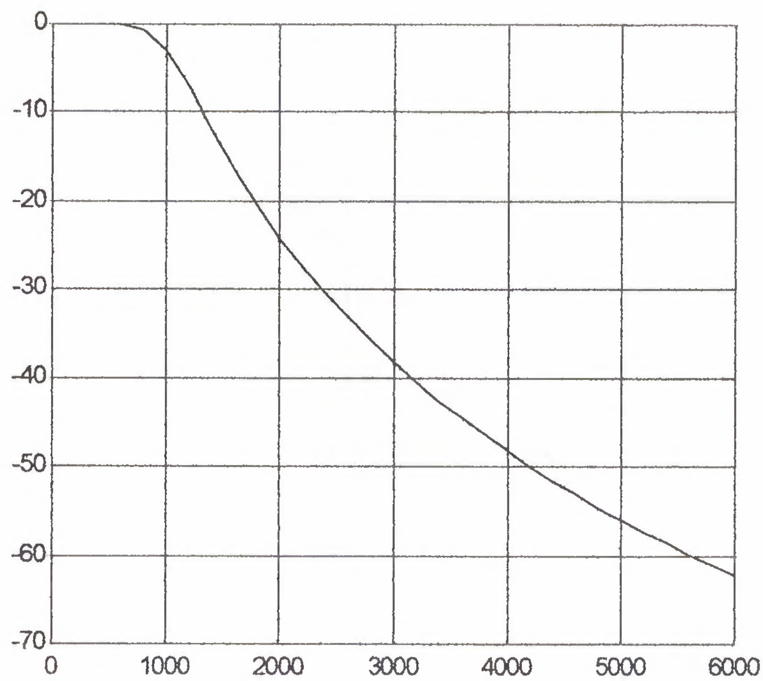


figure2.5

Example2.6. We consider first the design of a fourth order maximally flat analog lowpass filter with a 3-dB cutoff frequency at $\Omega = 1$ The pertinent MATLAB program is given below.

```
» N=4;
» Rp=2;
» Fp=1000;
» [num,den]=cheby1(N,Rp,Fp,'s')
num = 1.0e+011 *
 0        0        0        0    1.6345
den = 1.0e+011 *
 0.0000   0.0000   0.0000   0.0052   2.0577
» h=freqs(num,den,omega);
» gain=20*log10(abs(h));
» plot(omega,gain)
» grid
```

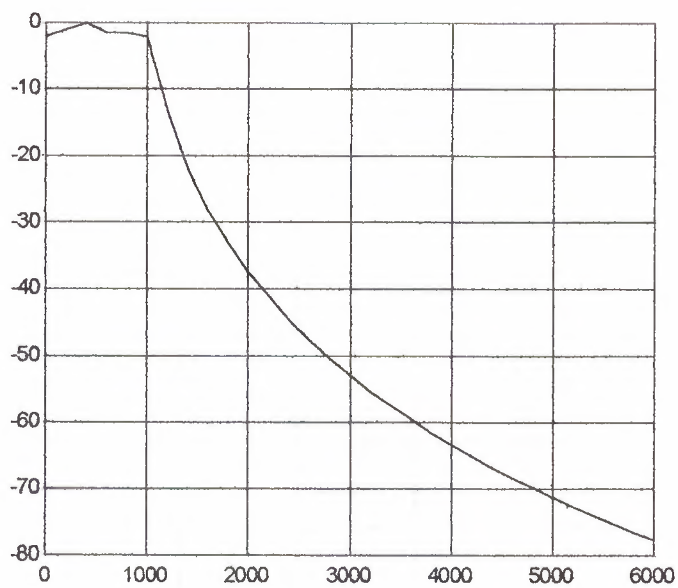


figure2.6

Example2.7. We now use MATLAB to complete the design of the Butterworth lowpass filter of Example2.1. To this end we modify Program2_1 of Example2.6 as indicated below in Program2_1 During execution, the program asks for the order of the filter and the 3-dB cutoff angular frequency (determined in Example2.3. to be equal to 4 and 7947.77, respectively) to be typed in, It then generates the gain response plot,As can be seen from this plot, the 1-dB passband edge is at 1 kHz, as desired, and at 5 kHz the attenuation is more than 40 dB, as expected.

```
» Rp=2;
» N=4;
» Fp=1000;
» Fs=5000;
» Rp=1;
» Rs=40;
» [num,den]=ellip(N,Rp,Rs,Fp,'s')
num =1.0e+011 *
0.0000  -0.0000   0.0000  -0.0000   3.2196
den =1.0e+011 *
0.0000   0.0000   0.0000   0.0080   3.6125
» h=freqs(num,den,omega);
» gain=20*log10(abs(h));
» plot(omega,gain)
» grid
```

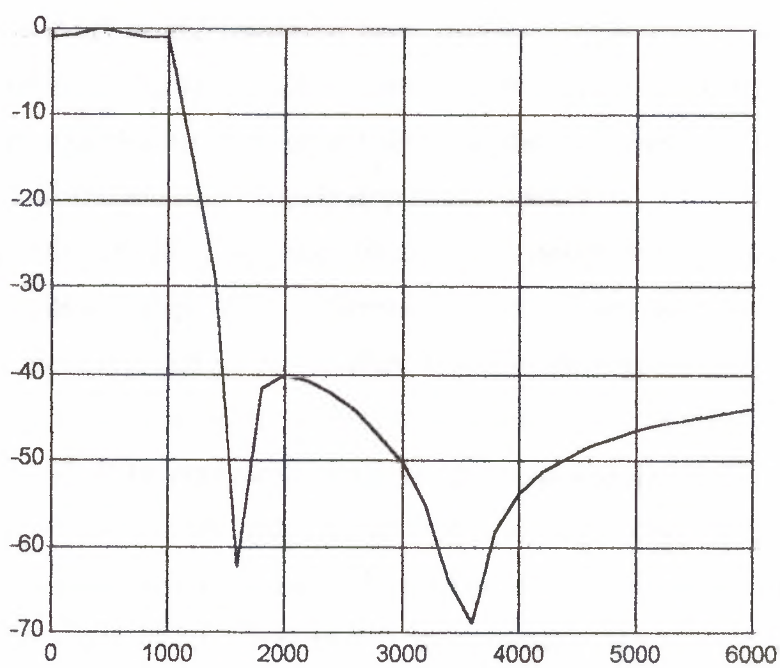


figure2.7

stopband attenuation of 40 dB, requires a Butterworth filter of order 29, a Chebyshev type 1 or 2 filter of order 10,

Example2.8. We next illustrate the design of a Type 1 Chebyshev lowpass filter meeting the specifications of Example2.1. The corresponding MATLAB program is given below in Program 2_3. As the program is run, it asks for the order of the filter, the passband edge angular frequency (determined in Example 2.4 to be equal to 3 and 6283. 18. respectively), and the passband ripple (1 dB) to be typed in. It then generates the gain response plot.

```
» [num,den]=butter(8,1,'s');
» omega=0:.05:1.5;
» h=freqs(num,den,omega);
» gain=20*log10(abs(h));
» plot(omega,gain);grid;hold on
» [num,den]=ellip(8,1,40,1,'s');
» h=freqs(num,den,omega);
» gain=20*log10(abs(h));
» plot(omega,gain);grid
» plot(omega,gain);grid; hold on
» [num,den]=cheby2(8,40,1,'s');
» h=freqs(num,den,omega);
» gain=20*log10(abs(h));
» plot(omega,gain);grid; hold on
» [num,den]=cheby1(8,1,1,'s');
» h=freqs(num,den,omega);
» gain=20*log10(abs(h));
```

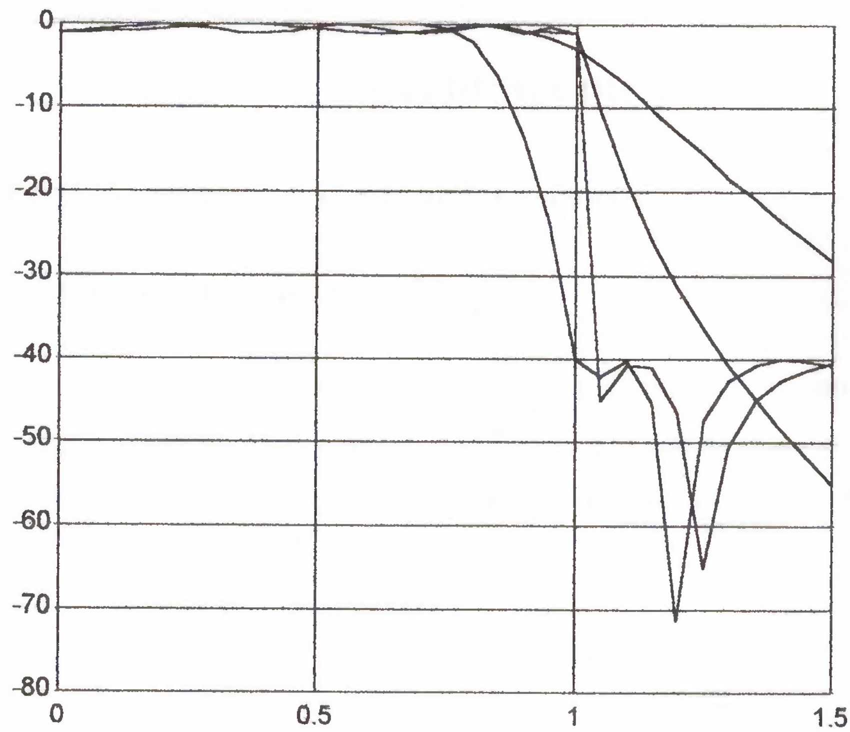


figure2.8

Chapter 3

3. FILTERING CIRCUIT

3.1. FFT, Filtering and MATLAB Analysis

3.1.1. Things The FFT Can Do

Once the waveform has been acquired and digitized, it can be fast-Fourier-transformed to the frequency domain. The FFT results can be either real and imaginary, or magnitude and phase, functions of frequency. The choice of output format belongs to the user.

Since the FFT generates the frequency spectrum for a time domain waveform, some fairly simple applications, e.g., harmonic analysis, distortion analysis, vibration analysis, and modulation measurements, might suggest themselves immediately.

Another important area is that of *frequency response estimation.* A linear, time-invariant system can be stimulated with an impulse function. Its output, the *impulse response*, can then be acquired and fast-Fourier-transformed to the frequency domain. The FFT of the impulse response, referred to as the *frequency response function*, completely characterizes the system.

Once a system's frequency response function is known, one can predict how that system will react to any waveform. This is done by *Convolution.*

An important aspect of the FFT is that *convolution* can easily be performed through frequency-domain multiplication. Let's say you know a system's impulse response, given by $h(t)$, and an input waveform given by $x(t)$. The output, say $y(t)$, caused by $x(t)$, can be computed in the classical manner by the convolution integral. But this is tedious and slow. An easier and faster approach is to FFT $x(t)$ and $h(t)$ to the frequency domain. Then the product of their frequency domain functions can be formed, giving $Y(f) = X(f)\ H(f)$. Forming this product corresponds to time domain convolution, and the convolution result can be obtained by inverse-Fourier-transforming (IFT) the $Y(f)$ function back to the time domain.

Correlation is another useful operation that the FFT makes easier. Mathematically, correlation looks and is performed in a manner similar to convolution. The difference is that one of the frequency domain functions is conjugated before the frequency domain product is formed.

Although the operations of convolution and correlation may look similar, their applications are not. Correlation is a sort of searching or looking for similarities between two waveforms. When two waveforms have absolutely no similarity, like uncorrelated noise, their correlation function is zero. On the other hand, correlation two waveforms that are exactly alike produce a perfect correlation function.

This property of finding similarities makes correlation a useful tool for detecting signals that are hidden or masked by other signals.

Another useful property of correlation is its ability to indicate delay. This is particularly useful in measuring things like path delay, path diversity, and echo return times.

Refer to the M-file list for an example of how to perform the FFT using Matlab, and also an m-file, which analyses .wav, files using the FFT.

3.1.2. Filtering

When a signal is measured as a time signal it can be converted to a spectrum. This spectral analysis shows the amplitude of the various frequencies contained within the signal. On this spectrum it is usual for a resonance to occur. The resonance is seen by comparatively large amplitude at a specific frequency. This frequency is of interest in terms of the device operating conditions.

In order to further study the resonance it is possible to employ a band pass filter for a specified frequency range. This filter allows only the frequencies within the given band to pass. This method eliminates the noise, which occurs when sampling. It should be noted that the noise present is not only electrical but may be other resonance's, aliasing, and other components of the machine or even other devices in close proximity.

3.1.3. Anti-Aliasing Filters

Aliasing is a problem when sampling a vibration. Filters can be used to avoid aliasing in signals containing many frequencies by subjecting the analog signal *x(t)* to an *antialiasing* filter. An antialiasing filter is a low-pass (i.e., only allows low frequencies through) sharp cutoff filter. The filter effectively cuts off frequencies higher than about half the maximum frequency of interest, which is also called the *Nyquist frequency*. This means that some prior knowledge of the nature of the spectrum is often required before the exact sampling rate is determined

This antialiasing filter must be employed *before* the signal is digitized. It is no good trying to use a low pass filter on the *digitized* signal because the aliasing effects occur *because of* the sampling process. Any aliasing effects would already be stored in the digitized signal and cannot be removed by low pass filtering as the effects appear as low frequencies in the signal.

It should be noted that the SoundBlaster sound card, and most of its clones, do not include antialiasing circuitry in their design. One must include a low pass filter in the data acquisition circuit when connecting to a soundcard, or at the very least it must be ensured that no frequencies greater than half the sampling rate appear in the input signal.

3.2. M-File List and Instructions

3.2.1. What is an m-file?

An M-file is script, or program, written in the MATLAB language. MATLAB, short for MATrix LABabortory, was developed by as a numeric-processing tool.

3.2.2. How do I download these m-files?

Simply click on an M-file to download it.

If you have the problem of your browser not requesting a location to save the file but instead display it in this frame, then you have to alter your browser setup. In the case of Netscape,

- From the menu, select **Options General Preferences....**

  This will bring up the General Preferences Panel.
- Select the **Helpers** tab on this General Preferences Panel.

  This will bring up the helpers panel.
- Select the **Create New Type...** button.

  This will bring up a new dialogue box.
- Type application into the Mine Type field. Then type mfile into the Mine Subtype field. Finally, select the OK button. You will be taken back to the helper's panel.
- On the helper's panel enter m at the File Extensions: line.
- Change the current Action to Save to Disk. Once all changes are entered, select the OK Button.

  This will mean that once an mfile is selected from below, you will be prompted to give a saving location on your disk.

3.2.3. What m-files are available for download?

Gibb m

Gibbs Phenomenon.

Aliasing m

An example of aliasing in the time and frequency domains.

Fft-ex m

An example of how to use the FFT in Matlab.

Wav-nlvs m

Wave File analysis.

3.3. Discrete Fourier Transform and the FFT

3.3.1. Introduction

The *Fourier Transform* provides the means of transforming a signal defined in the time domain into one defined in the frequency domain. When a function is evaluated by numerical procedures, it is always necessary to sample it in some fashion. This means that in order to fully evaluate a Fourier transform with *digital operations*, it is necessary that the time and frequency functions be sampled in some form or another. Thus the digital or *Discrete Fourier Transform* (DFT) is of primary interest.

3.3.2. The Fourier Transform

The Fourier transform is used to transform a continuous time signal into the frequency domain. It describes the continuous spectrum of a nonperiodic time signal. The Fourier transform *X(f)* of a continuous time function *x(t)* can be expressed as

$$X(f)=\int_{-\infty}^{\infty}x(t)e^{-i2\pi ft}dt \qquad (3.1)$$

The inverse transform is

$$x(t)=\int_{-\infty}^{\infty}X(f)e^{i2\pi ft}df \qquad (3.2)$$

3.3.3. The Discrete Fourier Transform

This is used in the case where both the time and the frequency variables are discrete (which they are if digital computers are being used to perform the analysis). Let x (nT) represent the discrete time signal, and let X (mF) represent the discrete frequency transform function. The Discrete Fourier Transform (DFT) is given by

$$X(mF)=\sum_{n}x(nT)e^{-inm2\pi FT} \qquad (3.3)$$

Where

$$x(nT)=\frac{1}{N}\sum_{m}X(mF)e^{inm2\pi FT} \qquad (3.4)$$

3.3.4. The Fast Fourier Transform

The *fast Fourier transform* (FFT) is simply a class of special algorithms which implement the discrete Fourier transform with considerable savings in computational time. It must be pointed out that the FFT is not a different transform from the DFT, but rather just a means of computing the DFT with a considerable reduction in the number of calculations required.

3.3.5. Approximation of Continuous Time Transforms With The DFT

The approximations involved when using the DFT in the analysis of continuous time systems must be carefully understood. There are problems that arise in the process that may lead to erroneous results unless proper precautions are taken.

While the mathematical properties of the DFT are exact, the DFT is seldom of interest as the end goal. It is usually employed to transform data, which may arise from either an actual continuous time process, or perhaps a discrete time process, which is being analyzed from a continuous time system approach. The DFT is usually used to approximate the Fourier transform of a continuous time process, and it is necessary to understand some of the limitations inherent in this approach.

There are three possible phenomena that result in errors between the computed and the desired transform. These three phenomena are (a) *aliasing*, (b) *leakage*, and (c) *the picket-fence effect.*

(a) *Aliasing.* The only solution to the aliasing problem is to ensure that the sampling rate is high enough to avoid any spectral overlap, or to use an *anti-aliasing filter.*

(b) *Leakage.* This problem arises because of the practical requirement that we must limit observation of the signal to a finite interval. The process of terminating the signal after a finite number of terms is equivalent to multiplying the signal by a *window function.* The net effect is a distortion of the spectrum. There is a spreading or leakage of the spectral components away from the correct frequency, resulting in an undesirable modification of the total spectrum.

The leakage effect cannot always be isolated from the aliasing effect because leakage may also lead to aliasing. Since leakage results in a spreading of the spectrum, the upper frequency may move beyond the Nyquist frequency, and aliasing may then result. The best approach for alleviating the leakage effect is to choose a suitable window function that minimizes the spreading.

(c) *Picket-Fence Effect*. This effect is produced by the inability of the DFT to observe the spectrum as a continuous function, since computation of the spectrum is limited to integer multiples of the fundamental frequency F (reciprocal of the sample length). Observation of the spectrum with the DFT is analogous to looking at it through a sort of "picket-fence," since we can observe the exact behavior only at discrete points. The major peak of a particular component could lie between two of the discrete transform lines, and the peak of this component might not be detected without some addition processing.

One procedure for reducing the picket-fence effect is to vary the number of points in a time period by adding zeros at the end of the original record, while maintaining the original record intact. This process artificially changes the period, which in turn changes the locations of the spectral lines without altering the continuous form of the original spectrum. In this manner, spectral components originally hidden from view can be shifted to points where they can be observed.

To summaries this section, the DFT algorithm can be used to approximate the transform of a continuous time function, subject to the following limitations and difficulties.

- The signal must be band limited, and the sampling rate must be sufficiently high to avoid aliasing.
- If it necessary to limit the length of the signal for computational purposes, the spectrum will be degraded somewhat by the leakage effect. Leakage is most severe when the simple rectangular window function is used.
- Components lying between discrete frequency lines are subject to error in magnitude due to the "picket-fence" effect.
- The magnitude level may be different from that of the continuous-time transform due to the variation in definitions.

9 element High Power Low Pass Filter Plan Review

(Plan is in PDF format, size is 17KB)

Free Radio Berkeley 9 element High Power Low Pass Filter

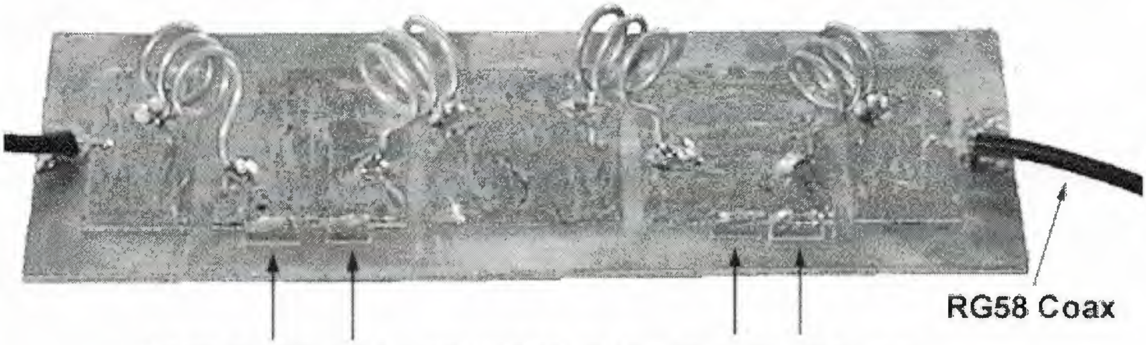


figure 3.3.1

A **Low Pass Filter** (LPF) is used after an RF amplifier to prevent the transmission of harmonic This design, which appears on Free Radio Berkley's web-site, is unusual in that it does not use discrete capacitors. In this design, the capacitive elements of the filter been made up by the capacitance formed by large areas of copper with a ground plane underneath. The filter is a 9 pole Chebyshev (each reactive element contributes one pole).

3.4. LC AND CRYSTAL FILTER SOFTWARE

3.4.1. What is this LC and crystal filter software?

Neil Heckt of Almost All Digital Electronics as an aid for filter designers to simply plug in various filter parameters and hey presto wrote the LC and crystal filter software! There is your finished design. Shortly I will present you with the design example of a crystal filter of the crystal ladder filters variety as designed on my evaluation copy of the downloaded software.

3.4.2. Designing a crystal filter

Unfortunately you can never get away from the fact that you must first of all measure the parameters of the low cost crystals you are using.

The usual design procedure proceeds as follows:

Obtain a good selection of the same frequency surplus crystals. Let's say we can obtain very cheaply some 5 Mhz crystals. We need to determine the crystal parameters in figure 1 below.

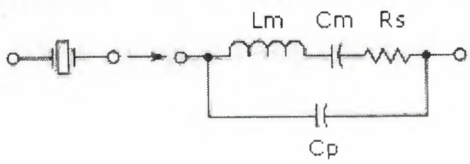


Figure 3.4.1 - parameters of crystal for a crystal filter

Now your crystal parameters can be measured relatively easy, albeit indirectly. The crystal is placed between an accurate and stable signal source of known impedance and a load of known impedance, usually a 50-ohm detector. The signal source needs to be at least 10 mW, stable, preferably 50 ohms, capable of fine tuning and monitored by a quality frequency counter with 1Hz resolution.

The crystal parameters starting with Cp can now be measured. Cp can be easily determined with a capacitance meter operating at a frequency far removed from the crystal frequency; it's as simple as that.

The signal source is slowly adjusted until a peak response is noted; this is series resonance, Fs where both the inductive reactance and capacitive reactance of the crystal cancel one another. The crystal is then replaced by a small value variable resistor and adjusted for a similar response in the detector. The value determined by the variable resistor is Rs in figure 1 above.

Next the crystal is reinserted and swept both sides of centre frequency to determine the 3 dB points, which gives us the loaded bandwidth Bw, or indirectly Q_L.

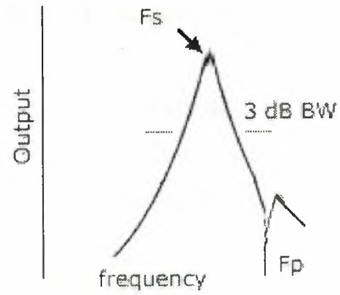


Figure 3.4.2 - crystal loaded bandwidth

As the crystal is slowly swept slightly higher in frequency, maybe only a 1,000 Hz or more a pronounced dip will be noted, this is the anti resonance or parallel resonant, Fp frequency where the effect of Cp effectively in series with Cm, resonates with Lm.

3.5. LC BAND PASS FILTERS

3.5.1. What are band pass filters?

LC Band pass filters are usually LC filters containing resonator combinations of inductance and capacitance, which are designed mathematically to respond to design frequencies while rejecting all other out of band frequencies. Because LC bandpass filters have inherent limitations these statements should not be taken too literally.

Now we move from the simple to the complicated. By this stage you should be able to understand:

Unloaded Q (Q_U)

Loaded Q (Q_L)

Reactance.

LC circuit combination for any given frequency.

LC Band pass filters are derived from tables named after the mathematicians who did the original calculations.

The main filters considered are:

Butterworth

Chebyschev

Bessel

Gaussian

Because of ease of alignment we will only consider the two and three resonator Butterworth LC bandpass filters of the relative narrow band variety. Here the term narrow band is a relative expression. Do NOT expect to design such a filter at 9 Mhz with a bandwidth of 3 KHz.

3.5.2. Designing LC Bandpass filter

Assume our source is from a proper 40-meter antenna of 50 ohms impedance and our load is a gee-whiz-bang-all-singing-all-dancing passive double balanced mixer, which also needs to see 50 ohms.

You have two options.

(a) Transformer coupling which is a turn's ratio of about 9:1 leaving us with something like an almost impractical 2.7 turns coupling.

(b) Capacitive coupling again. Where Cc is calculated by calculating reactance's: This 50 pF is then subtracted from both C_0's at either end to reduce that capacitance from 150 pF to 100 pF.

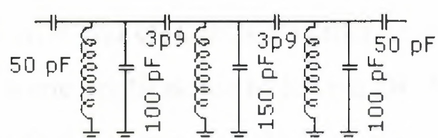


Figure 3.5.1 - LC band pass filter schematic

3.6. HIGH PASS FILTERS

3.6.1. What are high pass filters?

Assuming you have mastered the design of low pass LC filters we will now proceed to the design of a high pass filters. A high pass filter is simply the transformation of a low pass filter. For our purposes, we will say we need a five-pole butterworth filter with a cut off frequency Fc at 2000 KHz. That is we want to pass all frequencies above 2000 KHz (2 Mhz) but attenuate those below 2000 KHz.

Perhaps this might be required for the antenna input to a receiver where AM Radio interference is proving troublesome.

3.6.2. Design Procedure for high pass filters

If you have done the tutorial on low pass filters and are confused by what comes next, be aware there are literally hundreds of low pass filter types. However all low pass filters transform to high pass filters.

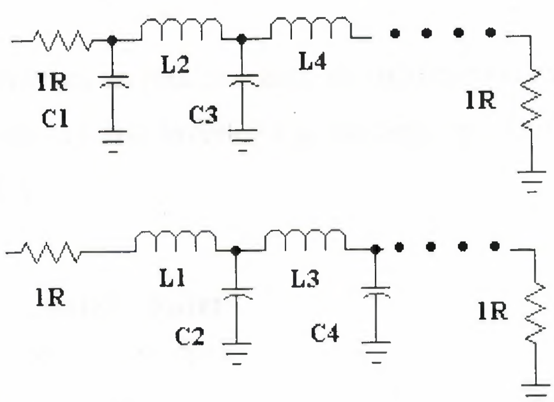


Figure 3.6.1 - low pass filters - equal terminations

Which type you choose is a matter of choice which may well be influenced by your needs in some applications to have a DC blocking capacitor in the input or output of the final finished high pass LC filter. In this case use schematic 2.

In the two schematics shown in figure 1 the principal difference is the placement of the first capacitor, denoted either C1 or C2.

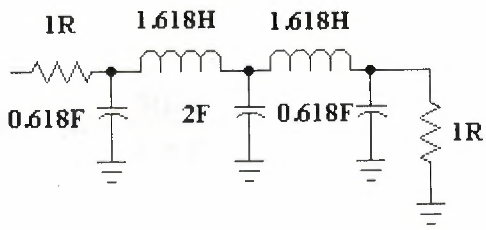


Figure 3.6.2 - low pass filters - equal terminations - normalized to 1 Hz

Notice that this low pass filter is normalized to 1 ohm impedance both in and out, a frequency of 1 Hz and capacitor values are expressed in Farads while Inductor values are in Henries.

3.6.3. Transformation to High Pass Filter Prototype

All right we have a low pass filter prototype, what now? We simply want to do the opposite to a low pass with our high pass filter, so we do the opposite and invert everything. Replace each component with it's opposite.

A capacitor becomes an inductor and, an inductor becomes a capacitor and, at the same time the values are also inverted e.g. the first capacitor of 0.618F becomes an inductor of 1 / 0.618H. Cool?

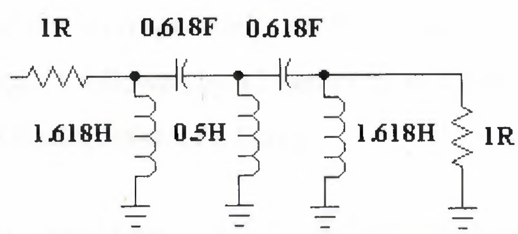


Figure 3.6.3 - transform low pass filter to high pass filter

3.6.4. Component calculations at Fc and at Zo - Frequency and Impedance scaling

This is the truly simple part if you like doing basic sums on the calculator. If not, then you're in for some bother.

The transformation is effected using the following basic, yet simple formulas for transformation LC:

$$C = \frac{C_n}{2\pi f_c R} \quad \text{AND} \quad L = \frac{RL_n}{2\pi f_c}$$

Here C is the final capacitor value, L is the final inductor value, Cn and Ln are the prototype element values in Fig 3, R is your final impedance aloe and fc is the final cut off frequency. It's as simple as that!

So for a cut off of 2000 kHz and a 50 ohms impedance the calculations for the first capacitor and inductor we encounter become, as a worked example for you.

$$C = \frac{C_n}{2\pi f_c R} = \frac{0.618F}{6.2832 \times 2{,}000{,}000 \times 50} = 9.836^{\times 10^{-10}} = 984 \text{ pF} \quad .....(3.6)$$

$$L = \frac{RL_n}{2\pi f_c} = \frac{50 \times 1.618H}{6.2832 \times 2{,}000{,}000} = 6.438^{\times 10^{-6}} = 6.438 \text{ uH} \quad ...(3.7)$$

Note that the original prototype is always expressed in terms of 1 ohm, 1 hertz (Hz), Farads and Henries.

When you do your sums you get back to numbers with negative exponents, they are the -10 and the -6 respectively. To bring capacitance to pF we multiply by exponent 12 (that's number 1 followed by 12 zeroes as in 1,000,000,000,000). Why? Because 1 Pf is one 1,000,000,000,000th of a Farad.

To bring inductance to uH we multiply by exponent 6 (that's number 1 followed by 6 zeroes as in 1,000,000). Why? Because 1 uH is one 1,000,000th of a Henry.

Your final filter comes out as follows:

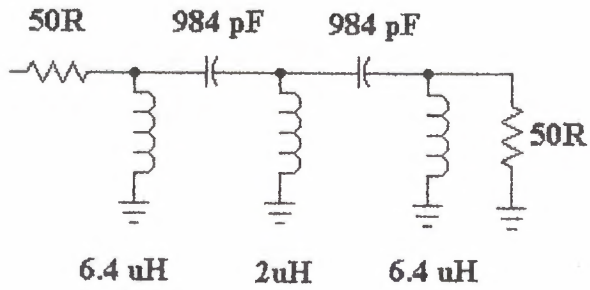


Figure 3.6.4 - final component calculations - high pass filter

3.6.5. Paranoia to avoid with high pass filter values

Firstly don't use an unnecessary precision with your values in high pass filters. A capacitance calculated as 983.5752483 pF is totally irrelevant. In the "real world we would use a standard 1000 pF capacitor, remembering it's tolerance is going to be +/- 5% anyway. Consider also, it is doubtful any impedance will be precisely 50 ohms. Finally, for this type of filter triodes are ideal and cheap to use as inductors.

3.7. IF AMPLIFIER FILTERS

3.7.1. What are IF amplifier filters?

For reasons of clarity, formulas will often be .gif files, when you see the first one you will realize why.

At the outset please understand that the best response you can possibly get from if amplifiers filters would be a 1% fractional bandwidth and even then you would be pushing it very hard.

3.7.2 Designing IF Amplifier filter

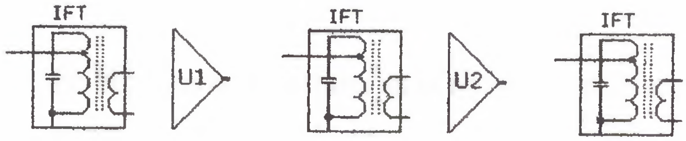


Figure 3.7.1 - if amplifier filter block diagram

Transformers separated by two stages of amplification, which could be valves, transistors or integrated circuits. We will not consider the actual active or amplifier stages here now, just the I.F. transformers and the circuit as a whole as an if amplifier filtering block.

These called "synchronously tuned filters" because an active device couples each stage. From earlier filter tutorials you will remember the filter bandwidth determined the required bandpass Q or Qbp. An example we will use throughout this tutorial will be an IF stage from a typical transistor radio at 455 KHz with a bandwidth of 10 KHz.

If our IF is 455 KHz and bandwidth is 10 KHz then Qbp = 455 / 10 = 45.5 This is a high number but single filter synchronously tuned stages (as in Fig 1) offer a relaxation on Qbp in accordance with the following formula. Here's why it's a .gif file.

$$\sqrt[m+1]{2^{1/2}-1} = \sqrt[1+1]{2^{1/3}-1} \qquad \text{...(3.8)}$$

3.7.3 Determining impedance of if amplifier filters

You will recall that Z or more correctly R = (2 * pi * Fo * L * Qbp) Typical 455 KHz IF transformers are nominally 680 uH variable inductors resonated by 180 pF capacitors. In this case we get (2 * 3.1416 * 0.455 * 680 uH * 23.2) = 45,101 which is a typical impedance for that kind of transformer.

If the collector load required for the transistor was say 10K then the transformer would be centre-tapped. If the next stage wanted to see 1K then the coupling turns winding would be the square root of 45K / 1K = 6.7 That means the coupling winding or secondary would have 1 / 6.7th number of turns as the primary.

From here on it is a matter of simple algebra to plug in the known to derive the unknowns.

An absolutely critical feature, as in all filters, are the terminating impedances.

3.8. LOW PASS FILTERS

3.8.1. What are low pass filters?

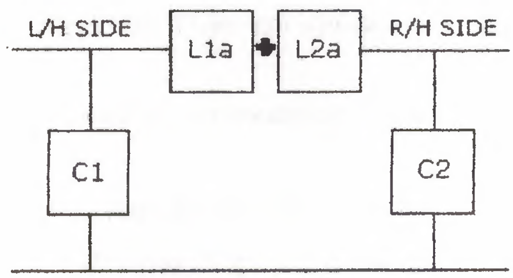


Figure 3.8.2 - two L network low pass filters

Where the reactance's are C1 = C2 = 50 ohms each and L1a = L2a = 50 ohms each or a total of 100 ohms.

The attenuation of this particular filter is given by the equation:

$$\mathbf{A_{dz} = 10 \log [\, 1 + (W / W_c)^{2n} \,]} \qquad \text{.......................................(3.9)}$$

- **W** = the frequency of desired attenuation
- $\mathbf{W_c}$ = the cutoff frequency ($\mathbf{W_{3db}}$) of the filter
- **n** = the number of elements in the filter

For our three (**n**) element filter above with a $\mathbf{W_c}$ of 7.5 Mhz and checking out the first harmonic (**W**) at 15 Mhz we find the attenuation at 15 Mhz is a mere 18.13 **dB**.

Obviously that number would not over excite you yet it is a fact of life. How many of you have thought such a LPF would yield stunning results?

A good example to investigate is the same filter used as the input to a receiver with an IF of 455 KHz. Our local oscillator runs at 7500 KHz + 455 KHz or 7955 KHz. An image frequency would be at 8410 KHz.

Well what is the use of such a low pass filter? Firstly if you consider reducing interference to the low VHF TV band you can get acceptable performance. Do a calculation of the attenuation at 50 Mhz. Secondly the filter is excellent as an impedance converter or matching device. In this circumstance consider any attenuation benefits accruing to be entirely a bonus.

CONCLUSION

The importance of analog filters is declining day after day, but the digital filters are growing widely.

The study of analog filters is of great importance because they provide the basis of the study of digitals filters. And anti-aliasing filters introduced to the signal processing system is based on analog filtering. Analog and digital conventional filters provide filtering if there is no overlap between spectrum signals and noise.

The main design of filters is simple as the low pass filters (LPF) gives the basis of other filter designs that becomes more complex. High pass and band pass filters are an upgrade or transformation of low pass filters.

Wireless and personal communication systems are increasingly being regarded as essential communication tool for future. From this point of view as new network infrastructure are implemented and competition between them increases.

The requirements of high voice quality from network provider are required, solution of this need investigation of filtering in the wireless and personal communication system where frequency scattering is one of the important problems.

The design and implementation of filter circuits is a challenge for every design engineer, and it requires a skill that cannot be learned by just taking classes and reading the necessary books. It is not `textbook knowledge' that makes a good designer, but the experience that only can be obtained by practice and exposure to design projects.

REFERENCES.

[1] Signals and systems, by ALAN V. OPPENHEIN, ALAN S.WILLSKY.

[2] Communication systems, by SIMON HAYKIN.

[3] Principles of communication systems, by HERBERT TAUB, DANALD L. SCHILLING.

[4] Digital communications, by SIMON HAYKIN.

[5] Analog and digital communication systems, by MARTIN S.

[6] http://www.irational.org/sic/radio/frb-9pole-lpf.html

[7] http://www.looksmart.com/eus1/eus317829/eus317864/eus77824

[8] http://www.airfilters.net

[9] http://www.dewinters.myweb.nl/filter.htm

[10] http://www.electronics-tutorials.com/filters/filter-software.htm

[11] http://www.thefiltermaster.com/indtestimonial.htm

[12] http://www.bluehighways.com/tifap/

[13] http://dcwi.com/~bmills/filter.html

[14] http://www.tradespan.co.za/tools/filter_elements.html