Chapter 1

INTRODUCTION

This chapter describes the roles and basic concepts of analog filters, their applications in electrical engineering, and certain fundamental ideas and terminology associated with them.

The term *filter* is used in many different ways in electrical engineering. An algorithm in a computer program that makes a decision on which commands and how certain commands are executed performs a filtering function. A decision technique that estimates the input signal from a set of signals and noise is known as optimal filtering. In analog and digital signal processing, filters eliminate or greatly attenuate the unwanted portion of an input signal. These analog and digital filtering processes may be performed in real-time or in off-line situations.

1.1 Preliminary Remarks

In this volume, we are dealing with filters that are physical continuous-time subsystems that perform certain input-output relationships in real time. Filters in this context are so widely used in electronic systems, such as electronics, military ordnance, telecommunications, radar, consumer electronics, instrumentation systems, that it is difficult to imagine any of these applications not containing components that can be identified as filters.

The development of filters started in the early part of the twentieth century. The early progress of filters was primarily associated with applications in telephony. The methods used were somewhat heuristic and empirical. Although later theoreticians could demonstrate that those earlier designs were suboptimal, those earlier achievements were not too far from the theoretical optimum in performance. In other words, when theoreticians were able to accomplish the optimum filter design, the improvements were really quite modest. This really makes the achievements of the earlier filter design without sophisticated mathematical means quite remarkable. However, in most cases, the more modern design techniques, in addition to rendering filters that are optimum in some sense, also employed a synthesis philosophy that is mathematically rigorous and scholastically satisfying.

The major progress in filter theory was largely accomplished in the 1930‘s and 1940’s. The theoretical thoroughness and elegance of this body of knowledge is, to this day, one of the most beautiful and admirable intellectual achievements in electrical engineering.

Those elegant and successful studies of filters were, however, limited to filters made of lossless elements - inductors, capacitors, and mutual inductances. In the late 1940‘s a new type of analog filters emerged - the active filters. Initial efforts were motivated by the low-frequency applications of filters in which inductors become too costly and their weights and volumes become excessive. In some of those applications, even the use of vacuum tubes could prove to be superior to the use of inductors. Because of the availability of active elements and the advent of solid-state devices and technologies, active filters eventually proved to be practical and attractive under many circumstances. In many situations, active and passive filters are equally adept in their suitability. Under other circumstances, they complement each other. On still other occasions, one type is clearly superior to the other. Hence, both passive and active filters have their places in electronic technology.

The development of active filters has been quite different from that of passive filters. Since passive filters are limited in terms of the types of components used, the available circuits and circuit configurations are rather limited. On the other hand, since there are large numbers of available active devices and configurations, the types of circuits suitable for use as active filters are also very large. Since the 1960’s until now, literally hundreds of active filter circuits were proposed. For various reasons, mostly practical ones, the commonly used active filters appear to be limited to a few more popular configurations. This is not to say that other circuits cannot perform the same filtering tasks. Rather, these popular circuits are quite adequate for almost all engineering needs. Therefore, there is no reason to try to use other known circuits or invent new ones. Our study of active filters here will be limited to circuits that have been proven practical, are widely used for various reasons, or pedagogically beneficial.

1.2 The Analog Filter

The term *analog filter* shall be used to mean that branch of filter theory that makes use of linear time-invarying elements to perform certain tasks on continuous-time analog signals. The term analog signals refers to signals that have not been quantized or digitized in their strength as functions of time.

An analog filter is typically a single-input single-output system as shown in Fig. 1.1. In Fig. 1.1(a), the input $x(t)$ and output $y(t)$ are both spec-

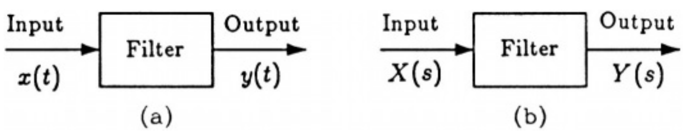


Figure 1.1: Representation of a filter (a) in the time domain and (b) in the frequency domain.

ified in the time domain. Both $x(t)$ and $y(t)$ may be either a voltage or a current. For this type of situation, the filter is often referred to as the *pulse-forming network* as it focuses on the waveshape aspects of the output-input relationship. In Fig. 1.1(b), the input-output relationship is governed by the network function $H(s)$ in the complex frequency domain where [1]

$$H(s) = \frac{Y(s)}{X(s)} \tag{1.1}$$

in which $s = \sigma + j\omega$ and $X(s)$ and $Y(s)$ may be regarded as the Laplace transform of $x(t)$ and $y(t)$ respectively.

In this volume, we shall concentrate on the situation formulated in Fig. 1.1(b). In reality, most time-domain application requirements of Fig. 1.1 (a) are translated into their equivalent in the frequency domain and then solved as if it were a frequency-domain problem [Su2].

In the frequency domain, our focus is generally directed toward either the *magnitude* and/or the *phase* of the network function on the j axis of the s plane. Or

$$H(s)\Big|_{s=j\omega} = H(j\omega) = |H(j\omega)|\, e^{j\phi(\omega)} \tag{1.2}$$

[1] In some filter literature, the reciprocal of this formalism is used. The reader should not have any difficulty in making the distinction between these two conventions.

where $|H(jw)|$ is the *magnitude function* and $\phi(\omega)$ is the *phase function.*

With the formulation of the function in (1.1), the magnitude function is the *gain* function.[2] This function is frequently expressed in dB, viz.

$$\alpha(\omega) = 20 \log |H(j\omega)| \text{ dB} \tag{1.3}$$

The phase function given in (1.2) is the phase *lead* function - the phase angle by which the output signal leads the input signal. Another important function is the *group delay*

$$T_d(\omega) = -\frac{d}{d\omega}\phi(\omega) \tag{1.4}$$

The phase function and the group delay function have profound time-domain ramifications as they have a direct effect on the waveshape of the output signals.

As a practical matter, the magnitude functions and the phase functions are usually dealt with separately. The reason for this approach is that the realization of a network function to furnish both a desirable magnitude function and a phase or delay function is simply too difficult, sometimes impossible, mathematically. There is a certain basic and inherent interrelationship between these two functions and they cannot be specified entirely independently.

In some applications, the magnitude function is the only one that matters. In that case, we simply accept the delay function that accompanies the realized magnitude function. If both the magnitude and delay functions are important, we usually prefer to realize the given magnitude function, as best we can, first. Then if the accompanying delay function is not satisfactory, we introduce additional networks, known as *delay equalizers* or *phase linearizers,* to improve the delay function.

1.3 Ideal and Approximate Filter Characteristics

In Fig. 1.2, five ideal filter magnitude characteristics are shown. They are (a) *lowpass,* (b) *highpass,* (c) *bandpass,* (d) *bandreject,* and (e) *allpass* filters. The first four types of characteristics are used for their frequency-selective properties. The allpass filter is used chiefly for its phase-linearization or delay-equalization capabilities. These ideal char-

[2] If $H(s)$ is defined as $X(s)/Y(s)$, then $|H(j\omega)|$ would the *loss function.*

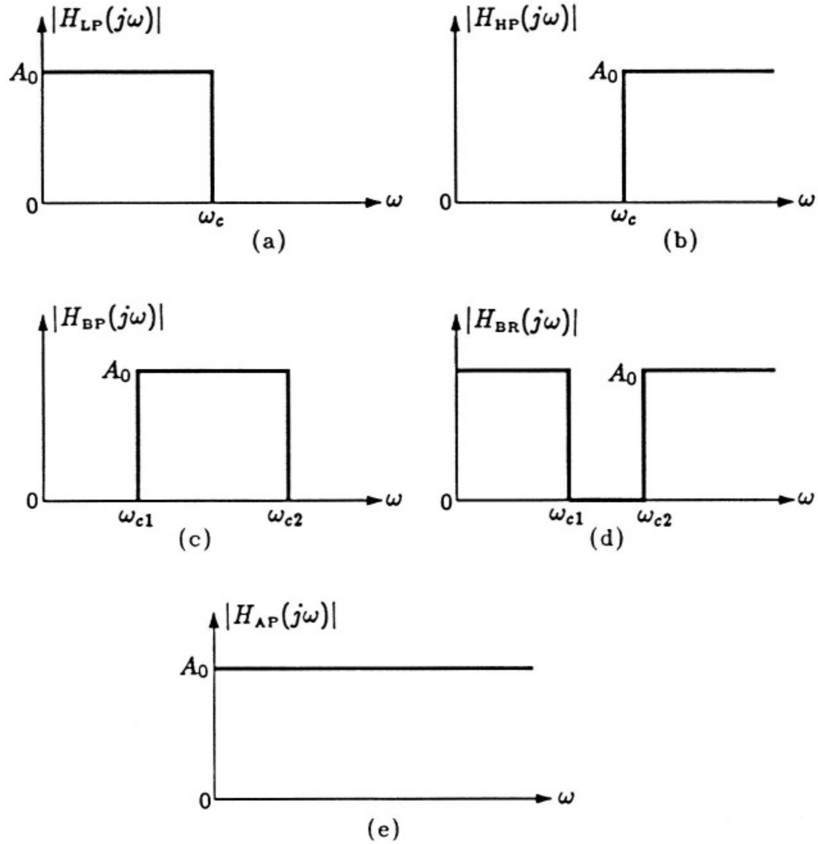


Figure 1.2: Magnitude characteristics of ideal filters.

acteristics are not realizable with finite networks. Hence all real-world filters can have only magnitudes that approximate these characteristics.

Take the approximate lowpass characteristic. We divide the frequency axis into three segments as shown in Fig. 1.3. The region $0 < \omega < \omega_p$ is considered to be the *pass band* in that the gain is relatively high so signals in this range are 'passed through.' Another phenomenon that we are forced to accept is that the gain in this region cannot be constant. Some variation is inevitable. As long as a characteristic is confined to the shaded box bound by $0 < \omega < \omega_p$ and $A_1 < |H(j\omega)| < A_0$, we normally consider it to have the same degree of approximation in the pass band. Two hypothetical characteristics are shown in Fig. 1.3. The amount of variation in the pass band is indicated by the maximum gain

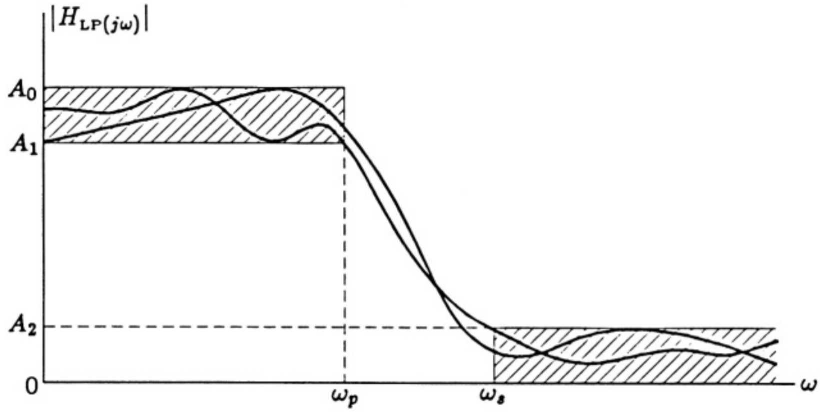


Figure 1.3: Approximate lowpass magnitude characteristics.

variation known as the *passband ripple* or *passband variation,* α_p, where

$$\alpha_p = 20 \log \left[\frac{A_0}{A_1}\right] \text{ dB} \tag{1.5}$$

At the other end of the ω axis, the range $\omega_s < \omega < \infty$, in which the gain is made very low, is called the *stop band.* Signals in this range are suppressed by the filter as much as practical - ideally with zero gain. Again, some finite gain and variation are inevitable. It is desirable that the maximum gain in this range, A_2, is made as low as practical. All magnitude characteristics that fall within the rectangle $\omega_s < \omega < \infty$ and $0 < |H(j\omega)| < A_2$ are considered to have the same degree of approximation in the stop band. The degree signals in this range are suppressed is indicated by the *stopband attenuation,* α_s, where

$$\alpha_s = 20 \log \left[\frac{A_0}{A_2}\right] \text{ dB} \tag{1.6}$$

The region $\omega_p < \omega < \omega_s$ is known as the *transition band.* Usually how a characteristic varies in this region is not given too much emphasis in selecting a filter characteristic. The quantity ω_s/ω_p is known as the *transition-band ratio.*

This general convention and terminology are easily extended to high-pass, bandpass, and bandreject characteristics. It can be argued that techniques are available by which these other three types of filters can be derived from basic lowpass ones. (Details of the relationship among these four types of filter characteristics will be taken up in Chapter 4.) This is not to say that occasions do not arise on which these other three types of filters are not to be designed by special methods not based on transforming lowpass prototype ones. These special techniques are indeed available and may be necessary to be used for special-purpose filters. However, for the purposes of this volume we will not enter the realm of these special techniques of filter specification and design.

In general, it is desirable for a filter to have the following characteristics:

1. Small passband ripple or variation, α_p.
2. Large stopband attenuation, α_s.
3. Low transition-band ratio, or $\omega_s/\omega_p \Rightarrow 1$.
4. Simple filter network. This is directly related to the order of the network function $H(s)$.

As with many engineering designs, these goals are mutually conflicting. For example, if we hold the order of the network function fixed, a small α_p can only be achieved at the expense of either a larger transition-band ratio or a lower stopband attenuation. Hence the choice of a suitable filter magnitude characteristic is a matter of compromise or trade-off of these mutually competing properties.

1.4 MATLAB

The software MATLAB is used as the standard computational tool in this volume.[3] A brief introduction to MATLAB is given here. In addition to the basic MATLAB, three toolboxes are necessary. They are the Signal Processing, Control System, and Symbolic Math Toolboxes. For readers who do not have a working knowledge of MATLAB, they should consult the user's guide or attend tutorial sessions offered by their institutions. We shall give a few examples to show how MATLAB is formulated.

The basic working units in MATLAB are matrices. The elements of the matrices may be real, complex, or symbolic. The matrices may be column vectors, row vectors, or scalars. For our purposes, almost all units are vectors or scalars.

For example, if we wish to multiply two polynomials $f_1(s) = 5s^3+4s^2+2s+1$ and $f_2(s) = 3s^2+5$, MATLAB uses the convolution of two number

[3] Other comparable software packages may be preferred by some readers. There is no compelling reason why they should not use their favorite software tools as long as those tools can perform the same computations.

sequences to obtain the coefficients of the product polynomial. The steps are

```
% Give the coefficients of the first polynomial
» f1 = [5 4 2 1];
% Give the coefficients of the second polynomial
» f2 = [3 0 5];
% Perform the convolution of the two number sequences
» f3 = conv(f1,f2)
f3 = 15 12 31 23 10 5
```

Lines proceeded by the percent symbols are comment lines and have no effect on the program. The final answer **f3** is a row vector and renders the result $f_3(s) = f_1(s) \times f_2(s) = 15s^5 + 12s^4 + 31s^3 + 23s^2 + 10s + 5$.

Using the commands from the Symbolic Math Toolbox to perform the multiplication of the same two polynomials, we will have

```
% Declare the symbolic variables
» syms s
% Define the two polynomials
» f1 = 5*s^3 + 4*s^2 + 2*s + 1;
» f2 = 3*s^2 + 5;
% Multiply the two polynomials
» f3 = f1*f2;
» f3 = expand(f3)
f3 = 15*s^5 + 12*s^4 + 31*s^3 + 23*s^2 + 10*s + 5
```

Another command that is used frequently in this volume is `roots` which gives the zeros of a polynomial whose coefficients are given as a row vector. Using the row vector of `f3` above

```
% The sym2poly command returns a row vector of
% a polynomial coefficients
» A = sym2poly(f3);
» z = roots(A)
z =
   -0.0000 + 1.2910i
   -0.0000 - 1.2910i
   -0.6553
   -0.0723 + 0.5477i
   -0.0723 - 0.5477i
```

It's implicit that the first value of `z` is `z(1)`, the second value `z(2)`, etc. Conversely, the command `poly` gives the polynomial coefficients whose zeros are known. Thus

```
» f4 = poly([z(3) z(4) z(5)])
f4 = 1 0.8 0.4 0.2
```

The two vectors `f4` and `f1` differ only by a constant multiplier. Obviously, they have the same zeros.

A row vector that consists of a number of equally spaced elements can be generated by the command `x = a:b:c`, in which `a` is the starting value, `b` is the spacing between element values, and `c` is the last value. Hence `x = 0:0.5:4` gives `x = 0 0.5 1 1.5 2 2.5 3 3.5 4`. The default value for `b` is `1`. This command is particularly convenient when a large number of points are to be generated for the plot of a curve.

MATLAB executes each command as it is entered from the keyboard. Those followed by a semicolon are executed but the answers are not displayed. This makes it difficult to debug a program if the program contains a sizable number of statements. Also, when a function is relatively long, it may be difficult to enter it through the keyboard without making mistakes. In those instances, it is best to prepare the sequence of statements in the Editor/Debugger and trial-run the program. If the program contains errors, the Debugger will cite them and the program can be modified in the Editor. The content of the program can be saved and retrieved as an 'M-file' - a file with the letter 'm' as its extension.

The reader should refer to the user's guide for other features and basic commands of MATLAB. Those commands that are particularly useful for our purposes will be introduced and explained as we use them throughout the volume.

1.5 Circuit Analysis

It is assumed that the reader has a basic knowledge and ability in circuit analysis and is capable of obtaining a network function for a given network. The analysis of a filter circuit is usually the first step in the study of a filter. In dealing with filters, certain analysis techniques are particularly suitable and efficient. We shall refresh the reader's memory with a couple of examples.

EXAMPLE 1.1. Obtain the transfer function $H(s) = E_2/E_1$ of the ladder network shown in Fig. 1.4.

SOLUTION 1. Node analysis

We set up the circuit for node analysis as shown in Fig. 1.5.

Chapter 2

THE APPROXIMATION

As was mentioned in Chapter 1, the first step in the design of a normalized filter is to find a magnitude characteristic, $|H(j\omega)|$, such that the set of specifications of an application is satisfied. Usually it is more convenient to deal with $|H(j\omega)|^2$ instead. As should be well known to the reader, a network function, $H(s)$, must be a real rational function in s (the ratio of two polynomials with real coefficients), $|H(j\omega)|^2$ can be obtained by

$$|H(j\omega)|^2 = H(s)H(-s)\Big|_{s=j\omega} = H(j\omega)H(-j\omega) = H(j\omega)H^*(j\omega) \quad (2.1)$$

Hence $|H(j\omega)|^2$ is an even function and is a rational function in ω^2. For example, if

$$H(s) = \frac{s+2}{s^3+2s^2+2s+3}$$

$$H(s)H(-s) = \frac{s^2-4}{s^6-8s^2-9}$$

$$|H(j\omega)|^2 = \frac{\omega^2+4}{\omega^6-8\omega^2+9}$$

Mathematically it's a great deal easier to deal with $|H(j\omega)|^2$ instead of the irrational function

$$|H(j\omega)| = \frac{\sqrt{\omega^2+4}}{\sqrt{\omega^6-8\omega^2+9}}$$

The study of methods of finding a real rational function in ω^2 to approximate an arbitrary magnitude or magnitude-squared characteristic is a rather specialized area. Numerous techniques using various criteria to achieve the approximation are available. These general techniques are usually needed when the magnitude characteristics do not have any standard pattern. The approximation of these general magnitude characteristics is more for magnitude equalization (as opposed to filtering) purposes. For our purposes, we are mostly interested in the magnitude characteristics that approximate the ideal filter characteristics of various types shown in Fig. 1.2. For convenience, we shall concentrate on the characteristics that approximate the ideal lowpass characteristic. In particular, we shall deal with those types of nonideal lowpass characteristics that are in common use - the Butterworth, the Chebyshev, and the elliptic-function magnitude characteristics.

The general approach in obtaining a lowpass characteristic is to seek a function of the form

$$|H(j\omega)|^2 = \frac{A_0}{1 + F(\omega^2)} \tag{2.2}$$

such that

$$F(\omega^2) \ll 1 \qquad 0 < \omega < \omega_p$$

$$F(\omega^2) \gg 1 \qquad \omega > \omega_s$$

This, in turn, makes $|H(j\omega)|^2 \approx A_0$ and $|H(j\omega)|^2 \leq A_0$ in the pass band and $|H(j\omega)|^2 \ll A_0$ in the stop band. These are the essential features of a lowpass characteristic that approximates the ideal lowpass characteristic.

2.1 The Butterworth Lowpass Characteristic

One of the simplest lowpass magnitude characteristics was first suggested by Butterworth in 1930 [Bu]. Because of its simplicity, it is often used when the filtering requirement is not too demanding.

2.1.1 The normalized Butterworth lowpass characteristic

One of the simplest choices for $F(\omega^2)$ in (2.2) is to make

$$F(\omega^2) = \omega^{2n} \tag{2.3}$$

where n is a positive integer. The magnitude-squared characteristic is

$$|H(j\omega)|^2 = \frac{1}{1+\omega^{2n}} \tag{2.4}$$

A filter that satisfies (2.4) is known as the *Butterworth filter of the nth order.* It is clear that

$$|H(j\omega)|_{\max} = H(0) = 1 \tag{2.5}$$

for any n. As ω increases, $|H(j\omega)|$ decreases monotonically. Also, for any n,

$$|H(j1)|^2 = \frac{1}{2} \tag{2.6}$$

Hence any $|H(j\omega)|$ given in (2.4) has the value $1/\sqrt{2}$ at $\omega = 1$. The gain at this point is 3.0103 dB below the maximum gain. This point is commonly referred to as the 3-dB or half-power point.

If we apply the binomial expansion of $|H(j\omega)|$, we can write

$$|H(j\omega)| = (1+\omega^{2n})^{-\frac{1}{2}} = 1 - \frac{1}{2}\omega^{2n} + \frac{3}{8}\omega^{4n} - \frac{5}{16}\omega^{6n} + \cdots \tag{2.7}$$

in the vicinity of $\omega = 0$. The first $2n-1$ derivatives of $|H(j\omega)|$ are zero at $\omega = 0$. Since $F(\omega^2)$ is of degree $2n$ in ω and we have made $|H(j0)| = 1$, (2.7) shows that we have made the $|H(j\omega)|$ curve as flat as possible at $\omega = 0$. This characteristic is often referred to as the *maximally flat magnitude characteristic.* Hence, in the range $0 < \omega < 1$, the higher n is, the flatter the characteristic is at the origin, and it approaches the ideal lowpass characteristic of Fig. 1.2(a) more closely.

For $\omega > 1$, the higher n is, the faster ω^{2n} increases and the faster $|H(j\omega)|$ decreases as ω is increased. Fig. 2.1 shows the $|H(j\omega)|$ characteristics for several values of n.

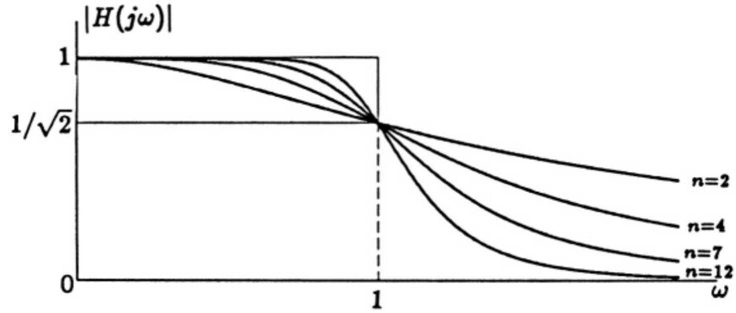


Figure 2.1: Normalized Butterworth magnitude characteristics for several values of n.

For $\omega \gg 1$, the Butterworth magnitude characteristic may be approximated by

$$|H(j\omega)|^2 \approx \frac{1}{\omega^{2n}} \tag{2.8}$$

$$\alpha \approx -10\log(\omega^{2n}) = -20n\log\omega \tag{2.9}$$

Hence the gain decreases at the rate of $20n$ dB/decade. The magnitude characteristics of Fig. 2.1 plotted as dB versus $\log\omega$ - the Bode plots - are shown in Fig. 2.2.

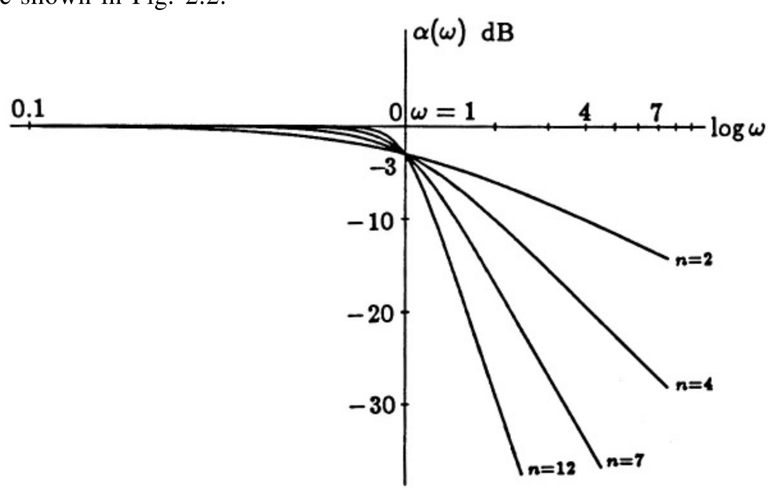


Figure 2.2: The Bode plots of the Butterworth magnitude characteristics of Fig. 2.1.

2.1.2 Using a normalized Butterworth characteristic for a filtering requirement

The squared magnitude function defined in (2.4) has the property that $|H(j1)| = 1/\sqrt{2}$ for any n. If the required filter has a passband variation of 3 dB, then we can simply make $\omega_p = 1$ rad/sec. In practice, the passband variation is not always 3 dB. Hence we may want to place ω_p and ω_s at the proper points on the frequency axis as shown in Fig. 2.3. The positioning of ω_p and ω_s is affected by the value of n. Hence we need

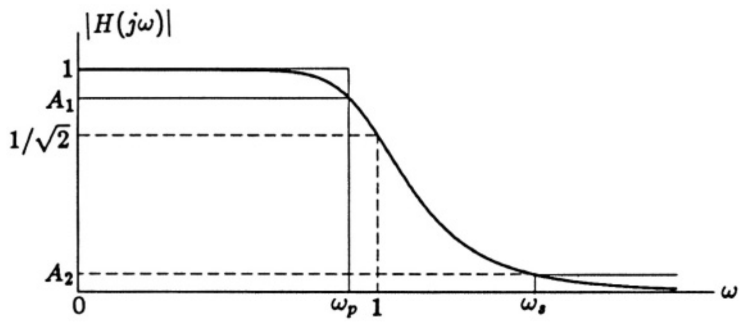


Figure 2.3: Locations of ω_p and ω_s for a Butterworth lowpass characteristic.

to adjust ω_p, ω_s, and n simultaneously. This necessary manipulation is best illustrated by an example.

EXAMPLE 2.1. Determine the Butterworth lowpass characteristic with the minimum n such that the following specifications are satisfied.

$$\alpha_p = 1 \text{ dB} \qquad \alpha_s = 25 \text{ dB} \qquad \omega_s/\omega_p = 1.5$$

SOLUTION We need

$$10\log(1+\omega_p^{2n}) = 1 \tag{2.10}$$

$$10\log(1+\omega_s^{2n}) = 25 \tag{2.11}$$

which leads to

$$\omega_p^{2n} = 10^{0.1} - 1 = 0.25893$$

$$\omega_s^{2n} = 10^{2.5} - 1 = 315.228$$

Since $\omega_s/\omega_p = 1.5$, it is necessary that

$$\left[\frac{\omega_s}{\omega_p}\right]^{2n} = 1.5^{2n} = \frac{315.228}{0.25893} = 1217.4$$

Solving, we get

$$n = 8.761$$

Since n must be an integer, we let

$$n = 9$$

From (2.10), we find[1]

$$\omega_p = 0.9277$$

For this value of ω_p and from (2.4) we get

$$\alpha_s = 10\log[1 + (1.5 \times 0.9277)^{18}] = 25.84 \text{ dB}$$

If we wish to relocate ω_p to 1 rad/sec we simply apply a frequency scaling factor $k_f = 1/0.9277$. The adjusted Butterworth lowpass magnitude characteristic function would be

$$|H_9(j\omega)|^2 = \frac{1}{1 + (0.9277\omega)^{18}}$$

2.2 The Chebyshev Lowpass Characteristic

Another commonly used standard lowpass characteristic is the Chebyshev lowpass characteristic. This class of characteristics makes use of the Chebyshev polynomial and produces an equal-ripple variation in the pass band. Outside the pass band, the gain also decreases monotonically, but at a faster rate than the Butterworth characteristics.

[1] With this choice, α_p is satisfied exactly and α_s is greater than specified. Some latitude is available here since n has been rounded off to the next higher integer.

Chapter 3

NETWORK FUNCTIONS

Once a magnitude characteristic has been chosen for a particular filter application, the next step is to determine a network function that not only has this magnitude characteristic, but is also realizable. By realizable, we mean that the function must be such that a workable network at least exists in theory, can be implemented with real-world components, and, barring unrealistic assumptions on these components, can be constructed and expected to perform the task accordingly. For example, the network function must be such that it has no pole in the right half of the s plane. If it does, the network will not be stable. The constructed network will either become nonlinear and function improperly or self destruct.

3.1 General Procedure

We now outline the procedure that will enable us to obtain a network function $H(s)$ when its $|H(j\omega)|$ (or equivalently, $|H(j\omega)|^2$) is given. In (2.1), we stated that for a given $H(s)$ its squared magnitude on the $j\omega$ axis can be obtained by

$$|H(j\omega)|^2 = H(s)H(-s)\Big|_{s=j\omega} \tag{3.1}$$

Our current objective is just the opposite. Here, we have an $|H(j\omega)|^2$ given and we write

$$H(s)H(-s) = |H(j\omega)|^2_{\omega^2=-s^2} = \left.\frac{A(\omega^2)}{B(\omega^2)}\right|_{\omega^2=-s^2} \tag{3.2}$$

If we let

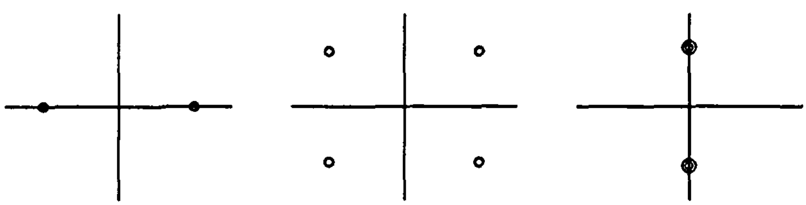
$$H(s) = \frac{P(s)}{Q(s)} \tag{3.3}$$

then we can write

$$P(s)P(-s) = A(\omega^2)\Big|_{\omega^2=-s^2} \quad \text{and} \quad Q(s)Q(-s) = B(\omega^2)\Big|_{\omega^2=-s^2} \tag{3.4}$$

The procedure for obtaining a $P(s)$ for a given $A(\omega^2)$ and that for obtaining a $Q(s)$ for a given $B(\omega^2)$ are mathematically identical. We shall arbitrarily choose to deal with $A(\omega^2)$ and $P(s)$.

Since $P(s)$ is a polynomial with real coefficients, its zeros must be either real or occur in conjugate pairs. The zeros of $P(-s)$ are the negative of those of $P(s)$. Hence the zeros of $P(s)P(-s)$ can occur only in groups each of which is one of the three types shown in Fig. 3.1.



(a) (b) (c)

In group type (a), the zeros are real. For each zero in one halfplane, there must also be another one in the other halfplane. Group type (b) contains complex zeros. They must occur in sets of four symmetric about both axes. We describe this type of group as possessing *quadrantal symmetry.* Group type (c) are j-axis zeros. In addition to being conjugate pairs, zeros of this type must be of even multiplicity. In Fig. 3.1, a conjugate pair of double zeros is indicated.

In obtaining a $P(s)$ for a given $A(\omega^2)$, we first replace every ω^2 in $A(\omega^2)$ with $-s^2$, or every w with s/j. Then $A(-s^2)$ is factored to reveal all its zeros. For real zeros, the corresponding factors will appear as $(s+a)(s-a)$. For complex zeros, the corresponding factors will appear as $(s^2+as+b)(s^2-as+b)$. For j-axis zeros, the factors will appear as $(s^2+\omega_i^2)^2$.

The construction of $P(s)$ is now clear. For each type (a) or (b) group of zeros, we allot those in either halfplane to $P(s)$, and those in the other halfplane to $P(-s)$. As far as $A(\omega^2)$ is concerned, which halfplane zeros are allotted to $P(s)$ doesn't matter. For a type (c) group of zeros, we allot one half of each multiple zero to $P(s)$ and $P(-s)$.

For example, if

$$A(-s^2) = (s+2)(s-2)(s^2+2s+5)(s^2-2s+5)(s^2+6)^2 \tag{3.5}$$

$P(s)$ could be any of the following.

$$(s+2)(s^2+2s+5)(s^2+6)$$

$$(s-2)(s^2+2s+5)(s^2+6)$$

$$(s+2)(s^2-2s+5)(s^2+6)$$

$$(s-2)(s^2-2s+5)(s^2+6)$$

Obviously, for each of the four choices, the resultant expressions $P(s)P(-s)$ are exactly alike.

The procedure for obtaining a $Q(s)$ for a given $B(\omega^2)$ is exactly the same as that for obtaining a $P(s)$ for a given $A(\omega^2)$. However, since $Q(s)$ must be Hurwitz,[1] we must always choose the left-halfplane zeros from each group. Hence, if (3.5) were a $B(-s^2)$, then the only choice for $Q(s)$ would be $(s+2)(s^2+2s+5)(s^2+6)$.

The various ways in which the zeros of $A(-s^2)$ from each group are allotted to $P(s)$ do not affect $A(\omega^2)$. They do have effects on the phase function of the resultant $H(s)$. When only zeros in the left halfplane are included in $P(s)$, the $H(s)$ will have the lowest phase at each frequency. The network function so formed is called the *minimum-phase* function. If right-halfplane zeros are included in $P(s)$, the network function is a *nonminimum-phase* function.

EXAMPLE 3.1. Obtain the network function whose magnitude squared is

$$|H(j\omega)|^2 = \frac{\omega^4+1}{\omega^2(\omega^4+6\omega^2+25)}$$

[1]A polynomial is Hurwitz if its zeros are all in the open left halfplane. This definition does not allow zeros to lie on the imaginary axis. In denominators of network functions, simple zeros are sometimes allowed to lie on the imaginary axis. In such situations, the polynomials are no longer, strictly speaking, Hurwitz. Polynomials with no right-halfplane zeros and with simple zeros on the imaginary axis are sometimes referred to as *modified Hurwitz polynomials*. In this volume, modified Hurwitz polynomials are simplified referred to as Hurwitz polynomials.

SOLUTION We have

$$H(s)H(-s) = \frac{s^4+1}{-s^2(s^4-6s^2+25)}$$

$$= \frac{(s-\underline{/45^\circ})(s-\underline{/-45^\circ})(s-\underline{/135^\circ})(s-\underline{/-135^\circ})}{-s^2(s+2+j)(s+2-j)(s-2+j)(s-2-j)}$$

Hence we can have either

$$H(s) = \frac{(s+0.707107+j0.707107)(s+0.707107-j0.707107)}{s(s+2-j)(s+2+j)}$$

$$= \frac{s^2+1.41421s+1}{s(s^2+4s+5)}$$

or

$$H(s) = \frac{(s-0.707107+j0.707107)(s-0.707107-j0.707107)}{s(s+2-j)(s+2+j)}$$

$$= \frac{s^2-1.41421s+1}{s(s^2+4s+5)}$$

EXAMPLE 3.2. Obtain the network function whose magnitude squared is

$$|H(j\omega)|^2 = \frac{\omega^2+9}{\omega^6-3\omega^4+12\omega^2+100}$$

SOLUTION We now have

$$H(s)H(-s) = \frac{s^2-9}{s^6+3s^4+12s^2-100}$$

$$= [(s+3)(s-3)]/[(s-1.78026)(s+1.78026)(s+1.12528+j2.08588)$$

$$(s+1.12528-j2.08588)(s-1.12528+j2.08588)(s-1.12528-j2.08588)]$$

Thus

$$H(s) = \frac{s\pm 3}{(s+1.78026)(s+1.12528+j2.08588)(s+1.12528-j2.08588)}$$

$$= \frac{s \pm 3}{(s + 1.78026)(s^2 + 2.25056s + 5.61716)}$$

$$= \frac{s \pm 3}{s^3 + 4.03081s^2 + 9.62374s + 10}$$

The computational steps to obtain the denominator of $H(s)$ can be carried out using MATLAB. The following are the commands and results.

```
% Specify denominator coefficient vector
» a = [1 0 3 0 12 0 -100];
% Obtain denominator zeros
» b = roots(a)
b =
-1.1253 + 2.0859i
-1.1253 - 2.0859i
 1.1253 + 2.0859i
 1.1253 - 2.0859i
 1.7803
-1.7803
% Form the denominator polynomial with the three
% left-halfplane zeros
» c = poly([b(1) b(2) b(6)])
c = 1 4.0308 9.6237 10
```

3.2 Network Functions for Butterworth Filters

For a network function whose magnitude is the Butterworth normalized lowpass characteristic, we have

$$H(s)H(-s) = \frac{1}{1 + (-s^2)^n} \tag{3.6}$$

We only have to deal with $B(-s^2)$. From (3.6), the zeros of $Q(s)Q(-s)$ are the roots of the equation

$$1 + (-s^2)^n = 0 \qquad \text{or} \qquad (-s^2)^n = -1 = e^{j(\pi + 2k\pi)} \tag{3.7}$$

where k is any integer. Equation (3.7) leads to

$$-s^2 = e^{j\frac{(2k+1)}{n}\pi} \qquad \text{or} \qquad s^2 = e^{j\left[\frac{(2k+1)}{n}\pi - \pi\right]} \tag{3.8}$$

Hence the $2n$ zeros of $Q(s)Q(-s)$ are

$$s_k = e^{j\left[\frac{(2k+1)}{2n}\pi - \frac{\pi}{2}\right]} \tag{3.9}$$

Alternately, if we write $s_k = \sigma_k + j\omega_k$, we have

$$\sigma_k = \cos\left[\frac{(2k+1)}{2n}\pi - \frac{\pi}{2}\right] = \sin\left[\frac{(2k+1)}{2n}\pi\right]$$

$$\omega_k = \sin\left[\frac{(2k+1)}{2n}\pi - \frac{\pi}{2}\right] = -\cos\left[\frac{(2k+1)}{2n}\pi\right]$$

Equation (3.9) indicates that the $2n$ zeros of $Q(s)Q(-s)$ are uniformly spaced around a unit circle whose angles are $360°/2n$ apart. If n is odd, two of the zeros will lie on the real axis. If n is even, all the zeros are complex conjugate pairs, and thus there will be none on the real axis. Figure 3.2 shows the zero distribution of $Q(s)Q(-s)$ for $n = 6$ and $n = 9$.

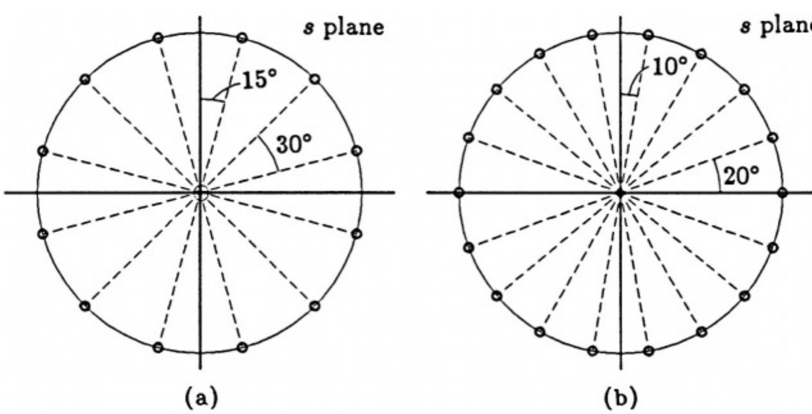


Figure 3.2: Zeros of $Q(s)Q(-s)$ for (a) $n = 6$ and (b) $n = 9$.

Since $Q(s)$ is the denominator of $H(s)$, it must be Hurwitz. Hence we must include only left-halfplane zeros of $B(-s^2)$ to form $Q(s)$. Polynomials formed to have left-halfplane zeros are known as *Butterworth polynomials.* Of course, zeros of $Q(s)$ are poles of $H(s)$.

EXAMPLE 3.3. Obtain the network function of the normalized Butterworth fourth-order lowpass filter.

SOLUTION The poles of $H(s)$ are shown in Fig. 3.3.

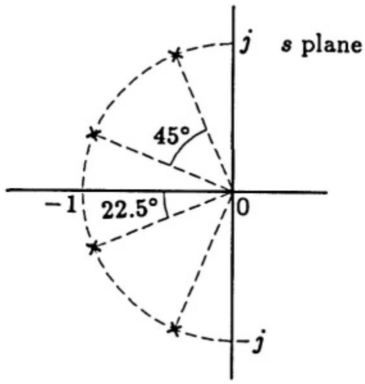


Figure 3.3: Poles of the fourth-order Butterworth lowpass filter.

Hence

$$H(s) = \frac{1}{(s - \angle 112.5^\circ)(s - \angle -112.5^\circ)(s - \angle 157.5^\circ)(s - \angle -157.5^\circ)}$$

$$= \frac{1}{(s + 0.382683 \pm j0.923880)(s + 0.923880 \pm j0.382683)}$$

$$= \frac{1}{(s^2 + 0.765367s + 1)(s^2 + 1.84778s + 1)}$$

$$= \frac{1}{s^4 + 2.61313s^3 + 3.41421s^2 + 2.61313s + 1}$$

This example can also be worked using MATLAB. Following the pattern of steps in Example 2, we have the following steps.

```
» a = [1 zeros(1,7) 1];
» b = roots (a)
b =
-0.9239 + 0.3827i
-0.9239 - 0.3827i
-0.3827 + 0.9239i
-0.3827 - 0.9239i
```

```
  0.3827 + 0.9239i
  0.3827 - 0.9239i
  0.9239 + 0.3827i
  0.9239 - 0.3827i
» c = poly([b(1:4)])
c = 1 2.6131 3.4142 2.6131 1
```

EXAMPLE 3.4. Obtain the network function of the normalized fifth-order Butterworth lowpass filter.

SOLUTION The poles of $H(s)$ are shown in Fig. 3.4. Hence

$$H(s) = \frac{1}{(s+1)(s - \underline{/108^\circ})(s - \underline{/-108^\circ})(s - \underline{/144^\circ})(s - \underline{/-144^\circ})}$$

$$= \frac{1}{(s+1)(s + 0.309017 \pm j0.951057)(s + 0.809017 \pm j0.587785)}$$

$$= \frac{1}{(s+1)(s^2 + 0.618034s + 1)(s^2 + 1.61803s + 1)}$$

$$= \frac{1}{s^5 + 3.23607s^4 + 5.23607s^3 + 5.23607s^2 + 3.23607s + 1}$$

Using MATLAB, we have the following steps.

```
» a = [1 zeros(1,9) -1];
» b = roots(a)
b =
-1.0000
-0.8090 + 0.5878i
-0.8090 - 0.5878i
-0.3090 + 0.9511i
-0.3090 - 0.9511i
  0.3090 + 0.9511i
  0.3090 - 0.9511i
  1.0000
  0.8090 + 0.5878i
  0.8090 - 0.5878i
» c = poly([b(1:5)])
c = 1 3.2361 5.2361 5.2361 3.2361 1
```

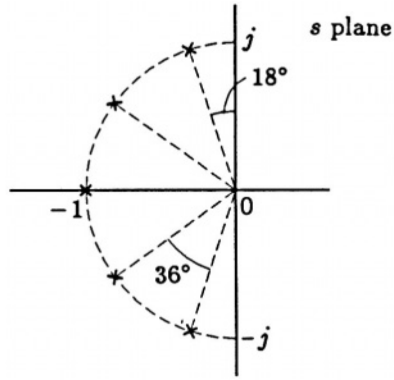


Figure 3.4: Poles of the fifth-order Butterworth lowpass filter.

It might be well to point out that normalized Butterworth polynomials have two interesting features. One of them is that the constant term is always unity. The other is that the coefficients of the polynomial are always symmetric.

A number of Butterworth polynomials are tabulated in Appendix A. Coefficients of these polynomials can also be obtained using the command buttap of MATLAB. This command has the format

```
[z,p,k] = buttap(n)
```

where n is the order of the filter function, `z` is the vector containing the zeros (it will always be empty), `p` is the vector containing the poles, and `k` is the proportionality constant (it will always be unity). Hence the Butterworth polynomials can be formed by using the poles returned by this command. The following are the steps by which the results of Examples 2 and 3 can also be obtained.

```
% Obtain the data for the 4th-order Butterworth function
» [z p k] = buttap(4);
% Obtain the denominator polynomial coefficients from
% the poles
» c = poly(p)
c = 1 2.6131 3.4142 2.6131 1

% Obtain the data for the 5th-order Butterworth function
» [z p k] = buttap(5);
» d = poly(p)
d = 1 3.2361 5.2361 5.2361 3.2361 1
```