

Analog Electrical Filters

Lawrence P. Huelsman
University of Arizona

I. Introduction to Filters
II. Types of Filter Functions
III. Types of Approximation
IV. Passive Filters
V. Active Filters
VI. Filter Design

GLOSSARY

Active RC filters A class of filters which uses elements of gain (typically operational amplifiers), resistors, and capacitors. It differs from passive filters in that it does *not* use inductors.

Approximation A process used to convert specifications for the magnitude of a network function from graphical form to algebraic form.

Complex frequency variable The variable s which is realized by using the Laplace transform.

Ladder filter An electrical filter having the form of alternating series and shunt elements.

Laplace transform A mathematical procedure used to convert expressions for time-domain signals (functions of t) to complex-frequency domain signals (functions of s).

Network function An albegraic expression used to relate an input quantity to an output quantity. It can be a function either of the sinusoidal frequency variable ω (radians/second) or the complex frequency variable s.

Operational amplifier An electronic device for amplifying time-domain voltage signals. It typically has a very high gain (ideally treated as infinite).

PSpice A general-purpose simulation program for PCs used to evaluate the performance of electrical circuits. The original version was named SPICE.

State variables A set of time-domain variables that are related to each other by a series of differentiations.

ANALOG ELECTRICAL FILTERS are devices used to process analog electrical signals of voltage and current. Analog signals are signals that are continuous in time and can take on any of a continuous spectrum of values. Digital signals, on the other hand, only occur at specified instances of time, as determined by a clock, and can only take on one of a discrete set of values.

I. INTRODUCTION TO FILTERS

A. History

Analog electrical filters were introduced under the name *electric wave filters* in 1915 by K. W. Wagner (Germany) and G. A. Campbell (United States). Originally they were

constructed with elements of resistance, capacitance, and inductance. Their major application was in the telephone industry as channel-bank filters in frequency-division multiplex telephone systems. The range of applications of analog filters was greatly extended through the introduction of active circuits to replace inductors starting in the 1950s. This resulted in the Active-RC class of filters, a term which is still used today. In the 1960s, analog filters used operational amplifiers as their active element(s). A major change in filter design occurred in the late 1970s and early 1980s with the development of switched-capacitor filters. The principle was originally reported by Maxwell in the late 1800s and used switches and capacitors to replace resistors in Active-RC filters. The result was a filter which could be fully fabricated in monolithic form on a single silicon chip. Analog filters are becoming increasingly important in application-specific integrated circuits (ASICs), which combine analog and digital signal processing functions in the same integrated circuit. The design of such circuits is usually referred to as mixed signal design.

B. Sinusoidal Signals

Analog electrical filters are used to process two main types of signals. The first of these is the sinusoidal signal. The second is the non-sinusoidal signal which may have almost any other time-domain behavior. In treating sinusoidal functions, the usual practice is to ignore the effect of initial conditions and the presence of transient effects. The result of this is called *sinusoidal steady-state*. The most common way of specifying analog electrical filters is through their sinusoidal steady-state properties. In this situation, we specify an input excitation $e(t)$ having the form:

$$e(t) = E_0 \cos(\omega t + \theta)$$

where ω is the radian frequency of the sinusoid in radians/second (abbreviated rad/s). This is related to f cycles/second (abbreviated Hz for Hertz) by $\omega = 2\pi f$. The result of passing the excitation $e(t)$ through an analog filter is a sinusoidal steady-state response $r(t)$ which has the form:

$$r(t) = R_0 \cos(\omega t + \phi)$$

To emphasize the sinusoidal properties, the excitation and response are now represented by *phasors*. Using $\mathcal{E}$ and $\mathcal{R}$ for the excitation and response phasors, we define:

$$\mathcal{E} = E_0 e^{j\theta} \qquad \mathcal{R} = R_0 e^{j\phi}$$

We now define the properties of the filter as given by a *network function* $N(j\omega)$, where:

$$N(j\omega) = \frac{\mathcal{R}}{\mathcal{E}}$$

One of the most frequently used filtering specifications is the magnitude of $N(j\omega)$. In terms of the phasors, this is

$$\frac{|\mathcal{R}|}{|\mathcal{E}|} = \frac{R_0}{E_0} = |N(j\omega)|$$

The specification for $|N(j\omega)|$ can be given in graphical or algebraic form. If it is given in graphical form, this may be converted to an algebraic form through the use of techniques referred to as *approximation*. The actual determination of the physical analog filter is done starting with the network function $N(s)$ (s is the *complex frequency variable*). This is found algebraically using the equation:

$$|N(j\omega)|^2_{\omega=s/j} = N(s)N(-s)$$

where the quantity $N(-s)$ is redundant. The network function $N(s)$ is the starting point for the actual synthesis of the physical analog electrical filter.

C. Magnitude/Phase Plots

Plots are frequently used to specify the graphical information which gives the properties of an analog filter. The most frequently used one is a plot of $|N(j\omega)|$ versus ω (or f). The ordinate is given in linear units of magnitude or in logarithmic units of dB (decibels), where:

$$dB = 20 \log_{10} |N(j\omega)|$$

Also of interest is the phase plot, a plot of arg $N(j\omega)$ versus frequency. The ordinate units are degrees or radians. Another type of plot is a plot of delay $D(\omega)$ (in seconds) versus frequency. Delay is defined as:

$$D(\omega) = -\frac{d \arg N(j\omega)}{d\omega}$$

II. TYPES OF FILTER FUNCTIONS

A. Low-Pass

Low-pass filter functions are used where it is desired to transmit signals of lower frequencies and block signals of higher frequencies. The desired band of low frequencies (starting with *dc*) is called the *passband*, and the band of higher frequencies is called the *stopband*. The frequency at which the passband and stopband meet is called the *cutoff frequency*. The passband is sometimes also referred to as the *bandwidth* of the filter. An idealized magnitude plot of a low-pass filter is shown as the first plot in Fig. 1. The ordinate is for $|N(j\omega)|$ and the abscissa is for ω. The cutoff frequency is shown as ω_c. Such an idealized characteristic is not physically realizable in actual filter form; it is only used to illustrate the general properties of a low-pass filter function. Network functions for low-pass filters generally have the form:

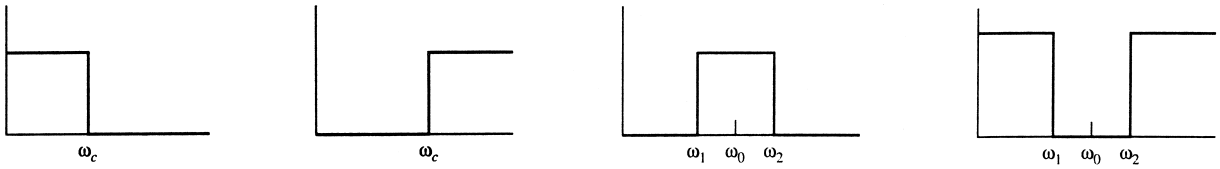


FIGURE 1 Ideal magnitude plots for low-pass, high-pass, band-pass, and band-reject filters.

$$N(s) = \frac{H}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

This is called an *all-pole* function. It is said to have n zeros at ∞. H is an arbitrary multiplicative constant which determines the *gain* of the function. A specific value of H is easily achieved by using a gain-determining amplifier following the portion of the filter that realizes the desired frequency properties.

B. High-Pass

High-pass filter functions are used where it is desired to transmit signals of higher frequencies and block signals of lower frequencies, just the opposite of what was the case for the low-pass filter. The desired band of high frequencies (up to ∞), is called the *passband*, and the band of lower frequencies is called the *stopband*. The frequency at which the passband and stopband meet is called the *cutoff frequency*. An idealized magnitude plot of a high-pass filter is shown as the second plot in Fig. 1. The ordinate is for $|N(j\omega)|$ and the abscissa is for ω. The cutoff frequency is shown as ω_c. Like the low-pass case, such an idealized characteristic is not physically realizable. Network functions for high-pass filters generally have the form

$$N(s) = \frac{Hs^n}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

Such a function is said to have n zeros at the origin. It can be derived from a low-pass function by replacing s with $1/s$. Such an operation is called a *low-pass to high-pass transformation*.

C. Band-Pass

Band-pass filter functions are used where it is desired to transmit signals in a certain band of frequencies and block signals of lower and higher frequencies. This is what is done in "tuning" a desired frequency, such as a radio or television signal. The desired band of frequencies is centered around a frequency ω_0 called the *center frequency*. The bands of frequencies below and above the passband are called the *stopbands*. The frequencies at which the passband and stopbands meet are called the *bandedge frequencies*. An idealized magnitude plot of a band-pass filter is shown as the third plot in Fig. 1. The ordinate is for $|N(j\omega)|$ and the abscissa is for ω. The bandedge frequencies are shown as ω_1 and ω_2. Network functions for band-pass filters generally have the form:

$$N(s) = \frac{Hs^{n/2}}{s^n + an - 1s^{n-1} + \cdots + a_1 s + a_0}$$

Such a function is said to have $n/2$ zeros at the origin and $n/2$ zeros at ∞. It can be derived from a low-pass function by replacing s with $(s^2+1)/s$. Such an operation is called a *low-pass to band-pass transformation*.

D. Band-Reject

Band-reject filter functions are used where it is desired to block signals in a certain band of frequencies and transmit signals of lower and higher frequencies. This is what is done in "blocking" an undesired frequency, such as an unwanted "hum" component in an audio signal. The band of frequencies to be blocked is called the stop band and is centered around a frequency ω_0 called the *null frequency*. The bands of frequencies below and above the stopband are called the *passbands*. The frequencies at which the passbands and stopband meet are called the *bandedge frequencies*. An idealized magnitude plot of a band-reject filter is shown as the fourth plot in Fig. 1. The ordinate is for $|N(j\omega)|$ and the abscissa is for ω. The bandedge frequencies are shown as ω_1 and ω_2. Network functions for band-reject filters generally have the form:

$$N(s) = \frac{H\left(s^2 + \omega_0^2\right)^{n/2}}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

Such a function is said to have $n/2$ zeros at the frequency ω_0 and a constant value at the origin and ∞. It can be derived from a high-pass function by replacing s with $(s^2+1)/s$.

III. TYPES OF APPROXIMATION

A. Butterworth

One of the simplest approximations for the magnitude of an all-pole, low-pass network is the Butterworth, named after British engineer S. Butterworth for his work on electronic amplifiers in 1930. This approximation is obtained

by adjusting the coefficients of the denominator polynomial such that as many derivatives of the magnitude-squared function as possible are set equal to zero at the origin. This criterion may be expressed as:

$$\left.\frac{d^i|N(j\omega)|^2}{d\omega^i}\right|_{\omega=0} = 0 \qquad i = 1, 2, \ldots, n-1$$

The resulting characteristic is referred to as having a maximally flat magnitude. It is monotonic (the derivative does not change sign) and goes to zero at infinite frequency. The magnitude function has the form:

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

where H is an arbitrary multiplicative constant, and where the function has been normalized for a cutoff frequency of 1 rad/s. The poles of the related $N(s)$ are located on the unit circle where they are equi-angularly spaced. Examples of second- and third-order network functions are

$$N_2(s) = \frac{H}{s^2+\sqrt{2}s+1} \qquad N_3(s) = \frac{H}{s^3+2s^2+2s+1}$$

Plots of $|N(j\omega)|$ for various orders are shown in Fig. 2. Note that as the order increases, the locus more closely approaches that of the ideal characteristic.

B. Chebyshev

Another approximation that is frequently used for all-pole, low-pass functions is the Chebyshev. The polynomials used in this approximation were originally used in the study of steam engines by the Russian P. L. Chebyshev in 1899. The approximation is obtained by using an nth order Chebyshev polynomial in the magnitude characteristic. This has the form:

$$|N(j\omega)|^2 = \frac{H^2}{1+\epsilon^2 C_n^2(\omega)}$$

where H is an arbitrary multiplicative constant, $C_n(\omega)$ is an nth-order Chebyshev polynomial, and ϵ is a constant used to adjust the passband attenuation. This approximation produces a magnitude characteristic which has an equal ripple property in the passband between the limits H and $H/\sqrt{1+\epsilon^2}$. Because of this, it is also referred to as an *equal-ripple function*. The passband is normalized for a cutoff frequency of 1 rad/s. In the stopband, the characteristic is monotonic. The poles of the related $N(s)$ are located on an ellipse. Examples of second- and third-order Chebyshev network functions which have a passband ripple of 1-dB are

$$N_2(s) = \frac{H}{s^2+1.097734s+1.102510}$$

$$N_3(s) = \frac{H}{(s^2+0.49417s+0.99420)(s+0.49417)}$$

Plots of typical $|N(j\omega)|$ for various orders are shown in Fig. 3. An advantage of the Chebyshev approximation over the Butterworth is that the order of the filter required to meet a given set of specifications is usually considerably lower. A disadvantage is that the transmission of time-domain signals such as pulses may show higher distortion.

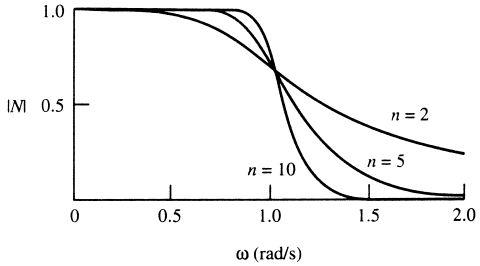


FIGURE 2 Magnitude of Butterworth functions.

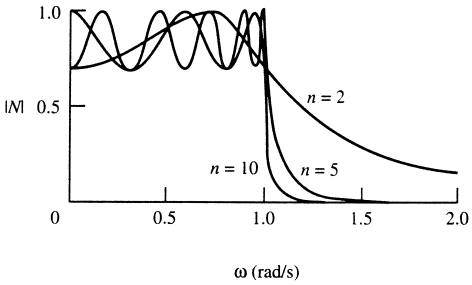


FIGURE 3 Magnitude of Chebyshev functions.

C. Inverse-Chebyshev

Some of the disadvantages of the Chebyshev filter regarding poor time-domain performance can be reduced by using the inverse-Chebyshev approximation. This consists of exchanging the characteristics of the passband and the stopband (i.e., making the passband monotonic and making the stopband equal ripple). The magnitude characteristic now has the form:

$$|N(j\omega)|^2 = \frac{\epsilon^2 C_n^2(1/\omega)}{1+\epsilon^2 C_n^2(1/\omega)}$$

where $C_n(\omega)$ is the Chebyshev polynomial referred to in the preceding section. This approximation produces a magnitude characteristic which has an equal-ripple property in the (normalized) stopband which starts at 1 rad/s. The order of filter required to meet a given set of specifications is the same as is required for the Chebyshev approximation. The filter realization, however, is more complex due to the requirement of having nulls (frequencies of zero transmission) in the stopband. An example of

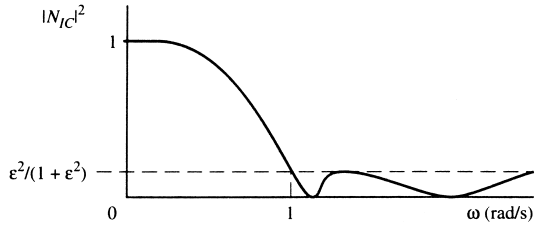


FIGURE 4 Magnitude of an inverse-Chebyshev function.

a third-order, low-pass, inverse-Chebyshev network function that has 30 dB attenuation in the stop band is

$$N(s) = \frac{H(s^2 + 1.33333)}{(s^2 + 0.44086s + 0.23621)(s + 0.53578)}$$

A plot of a typical fourth-order, inverse-Chebyshev function is given in Fig. 4. The constant ϵ is used to adjust the amount of stopband ripple. Although the inverse-Chebyshev filter provides improved time-domain performance over the regular Chebyshev one, it has the disadvantage that a more complex network requiring additional elements is required for its realization.

D. Elliptic

An approximation that is considerably more complex than either of the preceding ones is the elliptic. It is also called a Cauer approximation, named after German engineer Wilhelm Adolf Eduard Cauer for his work on network functions in 1931. Like the inverse-Chebyshev approximation, it has a numerator that is a function of s. Thus, it is not an all-pole function. Its magnitude characteristic has an equal ripple behavior in both passband and the stopband. Meeting a given set of specifications using an elliptic filter can be done with an even lower order filter than would be required using a Chebyshev one. Since the Chebyshev filter requires a lower order than the Butterworth, we see that increasing the mathematical complexity of the approximation leads to lower-order filters. An example of an elliptic approximation for a third-order filter with 1-dB ripple in the normalized passband (0–1 rad/s) and a minimum of 34-dB attenuation in an equal-ripple stopband starting at 2 rad/s is

$$N(s) = \frac{H(s^2 + 5.153209)}{s^3 + 0.974025s^2 + 1.244972s + 0.545678}$$

A plot of a typical sixth-order elliptic function is shown in Fig. 5. Like the inverse-Chebyshev approximation, the elliptic filter requires a more complex network structure for its realization than does either the Butterworth or the Chebyshev.

E. Thomson

A considerably different approximation than the preceding ones is required when time-domain performance is the dominant analog filtering criteria. To provide accurate transmission of time-domain signals requires that ideal phase, rather than the magnitude, be correctly approximated. The required specification is for linearly decreasing phase. From the discussion of Section I, this also implies constant delay. Also required is constant magnitude. Low-pass filters can meet these requirements over a limited frequency range through the use of the Thomson approximation originally proposed by W. E. Thomson in 1949. The network function uses Bessel polynomials. Examples of second- and third-order Thomson network functions are

$$N_2(s) = \frac{H}{s^2 + 3s + 3} \qquad N_3(s) = \frac{H}{s^3 + 6s^2 + 15s + 15}$$

These functions are normalized in the time domain rather than the frequency domain. Ideally they produce a 1-s delay at dc. Typical characteristics for the magnitude, phase, and delay of ideal (solid line) and band-limited (dashed line) Thomson filters are shown in Fig. 6.

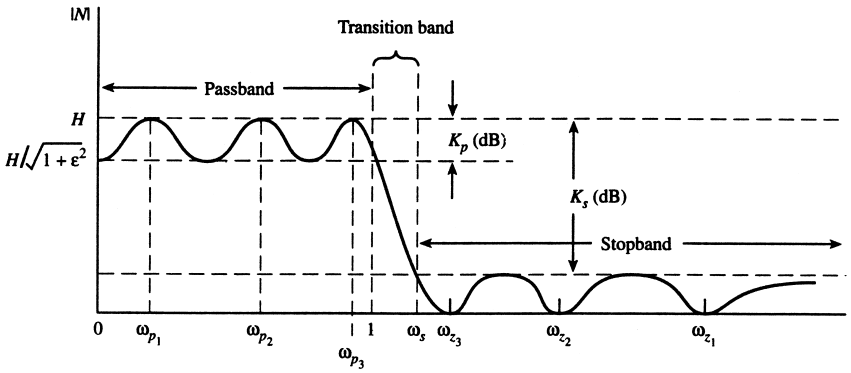


FIGURE 5 Magnitude of an elliptic function.

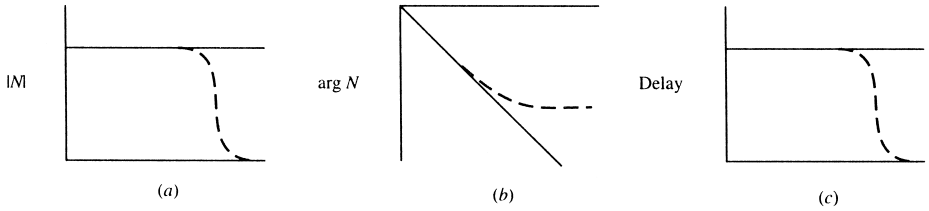


FIGURE 6 Magnitude, phase, and delay for a Thomson function.

F. Determination of Order

A fundamental problem in the use of analog electric filters is determination of the required order of the network function $N(s)$ and correspondingly the order of the filter realization. In $N(s)$, the term "order" refers to the highest degree of the denominator polynomial. In the filter realization, the term "order" refers to the minimum number of inductors and capacitors required. Some realizations, such as those for inverse-Chebyshev and elliptic, may require more than the minimum number. The order for the low-pass case can be determined by specifying four parameters, two in the passband and two in the stopband:

K_p—Maximum deviation in the passband. This typically occurs at the edge of the passband. In Chebyshev functions, it is the parameter that determines the passband ripple.
ω_p—Cutoff frequency that determines the end of the passband.
K_s—Minimum attenuation in the stopband. In elliptic functions, this is the frequency at which the equal-ripple stopband behavior begins.
ω_s—Frequency at which the minimum stopband attenuation begins.

The specifications are further illustrated in Fig. 7. The determination of order is completed by determining two constants Ω and M as:

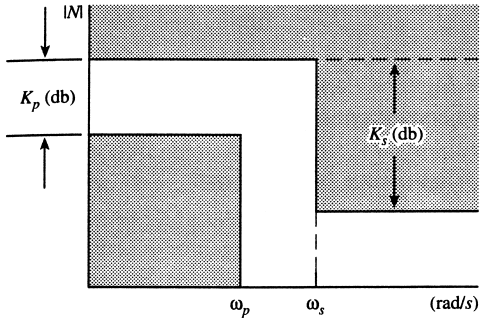


FIGURE 7 Filter specifications for determining order.

$$\Omega = \frac{\omega_s}{\omega_p} \qquad M = \sqrt{\frac{10^{K_s/10} - 1}{10^{K_p/10} - 1}}$$

For the Butterworth and Chebyshev cases, the order may be found as n_B and n_C, respectively:

$$n_B = \frac{\log M}{\log \Omega} \qquad n_C = \frac{\cosh^{-1} M}{\cosh^{-1} \Omega}$$

The procedure for determining the order of an elliptic function is considerably more complex and may be found in the literature. The determination of the order of a Thomson approximation is based on magnitude and delay specifications and is usually done using charts found in the literature.

IV. PASSIVE FILTERS

A. Ladder Filters

The most common form of passive filter topology is the ladder configuration. An example is shown is Fig. 8. Elements such as $Z_1, Z_3, \ldots, Z_n$, are called *shunt* elements. Elements such as $Z_2, Z_4, \ldots, Z_{n-1}$, are called *series* elements. Typically, the first (input) element of a ladder filter is a series resistor, and the last (output) element is a shunt resistor. The elements between the resistors are lossless elements of inductance and capacitance. The resulting configuration is referred to as a lossless, double-resistance, terminated ladder filter. The lossless shunt elements of such a filter produce transmission zeros at the frequencies where their impedance goes to zero. In effect, they act as a short circuit to such frequencies. The lossless series elements produce transmission zeros at the frequencies where their impedance goes to infinity. Thus, they act as

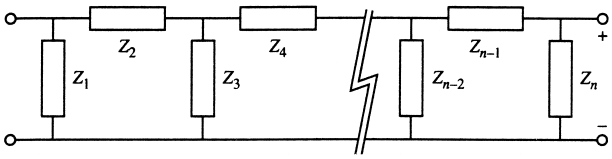


FIGURE 8 Ladder filter.

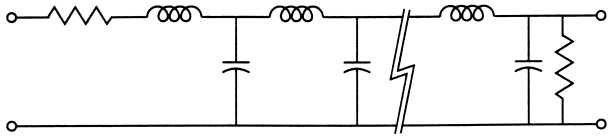


FIGURE 9 Low-pass filter.

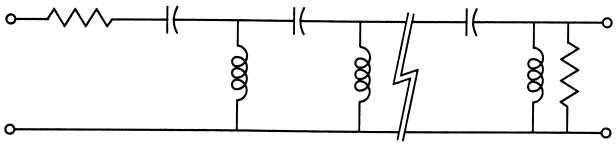


FIGURE 11 High-pass filter.

open circuits to prevent the transmission of such frequencies. These simple properties allow ladder networks to be easily designed to meet the various types of filter functions defined in Section II.

B. Low-Pass Filters

Butterworth and Chebyshev low-pass functions are all-pole functions, thus all their transmission zeros are located at infinite frequency. To realize such transmission zeros in a ladder network requires series inductors (which have infinite impedance at infinite frequency) and shunt capacitors (which have zero impedance at infinite frequency). An example all-pole, low-pass ladder filter is shown in Fig. 9. Inverse-Chebyshev and elliptic low-pass functions have transmission zeros located on the $j\omega$-axis, as well as at infinity. To realize the $j\omega$-axis zeros requires shunt elements consisting of the series connection of an inductor and a capacitor (which has zero impedance at its resonant frequency). Examples of both the even-order case and the odd-order case are shown in Fig. 10. Alternately, $j\omega$-axis zeros can be realized by series elements consisting of the parallel connection of an inductor and a capacitor (which has infinite impedance at its resonant frequency).

C. High-Pass Filters

Butterworth and Chebyshev high-pass functions are functions that have all their transmission zeros at the origin (dc). To realize such transmission zeros in a ladder network requires series capacitors (which have infinite impedance at zero frequency) and shunt inductors (which have zero impedance at zero frequency). An example high-pass, all-zeros-at-the-origin ladder filter is shown in Fig. 11. Inverse-Chebyshev and elliptic high-pass functions have transmission zeros located on the $j\omega$-axis, as well as at zero. To realize the $j\omega$-axis zeros requires shunt elements consisting of the series connection of an inductor and a capacitor, or series elements consisting of the parallel connection of an inductor and a capacitor just as was done for the low-pass case.

D. Band-Pass and Band-Reject Filters

Band-pass filters can easily be derived from the low-pass case by substituting series-LC circuits for the inductors and parallel-LC circuits for the capacitors. Thus, for the all-pole, low-pass filter (Butterworth or Chebyshev approximation) shown in Fig. 9, the corresponding band-pass filter would have series elements consisting of an inductor and capacitor (in series) and shunt elements consisting of an inductor and capacitor (in parallel). The same procedure may be applied to low-pass inverse-Chebyshev and elliptic approximations, although the resulting configuration is quite complex. Band-reject filters can easily be derived from the high-pass case by also substituting series-LC circuits for the inductors and parallel-LC circuits for the capacitors. Thus, for the high-pass, all-zeros-at-the-origin filter (Butterworth or Chebyshev approximation) shown in Fig. 11, the corresponding band-reject filter would have shunt elements consisting of an inductor and capacitor (in series) and series elements consisting of an inductor and capacitor (in parallel). The same procedure may be applied to high-pass inverse-Chebyshev and elliptic approximations. As was true in the band-pass case, these configurations are quite complex.

V. ACTIVE FILTERS

A. Single-Amplifier Filters

Filters comprised of a single active element coupled with passive elements of resistance and capacitance, the

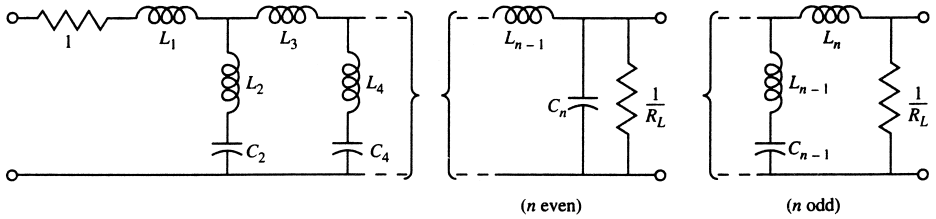


FIGURE 10 Filter which realizes $j\omega$-axis zeros.

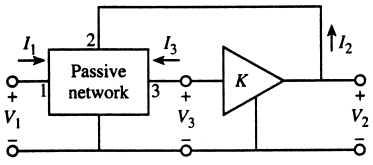


FIGURE 12 Active-RC filter configuration.

so-called Active-RC class of filters, were some of the first to be used in practical applications. The general structure consists of a passive resistor/capacitor (RC) network with a gain element used for output and for feedback. The general configuration is shown in Fig. 12. The triangle symbol with letter K represents an ideal voltage amplifier with infinite input impedance, zero output impedance, and voltage gain K. The voltage amplifier is realized by using an operational amplifier and a pair of feedback resistors. This configuration is frequently referred to as the Sallen and Key filter, and it is one of the earliest and most well-known active filter configurations. By proper choice of the R and C elements of the passive network, all of the usual filter functions can be realized. An example is the low-pass realization shown in Fig. 13. As an illustration of the use of this configuration, we may choose an equal-R, equal-C normalized design for which $R_1 = R_3 = 1\Omega$ and $C_2 = C_4 = 1$F. For these choices, the voltage transfer network function is

$$N(s) = \frac{V_2(s)}{V_1(s)} = \frac{K}{s^2 + (3 - K)s + 1}$$

For example, setting $K = 1.586$ makes the denominator polynomial $s^2 + \sqrt{2}s + 1$, which realizes a second-order Butterworth function. This configuration is primarily used for second-order network functions. If higher-order functions are required, the second-order functions may be connected one after the other. Such a connection is called a *cascade* filter configuration. This type of filter uses positive feedback to realize the network function and has the disadvantage of having high sensitivities to element tolerances and to changes in element values.

Another single-amplifier filter has a general configuration similar to the one shown in Fig. 12 but uses an inverting operational amplifier as the gain element. It is referred to as an *infinite-gain, single-amplifier* filter. An example of a low-pass realization is shown in Fig. 14. The determination of the values of the R and C elements is more complex than is the case for the Sallen and Key filter; however, due to the negative feedback employed, the filter has considerably lower sensitivities than the Sallen and Key one. The basic filter configuration can be modified to include both positive and negative feedback which makes it possible to realize inverse-Chebyshev and elliptic filter functions (for which the numerator of the network function is a polynomial). Like the Sallen and Key filter, network functions of higher than second order are usually realized using cascades of second-order functions. For high-frequency applications, both types of single-amplifier filters must be compensated to take account of the limiting effects of the gain band width of the operational amplifier.

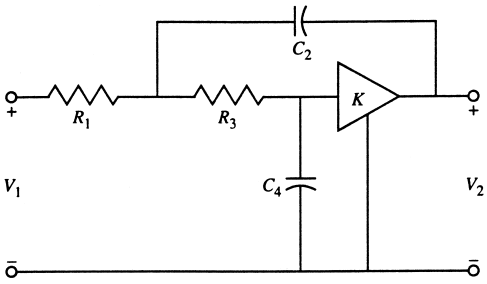


FIGURE 13 Low-pass Active-RC filter.

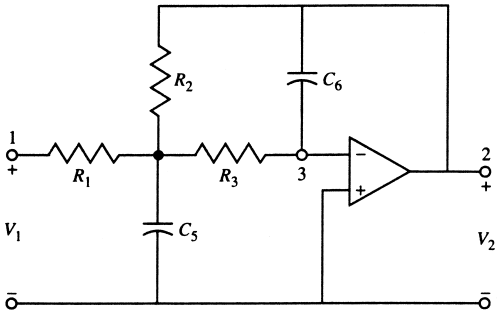


FIGURE 14 Infinite-gain, single-amplifier, low-pass, Active-RC filter.

B. Multiple-Amplifier Filters

Many of the disadvantages of the single-amplifier filter such as high sensitivity (Sallen and Key), cumbersome design (infinite-gain, single-amplifier) can be eliminated by using a multiple-amplifier design, typically one containing three operational amplifiers. The most well-known configuration is called variously the *state-variable* filter, the *KHN* (Kerwin-Huelsman-Newcomb) filter, or the *universal active* filter. It simultaneously makes available a low-pass, band-pass, and high-pass output. A fourth operational amplifier is sometimes used as a summer to combine these outputs. For example, summing the high-pass (numerator s^2 in the network function) and low-pass (numerator H in the network function) outputs gives a numerator $(s^2 + H)$ which realizes an inverse-Chebyshev or elliptic function. An example of the state-variable filter configuration is shown in Fig. 15. The high-pass, band-pass, and low-pass outputs are identified in the figure as V_{HP}, V_{BP}, and V_{LP}, respectively. To realize a normalized second-order Butterworth low-pass filter, the element values would all be chosen as unity with the exception of

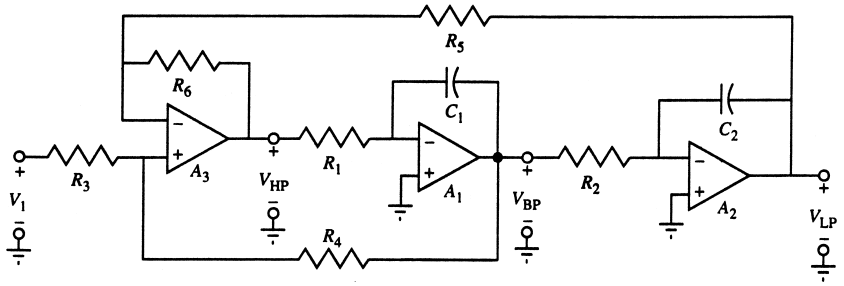


FIGURE 15 State-variable filter configuration.

$R_4 = 0.414\Omega$. The numerator constant is $H = 0.586$. Like the single-amplifier filter, the multiple-amplifier filter realizes second-order network functions. Higher-order filter requirements are produced by cascading second-order sections. The state-variable filter is available commercially from several manufacturers. An example is the UAF-41 (Burr-Brown). Among the disadvantages of the multiple-amplifier filter is its limited use in high frequency applications due to the high-frequency limitations (gain-bandwidth) of the operational amplifiers.

Another multiple-amplifier filter that has the advantage of a simplified design procedure is the Tow-Thomas filter. It provides a low-pass and a band-pass output. The basic configuration is shown in Fig. 16. The low-pass voltage transfer function realized is

$$\frac{V_{LP}(s)}{V_1(s)} = \frac{1/R_2R_4C_1C_2}{s^2 + s/R_1C_1 + 1/R_2R_3C_1C_2}$$

$$= \frac{H}{s_2 + (\omega_n/Q)s + \omega_n^2}$$

In the right member of this equation, the quantity ω_n is called the *bandwidth*. It defines the cutoff frequency for the filter. The quantity Q is called the *quality factor*. It determines the amount of peaking that occurs in the magnitude characteristic for the passband. If we compare the second and third members of the above relation we see that this filter configuration may easily be tuned by varying single resistors which can be implemented as potentiometers. To do this, we vary R_4 to change the value of the multiplicative constant H, R_1 to vary the value of Q, and R_3 to vary the value of ω_n. Thus, the characteristics of the low-pass function are easily tuned by three non-interacting controls.

C. OTA Filters

A quite different type of multiple-amplifier filter is one using *operational transconductance amplifiers* (OTAs). The conventional operational amplifier is ideally characterized as having infinite input resistance, zero output resistance, and a gain that approaches infinity. The realizations described in the preceding paragraph use large amounts of feedback to effectively make the circuit realization independent of the operational amplifier gain. The operational transconductance amplifier produces an output voltage proportional to an input current. In this sense, it directly parallels the basic operation of the solid-state devices used in its realization. The transconductance gain g_m can be varied over several decades by adjusting the bias to the circuit. The result is an active device whose transconductance can be used to replace the resistors in an Active-RC circuit and whose high-frequency performance is considerably higher than that of multiple-amplifier circuits using conventional operational amplifiers. An example of a second-order, low-pass filter using OTAs is shown in Fig. 17. The OTAs are represented by the triangle symbol with overlapping circles at the output terminals. A major disadvantage of the use of OTAs is the restricted differential

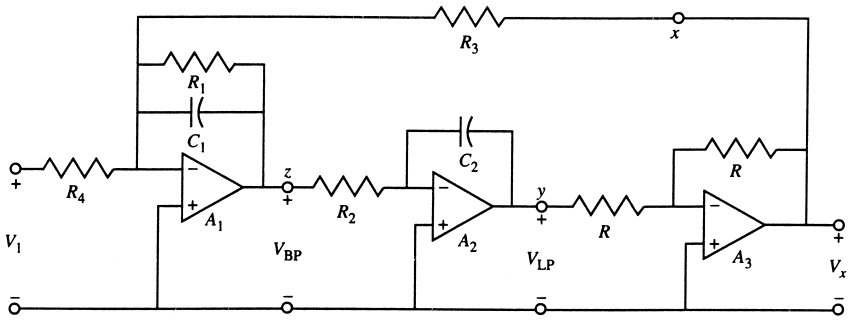


FIGURE 16 Tow-Thomas filter configuration.

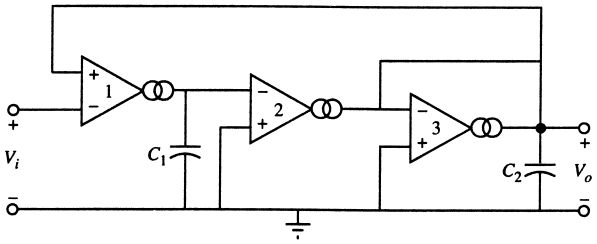


FIGURE 17 Operational transconductance amplifier (OTA) low-pass filter.

input swing required to maintain linearity. Many commercial OTAs are available, an example is the CA 3080 (Harris).

D. Direct Realizations

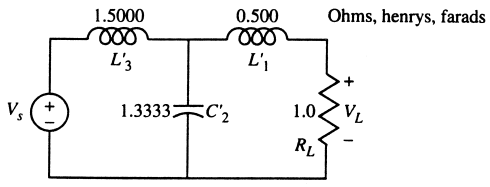
In addition to the cascade realizations of second-order filters described in the preceding sections, higher-order active filters can also be realized in a general form called a *direct* realization. Such a realization uses a passive, lossless-ladder filter as a prototype and replaces the passive reactive elements of inductance and capacitance by single-time-constant analog active circuits. Effectively, elements with impedance $Z_L(s) = sL$ and admittance $Y_C(s) = sC$ are replaced by integrators with voltage transfer functions $V_2/V_1 = N(s) = 1/sRC$, where R and C are the elements of the integrator. Such a filter configuration is frequently called a *leapfrog* filter because of the multiple negative feedback paths for signal flow. In effect, the leapfrog filter replaces the voltage and current variables of the passive filter configuration with a new set of voltage (only) variables in a manner similar to that used in old-time analog computers. Since the replacement of variables retains the low sensitivity properties of the original passive filter, Active-RC filters based on the leapfrog design have excellent sensitivity properties. An example of a third-order leapfrog filter is shown in Fig. 18. The single-resistance, terminated-LC ladder filter which realizes a third-order Butterworth approximation is shown in part (a). The resulting leapfrog configuration is shown in part (b).

E. Switched-Capacitor Filters

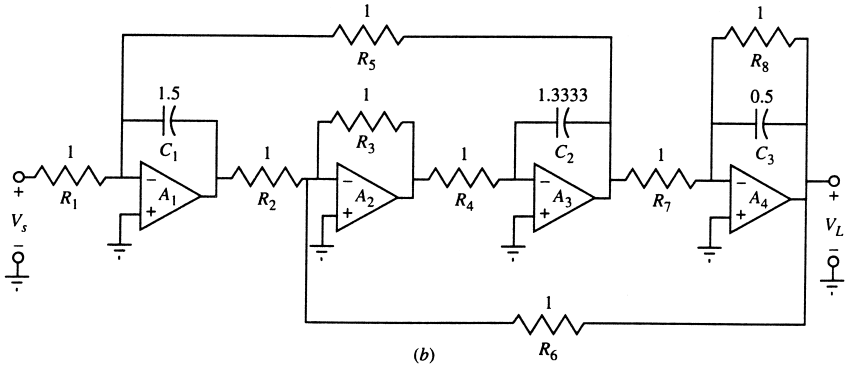
A type of filter that processes analog signals but is not continuous in its operation is the *switched-capacitor* (SC) *filter*. This class of filter is also referred to as a *sampled-data filter*. The basic concept of its operation is shown in Fig. 19, where a capacitor C_R is repeatedly switched between input and output terminals 1 and 2. If the switching is done at a clock frequency considerably higher (factors of 100 are frequently used) than the highest frequency component of the signal, the switched-capacitor element performs the circuit functions of a resistive element R_{eq}. Its value is

$$R_{eq} = \frac{1}{C_R f_c}$$

where f_c is the clock frequency. The implementation of a switched-capacitor is usually done in a parasitic



(a)



(b)

FIGURE 18 Leapfrog filter.

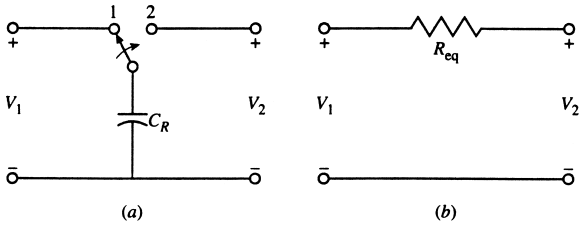


FIGURE 19 Switched-capacitor resistor equivalent.

insensitive manner using four metal-oxide silicon (MOS) switches driven by two clocks ϕ_1 and ϕ_2 which are phased at 180° and are non-overlapping. The circuit configuration is shown in Fig. 20. Part (a) shows the idealized configuration, and part (b) shows the actual realization. The parasitic capacitances eliminated by this configuration are shown as C_{P1} and C_{P2}.

The use of switched capacitors has major advantages in the fabrication of intergrated circuits. Typically, resistors are a difficult element to include in very large-scale integration (VLSI) design, requiring large amounts of chip area. A switched-capacitor replacement consisting of four switches and a capacitor, however, typically requires orders of magnitude less chip area. For example, a 1-pF capacitor, switched with $f_c = 100$ kHz realizes a 10-MΩ resistor in a chip area of approximately 0.01 mm^2. A realization by other techniques such as a polysilicon line or by diffusion would require about 100 times that chip area. A second major reason for the use of switched capacitors in filter realizations is that the frequency properties of the filter, usually determined by time constants of the form $R \times C$, are now determined by time constants $C/(C_R \times f_c)$. Thus, the time constant is determined by a *ratio* of capacitance values. Since typical integrated circuit fabrication techniques do not produce absolute element values reliably, but do produce ratios of values very precisely, highly accurate filter realizations can be produced using switched-capacitor techniques. A final advantage of switched-capacitor elements is that their value may be easily changed by simply changing the clock frequency. Thus, filter parameters such as center frequency, selectivity (Q), etc. can readily be changed using electrical signals.

An example of a fifth-order switched-capacitor filter that realizes a low-pass Butterworth magnitude characteristic is shown in Fig. 21. This is based on a leapfrog design. There are many commercially available switched-capacitor filters. Among the most well-known, general-purpose ones is the MF-10 (National Semiconductor), which provides two dual-integrator loop sections, each of which is capable of independently realizing a state-variable filter. Another example of a filter that realizes a specific configuration is the S3528 (Gould) low-pass, seven-pole elliptic filter realizations.

F. Switched-Current Filters

Although the use of switched-capacitor techniques has made the VLSI realization of active filters possible, it is not considered as an ideal solution. The difficulty arises from the fact that special processing steps are required to accommodate the floating capacitors used in switched-capacitor configurations. Since the analog filtering section typically consumes no more than a fifth of the total chip area in a mixed-signal VLSI design, the extra processing becomes difficult to justify. Other difficulties with switched-capacitor implementations are the result of modern trends to higher density of components in integrated circuits. The resulting limitations on the physical and electrical parameters result in poorer performance of the MOSFETs used as switches and thus lower the performance of the filters. More specifically, higher component densities produce higher power dissipation and higher electric fields. Both of these mandate the use of lower voltages which in turn results in degradation of MOSFET performance and smaller dynamic range for the analog signals.

The disadvantages described above are, in large measure, overcome by the use of switched-current (SI) filters. Note that SC filters are typically realized by using two integrator circuits, an inverting one and a non-inverting one. Such integrators also include the capability for summing the voltage signals for which they are used, but the summation operation requires the use of individual resistors. One of the main differences of the SI technique is that integrators are developed based on current signals rather than voltage ones. As a result, summation is readily accomplished by connecting the current signals to a common low-impedance integrator input.

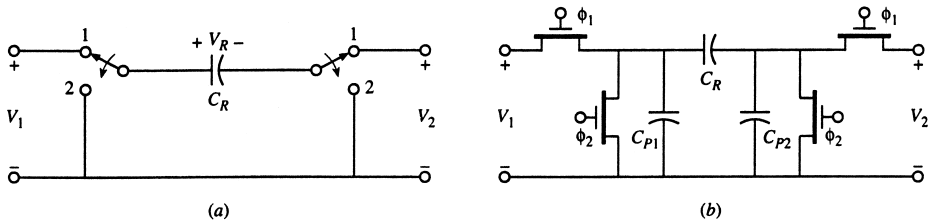


FIGURE 20 Parasitic-insensitive switched capacitor.

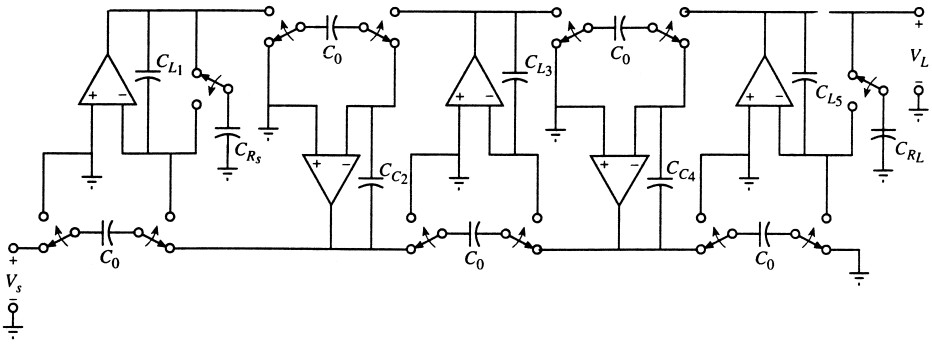


FIGURE 21 Switched-capacitor filter.

VI. FILTER DESIGN

Determination of the element values of double-terminated lossless ladder filters which meet specific approximation and order criteria is done through the use of tables or by using computer programs. Tables may be found in the texts listed in the Bibliography. Computer programs have been generated by various researchers. Most of these programs cover the realization of active filters as well as passive ones. The most well known of these programs are the following:

Ladder—This program performs approximation for low-pass, high-pass, band-pass, and band-reject filters using Butterworth, Chebyshev, elliptic, or Thomson techniques. It then produces passive, double-resistance, terminated ladder realizations for the specified function. It operates in the DOS environment.

Filter—This program performs approximation for low-pass, high-pass, band-pass, and band-reject filters using Butterworth, Chebyshev, elliptic, or Thomson techniques. The approximation operations are identical with those in the Ladder program described above. The program then produces active realizations for the specified function. Several different active filter configurations are available. It operates in the DOS environment.

FilterBB—This is a package of two programs. The first provides low-pass filter realizations of Sallen and Key and single-amplifier, infinite-gain configurations. The Sallen and Key stages can be designed either with unity gain or a specified finite gain value. The second program provides low-pass, high-pass, band-pass, and band-reject filter realizations using a state-variable configuration. Both programs operate in the DOS environment.

Winfilters—This program provides a general-purpose analog and digital filter design program. The analog design section includes both passive and active filter realizations. The filters can be low-pass, high-pass, band-pass, or band-reject. Approximation types include Butterworth, Chebyshev, Inverse-Chebyshev, elliptic, and Thomson. Graphics include pole-zero plots, magnitude, phase, and delay characteristics. The program generates SPICE netlists that can be used to verify circuit performance. This is a Windows-based program.

SEDIF—This program performs approximation for low-pass, high-pass, band-pass, and band-reject filters using Butterworth, Chebyshev, inverse-Chebyshev, or elliptic techniques. The designs are realized in active and passive forms. Plots of the magnitude, phase, and delay are provided. The program also plots the impulse and step response of the selected design. The program operates in the DOS environment.

Filter Designer—This program performs approximation for low-pass, high-pass, band-pass, and band-reject filters using Butterworth, Chebyshev, inverse-Chebyshev, or elliptic techniques. It was originally issued both as a demonstration version and as a commercial product. The designs are realized in active and passive forms; however, in the demonstration version, the circuit complexity is limited to third order. The program includes a section on switched capacitor design. The program operates in the DOS domain. The commercial program is no longer supported, but the demonstration version is available as freeware.

SEE ALSO THE FOLLOWING ARTICLES

ANALOG SIGNAL ELECTRONIC CIRCUITS • CIRCUIT THEORY • DIGITAL FILTERS • DIGITAL ELECTRONIC CIRCUITS

BIBLIOGRAPHY

Allen, P. E., and Sánchez-Sinencio, E. (1984). "Switched-Capacitor Circuits," Van Nostrand-Reinhold, New York.

Chen, Wai-Kai, ed. (1995). "The Circuits and Filters Handbook," CRC Press, Boca Raton, FL.

Huelsman, L. P. (1993). "Active and Passive Analog Filter Design," McGraw-Hill, New York.

Su, K. L. (1996). "Analog Filters," Chapman & Hall, London.



Analog-Signal Electronic Circuits

Donald T. Comer
David J. Comer
Brigham Young University

I. Introduction to Analog- and Mixed-Signal Electronic Circuits
II. The Analog Amplifier
III. Digital-to-Analog and Analog-to-Digital Converters
IV. Mixed-Signal System Applications

GLOSSARY

Analog-to-digital conversion The conversion of an input analog voltage or current to a digital output code that represents the value of the input analog signal at the time of conversion.

Analog signal A voltage or current that is continuous in time and carries information that is related to the signal.

Binary counter An electronic circuit that registers the occurrence of pulses on an input line and generates an output binary code representing the number of pulses that have occurred since a particular starting time.

Comparator circuit An electronic circuit that compares two input signals and generates one of two possible output levels, depending on which input signal has the largest magnitude.

Digital-to-analog conversion The conversion of an input signal, coded as an N-bit digital code, to an output analog voltage or current that has a value determined by the numeric value of the input code.

Op amp (operational amplifier) An electronic amplifier that has a very high voltage gain and allows the construction of circuits that perform mathematical operations such as summing or integration of input analog signals.

Sample-and-hold circuit An electronic circuit that samples a time-varying input analog signal under control of a digital sample gate and generates a constant output signal equal to the input signal at the time the gate signal is asserted. When the gate signal is not asserted, the output voltage holds constant at its previous value.

VLSI circuits An abbreviation for very large scale integrated circuits. These circuits are fabricated on or within a silicon substrate and may include 10,000 to several million transistor devices.

ELECTRONIC CIRCUITS may be classified as either analog or digital circuits on the basis of their intended application. For many years, the design of analog circuits was treated as a separate area from that of digital circuit design. Today, it is becoming common practice to design circuits that include both analog and digital features merged into a single product. This trend has been accelerated

by the availability of integrated circuit processes that allow the fabrication of both kinds of circuits on a common substrate. Circuits that combine both analog and digital features have come to be known as mixed-signal circuits.

I. INTRODUCTION TO ANALOG- AND MIXED-SIGNAL ELECTRONIC CIRCUITS

Analog electronic circuits are designed to perform processing functions on electronic signals. Analog signals are typically low-level, nonquantized, continuously variable signals that are produced by analog transmitters or transducers. For example, a radio station may transmit through its antenna a signal which consists of a sinusoidal signal that varies in amplitude in accordance with an imposed modulating signal. Several stations in a geographic area transmit simultaneously, each using a slightly different range of frequencies. The function of a radio receiver is to select the range of frequencies that includes the desired signal, amplify signals in this range, remove any unwanted components, and drive an audio speaker. The peak amplitude of the received signal may be about 50 μV at the output of the receiving antenna. This signal must be amplified to a voltage level sufficient to drive a speaker. The functions of selecting the desired frequency range (called filtering) and amplifying the signal are processes carried out by analog electronic circuits.

The electronic circuits of the early 1900s were implemented by using vacuum tubes, and their use was almost entirely centered around radio and communication applications. In the 1930s, amplifiers were also used in the construction of the transatlantic cable that was developed to provide telephone service between the United States and Europe. In the 1940s, analog electronic circuits were used in radar, sonar, and some weapon-control applications as well as in an expanded communication role.

With the invention of the bipolar junction transistor in 1951 and its rapid development in succeeding years, the vacuum tube began to be replaced by transistors in amplifier applications. By the 1960s, engineers had replaced vacuum tubes with transistors in most commercial analog electronic applications.

In 1964, Fairchild Semiconductor Corporation introduced the industry's first analog amplifier that was constructed by using integrated circuit technology. In this technology, all the resistors, capacitors, and transistors required to produce the amplifier were fabricated simultaneously on a common silicon substrate. This technology, originally known as integrated circuit technology, eventually evolved into VLSI (very large scale integration), in which thousands of transistors can be fabricated on a silicon substrate. In the 1980s, it became feasible to merge analog and digital circuits and to fabricate both on a common substrate. This became known as mixed-signal VLSI processing and is utilized today by a high percentage of commercial integrated circuit companies.

The most widely used substrate material is silicon. Early integrated processes concentrated on the fabrication of bipolar junction transistor (BJT) circuits. In recent years, processes involving the metal-oxide-semiconductor field-effect transistor (MOSFET) have become very popular.

The design of digital circuits on silicon can typically take advantage of computer automation and expert computer-automated design systems. This results from the fact that digital designs are generally very structured and involve the use of a limited number of common building blocks. However, the design of the analog portions of a mixed-signal circuit often uses unique cells that do not readily lend themselves to design automation and may produce unpredictable and often harmful interactions with the digital circuitry. Substrate noise, instability, and temperature drift are factors that often cause undesirable interactions between analog and digital sections of a mixed-signal integrated circuit. Therefore, special techniques are required in the design of these circuits.

II. THE ANALOG AMPLIFIER

A. Amplifier Specifications

The electronic amplifier is considered to be the fundamental building block of analog circuit design. To construct an amplifier, one must utilize a device that is capable of power gain; that is, this device must be capable of being activated by a low-power input signal to produce a higher-power output signal. The increased power is provided by a dc power supply.

The primary function of an amplifier is to generate an output signal that is a replica of the input, but at a higher signal strength. Amplifiers may be designed specifically for voltage gain, power gain, or current gain. The most common use of an amplifier is as a voltage-gain device. Figure 1 shows a typical amplifier diagram where an input signal, V_i, is applied to its input terminal and an output voltage, V_o, is generated at its output. The voltage gain, G_v, of the amplifier may be expressed as

$$G_v = \frac{V_o}{V_i}. \tag{1}$$

For example, an amplifier with a voltage gain of 100 would convert a 50-μV input signal to a 5-mV signal.

In many applications the dc level of the input or output is not important. Rather, it is the variational or alternating portion of the signal that contains the information to be

processed. Amplifiers intended only to transmit the variational portions of an input signal are referred to as ac coupled amplifiers while amplifiers that are intended to amplify and transmit dc as well as variational levels are referred to as direct coupled amplifiers.

In the typical amplifier circuit of Fig. 1, the resistance, R_L, represents the amplifier load. The power, P_o, delivered to the load can be expressed as

$$P_o = \frac{V_o^2(rms)}{R_L}, \tag{2}$$

where $V_o(rms)$ is the root-mean-square value of the output voltage. The power delivered to the input of the amplifier is

$$P_i = \frac{V_i^2(rms)}{R_i}, \tag{3}$$

where $V_i(rms)$ is the root-mean-square value of the input voltage and R_i is the input resistance of the amplifier.

The power gain of the amplifier of Fig. 1 may be expressed as

$$G_P = \frac{P_o}{P_i} = \frac{G_v^2 R_i}{R_L} \tag{4}$$

Other important parameters of an amplifier are its equivalent input noise and its bandwidth. The equivalent input noise is determined by measuring the noise voltage produced at the output, $V_o(noise)$, and dividing it by the voltage gain to give

$$V_i(noise) = \frac{V_o(noise)}{G_v}. \tag{5}$$

The equivalent input noise represents a lower limit on the input-signal voltage that can be amplified. If the input signal is smaller than the noise voltage, the signal will be obscured in noise, even after amplification.

The bandwidth of a wideband amplifier specifies the frequency range over which maximum amplification occurs. This maximum, or midband, voltage gain is normally measured at a single moderate frequency where the amplifier behaves close to ideal. The frequency of the input signal is then increased to a point where the amplifier gain

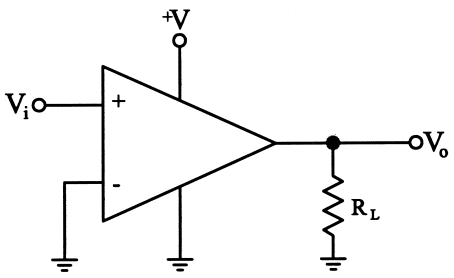


FIGURE 1 Typical voltage amplifier.

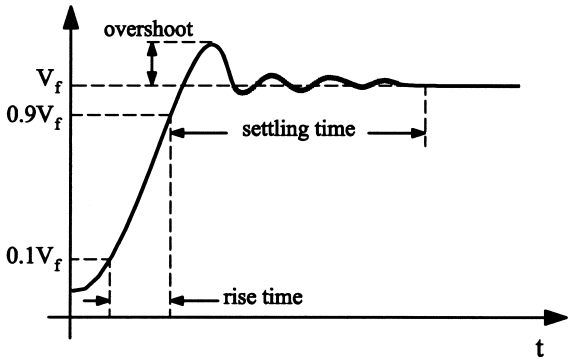


FIGURE 2 Typical amplifier step response.

decreases to 0.707 of its midband value. The frequency where this occurs is approximately equal to the bandwidth of the amplifier.

Some amplifiers are designed to have a high gain over only a very narrow band of frequencies. These stages are called narrowband, or tuned, amplifiers and are indispensable in communication applications.

The efficiency of an amplifier may be expressed as the ratio of the power delivered to the load to that drawn from the power supply. For the amplifier of Fig. 1, the efficiency, η, may be expressed as

$$\eta = \frac{V_o^2(rms)/R_L}{VI_{SS}}, \tag{6}$$

where V is the power-supply voltage and I_{SS} is the average current drawn from the power supply.

Other amplifier parameters that are important to some applications are the rise time, the settling time, and overshoot. Figure 2 shows an amplifier output waveform in which rise time, overshoot, and settling time are defined. In this case, the input to the amplifier is assumed to be a step function.

B. Types of Amplifiers

Several important types of amplifiers are the narrowband amplifier, the wideband amplifier, the power amplifier, and the operational amplifier, or op amp.

1. The Narrowband, or Tuned, Amplifier

The narrowband circuit is required in communication applications wherein signals contained in a small range of frequencies are to be amplified. Several radio and television stations may simultaneously transmit signals to a geographic area. Each signal has a different channel or carrier frequency and transmits information over a small range of frequencies centered about this carrier frequency. The receiving circuitry includes amplifiers that can be tuned to

the carrier frequency and amplify only the small range of frequencies near the carrier frequency. For example, the first amplifier stage in an FM radio receiver might have maximum amplification at a frequency of 100.3 MHz. The range of frequency over which near-maximum amplification occurs may extend from 100.2 to 100.4 MHz.

2. The Wideband Amplifier

The wideband stage amplifies signals falling within a certain broad range of frequencies. The highest frequency at which the circuit provides maximum amplification is several orders of magnitude above the lowest frequency of maximum amplification. Amplification is constant between these two frequencies. An example of a wideband amplifier is the video amplifier in television receivers, which amplifies signals from around 30 Hz to 4 MHz. Below 30 Hz and above 4 MHz, the amplification decreases. The ratio of upper-corner frequency to lower-corner frequency is 133,333 in this case. In a narrowband circuit, this ratio is near unity.

3. The Power Amplifier

Many amplifying stages are designed to have high values of voltage gain or current gain, but they produce an output signal with very small power. The first few stages of radio or television receivers generally produce output power levels in the milliwatt range.

Some applications require a high-power signal. Common examples of this are the signals that drive the audio speakers in a public address system, a high-fidelity radio receiver, or a compact disc (CD) player. Smaller audio speakers may require tens of watts while larger ones are driven by amplifier stages that produce hundreds of watts. These stages are called power amplifiers.

4. The Operational Amplifier, or Op Amp

The op amp was so named as a result of the mathematical operations this device can perform on electronic signals. The op amp, along with a small number of resistors, can generate an output signal or voltage that is proportional to the sum of several input signals. With a slightly different configuration of resistors, the op amp can generate an output signal that is proportional to the difference of two input signals. The addition of a capacitor can create a circuit having an output signal that is proportional to the integral of the input signal.

Several op amps, resistors, and capacitors can be interconnected or configured in a manner that allows differential equations to be simulated. The excitation function or independent variable is applied to the system of amplifiers and generates an output voltage that represents the dependent variable in the differential equation. A system allowing such simulations is called an analog computer.

The op amp is also a building block in circuits that convert between an analog signal and a corresponding digital code. A circuit that accepts an analog input signal and converts this signal to a corresponding N-bit output digital code is called an analog-to-digital converter, or ADC. A circuit that accepts an N-bit digital code and converts it to a corresponding analog output signal is called a digital-to-analog converter, or DAC.

Prior to the 1960s, op amps were very expensive precision circuits costing hundreds of dollars. In 1964, the first op amp was fabricated on an integrated circuit chip. Since that time, the integrated circuit op amp has evolved into a high-performance amplifier costing only a few cents.

Although the frequency performance of the op amp is somewhat limited, it is a very important component in electronic systems. It is used in many analog circuits as well as in circuits that combine analog and digital signals such as the ADC and DAC.

C. Op Amp Behavior

Figure 3 shows a common symbol for the op amp. This amplifier is designed to amplify the difference between the two input-signal voltages, V_1 and V_2. The output voltage for the circuit of Fig. 3 is given by

$$V_o = A(V_1 - V_2), \tag{7}$$

where A is the voltage gain of the op amp. Since the circuit amplifies the difference between the two input signals, it is referred to as a differential amplifier. Typical low-frequency voltage gains for a general-purpose op amp are 200,000–300,000 V/V.

If the voltage gain, A, is very large and the input resistance is also large, the circuit of Fig. 3 approaches an ideal op amp. When negative feedback is applied around an ideal op amp, a voltage node designated a virtual ground can be produced. Figure 4 shows one useful application of an op amp and illustrates the idea of a virtual ground.

In the circuit of Fig. 4, a signal that is proportional to the output voltage is returned, or fed back, to the inverting input of the op amp through the resistance R_2. This is

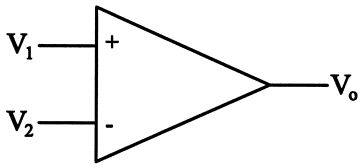


FIGURE 3 The operational amplifier (op amp).

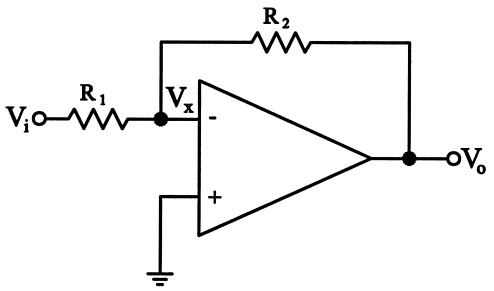


FIGURE 4 Inverting op amp.

called negative feedback and has the effect of reducing the voltage gain of the circuit. It also causes the node x to appear as a virtual ground. A virtual ground means that only a small voltage appears at this node and the resistance to ground is very small, approaching zero.

From Eq. (7), the voltage V_x may be determined as

$$V_x = \frac{V_o}{A}. \tag{8}$$

If the output voltage, V_o, is finite, which is guaranteed by the presence of negative feedback, then the voltage V_x will approach a value of zero volts as A approaches a large number. Thus, the voltage V_x is always a very small voltage. While it will not be shown here, it also has a very small resistance to ground and, therefore, satisfies the requirements of a virtual ground.

The virtual-ground node is sometimes referred to as a summing junction since it allows the currents flowing through any connected resistances to sum together and flow through the feedback resistor, R_2. For example, the current flowing into the node x through R_1 is V_i/R_1. The current flowing from node x through R_2 is $-V_o/R_2$. Assuming that no current flows into the input terminal of the op amp, the satisfaction of Kirchhoff's current law requires that

$$\frac{V_i}{R_1} = \frac{-V_o}{R_2}. \tag{9}$$

This establishes the voltage gain of the circuit at

$$G = \frac{V_o}{V_i} = \frac{-R_2}{R_1}. \tag{10}$$

Thus, for an ideal op amp, the gain is determined by the resistor ratio R_2/R_1 and is independent of the op amp gain A as long as it is large.

Practical op amps are typically constructed from transistor devices arranged in a manner such as that shown in Fig. 5. This typical op amp circuit is constructed from MOS (metal-oxide semiconductor) transistor devices and consists of an input differential stage (devices M_1 and M_2), a second gain stage (device M_9), and an output stage (device M_8). Other devices in this circuit provide dc bias current and active loads for the gain stages.

Actual op amps such as that shown in Fig. 5 differ from the ideal op amp defined by Eq. (7) and Fig. 3 in several ways. A practical op amp has a finite, rather than an infinite, input impedance and a finite, rather than zero, output impedance. Actual op amps also exhibit imbalances that cause the output to depend on input voltages and an input offset voltage, V_{OS}. This dependency can be expressed as

$$V_o = A(V_1 - V_2 - V_{OS}). \tag{11}$$

The input offset voltage, V_{OS}, is typically caused by mismatches in the input stage, for example, in devices M_1 and M_2 in the circuit of Fig. 5, and it may be positive or negative. This circuit may also function over a limited range of input voltages, sometimes called the common-mode input-voltage range. The offset voltage, V_{OS}, will also typically depend on both the common-mode input range and the power-supply voltage, $+V$. Thus, a practical op amp will include specifications on CMR (common-mode input range), CMRR (common-mode rejection ratio), and PSRR (power-supply rejection ratio). The bandwidth and gain of the op amp are usually specified, often as a gain–bandwidth product, which is the product of the low frequency gain and the bandwidth of the amplifier measured in an open loop (nonfeedback) condition.

III. DIGITAL-TO-ANALOG AND ANALOG-TO-DIGITAL CONVERTERS

The digital-to-analog converter, sometimes called a D/A converter or a DAC, is one of the most commonly used

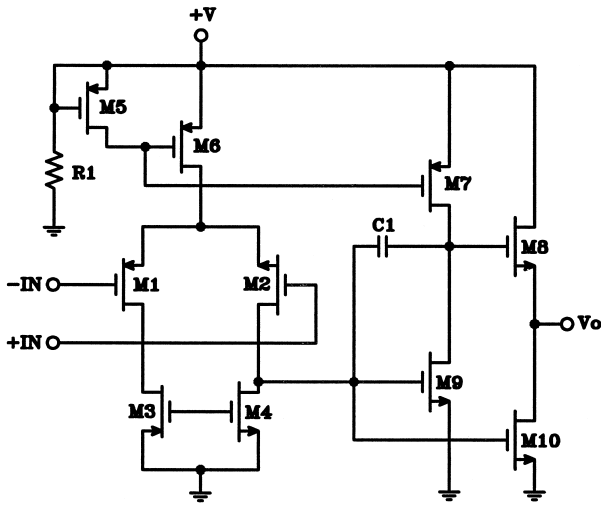


FIGURE 5 Typical MOS (metal-oxide semiconductor) op amp schematic.

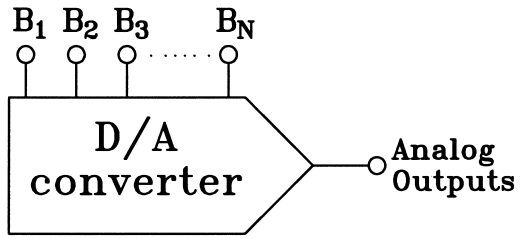


FIGURE 6 The digital-to-analog converter (DAC).

mixed-signal circuits; that is, it processes both analog and digital signals in the same circuit. Its input signals are digital, often derived from a microprocessor or a computer bus, while its output is an analog signal. Figure 6 shows a block diagram of the DAC. The input to this converter is a digital *word* of a prescribed number of bits, while the output is an analog signal uniquely related through coding to the input digital word.

A large number of D/A conversion techniques have been developed over the years. However, the DAC circuits most suitable for integrated circuit design may be classified into one of three basic categories: (1) current-scaling circuits, (2) voltage-scaling circuits, and (3) charge-scaling circuits. These approaches scale and sum precisely coded currents, voltages, or charge to generate the output analog signal. The first category is generally best suited for bipolar technology while the latter two are more widely used in CMOS (complementary metal-oxide–silicon) technology.

A. The Current-Scaling DAC

The first monolithic DAC to become widely used in industry was the DAC-08, originally designed by Precision Monolithics and Motorola. The DAC-08 uses the current-scaling approach.

Figure 7 illustrates a current-scaling DAC that uses precision current sources set up by a reference amplifier, which works with an $R-2R$ ladder and scaled n–p–n bipolar junction transistors to create binary-weighted current sources.

The reference amplifier in Fig. 7 consists of an op amp, A_1, working in conjunction with transistor Q_0 and resistor R_0 to produce a circuit that forces the current I_{REF} into the collector of Q_0. The devices Q_1 through Q_N can be designed to produce currents that are equal to, or precisely related to, I_{REF} by proper choice of R and the dimensions of the transistors. The circuit of Fig. 7 allows for switching between a true input, I_O, and its complement, I_O^*. This may be accomplished by a differential switch as illustrated in Fig. 8.

The DAC of Fig. 7 can be characterized in terms of the following equations, assuming that all inputs have a binary value of one or zero. The signal B_1 is assumed to be the control for the first-bit, or most-significant-bit (MSB), current, while B_N is the control for the least-significant-bit (LSB) current for an N-bit DAC. Although other types of coding are available, the most common is binary coding. The following equations are based on the assumption that the DAC is binary coded.

In Fig. 7, a current designated the full-scale current, I_{FS}, may be defined as

$$I_{FS} = KI_{REF}, \tag{12}$$

where K is a scaling factor determined by the ratio of R_0 to R.

The current I_1, or MSB current, is chosen to be exactly one-half the full-scale current. Thus I_1 is expressed as

$$I_1 = \frac{KI_{REF}}{2} = \frac{I_{FS}}{2}. \tag{13}$$

If all bit currents are assumed to be binarily scaled in descending order, then the output current can be expressed as

$$I_O = I_{FS}\left[\frac{B_1}{2^1} + \frac{B_2}{2^2} + \cdots + \frac{B_N}{2^N}\right] \tag{14}$$

or

$$I_O = \sum_{i=1}^{N} \frac{I_{FS}B_i}{2^i}, \tag{15}$$

where $[B_1 \cdots B_N]$ is the binary input code.

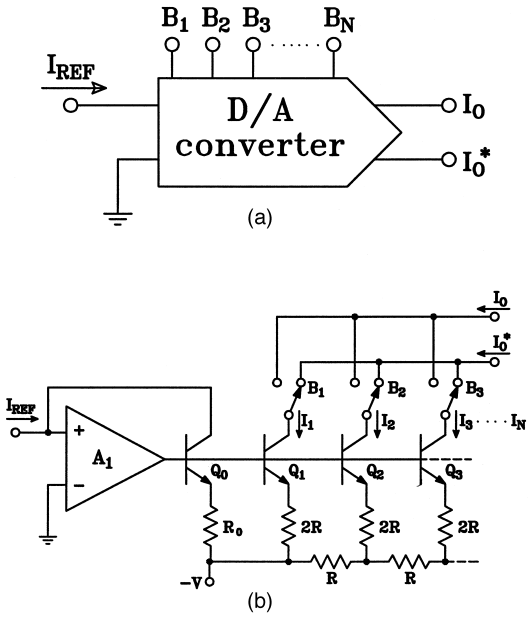


FIGURE 7 Current-scaling DAC: (a) block diagram; (b) schematic.

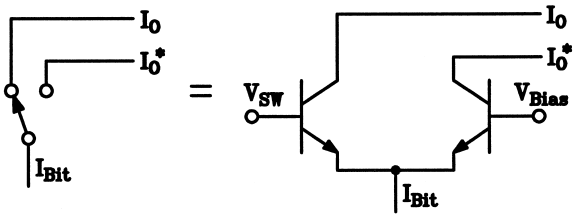


FIGURE 8 Differential DAC switch.

An alternative notation for this equation is

$$I_O = \frac{I_{FS}(n)}{2^N}, \tag{16}$$

where n is the decimal equivalent of the binary input code $[B_1 \cdots B_N]$.

1. DAC Specifications

Digital-to-analog converters are normally chosen for application within a given system on the basis of their resolution and accuracy, which, in turn, is generally determined by N, the number of input bits of the converter. The following is a listing and definitions of key DAC terms and specifications:

Resolution The resolution of a DAC is given by the number of bits, N. The resolution is the smallest increment of output that the DAC can produce. An 8-bit, DAC has a resolution of 8 bits, or one part in 2^8. This results in a percentage of 0.39%.

Linearity Linearity is the maximum allowable deviation from an ideal straight line drawn between the zero-scale and full-scale outputs. It is often given as a percentage or as a fraction of an LSB. For example, $(\frac{1}{2})$LSB linearity for an 8-bit DAC corresponds to 0.195%. Some manufacturers call this parameter the nonlinearity.

Monotonicity A DAC is said to be monotonic if each digital-code increment produces an output equal to or larger than that of the preceding code. A DAC is generally expected to be monotonic to increments as small as an LSB, but in general the smallest increment for which the DAC remains monotonic determines its monotonicity.

Settling time The settling time is measured from the instant of a digital-input-code change to the time that the analog output reaches its corresponding new value to within a specified error band. Usually the output is expected to settle to within an error band of $(\frac{1}{2})$LSB. Worst-case settling time is typically measured between the zero-scale and full-scale codes.

Differential nonlinearity The differential nonlinearity of a DAC is the maximum deviation from ideal of the analog output between any two adjacent codes. It is expressed as a percentage of full scale or in terms of an LSB value.

B. The ADC

Analog-to-digital, or A/D, conversion is the reverse of the operation previously described as D/A conversion. The function of an A/D converter, also called an ADC, is to measure an analog input signal and provide a quantized representation of the signal in the form of a digital word. Just as the DAC is considered a decoding device, the ADC can be considered an encoding device in the translation of signals between the analog and digital worlds. Figure 9 shows a block diagram representing an ADC.

The analog input signals consist of the input signal, V_i, as well as a precision reference, V_{REF}, assumed to be a dc voltage. The output consists of a digital word of N bits. In practice, the output may be generated either as a serial bit stream or in parallel as shown. However, for the purpose of definition, a parallel output with binary coding will be assumed, with B_1 representing the MSB and B_N representing the LSB. In the block diagram of Fig. 9, assuming signals are referenced to a common ground, the function of the ADC can be defined as that of generating binary words $[B] = [B_1, B_2, \ldots, B_N]$ according to the equation

$$n = \frac{V_i 2^N}{V_{FS}}, \tag{17}$$

where n is an integer and the decimal equivalent of $[B]$, and V_{FS} is the nominal full-scale voltage. The nominal full-scale voltage of the ADC is related to the reference voltage by the equation

$$V_{FS} = kV_{REF}, \tag{18}$$

where k is a constant scale factor. Alternatively, $[B]$ may be calculated from the equation

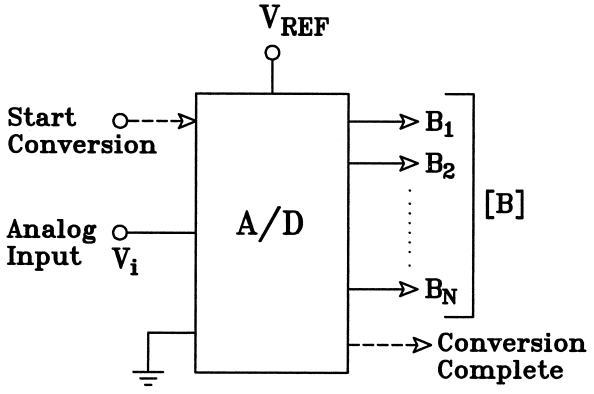


FIGURE 9 The analog-to-digital converter (ADC).

$$\left[\frac{B_1}{2^1}+\frac{B_2}{2^2}+\cdots+\frac{B_N}{2^N}\right]=\frac{V_i}{kV_{REF}}, \qquad (19)$$

where the binary coefficients of $[B]$ have a value of either one or zero as required to make the equality true.

Because an A/D conversion using any real circuit will require a finite time, it is customary to assume that the input signal is not changing during the conversion cycle. If V_i is likely to change, it is common practice to employ a sample-and-hold (S/H) circuit to capture and hold the input during the A/D conversion. In such a case, the digital output represented by Eq. (17) or Eq. (19) will correspond to the value of input that existed at the start of the conversion cycle. To clarify the point in time for which the conversion is valid, most ADCs include a start-conversion digital control signal and a digital status signal that indicates when the conversion is complete. These optional signals are shown as dashed lines on the block diagram of Fig. 9.

Although there are many practical methods of constructing an ADC, the type of converter described next serves as an instructive example.

1. The Counting ADC

Figure 10 shows a simplified diagram for a counting type of ADC which can be constructed from the following building blocks:

1. N-bit binary counter
2. N-bit current-scaling DAC
3. S/H circuit
4. Comparator circuit
5. D flip-flop and AND gate

The circuit also requires a clock signal.

The operation of the circuit can be described from the point in time that a start-conversion gate is applied. This

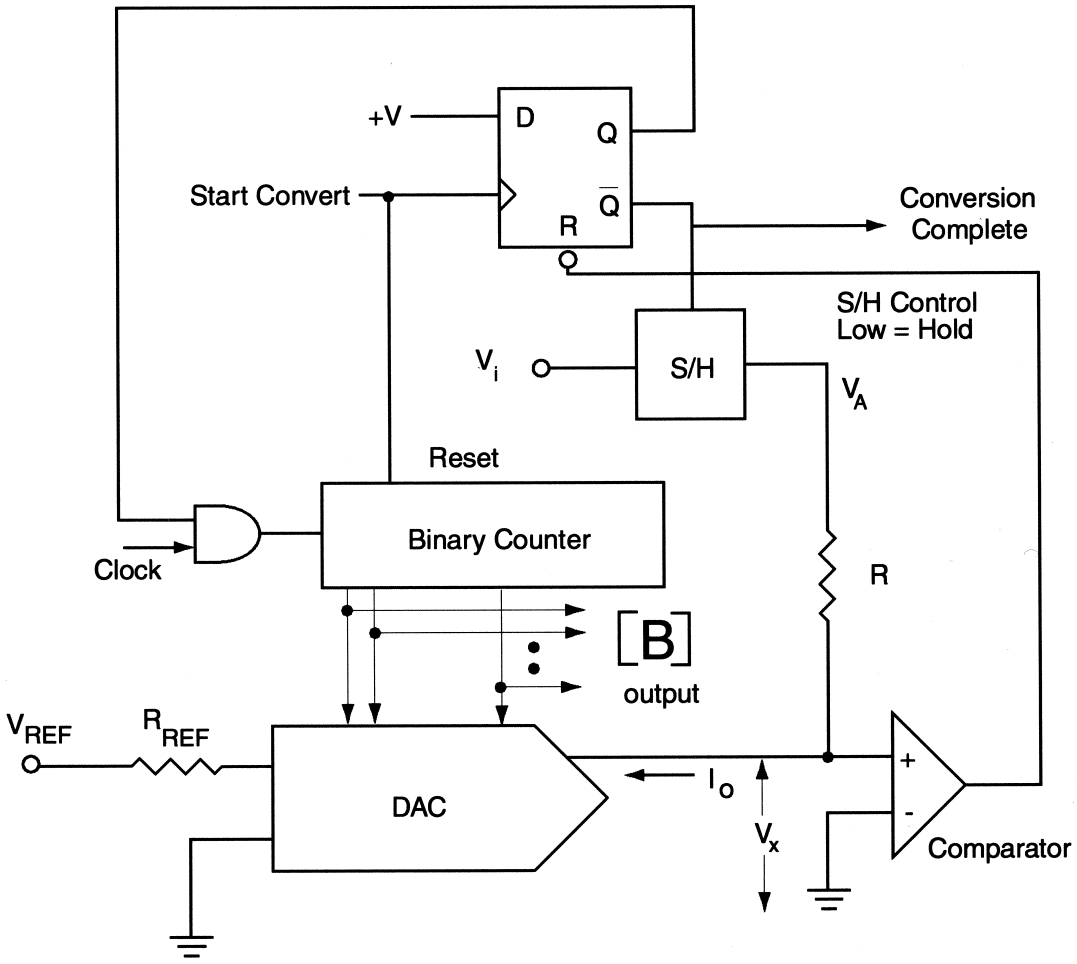


FIGURE 10 Staircase (counting) ADC. S/H, sample and hold.

signal causes the D flip-flop to set, which immediately switches the S/H circuit to a hold mode and gates the clock signal into the binary counter. The counter, which was previously reset to an all-zero count, begins a binary count-up sequence. Its output is applied to the DAC and will cause a staircase buildup of an output current, I_O. The node voltage, V_x, will initially set positive due to an assumed positive value of input, but V_x will decrease as I_O increases in binary LSB increments. The voltage V_x may be expressed as

$$V_x = V_A - I_O R. \tag{20}$$

When $I_O R$ equals or exceeds V_A, the comparator control node V_X will swing negative, which will drive the comparator output to a low state, which, in turn, will reset the flip-flop and cause the conversion-complete line to go high. The character stored in the binary counter at this time will correspond to the digital conversion of V_i.

This fact can be verified by noting that the circuit holds the binary code which makes

$$I_O R \geq V_i. \tag{21}$$

Using Eq. (16), which expresses I_O for the DAC circuit, and assuming that I_{FS} is given by V_{REF}/R_{REF}, one can write the inequality of Eq. (21) as

$$\frac{nKV_{REF}R}{R_{REF}2^N} \geq V_i. \tag{22}$$

If R is chosen equal to R_{REF}, then

$$n \geq \frac{V_i 2^N}{V_{REF}}, \tag{23}$$

which is the desired result. The significance of the equal-or-greater condition of Eq. (23) is that the binary code must advance past the point where V_x swings negative but will not advance more than a fraction of an LSB past before the conversion cycle is terminated. The accuracy and gain of the comparator will be a major factor in determining the accuracy of the ADC, along with the linearity of the DAC.

Because the digital count to the DAC always starts at a zero count and may have to cycle through all 2^N possible count combinations before finding the analog match, it is inherently slow. Furthermore, its conversion time is variable, with larger analog inputs requiring more time than that required by smaller inputs.

Other types of ADC circuits are used to decrease the time required to perform a conversion. Architectures available for higher conversion speeds include the successive-approximation ADC and the parallel, or flash, ADC.

2. ADC Specifications

The key specifications for an ADC are its resolution, linearity, and conversion time. Resolution and linearity have the same definitions as for the DAC. It must be emphasized, however, that if the output of an ADC is an N-bit code, it does not follow that the accuracy is one part in 2^N. Table I lists specifications for an ADC that are typical of those achieved by using VLSI technology.

TABLE I Typical VLSI ADC Parameters

Parameter	Specification
Resolution	12 bits
Linearity	$\pm(\frac{1}{2})$LSB
Differential nonlinearity	13 bits
Full-scale error	Adjustable to zero at 25°C
Full-scale temperature coefficient	20 ppm/°C
Analog input range	0–5 V or −2.5 to +2.5 V
Output code	Binary
Digital interface	TTL[a]
Conversion time	2 μsec

[a] TTL, transistor–transistor logic.

IV. MIXED-SIGNAL SYSTEM APPLICATIONS

The mixed-signal circuits described earlier may be used in a variety of ways to construct electronic systems with a range of choices between digital and analog approaches. This section illustrates a few systems that can be designed by using digital, analog, and mixed-signal building blocks.

A. Digital/Analog Waveform Generator

Complex analog waveforms are required in applications ranging from the synthesis of human speech to laboratory waveform test generators. Figure 11 shows a system capable of generating an arbitrary waveform from data stored in a digital memory and accessed sequentially in time by a clock-driven binary counter. The digital output characters are converted to analog by the DAC. Filtering is employed to smooth the quantized edges of the output waveform.

B. Digital Audio Systems

Audio recording, since the invention of the phonograph by Edison in 1877, has traditionally been accomplished by the storage of analog signals on cylinders, disks, or magnetic tape. However, in recent years, a transition toward digital storage has taken place, with audio signals being encoded as digital samples and stored as bits on tape or CD media. Continuous high-quality audio may be reproduced by reading the digital samples and converting back to analog by using a DAC followed by a smoothing

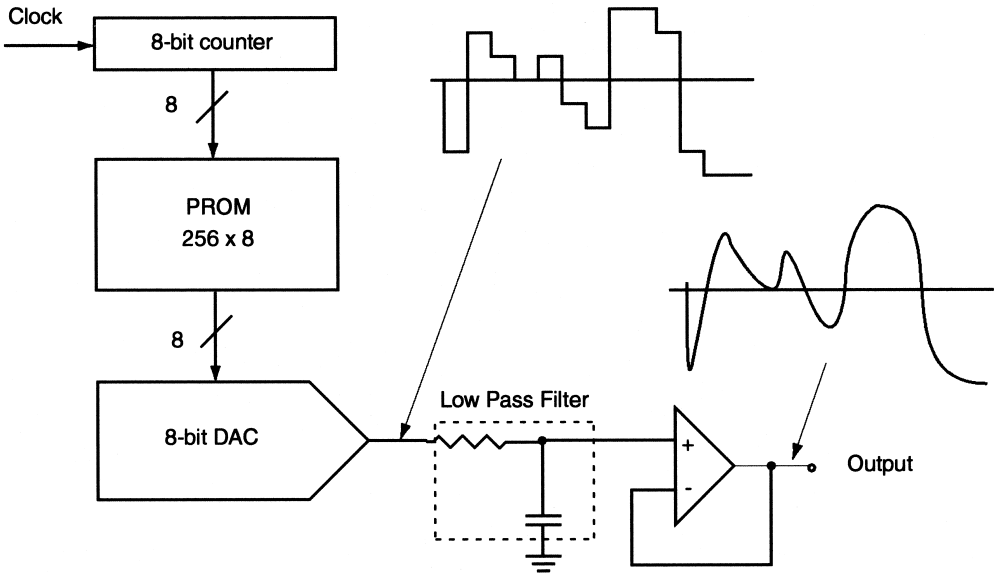


FIGURE 11 Digital/analog waveform generator. PROM, programmable read-only memory.

filter. Figure 12 suggests block diagrams for the recording and playback of digital audio.

The theoretical basis of digital audio is the sampling theorem, which states that an analog waveform with a band-limited spectrum of B Hz may be completely characterized by taking periodic samples at a rate of at least twice the bandwidth. This rate is referred to as the Nyquist sampling frequency. For example, a signal with a 20-kHz bandwidth can be characterized by taking periodic samples at a rate of 40 ksamples/sec or higher.

The advantages of a digital audio recording system over a traditional analog system include lower noise and

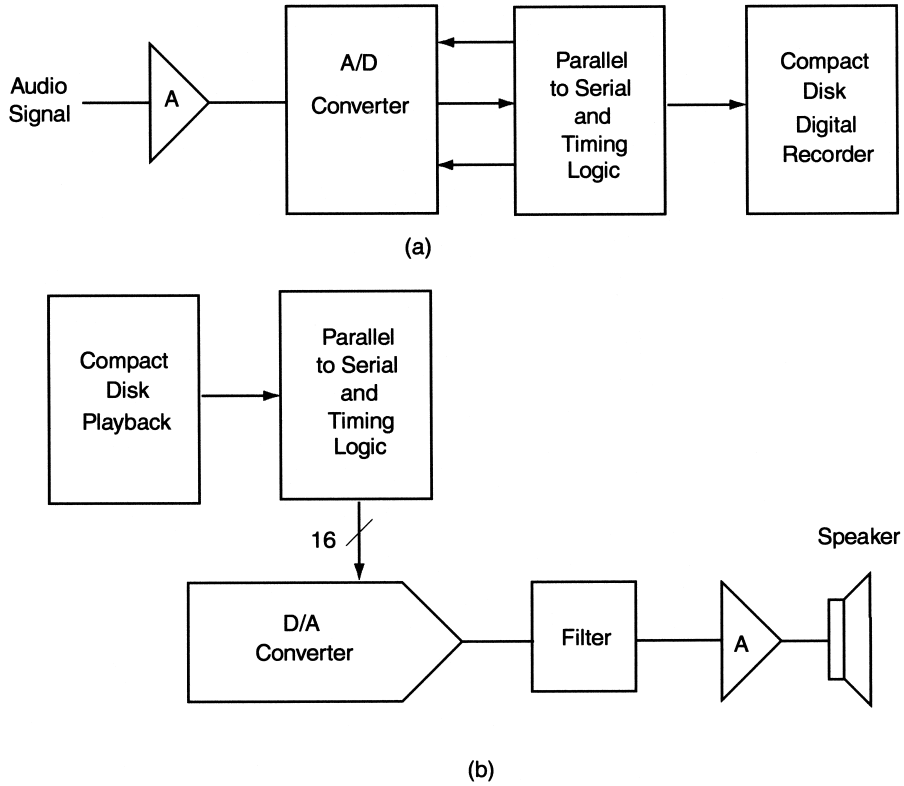


FIGURE 12 Digital audio recording and playback system: (a) recording; (b) playback.

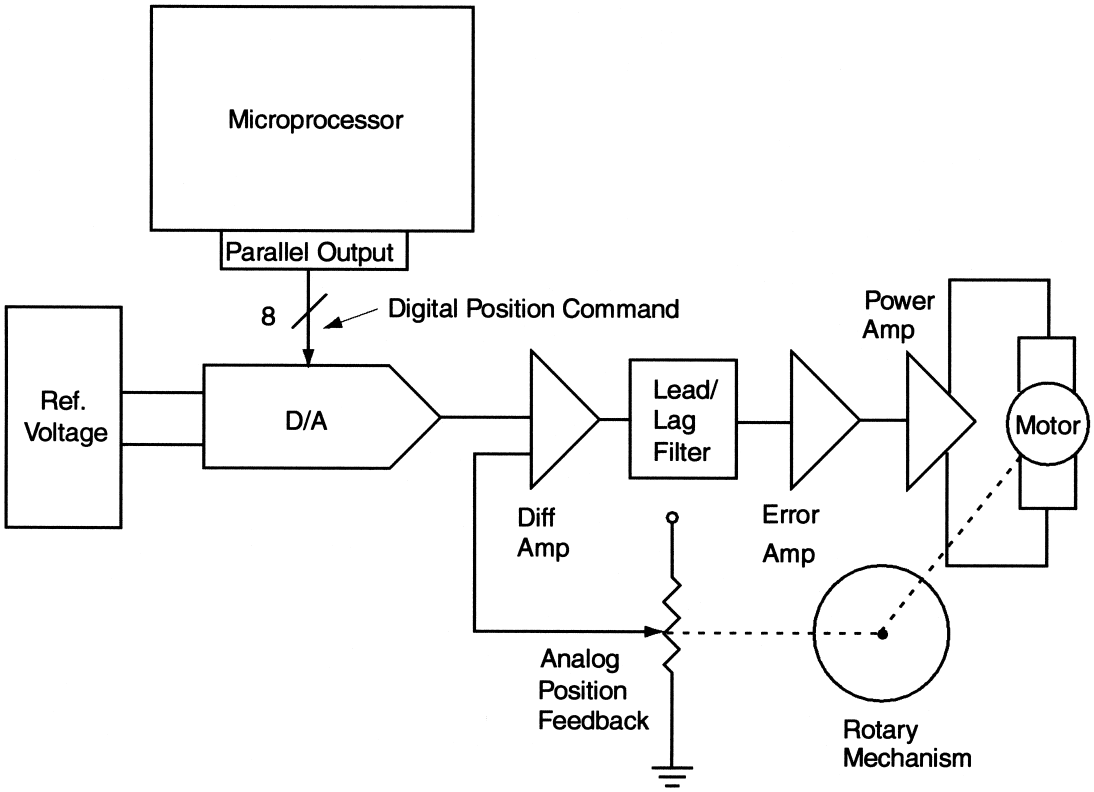


FIGURE 13 D/A rotary position system.

elimination of frequency modulation effects due to media transport variations.

C. Computer-Controlled Robotics

The use of computers and microprocessors to control mechanisms is an application that offers a very wide choice of analog versus digital techniques. Figure 13 shows a block diagram of a position servo system that might be used to control the rotation of a gun turret on a battleship, a cutting device on a milling machine, or the rotation of the head of a mechanical robot.

The system illustrated uses a position potentiometer with a shaft that is coupled to the rotary mechanism to generate position feedback. This is an analog method. Digital rotary shaft encoders are also available and the use of a digital feedback scheme would allow for a different position-system design.

SEE ALSO THE FOLLOWING ARTICLES

ANALOG ELECTRICAL FILTERS • CIRCUIT THEORY • DIGITAL FILTERS • DIGITAL ELECTRONIC CIRCUITS

BIBLIOGRAPHY

Comer, D. T. (1994). "Introduction to Mixed Signal VLSI," Array Publishing Co., New York.

Jaeger, R. C. (1997). "Microelectronic Circuit Design," McGraw-Hill, New York.

Johns, D. A., and Martin, K. (1997). "Analog Integrated Circuit Design," Wiley, New York.

Sedra, A. S., and Smith, K. C. (1998). "Microelectronic Circuits," 4th ed., Oxford Univ. Press, New York.

Tsividis, Y. (1996). "Mixed Analog–Digital VLSI Devices and Technology," McGraw-Hill, New York.



Circuit Theory

Krishnaiyan Thulasiraman
University of Oklahoma

I. Graph Theory: Basic Concepts and Results
II. Graphs and Electrical Networks
III. Loop and Cutset Systems of Equations

CIRCUIT THEORY is an important and perhaps the oldest branch of electrical engineering. A circuit is an interconnection of electrical elements: passive elements such as resistances, capacitances, inductances, active elements, and sources (or excitations). Two variables—namely, voltage and current variables—are associated with each circuit element. There are two aspects to circuit theory: analysis and design. In circuit analysis, we are interested in determination of the values of currents and voltages in different elements of the circuit, given the values of the sources or excitations. On the other hand, in circuit design, we are interested in the design of circuits, which exhibit certain prespecified voltage or current charactersitics at one or more parts of the circuit. In this chapter, we will confine our discussion to certain aspects of circuit analysis.

The behavior or dynamics of a circuit is described by three systems of equations determined by Ohm's law, Kirchhoff's voltage law, and Kirchhoff's current law. Ohm's law specifies the relationship between the voltage and current variables associated with a circuit element. This relationship is not specified for independent sources. Also, this relationship could be linear or nonlinear. If the relationship is linear, then the circuit element is called a *linear element*; otherwise, it is called a *nonlinear element*. A circuit is linear if it contains only linear elements besides independent sources. Kirchhoff's voltage specifies the dependence among the voltage variables in the circuit, and Kirchhoff's current law specifies the dependence among the current variables in the circuit.

The systems of equations determined by the application of Kirchhoff's voltage and current laws depend on the structure or the graph of the circuit. In other words, they depend only on the way the circuit elements are interconnected. Thus, the graph of a circuit plays a fundamental role in the study of circuits. Several interesting properties of circuits depend only on the structure of the circuits. Thus, the theory of graphs has played a fundamental role in discovering structural properties of electrical circuits.

In this chapter we shall develop most of those results that form the foundation of graph theoretic study of electrical circuits. A comprehensive treatment of these developments may be found in Swamy and Thulasiraman (1981). All theorems in this chapter are stated without proofs. Our discussion here follows closely our development in the Graph Theoretic Foundation of Circuit Analysis chapter in Chen (2001).

I. GRAPH THEORY: BASIC CONCEPTS AND RESULTS

Our development of graph theory is self-contained, except for the definitions of standard and elementary results from set theory and matrix theory.

• **Graph:** A graph $G = (V, E)$ consists of two sets, a finite set $V = (v1, v_2, \ldots, v_n)$ of elements called *vertices* and a finite set $E = (e_1, e_2, \ldots, e_n)$ of elements called *edges*.

• **Directed and undirected graph:** If the edges of G are identified with ordered pairs of vertices, then G is called a directed or an *oriented graph*; otherwise,it is called an *undirected* or an *unoriented graph*.

Graphs permit easy pictorial representations. In a pictorial representation each vertex is represented by a dot and each edge is represented by a line joining the dots associated with the edge. In directed graphs, an orientation or direction is assigned to each edge. If the edge is associated with the ordered pair (v_i, v_j), then this edge is oriented from v_i to v_j. If an edge e connects vertices v_i and v_j, then it is denoted by $e = (v_i, v_j)$. In a directed graph, (v_i, v_j) refers to an edge directed from (v_i, v_j). A graph and a directed graph are shown in Fig. 1. Unless explicitly stated, the term "graph" may refer to a directed graph or an undirected graph.

• **End vertices:** The vertices v_i and v_j associated with an edge are called the *end vertices* of the edge.

• **Parallel edges:** All edges having the same pair of end vertices are called *parallel* edges. In a directed graph parallel edges refer to edges connecting the same pair of verices v_i and v_j the same way from v_i to v_j or from v_j to v_i . For instance, in the graph of Fig. 1a, the edges connecting v_1 and v_2 are parallel edges. In the directed graph of Fig. 1b the edges connecting v_3 and v_4 are parallel edges. However, the edges connecting v_1 and v_2 are not parallel edges because they are not oriented in the same way.

• **Self loop:** If the end vertices of an edge are not distinct, then the edge is called a *self loop*. The graph of Fig. 1a has one self loop and the graph of Fig. 1b has two self loops.

An edge is said to be *incident on* its end vertices. In a directed graph the edge (v_i, v_j) is said to be *incident out* of v_i and is said to be *incident into* v_j. Vertices v_i and v_j are adjacent if an edge connects v_i and v_j.

• **Degee:** The number of edges incident on a vertex v_i is called the *degree* of v_i and is denoted by $d(v_i)$.

• **In-degree:** In a directed graph, $d_{in}(v_i)$ refers to the number of edges incident into vertex v_i, and it is called the *in-degree*.

• **Out-degree:** In a directed graph, $d_{out}(v_i)$ refers to the number of edges incident out of the vertex v_i.

• **Isolated vertex:** If $d(v_i) = 0$, then v_i is said to be an *isolated vertex*.

• **Pendant vertex:** If $d(v_i) = 1$, then v_i is said to be a *pendant vertex*.

A self loop at a vertex v_i is counted twice while computing $d(v_i)$. As an example,in the graph of Fig. 1a, $d(v_1) = 3$, $d(v_4) = 3$, and v_5 is an isolated vertex. In the directed graph of Fig. 1b, $d_{in}(v_1) = 3$, and $d_{out}(v_1) = 2$.

Note that in a directed graph, for every vertex v_i,

$$d(v_i) = d_{in}(v_i) + d_{out}(v_i)$$

Theorem 1

1. The sum of the degrees of the vertices of a graph G is equal to $2m$, where m is the number of edges of G.
2. In a directed graph with m edges, the sum of the in-degrees and the sum of the out-degrees are both equal to m.

The following theorem is known to be the first major result in graph theory.

Theorem 2

The number of vertices of odd degree in any graph is even.

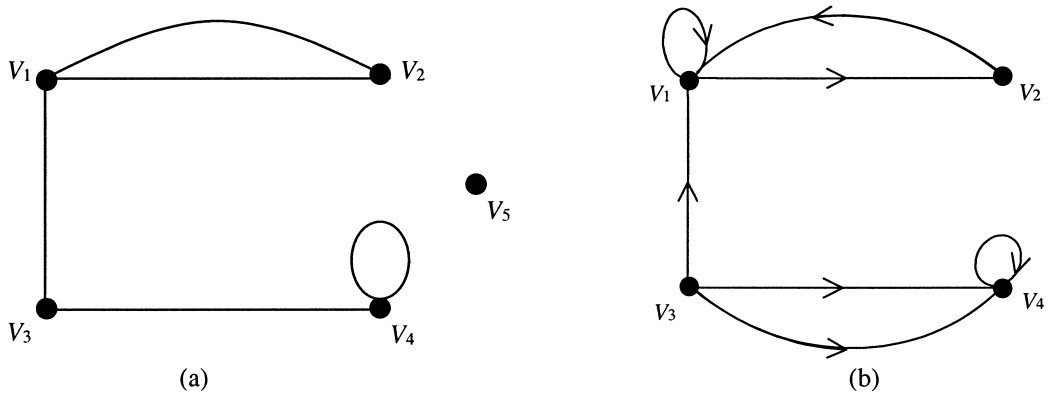


FIGURE 1 (a) An undirected graph, (b) a directed graph.

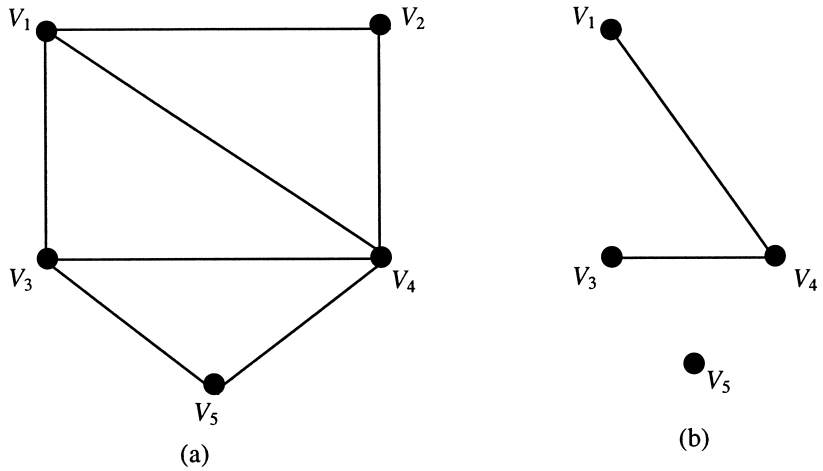


FIGURE 2 (a) Graph G; (b) subgraph G.

Consider a graph $G = (V', E')$. The graph $G' = (V', E')$ is a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$. As an example, a graph G and a subgraph of G are shown in Fig. 2.

- **Path:** In a graph G a *path* P connecting vertices v_i and v_j is an alternating sequence of vertices and edges starting at v_i and ending at v_j, with all vertices except v_i and v_j being distinct.
- **Directed path:** In a directed graph a path P connecting vertices v_i and v_j is a called a *directed path* from v_i to v_j if all the edges in P are oriented in the same direction as we traverse P from v_i toward v_j.
- **Circuit:** If a path starts and ends at the same vertex, it is called a *circuit*.
- **Directed circuit:** In a directed graph, a circuit in which all the edges are oriented in the same direction is called a *directed circuit*. It is often convenient to represent paths and circuits by the sequence of edges representing them. For example, in the undirected graph of Fig. 3a, P: e_1, e_2, e_3, e_4 is a path connecting v_1 and v_5, and C: $e_1, e_2, e_3, e_4, e_5, e_6$ is a circuit. In the directed graph of Fig. 3b, P: e_1, e_2, e_7, e_5 is a directed path, and C: e_1, e_2, e_7, e_6 is a directed circuit. Note that C: e_7, e_5, e_4, e_1, e_2 is a circuit in this directed graph, although it is not a directed circuit. Similarly, P: e_9, e_6, e_1 is a path but not a directed path.
- **Connected graph:** A graph is *connected* if there is a path between every pair of vertices in the graph; otherwise, the graph is not connected. For example, the graph in Fig. 2a is a connected graph, whereas the graph in Fig. 2b is not a connected graph.
- **Tree:** A *tree* is a graph that is connected and has no circuits.
- **Spanning tree:** Consider a connected graph G, A subgraph of G is a *spanning tree* of G if the subgraph is a tree and contains all the vertices of G. A tree and a spanning tree of the graph of Fig. 4a are shown in Figs. 4b and 4c, respectively.
- **Branches:** The edges of a spanning tree T are called the braches of T.

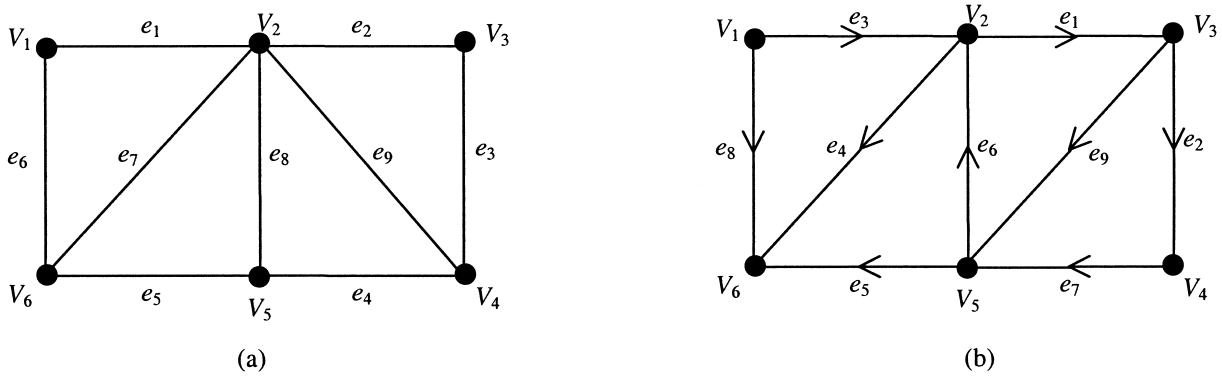


FIGURE 3 (a) An undirected graph; (b) a directed graph.

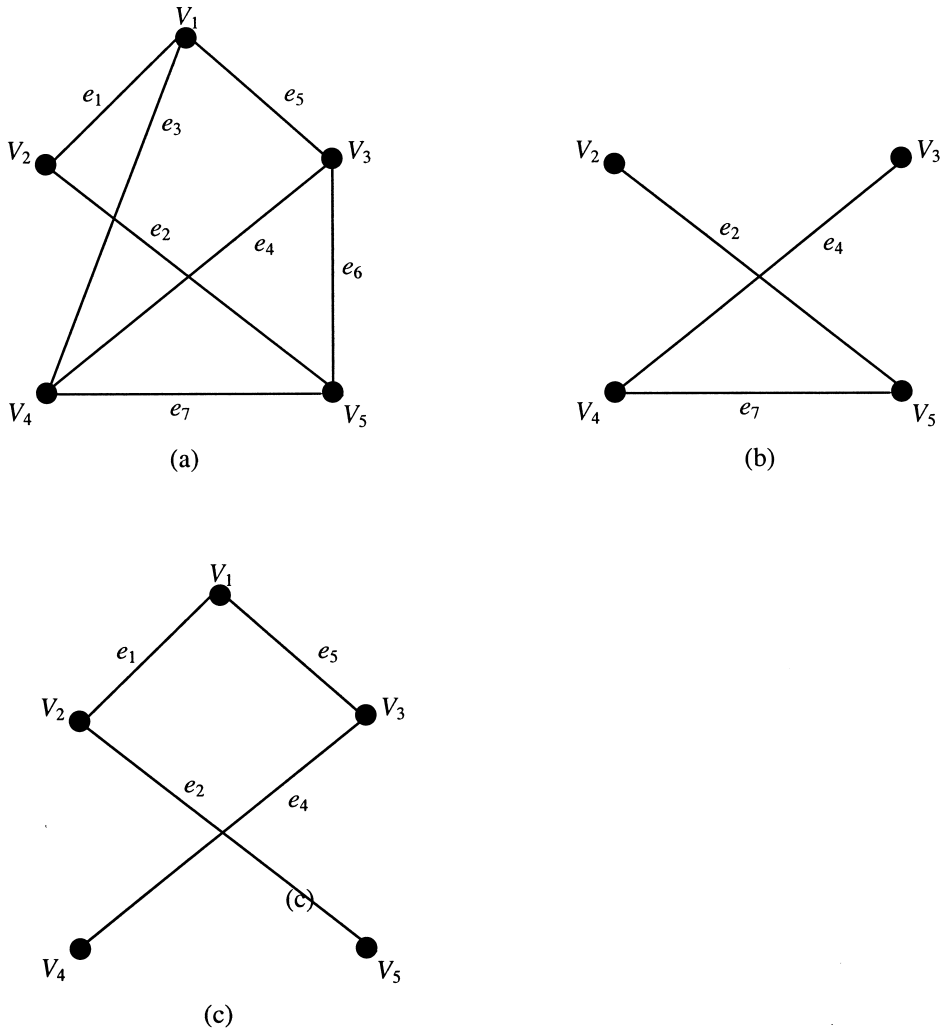


FIGURE 4 (a) Graph G; (b) a tree of graph G; (c) a spanning tree of G.

• **Cospanning tree:** Given a spanning tree of connected graph G, the cospanning tree relative to T is the subgraph of G induced by the edges that are not present in T. For example, the cospanning tree is relative to the spanning tree T if Fig. 4c consists of the edges e_3, e_6, e_7.

• **Chords:** The edges of a cospanning tree are called *chords*.

It can be easily be verified that in a tree exactly one path connects any two vertices. It should be noted that a tree is minimally connected in the sense that removing any edge from the tree will result in a disconnected graph.

Theorem 3

A tree on n vertices has $n - 1$ edges.

If a connected graph G has n vertices and m edges, then the rank ρ and nullity μ of G are defined as follows:

$$\rho(G) = n - 1$$
$$\mu(G) = m - n + 1$$

The concepts of rank and nullity have parallels in other branches of mathematics, such as matrix theory.

Clearly, if G is connected, then any spanning tree of G has $\rho = n - 1$ branches and $\mu = m - n + 1$ chords.

A. Cuts, Circuits, and Orthogonality

We introduce here the notions of a cut and a cutset and develop certain results which bring out the dual nature of circuits and cutsets. Consider a connected graph $G = (V, E)$ with n vertices and m edges. Let V_1 and V_2 be two mutually disjoint nonempty subsets of V such that $V = V_1 \cup V_2$; thus, V_2 is the complement of V_1 in V and vice versa. V_1 and V_2 are also said to form a partition of V. Then the set

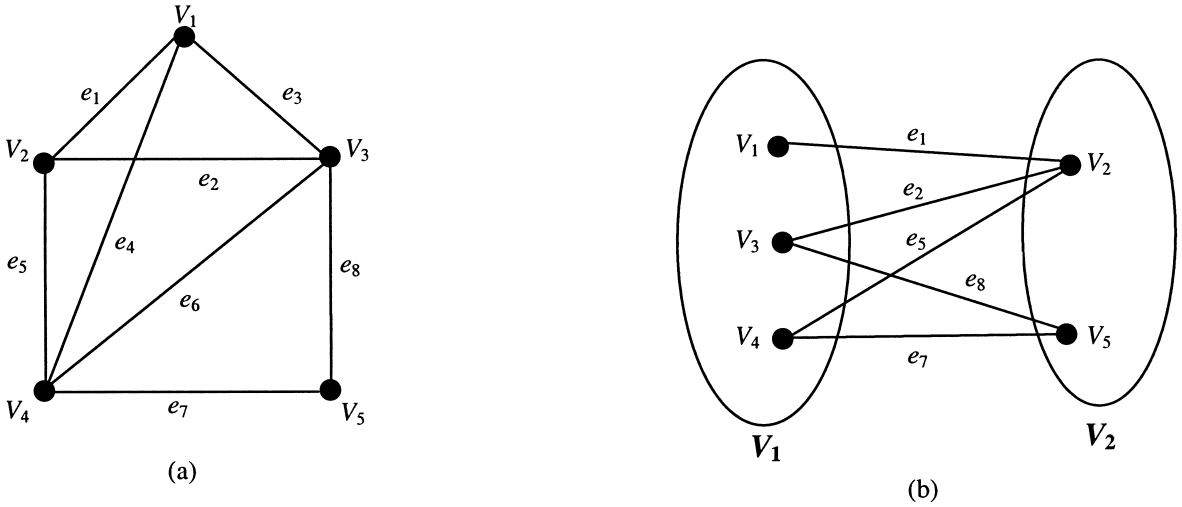


FIGURE 5 (a) Graph G; (b) cut $\langle V_1, V_2 \rangle$ of G.

of all those edges which have one end vertex in V_1 and the other in V_2 is called a *cut* of G. As an example, a graph and a cut $\langle V_1, V_2 \rangle$ of G are shown in Fig. 5.

The graph G' which results after removing the edges in a cut will not be connected. A *cutset* S of a connected graph G is a minimal set of edges of G such that removal of S disconnects G. Thus, a cutset is also a cut. Note that the minimality property of a cutset implies that no proper subset of a cutset is a cutset.

Consider a spanning tree T of a connected graph G. Let b be a branch of T. Removal of the branch b disconnects T into two trees, T_1 and T_2. Let V_1 and V_2 denote the vertex sets of T_1 and T_2, respectively. Note that V_1 and V_2 together contain all the vertices of G. We can verify that the cut $\langle V_1, V_2 \rangle$ is a cutset of G and is called the *fundamental cutset* of G with respect to branch b of T. Thus, for a given graph G and a spanning tree T of G, we can construct $n-1$ fundamental cutsets, one for each branch of T. As an example, for the graph shown in Fig. 5, the fundamental cutsets with respect to the spanning tree $T = [e_1, e_2, \ldots, e_6, e_8]$ are

Branch e_1: (e_1, e_3, e_4)
Branch e_2: (e_2, e_3, e_4, e_5)
Branch e_6: (e_6, e_4, e_5, e_7)
Branch e_8: (e_8, e_7)

Note that the fundamental cutset with respect to branch b contains b. Furthermore, branch b is not present in any other fundamental cutset with respect to T.

Next we identify a special class of circuits of a connected graph G. Again, let T be a spanning tree of G. Because exactly one path exists between any two vertices of T, adding a chord c to T produces a unique circuit. This circuit is called the *fundamental circuit* of G with respect to chord c of T. Note again that the fundamental circuit with respect to chord c contains c, and chord C is not present in any other fundamental circuit with respect to T. As an example, the set of fundamental circuits with respect to the spanning tree $T = (e_1, e_2, e_6, e_8)$ of the graph shown in Fig. 5 is

Chord e_3: (e_3, e_1, e_2)
Chord e_4: (e_4, e_1, e_2, e_6)
Chord e_5: (e_5, e_2, e_6)
Chord e_7: (e_7, e_8, e_6)

B. Incidence, Circuit, and Cut Matrices of a Graph

The incidence, circuit, and cut matrices are coefficient matrices of Kirchoff's equations which describe an electrical network. We next define these matrices and present some properties of these matrices which are useful in the study of electrical networks.

1. Incidence Matrix

Consider a connected directed graph G with n vertices and m edges and having no self loops. The *all-vertex incidence matrix* $A_c = [a_{ij}]$ of G has n rows, one for each vertex, and m columns, one for each edge. The element a_{ij} of A_c is defined as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } j\text{th edge is incident out of the } i\text{th vertex} \\ -1, & \text{if } j\text{th edge is incident into the } i\text{th vertex} \\ 0, & \text{if the } j\text{th edge is not incident on the } i\text{th vertex} \end{cases}$$

As an example, the A_c matrix of the directed graph in Fig. 6 is given below:

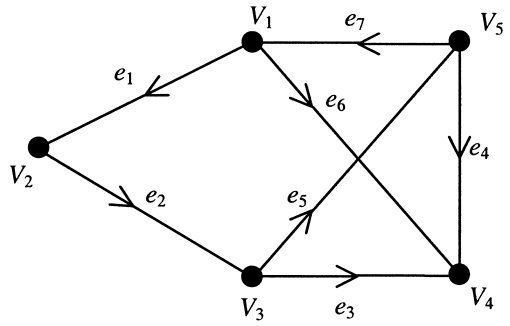


FIGURE 6 A directed graph.

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 1 \end{bmatrix} \end{array}$$

From the definition of A_c, it should be clear that each column of this matrix has exactly two nonzero entries, one $+1$ and one -1; therefore, we can obtain any row of A_c from the remaining rows. Thus,

$$\text{rank}(A_c) \leq n - 1$$

An $(n-1)$-rowed submatrix of A_c is referred to as an *incidence matrix* of G. The vertex which corresponds to the row which is not in A_c is called the *reference vertex* of A.

2. Cut Matrix

Consider a cut (V_a, V_b) in a connected directed graph G with n vertices and m edges. Recall that $\langle V_a, V_b \rangle$ consists of all those edges connecting vertices in V_a to V_b. This cut may be assigned an orientation from V_a to V_b or from V_b to V_a. Suppose the orientation of (V_a, V_b) is from V_a to V_a. Then the orientation of an edge (v_i, v_j) is said to agree with the cut orientation if $v_i \in V_a$, and $v_j \in V_b$.

The *cut matrix* $Q_c = [q_{ij}]$ of G has m columns, one for each edge, and has one row for each cut. The element q_{ij} is defined as follows:

$$q_{ij} = \begin{cases} 1, \text{ if the } j\text{th edge is in the } i\text{th cut and its orientation agrees with the cut orientation} \\ -1, \text{ if the } j\text{th edge is in the } i\text{th cut and its orientation does not agree with the cut orientation} \\ 0, \text{ if the } j\text{th edge is not in the } i\text{th cut} \end{cases}$$

Each row of Q_c is called the *cut vector*. The edges incident on a vertex form a cut. Thus it follows that the matrix A_c is a submatrix of Q_c. Next we identify another important submatrix of Q_c.

Recall that each branch of a spanning tree T of connected graph G defines a fundamental cutset. The submatrix of Q_c corresponding to the $n-1$ fundamental cutsets defined by T is called the *fundamental cutset matrix* Q_f of G with respect to T.

Let $b_1, b_2, \ldots, b_{n-1}$ denote the branches of T. Let us assume that the orientation of a fundamental cutset is chosen so as to agree with that of the defining branch. Suppose we arrange the rows and the columns of Q_f so that the ith column corresponds to the fundamental cutset defined by b_i. Then the matrix Q_f can be displayed in a convenient form as follows:

$$Q_f = [U \mid Q_{fc}]$$

where U is the unit matrix of order $n-1$, and its columns correspond to the branches of T. As an example, the fundamental cutset matrix of the graph in Fig. 6 with respect to the spanning tree $T = (e_1, e_2, e_5, e_6)$ is given below:

$$Q_f = \begin{array}{c} \\ e_1 \\ e_2 \\ e_5 \\ e_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_5 & e_6 & e_3 & e_4 & e_7 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

It is clear that the rank of Q_f is $n-1$. Hence,

$$\text{rank}(Q_c) \geq n - 1$$

3. Circuit Matrix

Consider a circuit C in a connected directed graph G with n vertices and m edges. This circuit can be traversed in one of two directions, clockwise or counterclockwise. The direction we choose for traversing C is called the *orientation* of C. If an edge $e = (v_i, v_j)$ directed from v_i to v_j is in C and if v_i appears before v_j as we traverse C in the direction specified by the orientation of C, then we say that the orientation agrees with the orientation of e.

The *circuit matrix* $B_c = [b_{ij}]$ of G has m columns, one for each edge, and has one row for each circuit in G. The element b_{ij} is defined as follows:

$$b_{ij} = \begin{cases} 1, \text{ if the } j\text{th edge is in the } i\text{th circuit and its orientation agrees with the circuit orientation} \\ -1, \text{ if the } j\text{th edge is in the } i\text{th circuit and its orientation does not agree with circuit orientation} \\ 0, \text{ if the } j\text{th edge is not in the } i\text{th circuit} \end{cases}$$

The submatrix of B_c corresponding to the fundamental circuits defined by the chords of a spanning tree T is called

the *fundamental circuit matrix* B_f of G with respect to the spanning tree T.

Let $c_1, c_2, c_3, \ldots, c_{m-n+1}$ denote the chords of T. Suppose we arrange the columns and the rows of B_f so that the ith row corresponds to the fundamental circuit defined by the chord c_i and the ith column corresponds to the chord c_i If, in addition, we choose the orientation of a fundamental circuit to agree with the orientation of the defining chord, we can write B_f as:

$$B_f = [U \perp B_{ft}]$$

where U is the unit matrix of order $m-n+1$, and its columns correspond to the chords of T.

As an example, the fundamental circuit matrix of the graph shown in Fig. 6 with respect to the tree $T = (e_1, e_2, e_5, e_6)$ is given below:

$$B_f = \begin{array}{c} \\ e_3 \\ e_4 \\ e_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_3 & e_4 & e_7 & e_1 & e_2 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 & 1 & -1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

It is clear from the above that the rank of B_f is $m-n+1$; hence,

$$\text{rank}(B_c) \geq m-n+1.$$

The following results constitute the foundation of the graph theoretic application to electrical circuit analysis.

Theorem 4 (orthogonality relationship)

1. A circuit and a cutset in a connected graph have an even number of common edges.
2. If circuit and a cutset in a directed graph have $2k$ common edges, then k of these edges have the same relative orientation in the circuit and the cutset, and the remaining k edges have one orientation in the circuit and the oppostite orientation in the cutset.

Theorem 5

If the columns of the circuit matrix B_c and the columns of the cut matrix Q_c are arranged in the same edge order, then

$$B_c Q_c = 0$$

Theorem 6

Rank$(B_c) = m-n+1$ and rank$(Q_c) = n-1$.

Note that it follows from the above theorem that the rank of the circuit matrix is equal to the nullity of the graph, and the rank of the cut matrix is equal to the rank of the graph. This result, in fact, motivated the definitions of the rank and nullity of a graph

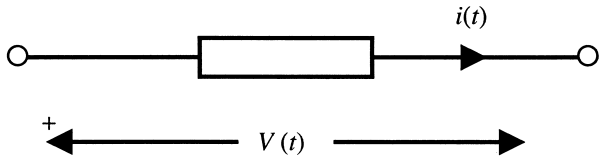


FIGURE 7 A network element with reference convention.

II. GRAPHS AND ELECTRICAL NETWORKS

An electrical network is an interconnection of electrical network elements such as resistances, capacitances, inductances, voltage and current sources, etc. Each network element is associated with two variables, the voltage variable, $v(t)$ and the current variable $i(t)$. We also assign reference directions to the network elements (see Fig. 7) so that $i(t)$ is positive whenever the current is in the direction of the arrow, and $v(t)$ is positive whenever the voltage drop in the network element is in the direction of the arrow. Replacing each element and its associated reference direction by a directed edge results in the directed graph representing the network. For example, a simple electrical network and the corresponding directed graph are shown in Fig. 8.

The physical relationship between the current and voltage variables of network elements is specified by Ohm's law. For voltage and current sources, the voltage and current variables are required to have specified values. The linear dependence among the voltage variables in the network and the linear dependence among the current variables are governed by Kirchoff's voltage and current laws.

Kirchoff's Voltage Law (KVL): The algebraic sum of the voltages around any circuit is equal to zero.

Kirchoff's Current Law (KCL): The algebraic sum of the currents flowing out of a node is equal to zero.

As an example, the KVL equation for the circuit 1, 3, 5 and the KCL equation for vertex b in the graph of Fig. 8 and

$$\begin{array}{ll} \text{Circuit } 1, 3, 5 & v_1 + v_3 + v_5 = 0 \\ \text{Vertex } b & -i_1 + i_2 + i_3 = 0 \end{array}$$

It can be easily seen that KVL and KCL equations for an electrical network N can be conveniently written as:

$$A_c I_e = 0$$

and

$$B_c V_e = 0$$

where A_c and B_c are, respectively, the incidence and circuit matrices of the directed graph representing N; I_e and V_e are, respectively, the column vectors of element currents and voltages in N. Because each row in the cut matrix Q_c can be expressed as a linear combination of the rows of

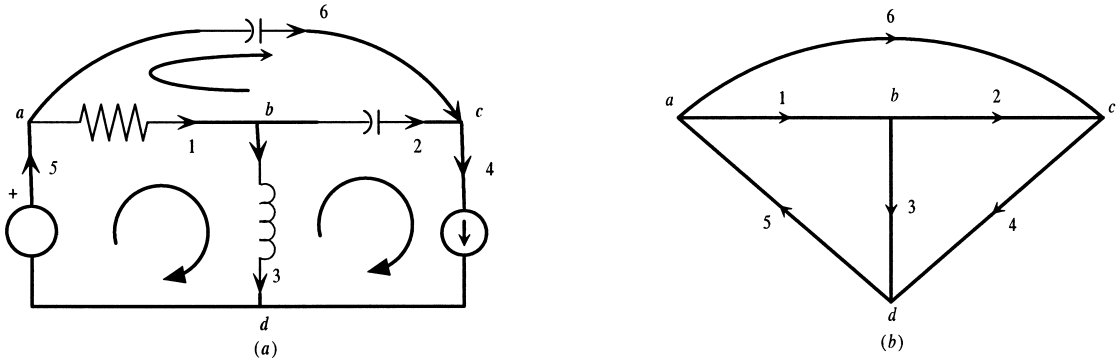


FIGURE 8 (a) An electrical network N; (b) directed graph representation of N.

the matrix, in the above we can replace A_c by Q_c; thus, we have

$$\text{KCL:} \quad Q_c I_e = 0$$
$$\text{KVL:} \quad B_c V_e = 0$$

Thus, KCL can also be stated as: The algebraic sum of the currents in any cut of N is equal to zero.

If a network N has n vertices and m elements and its graph is connected, then there are only $(n-1)$ linearly independent cuts and only $(m-n+1)$ linearly independent circuits. Thus, in writing KVL and KCL equations we need to use only B_f, a fundamental circuit matrix, and Q_f, a fundamental circuit matrix, respectively. Thus, we have

$$\text{KCL:} \quad Q_f I_e = 0$$
$$\text{KVL:} \quad B_f V_e = 0$$

We note that the KCL and KVL equations depend only on the way network elements are interconnected and not on the nature of the network elements. Thus, several results in electrical network theory are essentially graph theoretic in nature. Some results of interest in electrical network analysis are presented in the reminder of this chapter. In the following, a network N and its directed graph representation are both denoted by N.

Loop and Cutset Transformations

Let T be a spanning tree of an electrical network. Let I_c and V_t be the column vectors of chord currents and branch currents with respect to T.

1. Loop transformation:

$$I_e = B_f^t I_c$$

2. Cutset transformation:

$$V_e = Q_f^t V_t$$

If, in the cutset transformation, we replace Q_f by the reduced incidence matrix A, then we get the *node transformation* given below:

$$V_e = A^t V_n$$

where the elements in the vector V_n can be interpreted as the voltages of the nodes with respect to the reference node r. (Note: the matrix A does not contain the row corresponding to the node r.)

The above transformations have been extensively employed in developing different methods of network analysis. Two of these methods are described in the following.

III. LOOP AND CUTSET SYSTEMS OF EQUATIONS

As we observed earlier, the problem of network analysis is to determine the voltages and currents associated with the elements of an electrical network. These voltages and currents can be determined from Kirchoff's equations and the element voltage–current (in short, $v-i$) relations given by Ohm's law. However, these equations involve a a large number of variables. As can be seen from the loop and cutset transformations, not all these variables are independent Furthermore, in place of KCL equations we can use the loop transformation which invloves only chord currents as variables. Similarly, KVL equations can be replaced by the cutset transformation which involves only branch voltage variables. We can take advantage of these transformations to establish different sytems of network equations known as the loop and cutset systems.

In deriving the loop system we use the loop transformation in place of KCL, and in this case the loop variables (chord currents) will serve as independent variables. In deriving the cutset system we use the cutset transformation in place of KVL, and the cutset variables (tree branch voltages) will serve as the independent variables in this case. Consider a connected electrical network N. We assume that N consists of only resistances (R), capacitances (C), inductances (L) (referred to collectively as RCL) including mutual inductances, and independent

voltage and current sources. We also assume that all initial inductor currents and initial capacitor volatages have been replaced by appropriate sources. Further, the volatage and current variables are all Laplace transforms of the complex frequency variable s. In N there can be no circuit consisting of only independent voltage sources, for, if such a circuit of sources were present, then by KVL there would be a linear relationship among the corresponding voltages, violating the independence of the voltage sources. For the same reason, in N there can be no cutset consisting of only independent current sources. So there exists in N a spanning tree containing all the voltage sources but not current sources. Such a tree is the starting point for the development of both the loop and cutset systems of equations.

Let T be a spanning tree of the given network such that T contains all the voltage sources but no current sources. Let us partition the element voltage V_e and the element current vector I_e as follows:

$$V_e = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \quad \text{and} \quad I_e = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}$$

where the subscripts 1, 2, and 3 refer to the vectors corresponding to the current sources, RCL elements, and voltage sources, respectively. Let B_f be the fundamental circuit matrix of N, and Q_f the fundamental cutset matrix of N with respect to T. Then the KVL and the KCL equations can written as follows:

$$\text{KVL:} \quad B_f V_e = \begin{bmatrix} U & B_{12} & B_{13} \\ 0 & B_{22} & B_{23} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = 0$$

$$\text{KCL:} \quad Q_f I_e = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{21} & Q_{22} & U \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = 0$$

A. Loop Method of Network Analysis

• Step 1: Solve the following for the vector I_l (note that I_l is the vector of currents in the nonsource chords of T).

$$Z_l I_l = -B_{23} V_3 - B_{22} Z_2 B_{12} I_1 \tag{1}$$

where Z_2 is the impedance matrix of RCL elements and $Z_l = B_{22} Z_2 B_{22}$. Equation (1) is called the *loop system of equations*.

• Step 2: Calculate I_2 using:

$$I_2 = B_{12} I_1 + B_{22} I_l \tag{2}$$

then,

$$V_2 = Z_2 I_2 \tag{3}$$

• Step 3: Determine V_1 and I_3 using the following:

$$V_1 = -B_{12} V_2 - B_{13} V_3 \tag{4}$$

$$I_3 = B_{13} I_1 + B_{23} I_l \tag{5}$$

Note that I_1 and V_3 have specified values, since they correspond to current and voltage sources, respectively.

B. Cutset Method of Network Analysis

• Step 1: Solve the following for the vector V_b (note that V_b is the vector of voltages in the nonsource branches of T):

$$Y_b V_b = -Q_{11} I_1 - Q_{12} Y_2 Q_{22} V_b \tag{6}$$

where Y_2 is the admittance matrix of RLC elements and $Y_b = Q_{12} Y_2 Q_{12}$. Equation (6) is called the *cutset system of equations*.

• Step 2: Calculate V_2 using:

$$V_2 = Q_{12} V_b + Q_{22} V_3 \tag{7}$$

then,

$$I_2 = Y_2 Y_2 \tag{8}$$

• Step 3: Determine V_1 and I_3 using the following:

$$V_1 = Q_{21} V_b + Q_{21} V_3 \tag{9}$$

$$I_3 = -Q_{21} I_1 - Q_{22} I_2 \tag{10}$$

Note that I_1 and V_3 have specified values, since they correspond to current and voltage sources.

This completes the cutset method of network analysis. Next we illustrate the loop and cutset methods of analysis on the network shown in Fig. 9. The graph of the network is shown Fig. 9b. We choose the spanning tree T consisting of edges 4, 5, and 6. Note that T contains the voltage source and has no current source. The fundamental circuit and the fundamental cutset matrices with respect to T are given below in the required partioned form:

$$B_f = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \left[\begin{array}{c|cccccc} 1 & 0 & 0 & -1 & -1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 & -1 & 0 \end{array}\right] \end{array}$$

$$Q_f = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \left[\begin{array}{c|cccccc} 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ \hline -1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

This completes the cutset method of network analysis.

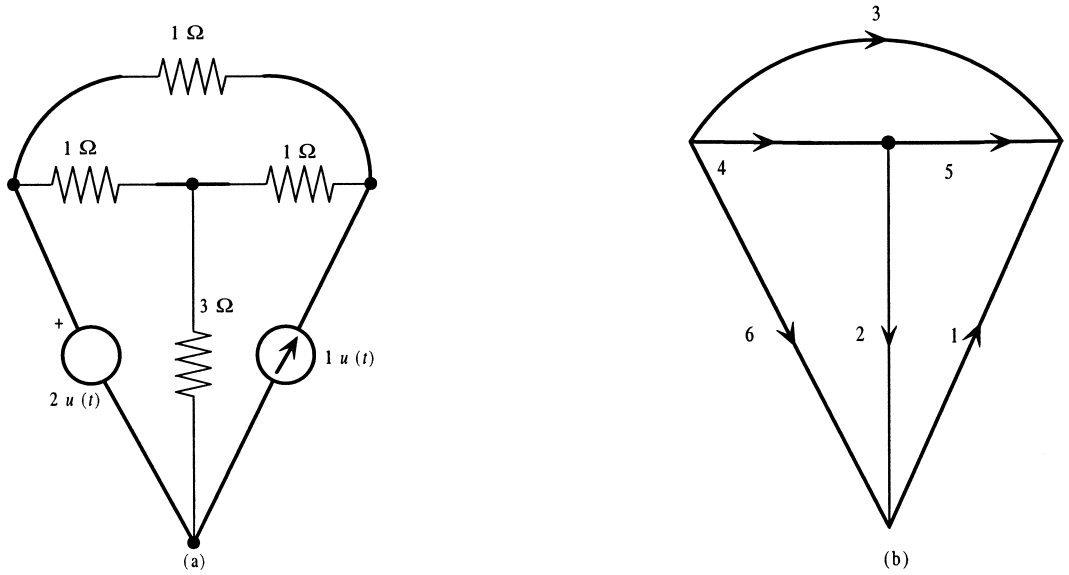


FIGURE 9 A network and its graph.

From these matrices we get:

$$B_{12} = \begin{bmatrix} 0 & 0 & -1 & -1 \end{bmatrix}$$

$$B_{13} = [1]$$

$$B_{22} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix}$$

$$B_{23} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$Q_{11} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$Q_{12} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$Q_{21} = [-1]$$

$$Q_{22} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

We also have

$$Z_2 = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Y_2 = \begin{bmatrix} 1/3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$v_6 = 2 \text{ volts}$$

$$i_1 = 1 \text{ ampere}$$

C. Loop Method Example

Edges 2 and 3 are nonsource chords. So,

$$I_l = \begin{bmatrix} i_2 \\ i_3 \end{bmatrix}$$

Substituting

$$Z_l = B_{22} Z_2 B_{22}^t = \begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix}$$

in Eq. (1), we get the following loop system of equations:

$$\begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

Solving for i_2 and i_3, we get:

$$I_l = \begin{bmatrix} i_2 \\ i_3 \end{bmatrix} = 1/11 \begin{bmatrix} 7 \\ -5 \end{bmatrix}$$

Using Eq. (2), we get:

$$I_2 = \begin{bmatrix} i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix} = 1/11 \begin{bmatrix} 7 \\ -5 \\ 1 \\ -6 \end{bmatrix}$$

Then, using $V_2 = Z_2 I_2$, we get:

$$V_2 = \begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = 1/11 \begin{bmatrix} 21 \\ -5 \\ 1 \\ -6 \end{bmatrix}$$

Finally, from Eqs. (4) and (5) we get:

$$V_1 = [v_1] = -27/11$$

$$I_3 = [i_6] = 4/11$$

D. Cutset Method Example

Edges 4 and 5 are the nonsource branches, so:

$$V_b = \begin{bmatrix} v_4 \\ v_5 \end{bmatrix}$$

Substituting

$$Y_b = \begin{bmatrix} 7/3 & 1 \\ 1 & 2 \end{bmatrix}$$

in Eq. (6), we get the following cutset system of equations:

$$Y_b V_b = \begin{bmatrix} 7/3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} v_4 \\ v_5 \end{bmatrix} = \begin{bmatrix} -1/3 \\ -1 \end{bmatrix}$$

Solving for V_b.

$$V_b = \begin{bmatrix} v_4 \\ v_5 \end{bmatrix} = 1/11 \begin{bmatrix} 1 \\ -6 \end{bmatrix}$$

From Eq. (7) we get:

$$V_2 = \begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = 1/11 \begin{bmatrix} 21 \\ -5 \\ 1 \\ -6 \end{bmatrix}$$

Using

$$I_2 = Y_2 V_2$$

we get:

$$I_2 = \begin{bmatrix} i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix} = 1/11 \begin{bmatrix} 7 \\ -5 \\ 1 \\ -6 \end{bmatrix}$$

Finally, using Eqs. (9) and (10),

$$V_1 = [v_1] = -27/11$$

$$I_3 = [i_6] = 4/11$$

This completes our illustration of the loop and cutset methods of circuit analysis.

Suppose a network N has no independent voltage sources. Then a convenient description of N with the node voltages as independent variables can be obtained as follows. Let A be the incidence matrix of N with vertex v_r as reference. Let us also partition A as $A = [A_{11} \;\; A_{12}]$, where the columns of A_{11} and A_{12} correspond, respectively, to the *RCL* elements and current sources. If I_1 and I_2 denote the column vectors of *RCL* element currents and current source currents, then KCL equations for N become:

$$A_{11} = -A_{12} I_2$$

We also have

$$I_1 = Y_1 V_1$$

where V_1 is the column vector of voltages of *RCL* elements and Y_1 is the corresponding admittance matrix. Furthermore, by the node transformation we have:

$$V_1 = A_{11}^t V_n$$

where V_n is the column vector of node voltages. So, we get from the KCL equations:

$$(A_{11} Y_1 A_{11}^t) V_n = -A_{12} I_2$$

The above equations are called *node equations*. The matrix $A_{11} Y_1 A_{11}^t$ is called the *node admittance matrix* of N.

FURTHER READING

For a more comprehensive discussion of other developments in graph theoretic concepts, please consult Chen (1972, 2001), Swamy and Thulasiraman (1981), and Watanabe and Shinoda (1999). For a very good treatment of liner circuits and other releated references, see Balabanian and Bickart (1981). Mitra (1974) provides a very good early work on active networks, while Chua et al. (1987) and Hasler and Neirynck (1986) are good sources for nonlinear network theory.

SEE ALSO THE FOLLOWING ARTICLES

ANALOG-SIGNAL ELECTRONIC CIRCUITS • DIGITAL ELECTRONIC CIRCUITS • ELECTROMAGNETICS • GRAPH THEORY • KALMAN FILTERS AND NONLINEAR FILTERS • NETWORKS FOR DATA COMMUNICATION • POWER ELECTRONICS

BIBLIOGRAPHY

Balabanian, N., and Bickart, T. (1981). "Linear Network," Matrix, Cleveland, OH.

Chen, W.-K. (1972). "Applied Graph Theory," North-Holland, Amsterdam.

Chen, W.-K. (2001). *In* "Electrical Engineering Handbook," Academic Press, San Diego, CA.

Chua, L. O., Desoer, C. A., and Kuh, E. S. (1987). "Linear and Nonlinear Circuits," McGraw-Hill, New York.

Hasler, M., and Neirynck, J. (1986). "Nonlinear Circuits," Artech House, Norwood, MA.

Mitra, S. K. (1974). "Analysis and Synthesis of Linear Active Networks," Van Nostrand-Reinhold, New York.

Swamy, M. N. S. S., and Thulasiraman, K. (1981). "Graphs, Networks, and Algorithms," Wiley-Interscience.

Watanabe, H., and Shinoda, S. (1999). "Soul of circuit theory: a review on research activities of graphs and circuits in Japan," *IEEE Trans. Circuits and Systems* **45,** 86–94.



Digital Electronic Circuits

David J. Comer
Donald T. Comer
Brigham Young University

I. Introduction to Digital Electronic Circuits
II. Using Switches To Generate Binary Voltage Levels
III. Types of Digital Circuits

GLOSSARY

Boolean algebra A type of algebra used to analyze logical relationships and logic circuits.
Digital code A collection on n digital signals that comprise an n-bit code.
Digital signal A voltage or current that can exist only at a finite number of levels or values. In many digital systems, the number of levels is two.
Integrated circuit or IC A small piece or chip of solid semiconductor material on or in which has been formed electrical components such as transistors, resistors, and capacitors. These components are interconnected within the chip to create a functioning electronic circuit or system.
Microprocessor An integrated circuit that includes the central processing unit of a computer (arithmetic unit plus the control unit) on a single IC chip.
Truth table A table that specifies the output values of a logic circuit as a function of all possible input combinations.
Voltage-controlled switch An electronic circuit that produces a high or a low level output, depending on the level of the input signal.

DIGITAL ELECTRONIC CIRCUITS are designed to respond to one or more input signals that have a finite number of voltage or current levels and produce one or more output signals that have a finite number of output current or voltage levels.

I. INTRODUCTION TO DIGITAL ELECTRONIC CIRCUITS

While the digital electronic circuit or digital circuit is used in other devices, the digital computer is the major consumer of these circuits. A single personal computer may contain several million transistors and associated electrical components to create the digital circuits that execute the computer's instructions. Examples of digital computer circuits are the flip-flop and logic gates such as the OR gate, NOR gate, AND gate, and NAND gate. These digital

circuits are also used in noncomputer applications as are the monostable and astable multivibrators and other timing circuitry.

A. Digital Circuit Operation

The operation of digital circuits is predicated on the use of a finite number of voltage or current levels. The most common number of levels used is two, leading to the use of a binary or binary-related numbering system. A digital circuit may accept input signals with voltages that are near either the 0-V level or the 3-V level. The only two possible output levels are also approximately 0 V and 3 V. One of these voltage levels, often the high level, is associated with the binary number 1 and the other level is associated with the binary number 0. These digital circuits form the building blocks from which large digital systems such as computers can be constructed.

B. Digital Signals Compared to Analog Signals

Until the electronic digital computer was born, most electronic systems accepted analog input signals and generated analog output signals. These signals are continuous in time and can take on any value between the minimum level and the maximum level. The variation from one signal value to another is a smooth continuous change as shown in Fig. 1(a). The digital signal makes abrupt transitions from one level to another as shown in Fig. 1(b). Although there is a finite switching time of the digital circuit, the transition is almost instantaneous. The digital circuit is designed such that all acceptable input digital input signals will drive the output to either one of two levels. The output will never exist at any intermediate value between the two levels except during the very short transition from one level to the other.

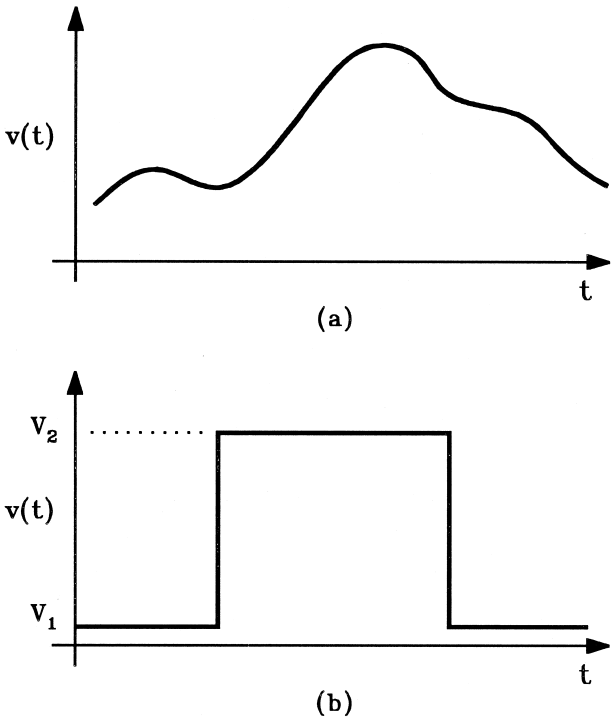


FIGURE 1 (a) Analog waveform. (b) Digital waveform.

In applications that require a signal to represent one of several numbers, the analog system can do so with much less circuitry. For example, an analog signal that is to represent numbers varying from 0 to 999 can be represented by 1000 different voltage levels. If the output signal can vary from a minimum of 0 to a maximum of 6 V, then each number will be represented by a voltage range of

$$\Delta V = \frac{6}{1000} = 0.006 \text{ V/number.} \quad (1)$$

The number 8 would correspond to a voltage that falls within the range 0.048 to 0.054 V.

While the output of a single analog circuit could represent 1000 different values, the very small voltage ranges for the different numbers would lead to accuracy problems. Most electronic circuits generate a small amount of electronic noise and some circuits pick up extraneous electronic noise from the electrical energy generated by communication circuits and other sources. Often this noise signal will exceed the voltage range of each number and cause the analog signal to be interpreted incorrectly. Furthermore, the transistors and other components used to construct circuits can cause the output voltage to depend on the temperature of the surroundings. The voltage change due to temperature may cause a shift in voltage greater than 0.006 V and the output number may again be misinterpreted.

The digital signal levels in a binary system may be 0 and 3 V. These levels are so widely separated that a small noise signal would cause no confusion in interpreting the signal level. Unfortunately, only two numbers can be represented by a single binary output. However, 10 binary outputs can be used to represent any binary number from 0 to 1023 by a digital code. Each output level can correspond to a different column of a 10-position binary number. The binary system is accurate and much more immune to noise problems, but requires more digital circuits than the analog system. Because integrated circuit technology can provide millions of circuits in a small volume, this disadvantage of digital circuits becomes less significant.

C. History of Digital Electronic Circuits

While mechanical or electromechanical computers were developed earlier, the first electronic digital computer was

constructed at the University of Pennsylvania between 1943 and 1946. This system, called the ENIAC (Electronic Numerical Integrator and Computer), contained over 18,000 vacuum tubes, weighed 30 tons, and occupied a 30- by 50-ft room. The size and power consumption were very large compared to the computers of today, but the worst factor in ENIAC's inefficiency was the unreliable operation of the vacuum tubes making up the digital circuits. During 1952, for example, approximately 19,000 vacuum tubes had to be replaced to keep the computer functional. Although reliability was a problem, ENIAC demonstrated the feasibility of the electronic computer.

The point-contact transistor was invented toward the end of 1947 and the junction transistor in 1951 at Bell Labs. The present bipolar junction transistor (BJT) differs only in geometrical details from the device developed in 1951. Because vacuum tubes were firmly entrenched as the electronic device of choice, other companies developed computers using this component in the early 1950s. Toward the latter part of the decade, computers based on the more reliable BJT began to appear. These systems were still somewhat bulky and quite expensive and were generally purchased only by larger companies or institutions.

By 1964, the integrated circuit (IC) became commercially available. These small digital circuits might include 10 to 20 transistors along with resistors and capacitors in a package similar in size to that of a single transistor. The transistors, resistors, and capacitors are created within or on a single piece of silicon. These digital circuits are referred to as small-scale integrated (SSI) circuits. As IC technology improved, the medium-scale integrated (MSI) circuit, then the large-scale integrated (LSI) circuit became available. The LSI chip may contain over 10,000 electronic components on a single chip.

With the capacity to create this number of components on a single chip, the first microprocessor was developed in 1971. This chip was based on a 4-bit binary number, but implemented the circuitry for the arithmetic unit and control unit of a digital computer on a single silicon chip using BJTs.

The 8-bit microprocessor was introduced in 1972. Within a few years, very-large-scale integration (VLSI) was achieved and 16 and 32-bit microprocessors were then developed. The larger systems are now generally based on the metal–oxide–silicon field-effect transistor (MOSFET). This device was invented in the early 1960s and has several advantages over the BJT. More devices can be placed in a given area of the chip and less power is required to operate the MOSFET. Ultra-large-scale integration (ULSI) now creates tens of millions of MOSFETs on a single chip, allowing the construction of very powerful personal computers.

While computers use a large percentage of the total number of digital circuits fabricated, the digital circuit is also important in communications and other areas. Cellular phones, high-definition television, and computer modems apply digital techniques in signal transmission. Guidance control of space vehicles and missiles rely on digital circuits. Electronic instruments and automotive applications such as anti-skid braking systems employ digital circuits for implementation.

II. USING SWITCHES TO GENERATE BINARY VOLTAGE LEVELS

A major effort in the digital electronics field is spent in developing devices that emulate the behavior of a voltage-controlled switch. In these devices, an input voltage or current controls the opening and closing of an output switch. These devices are used to create switching or digital circuits.

Two important elements in the realization of switches are the BJT and the MOSFET. Some of the factors involved in selecting a device relate to how closely the device emulates the switch, how small the device can be made, how fast the switch can change states, and how many supporting components are required.

A. The Ideal Switch

In digital circuits, a voltage-controlled switch is used to generate two different output voltage levels; one when the switch is closed, the other when the switch is open. Figure 2 shows a configuration to generate these two levels using an ideal manual switch.

When the switch is closed, the output terminal is connected directly to ground forcing an output voltage of 0 V. When the switch is opened, the voltage is pulled to V_1 through the load resistor.

B. The Nonideal Switch

Figure 3 shows a nonideal switch that has some low resistance in the closed or "on" condition rather than a short circuit. In the open or "off" condition, the resistance of this switch is large, but finite rather than infinite. The closed or on-resistance of the switch is designated R_{sc} and the open or off-resistance R_{so}.

The low voltage level of the circuit occurs when the switch is closed and is

$$V_{low} = V_1 \frac{R_{sc}}{R_{sc} + R_L}. \tag{2}$$

If R_{sc} is much smaller than R_L, this voltage will be near the 0-V level. The high voltage occurs when the switch is open and is

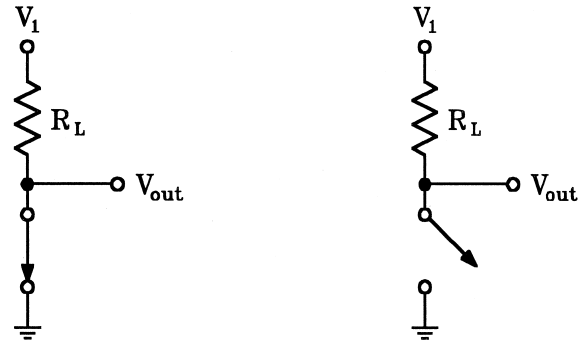


FIGURE 2 An ideal switch used to generate two voltage levels.

$$V_{high} = V_1 \frac{R_{so}}{R_{so} + R_L}. \tag{3}$$

If R_{so} is much larger than R_L, this voltage will be near the voltage V_1.

The nonideal switch can be made to behave much like the ideal switch if the on-resistance is low and the off-resistance is high.

Another method of generating two different voltage levels replaces the resistor of Fig. 2 with a second switch, as shown in Fig. 4. This configuration is called a complementary arrangement of switches. One switch will always be set to the open position when the other is closed and vice-versa.

For ideal switches it is easy to see that when the top switch is open and the lower switch is closed, the output voltage will be 0 V. When the lower switch opens and the upper switch closes, the output voltage will be equal to V_1.

For nonideal switches, these same results will be approximated, depending on the values of on-resistance and off-resistance of the switches. In fact, the complementary switch will generate voltage levels nearer 0 V and V_1 than the single switching circuit of Fig. 3 when nonideal switch resistances are present.

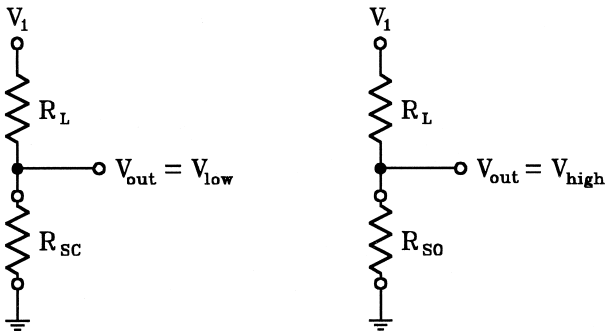


FIGURE 3 A nonideal switch.

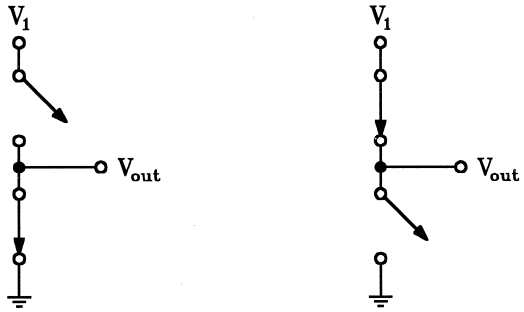


FIGURE 4 Complementary switch arrangement.

The two voltage levels are

$$V_{low} = V_1 \frac{R_{sc}}{R_{sc} + R_{so}} \tag{4}$$

and

$$V_{high} = V_1 \frac{R_{so}}{R_{so} + R_{sc}}. \tag{5}$$

If the open switch resistance is a large factor, perhaps 10,000, times the closed switch resistance, the lower level will be almost zero and the higher level will be almost equal to V_1.

C. Voltage-Controlled Switches

In digital circuits, the switches are replaced by high-speed electronic devices that can switch between a high resistance state and a low resistance state under control of an input signal. The two types of semiconductor switches used today are the BJT and the MOSFET. In the 1960s and 1970s, the BJT was often used in a configuration that emulates the single switch with a resistive load. The MOSFET is often used in a configuration that emulates the complementary switching behavior of the two switches in Fig. 4. These are called complementary MOS or CMOS circuits. These switching circuits form the basis of the digital circuits that are used to construct computers and other digital systems.

At the present time, CMOS circuits are by far the most commonly used in microcomputers. This is primarily due to the huge number of MOS devices that can be fabricated on a single chip, lowering the cost of the chip. The continually improving performance achieved with CMOS circuits also contributes to the popularity of the MOS device.

1. The BJT Switch

The bipolar junction transistor or BJT is manufactured in two different arrangements. One is called the npn device,

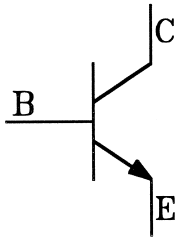


FIGURE 5 The symbol for an npn transistor.

the other is called the pnp device. The npn device is used most often in digital logic circuits.

The BJT consists of three different physical regions: the base, the emitter, and the collector as represented by the symbol for an npn device shown in Fig. 5. Each of the three distinct regions is electrically connected to a terminal, allowing connection of the regions to other circuit components. The emitter terminal is identified by an arrowhead that indicates the only possible direction of current flow for the npn device. The base terminal is the control for the device. If no voltage is applied from base to emitter, no current will flow from collector to emitter. If sufficient current is directed into the base terminal, leading to a voltage drop from base to emitter, a large current will flow from collector to emitter. These conditions allow the BJT to function as a controllable switch. The switch appears between the collector and emitter terminals. This switch is open if no current is applied to the base terminal and is closed when enough current is caused to flow into the base.

The BJT can operate in three different regions, the cutoff region, the active region, or the saturation region. For many switching applications, the device is either in the cutoff or saturation regions for most of the time with only short durations in the active region as a transition between the other two regions is made. Figure 6 shows an npn transistor in a common switching circuit configuration.

A high level voltage at the input saturates the transistor and the collector to emitter path approximates a closed switch. The output voltage is approximately zero. A low voltage level at the input leads to an approximate open switch between collector and emitter. The output is now at a voltage of V_1. This circuit is a voltage-controlled switch, emulating the configuration of Fig. 3, that is used in many different digital circuit applications.

Early BJT logic circuits such as the resistor–transistor–logic family (RTL) employed this basic configuration with little modification. Present day logic families such as transistor–transistor logic (TTL) and emitter-coupled logic (ECL) modify the basic configuration to speed the switching operation.

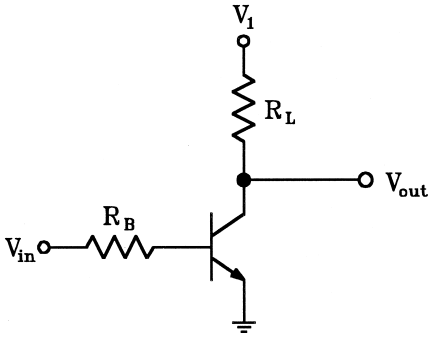


FIGURE 6 A switching circuit using an npn BJT.

2. The MOSFET or CMOS Switch

The MOS device consists of four geometric regions called the gate, the source, the drain, and the substrate. A channel can exist between the source and drain regions. This device can be fabricated as an n-channel (nMOS) device or as a p-channel (pMOS) device. The symbols for both devices are shown in Fig. 7. The substrate is often tied to the source terminal in simple digital circuits, thus the substrate is not shown in the symbols of Fig. 7.

The resistance from drain to collector is controlled by the voltage applied between gate and source terminals. For the nMOS device, a zero or low value of V_{GS} causes a very large resistance to exist between source and drain. When V_{GS} exceeds some relatively small voltage called the threshold voltage, the resistance between source and drain becomes much smaller. For the pMOS device, a negative value of V_{GS} must be applied to lead to a low resistance between source and drain terminals. A very small negative value of V_{GS} leads to a high resistance between source and drain.

The MOS device approximates a switch better than the BJT does in the off state. It does not approximate a switch as closely as the BJT does in the on state. The resistance between source and drain in the off state may be in the range of hundreds of megohms for the MOS device. The resistance in the on state may be a few hundred to a few thousand Ohms for the MOSFET.

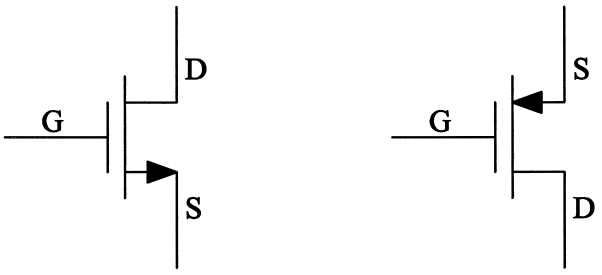


FIGURE 7 Symbols for the MOSFET: (a) nMOSFET; (b) pMOSFET.

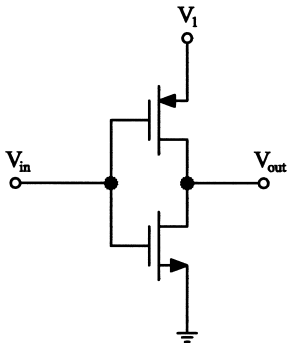


FIGURE 8 A complementary MOSFET (CMOS) switch.

The MOSFET is forced into the off state when the magnitude of the gate to source voltage is less than the threshold voltage of the device. This value is determined in the construction process and may be 0.8 V for the nMOS device and −0.8 V for the pMOS device. When the gate to source voltage for the nMOS device exceeds the threshold voltage, current flows from drain to source and the device is in the on state.

The basic switch configuration of the MOSFET switch corresponds to the complementary arrangement shown in Fig. 4. A pMOS device is used for the upper switch and an nMOS device is used for the lower switch as shown in Fig. 8. This is called a CMOS switch.

When the input voltage is at 0 V, the nMOS device is in the off state since the gate to source voltage is 0 V. The pMOS device has $-V_1$ V from gate to source and this device is in the on state. The output voltage is at V_1 V. This state is depicted in Fig. 9.

If the gate voltage changes to a high voltage, say V_1 V, the nMOS device will turn on and the pMOS device will turn off. This leads to the situation depicted in Fig. 10.

The key concept in digital BJT or CMOS circuits is that of a voltage controlled switch. The state of the switch or switches, as in the case of the CMOS circuit, can be controlled to generate either a high or low output voltage. The gate voltage controls the state of the CMOS circuit and draws no current at low and intermediate frequencies.

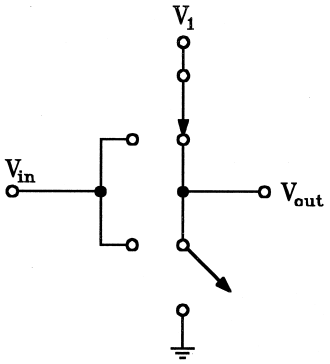


FIGURE 9 Model of the CMOS circuit for $V_{in} = 0$ V.

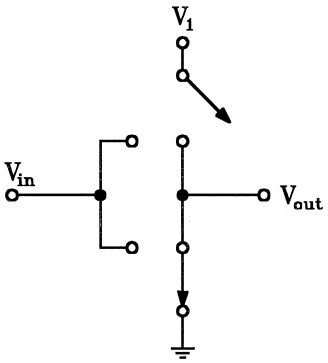


FIGURE 10 Model of the CMOS circuit for $V_{in} = V_1$ V.

III. TYPES OF DIGITAL CIRCUITS

The simple switching circuits discussed in the preceding paragraphs form the basis of many useful digital circuits. The digital computer applies digital circuits as building blocks to construct the subsystems that make up the computer.

An important group of circuits for the computer are called logic circuits. A simple logic circuit accepts several binary inputs and produces a binary output that depends on the combination of the binary input bits. Circuits such as inverters, AND gates, OR gates, NAND gates, and NOR gates are classified as combinational logic circuits. In addition, the bistable flip-flop circuit is also used in computer systems, often to act as a memory to store a binary number or bit. This circuit forms the basis of a sequential digital system. The outputs of n logic circuits can be grouped together to represent a digital code. In computers, these codes may represent numbers in some binary-related code or they may represent letters in some code such as the ASCII code.

Digital circuits are also used in noncomputer applications, for example, as controllers. These systems produce control signals that trigger certain events, based on the input signals and the time duration from some reference signal. One popular version of a controller is the digital state machine that is composed of logic gates and flip-flops.

Other digital circuits are used in a computer as well as in noncomputing digital systems. The monostable and astable multivibrators are examples of digital circuits used in both types of digital systems.

A. Digital Circuits for Computing Applications

1. Inverters

Essentially every personal computer now made is based on CMOS or closely related logic circuits. Computer chips are fabricated with tens of millions of MOSFET devices

on each chip. These devices now operate at high switching speeds and with very little power consumption. For these reasons CMOS logic circuits have become the logic family of choice for personal computers and other digital systems. Consequently, BJT logic circuits such as transistor–transistor logic (TTL), emitter-coupled logic (ECL), and current-mode logic (CML) are not discussed here even though they are still used in several applications.

The CMOS switching circuit shown in Fig. 8 is often called an inverter. If the input voltage is low, the nMOS device is in cutoff and the pMOS device is in the conducting state with a low resistance between drain and source. The output voltage will be at the high voltage level. When the input voltage level moves positive to the high level, the pMOS device shuts off while the nMOS device now has a small resistance between drain and source. The output voltage is at 0 V for this input condition.

It has been mentioned earlier that the CMOS switch or inverter has a rail-to-rail swing at the output. For a 3-V supply, the high output level of the inverter is 3 V and the low level output is 0 V. This significant result guarantees that any circuit driven by the output of the inverter will have a high input level of 3 V and a low input level of 0 V. There is no degradation of voltage level through a CMOS inverter stage. This same result also applies to the CMOS NAND and NOR gates to be considered next.

2. Logic Gates

A key concept related to circuits is the use of Boolean variables to express inputs and outputs of logic gates. It is conventional to express logic operations in terms of Boolean variables and manipulate these expressions with Boolean algebra. For example, the two-input OR gate of Fig. 11 uses the variables A and B to represent inputs and X to represent the output. While the actual voltage levels of an input may take on the values of 0 and 3 V at different times, we associate the high voltage level with the binary number 1 and the low level with binary 0, assuming a positive logic convention. The inputs and output then take on values of 0 or 1.

The Boolean algebra expression used to define an OR gate is

$$X = A + B. \tag{6}$$

The " + " sign denotes the "OR" operation rather than the usual addition. This equation is read, "The output X is true (or equal to 1) when either input A OR input B is true (or equal to 1).

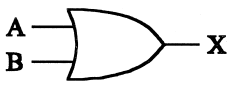


FIGURE 11 Symbol for an OR gate.

The Boolean algebra expression used to define an AND gate is

$$X = A \cdot B \tag{7}$$

The "·" sign denotes the "AND" operation rather than the usual multiplication. This equation is read, "The output X is true (or equal to 1) when both input A AND input B are true (or equal to 1). Often the "·" symbol is deleted between the variables in the AND expression. There can be more than two inputs to a logic gate.

The output of a gate can be inverted (complemented) to provide a different logic function. If the output of the OR gate is inverted, it becomes a NOR gate. If the output of the AND gate is inverted, it becomes a NAND gate. The corresponding Boolean expressions for these gates are

$$X = \overline{A + B} \tag{8}$$

and

$$X = \overline{A \cdot B}. \tag{9}$$

a. The NOR gate. A truth table is often used to specify the output of a logic gate for all possible combinations of input signals. The truth table for a two-input NOR gate is

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

When either input is a logic 1, the output will be 0. The only combination of inputs that results in an output of 1 is $A = 0$ and $B = 0$.

The two-input NOR gate can be implemented in CMOS as shown in Fig. 12. The general logic symbol for the NOR gate is also indicated in this figure. The small circle at the output represents an inversion following the OR gate to create the NOR gate.

When both inputs are at 0 V or logic 0, both nMOS devices are off and both pMOS devices are on. This input condition leads to 0-V gate-to-source voltages for both M1 and M2 and -3-V gate-to-source voltages for M3 and M4. The output voltage will be very near 3 V or logic 1. Typical values of on resistance and off resistance are 10 kΩ and 200 MΩ, respectively.

When one or both inputs switch to logic 1 or 3 V, at least one nMOS device turns on while at least one pMOS device turns off. In either case, a very high resistance exists between the power supply voltage and output and a very low

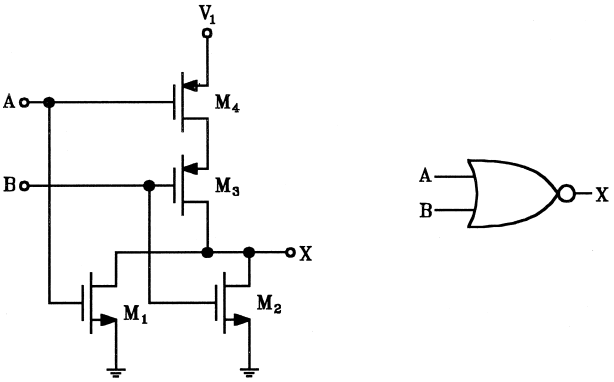


FIGURE 12 (a) Two-input CMOS NOR gate. (b) Symbol for the NOR gate.

resistance exists between output and ground. This forces the output to be at 0 V or logic 0.

The NOR gate can be extended to a higher number of inputs by adding a complementary pair of devices for each additional input. The additional pMOS device is placed in series with the other pMOS devices and the additional nMOS device is placed in parallel with the other nMOS devices.

b. The NAND gate. A two-input NAND gate satisfies the truth table

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

The logical expression for this gate is

$$X = \overline{A \cdot B} \tag{10}$$

For a positive logic system in which a high voltage level is defined as a logic 1, the NAND gate output is at the high level for all but one input combination. If both inputs are high, the output should be low. The CMOS gate of Fig. 13 satisfies this truth table. Also shown in the figure is the logic symbol for the NAND which is equivalent to an AND gate followed by an inversion.

For the combination of inputs $A = B = 1$, both M1 and M2 will be turned on and both M3 and M4 will be off. A very high impedance exists between the power supply and the output terminal while a low impedance exists between the output terminal and ground.

For any other input condition, the impedance from power supply to output will be low while the impedance from output to ground will be high.

The NOR gate and NAND gate make up a set of gates that are said to be functionally complete. This means that any possible combinational logic function can be realized using nothing more than these two types of gates.

3. Flip-Flops

The circuits considered in the preceding paragraphs are combinational logic circuits. These systems have outputs that depend only on the applied input signals. For a given set of inputs, the outputs will always be the same. Combinational circuits are important, but so is another class of digital circuits referred to as sequential circuits.

A sequential circuit depends not only on the present inputs but also on past history of the inputs and time. The basis of sequential circuits is the flip-flop. The flip-flop is perhaps the most important single type of circuit in the digital field. The flip-flop belongs to a class of circuits called bistable multivibrators.

Flip-flops operate in one of two modes: direct or clocked. Direct-mode flip-flops respond directly to applied inputs. The outputs change as a direct result of the inputs. In clocked flip-flops, a change of input has no effect on the output until a clock signal is applied. When a clock transition from one voltage level to another occurs, the output changes to a value dictated by the input levels. The SR flip-flop and gated D flip-flop are examples of direct-mode operation devices. The toggle, clocked D, and JK flip-flops are clocked devices. Some clocked flip-flops also have direct inputs, allowing operation in either mode.

a. The direct input SR flip-flop. This form of flip-flop has two input lines and either one or two outputs. The output line that is always present is often labeled Q. Generally, a second output called $\bar{Q}$ will also be present. One input line is used to set the device to the $Q = 1$ state while the other input sets the device to the $Q = 0$ state. These inputs are called the SET or S input and the RESET or R input; hence the name SR flip-flop. The outputs Q

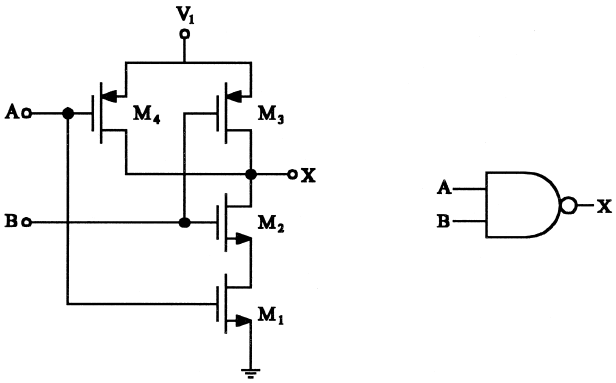
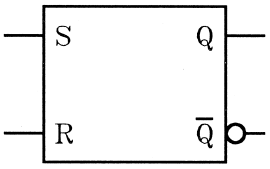


FIGURE 13 (a) Two-input CMOS NAND gate. (b) Symbol for the NAND gate.



S	R	Q	$\overline{Q}$
0	0	Q_0	$\overline{Q}_0$
0	1	0	1
1	0	1	0
1	1	0	0

FIGURE 14 The SR flip-flop and characteristic table.

and $\bar{Q}$ make up what is called a double-rail output; that is, $\bar{Q}$ always equals the complement of Q except when both S and R are asserted simultaneously. This condition is normally avoided in actual circuit operation. The symbol for the SR flip-flop is shown in Fig. 14 along with the characteristic table. The notation Q_o and $\bar{Q}_o$ refers to the states of outputs Q and $\bar{Q}$ before the input conditions are applied.

From the characteristic table we see that Q may equal 0 or 1 for the input condition of $S = R = 0$, depending on the input conditions prior to setting both inputs low. This is a distinct difference from combinational circuits which always produce a unique output for a given set of inputs.

When both S and R are asserted, Q and $\bar{Q}$ are driven low. If both S and R are returned simultaneously to the low level, it is impossible to determine if Q will assume the 1 state or the 0 state. Generally the input condition of $S = R = 1$ is avoided unless the following input condition is either $S = 1$ and $R = 0$ or $S = 0$ and $R = 1$. In either of these cases, the output will be uniquely determined.

The SR flip-flop can be constructed from NOR gates as shown in Fig. 15. For this latch, applying a 1 on S (high level) and a 0 on R results in $\bar{Q}$ being asserted low. Since $\bar{Q}$ drives an input to the upper NOR gate, both inputs are low, resulting in a high value for Q. When S returns to a low value, the high level of Q continues to drive the lower NOR gate to keep $\bar{Q}$ at a low level. If R is now asserted, the upper NOR gate drives Q low which now allows $\bar{Q}$ to return to a high level.

The SR flip-flop can function as a 1-bit memory. To store a 1 in this device, the S input is asserted. This input can then be deasserted, and the flip-flop remains in the $Q = 1$ state as long as the two inputs remain deasserted. The 0 bit can be stored by asserting R. Several SR flip-flops can be combined to form a storage register for several bits.

b. The clocked JK flip-flop. This bistable circuit has two gating inputs along with a clock input. The voltage level of the gates determine the output state to which the clock input will shift the flip-flop. Often direct set and reset inputs are provided that can override the clock input. For clocked operation the R and S inputs are set to the inactive level; but if a certain output condition is to be preset into the flip-flop before clocked operation takes place, these inputs can be used. The J and K inputs are called gates.

In the clocked mode of operation, the gating conditions are summarized in Fig. 16. The condition of both J and K low is called the "no change" condition. The toggle condition corresponds to both J and K high. The set condition occurs when $J = 1$ and $K = 0$ while the reset condition occurs when $J = 0$ and $K = 1$. It must be emphasized that the voltage levels applied to the J and K gates do not directly cause output changes. Levels on these gates determine to what level the clock transition drives the output. Changes on the J and K inputs cannot cause a change in Q if no clock pulse is applied.

There is a similarity between the function table of Fig. 16 and the SR table of Fig. 14. One obvious difference is the condition of $J = K =$ high leads to the toggling of the JK whereas the condition of $S = R =$ high should be avoided for the SR flip-flop. A more subtle difference is that the SR flip-flop operates in direct response to the S

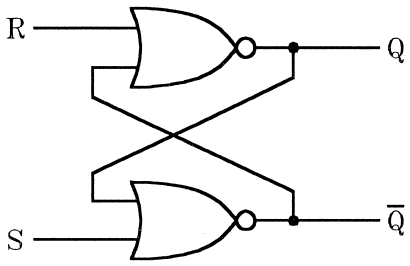
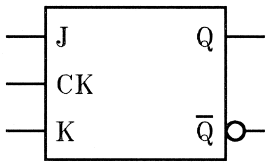


FIGURE 15 NOR gate configuration of an SR flip-flop.



S	R	Q	$\overline{Q}$	Condition
0	0	Q_0	$\overline{Q}_0$	No Change
0	1	0	1	Reset
1	0	1	0	Set
1	1	$\overline{Q}_0$	Q_0	Toggle

FIGURE 16 Gating conditions for a clocked JK flip-flop.

and R inputs while the levels on J and K determine the state that will exist after a clock transition occurs.

4. Frequency Limitations on Digital Circuits

A very significant factor in digital logic circuit performance is switching speed. Shorter switching times allow the execution of more operations per second by the computer. At the circuit level, rise and fall times and propagation delay times are measures of the speed of an individual circuit. The rise time is measured by applying a very abrupt input that will drive the output to the high voltage level. The time taken for the output to move from 10 to 90% of the total transition voltage is the rise time. The fall time measures the time to cross this same voltage interval when the output is driven from its high level to the low level. The propagation delay time is the time difference between the input transition and the output transition. It is measured between the 50% points of input and output transitions.

Although not the best measure of execution speed, the speed of the master clock of a computer gives an indication of the switching times required by the internal logic circuits. For a logic operation to be performed in one clock period, the clock cycle duration must exceed the propagation delay time and the rise time of the circuit performing the operation. By the year 2000, personal computer clocks reached frequencies of 1 GHz, improving from the low MHz range in two decades.

The switching speed of a circuit is limited by the parasitic capacitances associated with the transistors and connecting leads. The control of current flow through a BJT or a MOSFET is accomplished by means of a control voltage. As this voltage changes, a change in electronic charge takes place at various points within the device. Since capacitance can be defined as

$$C = \frac{dQ}{dV}, \tag{11}$$

these charge effects can be represented or modeled by capacitors. These capacitors are called parasitic because they are inadvertent results of the fabrication of the devices. In addition, the interconnections between devices on a chip and between chips introduce capacitance that also limits switching speeds.

In general, as MOSFET device size becomes smaller, the parasitic capacitance becomes smaller and higher switching speeds are possible. For years, as photolithographic technology improved to allow smaller devices, computer clock speeds continued to increase.

Another method of minimizing parasitic capacitances applies principles of physics rather than size reductions. The bipolar heterojunction transistor uses different but structurally similar materials, such as germanium and silicon, to decrease parasitic capacitance. Operation of these devices in the tens of GHz range was accomplished by the year 2000.

5. High Frequency Digital Circuits

Various configurations and devices have been developed to decrease the switching times of digital circuits. Most personal computers now use dynamic MOSFET circuits to achieve higher operating speeds. These circuits represent the two binary digits by the presence or absence of charge on a capacitor. To minimize the time to deposit or remove charge, only the small parasitic capacitance associated with the MOS devices is used.

In BJT high speed design using silicon BJTs, both emitter-coupled logic (ECL) and current mode logic (CML) are configurations that are popular. An improved device is the heterojunction BJT that allows very high speed operation in supercomputers.

B. Digital Circuits for Noncomputing Applications

Two important circuits used in timing applications are the monostable multivibrator and the astable multivibrator. The monostable multivibrator generates an output pulse of predetermined duration in response to a short trigger pulse at the input. This duration is set by the values of certain timing components in the circuit. The astable multivibrator produces a rectangular waveform at the output and requires no input signal. The frequency of the output signal is determined by the values of certain timing components in the circuit.

The astable is often referred to as a free-running multivibrator or as a clock circuit and the monostable is called a one-shot circuit. The flip-flop discussed earlier is classified as a bistable multivibrator. The basic difference among the types of multivibrator is the number of stable states associated with each type. The bistable has two stable states, the monostable has one stable state (and one quasi-stable state), and the astable has no stable states (two quasi-stable states). The waveforms of Fig. 17 show typical input and output signals that might be expected from the astable and the one-shot multivibrators.

The output of the one-shot remains at a specific level while in the stable state until an input signal initiates the quasi-stable state. The output voltage switches to another level in this state and remains at that level until the output reverts back to the original value. The time that the circuit remains in the quasi-stable state is called the period of the one-shot and is determined by circuit components. In the bistable and monostable multivibrators, the output signal levels are independent of the amplitudes of the input

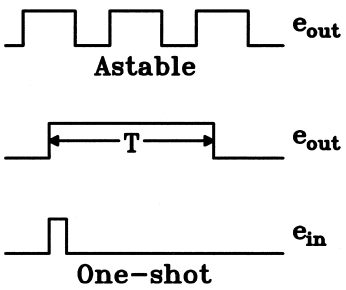


FIGURE 17 One-shot and astable waveforms.

signals, assuming these amplitudes are large enough to initiate switching.

The astable requires no input while the output continually switches between the two quasi-stable states. The length of time spent at each voltage level is determined by certain components of the circuit.

The one-shot circuit is used to create a pulse with a specific width in response to the input trigger signal. The pulse width may be as small as a few nanoseconds or as long as a thousand seconds. Such timed pulses are important in the control of machines or appliances.

The astable output waveform is used as a clock signal in low accuracy applications. For more precise applications, crystal controlled astables or oscillators are used.

SEE ALSO THE FOLLOWING ARTICLES

ANALOG ELECTRICAL FILTERS • ANALOG-SIGNAL ELECTRONIC CIRCUITS • BOOLEAN ALGEBRA • CIRCUIT THEORY • DIGITAL FILTERS • MATHEMATICAL LOGIC

BIBLIOGRAPHY

Comer, D. J. (1995). "Digital Logic and State Machine Design," (3rd edition, Saunders College Publishing, New York.

Martin, K. (2000). "Digital Integrated Circuit Design," Oxford University Press, New York.

Rabaey, J. M. (1996). "Digital Integrated Circuits," Prentice Hall, Upper Saddle River, NJ.

Uyemura, J. P. (2000). "A First Course in Digital Systems Design," Brooks/Cole, Pacific Grove, CA.

Wakerly, J. F. (2000). "Digital Design Principles and Practice," (3rd edition), Prentice Hall, Upper Saddle River, NJ.



Digital Filters

N. K. Bose
Pennsylvania State University

I. Introduction
II. Wave Digital Filters
III. LSI Filtering of Multidimensional Discrete Filters
IV. Velocity Filtering
V. Filter Banks
VI. Conclusions

GLOSSARY

Filter banks Digital multirate filter banks, frequently used for compression of speech, image, and video signals, contain analysis and synthesis blocks for filtering, performing sampling rate conversion by decimation and interpolation, as well as coding, transmission, and decoding.

Multidimensional LSI filters Linear shift-invariant (LSI) digital filters for processing spatiotemporal signals in wavenumber–frequency space as well as for separating signals propagating with different velocities. Such types of filters are used, respectively, in image or video signal processing and in geophysical (including seismic) signal processing and span a variety of applications. Linear shift-varying as well as well as nonlinear filters (including the well-documented polynomial filters) can also be multidimensional.

Wave digital filter Low-sensitivity digital filter obtained from a doubly terminated analog passive prototype by replacing each type of analog network element by an appropriate type of digital realization.

Wavenumber Spatiotemporal angular frequency vector whose components are the variables in the Fourier transform of a spatiotemporal signal.

Velocity filter Filter capable of discriminating between signals collected by an array of sensors based upon their velocities of propagation.

I. INTRODUCTION

This article documents the present status of the topic of digital filters which, over the last three decades developed from a research discipline into an essential technology, spurred by the incredible advances in microminiaturization that facilitated the implementation of this digital signal processing workhorse in an expanding horizon of applications. Digital filters are now ubiquitous in the processing of spatial and temporal signals of diverse physical origin like in radio astronomy, seismology, bioengineering, speech and audio, image and video. The demands of the information age and the accompanying challenges for reliable, robust, high-speed communication of multimedia signals that include text, data, sound, image and video between remote locations assure the

continuance of our dependence on digital filter technology well into the future.

The term, "digital filters," describing software or hardware designed to extract information from degraded data, appears in the title of several texts (Antoniou, 1993; Bose, 1993a; Hamming, 1989) and the subject it covers is treated extensively in works on digital signal processing. (Mitra and Kaiser, 1993; Bose and Rao, 1993). The heavy interest in this area relates to its importance in the attempt to attain robust, reliable, high-speed transmission and reception of information-bearing multimedia signals corrupted by noise or otherwise degraded during passage between multiusers and facilities. The increased sharing of spectrum and the need for multiplexing translate into a higher likelihood of user interference, and to accommodate a large number of users in the available spectrum, techniques for interference rejection become necessary. The design of an optimum filter for minimizing a chosen error criterion requires a priori information about the statistics of the data to be processed and an adaptive filter is needed when only partial knowledge of the signal characteristics is available. Interference-rejection techniques often need to be adapative because of the changing nature of interference and channel. The signals, weak or strong, could be spatial as well as temporal and may be of diverse physical origin.

The domain of application of digital filtering has expanded at a very rapid rate and the present and future needs of signal processing for communications in defense, surveillance, health care, and consumer electronics provide significant challenges to the objective of developing theory and practice, spurred by progress in microminiaturization and computing capabilities. For example, digital camera images often suffer from color noise and artifacts, which can be removed by filter plug-ins. The task of digital television bandwidth reduction involves two major data compression approaches: lossless (involving redundancy removal), which can compress most images in the range of 2:1 or 3:1, and lossy (discarding some information with reduced quality at higher compression), which can compress moving images greater than 100:1. Transform coding by discrete cosine transform (DCT), employed in the JPEG standard for still images, suffers from blocking effects due to the 8-by-8 block size at, say over 16:1 compression (because of the short basis functions for reconstruction). The blockiness is especially percieved in the blue and red channels in the flat regions of an image. On the other hand, artifacts like "mosquito noise" are conspicuous in areas of image with lots of detail. Most of the luminance information is carried in the green channel, while the noise and artifacts that are produced in digital cameras and JPEG compression occur primarily in the chrominance information that is dominant in the blue and red channels. After converting the RGB color images via RGB/YIQ transformation to chrominance (I, Q) and luminance components (Y), the chrominance is filtered so that the luminance details can be transferred to the blue and red channels, and then the image is converted back to RGB via the YIQ/RGB transformation. The so-called Quantum Mechanic filter uses an adaptive median-like filter (nonlinear) for removing large amounts of noise without destroying thin color details. Subband coding using wavelets (tree-structured filter banks) avoids blocking at medium bit rates because basis functions have variable lengths; at low bit rate, however, high wavenumbers (textures) are removed, causing ringing.

II. WAVE DIGITAL FILTERS

Considerable attention has been given to the problem of realizing low-sensitivity digital filters. As a rule of thumb, the simpler the realization procedure, the poorer the sensitivity of filter performance to elemental variations due, say, to finite arithmetic constraints like coefficient quantization with a specified number of bits. This is justified from the simultaneous increasing complexity of procedures for realization and improved passband sensitivity as one proceeds from the direct to the cascade and parallel structures culminating in the lattice-ladder realization of a digital filter transfer function. Fettweis (1971) considered the possibility of departure from the traditional digital domain approaches and showed how the richly documented synthesis techniques for analog circuits that can be used to generate a prototype analog filter structure could be transformed to a corresponding digital filter structure capable of capturing the low-sensitivity characteristics of the passive analog prototype. Expectation of low sensitivity to changes in multiplier coefficients was a major motivation for developing the principles of wave digital filtering.

A digital filter generated from an analog filter by the appropriate nesting of multiport filters (with associated port resistance at each port) called adaptors (realizable either in parallel or series form), along with elements (which could be one-port delays, reciprocal two-port unit elements, as well as quasi-reciprocal lines) and sources, are referred to as wave digital filters, motivated by the theory of wave propagation along a transmission line. The adaptors are built with adders and multipliers. A unconstrained (i.e., the port resistances are not constrained) series or a parallel n-port adaptor requires $n-1$ multipliers and $3n-3$ adders, while a constrained (made reflection-free at one port by appropriately relating the port resistances) adaptor rquires $n-2$ multipliers and $3n-5$ adders. The adaptors, which are the basic building blocks of wave digital filters, require some care in interconnection akin to that needed in the matching of resistors for maximum power transfer

to a resistively terminated lossless two-port from a source with resistance.

The property of low sensitivity in the passband of a generic filter structure composed of a lossless two-port between a source with resistance and a resistive termination is expected to be carried over to a digital filter derived from the generic prototype. This expectation was confirmed by Fettweis (1972), who introduced the concepts of pseudopassivity and pseudolosslessness for digital filters as counterparts of the properties of passivity and losslessness associated with classical analog filter theory. The variety of distinct topologies in classical filter synthesis leads to several classes of digital filter structures that keep intact the excellent passband sensitivity characteristics of the analog doubly terminated lossless reference filters. Furthermore, the good stopband insensitivity to parameter variation of analog ladder filters motivated particular attention to this subclass of generic passive filter structures. It was shown that any LC ladder filter structure can be very simply translated into a corresponding digital structure where the number of multipliers exceeds the number of inductors and capacitors in the doubly terminated lossless two-port by one (Sedlmeyer and Fettweis, 1973). The adaptors used in the realization are three-ports, all unconstrained except one. Thus, the number of multipliers could be significantly larger than the order of the ladder filter, whose series or shunt arms are designed to realize the appropriate transmission zeros of the filter type. In wave digital filters designed from the symmetric lattice reference filters, the stopband insensitivity is poorer than ladder filters but the number of multipliers does not exceed the filter order and the passband insensitivity may even be better than for ladder structures. The wave digital counterpart of lattice-ladder filters is also possible. An excellent review of other aspects of wave digital filter theory and principles not mentioned here is available (Fettweis, 1996).

Before concluding this section, it is important to mention that a properly designed wave digital filter is free of zero-input limit cycles, does not exhibit zero-input parasitic oscillations even under looped conditions, offers satisfactory sensitivity properties in passbands as well as stopbands, and has good dynamic range. The principles of wave digital filtering generalize naturally to multidimensions, where, however, the absence of general synthesis techniques makes design of reference filters much more difficult in comparison to the one-dimensional case.

III. LSI FILTERING OF MULTIDIMENSIONAL DISCRETE FILTERS

Digital filtering is used to process discrete data obtained either from sampling continuous signals or in some other manner. Its range of application is extensive, including the processing of geophysical, biomedical, pictorial, sonar, and radar data. Most of the discussion in this section applies to multidimensional problems, though for the sake of brevity in exposition, the 2D case is emphasized.

Various strategies exist to sample band-limited multidimensional signals. Petersen and Middleton (1962) showed that rectangular sampling is a special case of a more general sampling strategy by which a band-limited waveform is sampled on a nonorthogonal sampling raster. Another special case of this general strategy is discussed in depth by Dudgeon and Mersereau (1984), who point out the advantages of hexagonal sampling. For more information on the subject of rectangular sampling along with reconstruction formulas for recovery of the analog signal from its discrete samples, see Zayed (1993).

The filtering of the discrete spatiotemporal multidimensional signals by linear shift-invariant (LSI) systems, defined next, will be considered in this section; the mathematical tools needed for the purpose will first be briefly summarized. For additional details, see Bose (1982).

A. Mathematical Tools

The notations and definitions, given for the 2D case for brevity, are directly generalizable to n dimensions. The reader should have no difficulty in extracting the corresponding 1D counterparts (Bose, 1993b).

Definition 1. A 2D discrete system whose input/output relationship $y[k_1, k_2] = T[x[k_1, k_2]]$ is characterized by the operator $T[\cdot]$ is said to be linear if and only if for arbitrary inputs $x_1[k_1, k_2]$ and $x_2[k_1, k_2]$ and arbitrary complex constants c_1 and c_2,

$$\begin{aligned} &T[c_1 x_1[k_1, k_2] + c_2 x_2[k_1, k_2]] \\ &\quad = c_1 T[x_1[k_1, k_2]] + c_2 T[x_2[k_1, k_2]]. \end{aligned} \tag{1}$$

The system characterized by $T[\cdot]$ is shift-invariant if and only if

$$y[k_1 - k_0, k_2 - l_0] = T[x[k_1 - k_0, k_2 - l_0]] \tag{2}$$

for all $x[k_1, k_2]$, with k_0, l_0 arbitrary integers.

A system satisfying both the above properties is linear shift-invariant (LSI).

The mathematical tool used in the study of multidimensional LSI systems is the multidimensional z-transform. In the case of 2D systems, the z-transform of a sequence $\{x[k_1, k_2\}$ is defined to be

$$\begin{aligned} Z[x[k_1, k_2]] &\triangleq X(z_1, z_2) \\ &\triangleq \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2] z_1^{-k_1} z_2^{-k_2}. \end{aligned} \tag{3}$$

An alternative definition replaces $z_1^{-k_1} z_2^{-k_2}$ with $w_1^{k_1} w_2^{k_2}$ in (3), but no essential conceptual difference arises due to these two possibilities. The expository advantage of working with the complex variables $w_1 \triangleq z_1^{-1}$ and $w_2 \triangleq z_2^{-1}$ is context dependent. In combinatorial studies, $X(z_1, z_2)$ is referred to as a generating function. The values of z_1, z_2 for which the double summation in (3) converges absolutely constitute the region of convergence, referred to as the *Reinhardt domain*. This domain is completely specified by the magnitudes $|z_1|, |z_2|$, respectively, of the complex variables z_1, z_2,

The inversion formula associated with (3) is

$$x[k_1, k_2] = \frac{1}{(2\pi j)^2} \oint_{C_1} \oint_{C_2} X(z_1, z_2) z_1^{-1-k_1} z_2^{-1-k_2} \, dz_1 \, dz_2. \tag{4}$$

In (4), the closed contours C_1, C_2 must be in the region of convergence of $X(z_1, z_2)$ in (1). The Reinhardt domain must be specified before the sequence $x[k_1, k_2]$ can be uniquely calculated from its z-transform $X(z_1, z_2)$ as in the one-dimensional case.

Any 2D LSI system is completely specified by its impulse response

$$h[k_1, k_2] = T[\delta[k_1, k_2]],$$

where the 2D unit impulse function is

$$\delta[k_1, k_2] = \begin{cases} 1 & \text{if} \quad k_1 = k_2 = 0, \\ 0 & \text{otherwise.} \end{cases}$$

The output $y[k_1, k_2]$ of any 2D LSI system with impulse response sequence $\{h[k_1, k_2]\}$ and specified input $\{x[i_1, i_2]\}$ is given by the 2D discrete convolution

$$y[k_1, k_2] = \sum_{i_1=-\infty}^{\infty} \sum_{i_2=-\infty}^{\infty} x[i_1, i_2] h[k_1 - i_1, k_2 - i_2]. \tag{5}$$

The input–output relationship of a 2D LSI system is defined by a difference equation of the form

$$\sum_{\substack{i_1 \\ (i_1, i_2) \in I_b}} \sum_{i_2} b[i_1, i_2] y[k_1 - i_1, k_2 - i_2] = \sum_{\substack{i_1 \\ (i_1, i_2) \in I_a}} \sum_{i_2} a[i_1, i_2] x[k_1 - i_1, k_2 - i_2], \tag{6}$$

where I_b (output mask) and I_a (input mask) denote, respectively, the finite-arearegions of support for arrays $\{b[i_1, i_2]\}$ and $\{a[i_1, i_2]\}$. With $b[0, 0] = 1$, we can write (6) as

$$y[k_1, k_2] = \sum_{\substack{i_1 \\ (i_1, i_2) \in I_a}} \sum_{i_2} a[i_1, i_2] x[k_1 - i_1, k_2 - i_2] - \sum_{\substack{i_1 \\ (i_1, i_2) \in I_b \\ (i_1, i_2) \neq (0,0)}} \sum_{i_2} y[k_1 - i_1, k_2 - i_2]. \tag{7}$$

B. Two-Dimensional FIR Filter Principles

Finite impulse response (FIR) digital filters are extensively used in one-dimensional and multidimensional signal processing because of their inherent stability and linear phase (subject to appropriate symmetrization) properties. The importance of phase is especially underscored in multidimensions (Huang *et al.*, 1975). The respective generalizations of one-dimensional design techniques (Bose, 1993a; Hamming, 1989) are briefly summarized next.

1. Windowing Scheme

Perhaps the simplest and most straightforward design technique is the windowing technique, which is a judicious mixture of art and science. Various windows have been discovered and multidimensional design often exploits the documented one-dimensional case. Huang (1972) related the design of 2D and 1D windows in the following manner.

Fact: If $\omega(x)$ is a good symmetrical 1D window, then $\omega(\sqrt{x_1^2 + x_2^2})$ is a good circularly symmetrical 2D window.

The approximation is especially good if the Fourier transform of the window bisequence (this transform is called the window wavenumber response) has a support which is much smaller than the extent of band-limitedness of the low-pass, 2D, circularly symmetric filter wavenumber response. Let the ideal 2D, circularly symmetric wavenumber response be defined by

$$H(\omega_1, \omega_2) = \begin{cases} 1 & \omega_1^2 + \omega_2^2 \leq B^2, \\ 0 & \text{otherwise.} \end{cases}$$

Let the Fourier transform of the 2D window $\omega[x_1, x_2]$ used to truncate the 2D impulse response, $h[x_1, x_2]$, be $W(\omega_1, \omega_2)$. Then, the wavenumber response of the truncated bisequence, denoted by $f[x_1, x_2] = h[x_1, x_2] w[x_1, x_2]$, is the 2D convolution

$$F(\omega_1, \omega_2) = H(\omega_1, \omega_2) ** W(\omega_1, \omega_2).$$

Clearly, if the "width" of $W(\omega_1, \omega_2)$ is much smaller than the "width" of $H(\omega_1, \omega_2)$, then $F(\omega_1, \omega_2)$ becomes

essentially the convolution of $W(\omega_1, \omega_2)$ and a 2D step function. Therefore, if

$$\omega[x_1, x_2] = \omega\left(\sqrt{x_1^2 + x_2^2}\right),$$

then $F(\omega_1, \omega_2)$ is approximately of the functional form $F(\sqrt{\omega_1^2 + \omega_2^2})$.

2. Frequency-Sampling Scheme

In one dimension the Lagrange formula (Bose, 1993a) gives a polynomial $P(z)$ of degree n which interpolates over $n+1$ distinct points $\{z_0, z_1, \ldots, z_n\}$. Specifically, if the frequency samples at $\{z_0, z_1, \ldots, z_n\}$ are specified by $P(z_k) = w_k, k = 0, 1, \ldots, n$, then

$$P(z) = \sum_{k=0}^{n} w_k \ell_k(z),$$

where

$$\ell_k(z) = \frac{\prod_{i \neq k}(z - z_i)}{\prod_{i \neq k}(z_k - z_i)}.$$

In 2D a similar interpolation is possible. Let $\{z_0, z_1, \ldots, z_n\}$ be $n+1$ distinct points. Let $\{w_0, w_1, \ldots, w_m\}$ be a second such set of $m+1$ points. Set (Davis, 1975)

$$P(z) = \prod_{i=0}^{n}(z - z_i),$$

$$Q(w) = \prod_{j=0}^{m}(w - w_j),$$

$$P_j(z) = \frac{P(z)}{(z - z_j)},$$

$$Q_k(w) = \frac{Q(w)}{(w - w_k)}.$$

The $(m+1)(n+1)$ polynomials

$$\ell_{jk}(z, w) = \frac{P_j(z) Q_k(w)}{P_j(z_j) Q_k(w_k)}$$

satisfy

$$\ell_{jk}(z_r, w_s) = \delta_{jr}\delta_{ks}.$$

Hence,

$$P(z, w) = \sum_{j=0}^{n}\sum_{k=0}^{m} \mu_{jk}\ell_{jk}(z, w)$$

is a bivariate polynomial of total degree $\leq m+n$ which satisfies the $(m+1)(n+1)$ interpolation conditions

$$P(z_j, w_k) = \mu_{jk}, \quad j = 0, 1, \ldots, n; \quad k = 0, 1, \ldots, m.$$

Thus, frequency sampling 2D FIR filters can be designed (Bose, 1982, pp. 253–255).

3. Minimax FIR Filters

The Parks–McClellan method for optimal (in the minimax sense) FIR filter design is based on the validity of a univalence condition referred to as the Haar condition (Bose, 1993a). This condition provides the uniqueness of the best approximation, a useful criterion (for convergence) in iterative implementation via the Remez exchange algorithm (Bose, 1993a). The Haar condition does not hold, in general, in the 2D case. Let the monomials in two variables z_1, z_2 be given as follows: $P_0(z_1, z_2) = 1$, $P_1(z_1, z_2) = z_1$, $P_2(z_1, z_2) = z_2$, $P_3(z_1, z_2) = z_1^2$, $P_4(z_1, z_2) = z_1 z_2$, $P_5(z_1, z_2) = z_2^2$, $P_6(z_1, z_2) = z_1^3, \ldots$. It is not always possible, having been given n arbitrary distinct points (z_{1i}, z_{2j}), to find a linear combination of $P_0, \ldots, P_{n-1}$ that takes on preassigned values at these points (Davis, 1975, p. 27).

4. Multidimensional FIR Filter via Transformation

To exploit the procedures richly documented for 1D digital filter design in the corresponding 2D case, a type of 1D-to-2D frequency/wavenumber transformation was proposed by McClellan (1973). This transformation maps a 1D zero-phase frequency response to a 2D frequency (wavenumber) respose and is suitable for the design and implementation of 2D digital filters whose wavenumber response magnitudes approximate either the low-pass circularly symmetric or fan characteristics (Dudgeon and Mersereau, 1984). The circularly symmetric low-pass design transformation was generalized to multidimensions through a simple but clever change of variables by Fettweis (1977).

C. Two-Dimensional IIR Filter Principles

The high implementational complexity of FIR filters is especially felt for those with sharp roll-off frequency (wavenumber) characteristics. The transition width of a 1D FIR filter is inversely proportional to the length of its impulse response sequence. For 2D FIR filters that approximate a diamond-shaped magnitude of wavenumber response, the filter support size is believed to be inversely proportional to the square of the transition width (Carrai *et al.*, 1994). The advantage of achieving reduction in computational complexity of implementation via IIR filtering is offset, however, by the difficulties with stabilization, nonlinearity of the phase response, and various problems encountered in implementation methods. Recursibility issues also have to be dealt with. In spite of

these and other drawbacks a number of design techniques have been developed in both the traditional wavenumber domain as well as in the spatial domain (where local state-space models are used). Global state-space models for multidimensional filters are, in general, infinite dimensional. Only the important subclass of *recognizable* rational filter transfer functions (where the denominator polynomial is expressible as a product of univariate polynomials) have finite-dimensional global state-space models.

The wavenumber-domain IIR design methods can be grouped into two categories, those based on spectral transformations and those based on computer-aided optimization including iterative schemes. In the former approach a 2D IIR filter is designed from a 1D (or even a 2D) prototype (which may be either a stable analog or a stable digital filter) by transformations that besides keeping the stability property invariant also succeed in achieving the specified wavenumber response characteristics. In the computer-aided optimization approaches, a nonlinear optimization procedure iteratively adjusts the filter transfer function parameters untill a specified error criterion is minimized. Details on the various techniques are available in standard texts (e.g., Dudgeon and Mersereau, 1984; Bose, 1982).

The support of the impulse response of a recursive 2D digital filter need not be confined to a quarter-plane. In fact, wavenumber responses of quarter-plane filters are quite restricted (Bose, 1982). The condition for recursive computation of the output bisequence in Eq. (6) in response to a finite input bisequence can be met by cone (wedge) filters (to be described below) and the limiting case of cone (wedge) filters, namely half-plane filters. These filters are much more versatile than quarter-plane filters.

D. Recursibility

The conditions for recursibility in multidimensions have been discussed in sufficient generality (Dudgeon and Mersereau, 1984; Bose, 1982). Here the objective is to discuss important special cases which are not only easily understood and appreciated, but also bring out key features and methods in 2D recursive filter analysis such as unit impulse response computation of IIR filters and their stability. Consider the non-quarter-plane digital filter characterized by the transfer function

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{1}{1 + 0.5z_1z_2^{-1} + z_1^{-1} + z_2^{-1} + z_1^{-2}z_2}, \quad (8)$$

where $Y(z_1, z_2)$, and $X(z_1, z_2)$ are, respectively, the transforms of the output and input bisequences. It is easy to verify that the unit impulse response $\{h[k_1, k_2]\}$ is recursively computable with zero-boundary conditions from the 2D difference equation

$$h[k_1, k_2] = \delta[k_1, k_2] - \frac{1}{2}h[k_1 + 1, k_2 - 1] - h[k_1 - 1, k_2] - h[k_1, k_2 - 1] - h[k_1 - 2, k_2 + 1]. \quad (9)$$

The support, $S_{p,q,r,t} \triangleq \text{supp}\{h[k_1, k_2]\}$, of the impulse response is in the causality cone $H_{1,1,1,2}$ defined by the intersection of the two half-planes,

$$H_{1,1}\colon\ x_1 + x_2 \geq 0 \qquad \text{and} \qquad H_{1,2}\colon\ x_1 + 2x_2 \geq 0.$$

It is important to note that if $\mathcal{Z}$ denotes the set of integers and $\mathcal{Z}^2 \triangleq \mathcal{Z} \times \mathcal{Z}$, then there exists an index map

$$\phi\colon\ H_{p,q,r,t} \cap \mathcal{Z}^2 \rightarrow Q_1 \cap \mathcal{Z}^2$$

from the causality cone $H_{p,q,r,t} = H_{p,r} \cap H_{q,t}$ with support $S_{p,q,r,t}$ (p, q, r, t are nonnegative integers) to the first quadrant quarter-plane Q_1 with support $S_{1,0,0,1}$.

The map referred to is bijective, i.e., isomorphic (one-to-one and onto) and is described by the relation

$$[n_1, n_2] = \phi[k_1, k_2] = [pk_1 + rk_2, qk_1 + tk_2]. \quad (10)$$

The associated inverse map, with $pt - qr = 1$, is

$$[k_1, k_2] = \phi^{-1}[n_1, n_2] = [tn_1 - rn_2, -qn_1 + pn_2]. \quad (11)$$

Clearly,

$$\phi[t, -q] = [pt - qr, qt - qt] = [1, 0],$$
$$\phi[-r, p] = [-pr + pr, -qr + pt] = [0, 1],$$

which implies that points $[t, -q]$ and $[-r, p]$ in $H_{p,q,r,t}$ are mapped to points $[1, 0]$ and $[0, 1]$, respectively, in Q_1. The multidimensional counterpart of the 2D mapping and its inverse can be found in Bose (1985), Chapter 5.

The quarter-plane filter transfer function obtained from the image of the map is

$$\hat{H}(z_1, z_2) \triangleq \sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} h\left[\phi^{-1}[n_1, n_2]\right] z_1^{-n_1} z_2^{-n_2},$$

where $\phi^{-1}[n_1, n_2]$ is defined in (11). Then

$$\begin{aligned}\hat{H}[z_1, z_2] &= \sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} h[tn_1 - rn_2, -qn_1 + pn_2] z_1^{-n_1} z_2^{-n_2} \\ &= \sum_{\text{supp}\{h[k_1,k_2]\}}\sum h[k_1, k_2] z_1^{-(pk_1+rk_2)} z_2^{-(qk_1+tk_2)} \\ &= \sum_{\text{supp}\{h[k_1,k_2]\}}\sum h[k_1, k_2]\left(z_1^p z_2^q\right)^{-k_1}\left(z_1^r z_2^t\right)^{-k_2} \\ &= H\left(z_1^p z_2^q, z_1^r z_2^t\right).\end{aligned}$$

Notice that for notational convenience, below the z-transform involves positive powers of the indeterminates, $w_1 \triangleq z_1^{-1}$, $w_2 \triangleq z_2^{-1}$. Therefore,

$$\hat{H}[z_1, z_2] = \sum_{\text{supp}\{h[k_1,k_2]\}} \sum h[k_1, k_2]\left(z_1^p z_2^q\right)^{-k_1}\left(z_1^r z_2^t\right)^{-k_2}$$

$$= \sum_{\text{supp}\{h[k_1,k_2]\}} \sum h[k_1, k_2]\left(w_1^p w_2^q\right)^{k_1}\left(w_1^r w_2^t\right)^{k_2}.$$

The transfer function in the complex variables w_1, w_2 of the first quadrant quarter-plane filter obtained from $H(z_1, z_2)$ in (8) is

$$G[w_1, w_2] = \frac{1}{1 + 0.5z_1z_2^{-1} + z_1^{-1} + z_2^{-1} + z_1^{-2}z_2}\Bigg|_{\substack{z_1=w_1^{-1}w_2^{-1} \\ z_2=w_1^{-1}w_2^{-2}}}$$

$$= \frac{1}{1 + 0.5w_2 + w_1w_2 + w_1w_2^2 + w_1}.$$

Apply a first-quadrant quarter-plane stability test [see Chapter 3 in Bose (1982) for a comprehensive account] on $G(w_1, w_2)$, i.e., check whether or not $A(w_1, w_2) \neq 0$, $|w_1| \leq 1$, and $|w_2| \leq 1$, where $A(w_1, w_2) \triangleq 1 + 0.5w_2 + w_1 + w_1w_2 + w_1w_2^2$. Clearly $A(-1, 0) = 0$, which implies that $G(w_1, w_2)$ is BIBO unstable. Consequently, $H(z_1, z_2)$ in (8) is also a BIBO-unstable general support recursive digital filter. Indeed, from (9) it is straightforward to show that along the boundaries of the causality cone that provides the region of support of the filter,

$$h[-k, k] = \left(-\frac{1}{2}\right)^k, \qquad k \geq 0,$$

$$h[2k, -k] = (-1)^k, \qquad k \geq 0.$$

Subsequently, it is easy to show after some algebraic manipulation that

$$h[-k, k+1] = -\frac{3}{4}(k+1)\left(-\frac{1}{2}\right)^k, \qquad k \geq -1,$$

$$h[2k+1, -k] = \frac{1}{2}k(-1)^k, \qquad k \geq 0,$$

$$h[-k, k+2] = \left[-\frac{15}{2} + \frac{27}{4}(3+k) - \frac{9}{4}(3+k)^2\right] \times \left(-\frac{1}{2}\right)^{k+3}, \qquad k \geq -2,$$

$$h[-k, k+3] = \left[-\frac{27}{2} + \frac{153}{8}(4+k) - \frac{27}{4}(4+k)^2 + \frac{9}{8}(4+k)^3\right]\left(-\frac{1}{2}\right)^{k+4}, \qquad k \geq -3,$$

etc. Better still, a plot of the impulse response $h[k_1, k_2]$ obtained by setting up a recursivecomputation on the preceding set of equations shows how it builds up, especially in the first quadrant.

IV. VELOCITY FILTERING

A two-dimensional signal often exhibits orientation in space called direction. When a signal has components along a direction, its Fourier transform also contains high-magnitude components along a well-defined direction. The so-called fan filter can be used as a directional filter for discriminating between signals along different directions. In array signal processing, the directional property of signals can be interpreted as velocity. Velocity filtering is essentially unique to seismology, at least to date, since the velocity of propagation is not constant for all waves. In radar, radio astronomy, and to a large extent also in sonar, the velocity is essentially constant.

Seismic waveforms consist of two distnct components. One is a signal reflected from subsurface formations and the other is a surface signal often referred to as ground roll. The information about subterranean deposits like oil and hydrocarbon in sedimentation layers is embedded in the reflected component, whereas the ground roll component transmitted along the surface of the earth is undesirable since it degrades the reflected signal of interest. The surface wave has a lower velocity of propagation than the reflected wave and can be targeted for attenuation by filtering (see Bose, 1993a, pp. 480–481).

Velocity filters provide a means for discriminating signals from noise or other undesired signals because of their different apparent (move-out) velocities. Thus, the use of velocity filters makes it possible selectively to enhance or attenuate signals occupying the same frequency ranges as noise or undesired signals. Velocity filtering can be performed by multidimensional filters included in the array processing scheme.

A. Signal and Noise Characteristics

Apparent velocity of arriving signals depends upon the wave type, source location, and structure beneath the recording site.

(a) Teleseismic signals: These are assumed to propagate coherently across the array. Observed apparent velocities usually lie above 8 km/sec (P-waves) and above 4 km/sec (S-waves). P-waves show predominant frequencies around 1 Hz and S-waves usually have lower peak frequency.
(b) Microseismic noise: This propagates with velocities from about 2.5 to 4 km/sec. Dominant frequencies occupy a broad low-frequency range. Depending

upon the distance to the source and sensor spacing, some portions of microseismic noise may propagate coherently across the array.

(c) Incoherent noise: Various human activities (factories, traffic, construction), action of wind, smaller water basins, etc., generate high-frequency noise above 1 Hz. Corresponding apparent velocities vary for different sources, but in general have low values, around 1 km/sec. For sensor spacing larger than several kilometers, it is most likely that the high-frequency noise propagates incoherently across the array, irrespective of whether the location of the noise source is inside or outside the array pattern.

(d) Signal-generated noise: This is noise generated by the desired signal itself in the vicinity of the sensor as multiple reflections and mode conversions. It has a wide range of apparent velocities, depending upon the local structure, and it may propagate coherently across the array.

B. One-Dimensional Arrays for Velocity Filtering

In Fig. 1, P-waves with apparent velocities in the range from 8 km/sec to infinity occupy the triangular zone labeled "signal zone." All kinds of coherent noise with apparent velocities between 2.5 and 4 km/sec are distributed within the wedge marked "coherent noise."

Consider a 1D array and horizontal wave propagation parallel to the line of sensors. An adequate separation of the signal from the ambient noise may be achieved by velocity filtering, which is implemented via fan filters. For a fan filter the pass and rejection zones become defined as straight lines through the origin in the frequency–wavenumber, $(f,\ k)$, plane. One of the major applications of fan filters is in directional filtering, where the filter can pass the signals along prescribed directions and reject

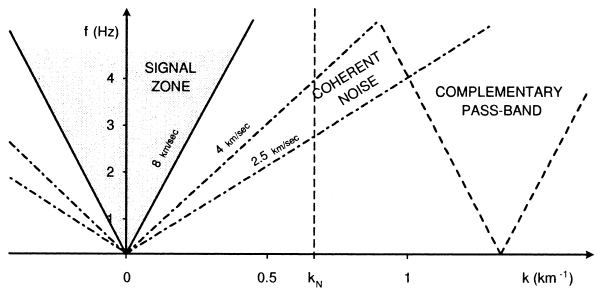


FIGURE 1 Idealized distribution of the signal and noise components in the frequency versus wavenumber plane. Space sampling period $\Delta x = 0.8$ km and time sampling period $\Delta t = 0.1$ sec are typical.

along others. Besides sampling there are no other limitations upon frequency and wavenumber intervals. Provided that the coherent noise occupies a region in the $(f,\ k)$ plane that differs from that of the signal, the velocity filtering performs a perfect discrimination against the coherent noise.

Assume that it is desired to pass waveforms with wavenumbers within the range $-|f|/V \le k \le |f|/V$. Outside this wavenumber range all waveforms are rejected. The 2D transfer function is then defined as

$$H(f,k) = \begin{cases} 1 & -|f|/V \le k \le |f|/V \\ 0 & \text{elsewhere.} \end{cases} \tag{12}$$

The time–space impulse response of the filter may be expressed in terms of the inverse 2D Fourier transform of $H(f,\ k)$,

$$h(t,x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(f,k)\ \exp \jmath 2\pi(ft - kx)\,df\,dk. \tag{13}$$

Due to the periodicity of $H(f,k)$ in wavenumber, we must limit ourselves to the resolvable wavenumber band only:

$$-k_N \le k \le k_N. \tag{14}$$

It follows from the basic relation

$$f = Vk \tag{15}$$

that for a given apparent velocity V and frequencies $|f| > Vk_N$ in the (f,k) plane, we leave the primary and enter the complementary pass zone. Therefore, the frequency limits for a meaningful transfer function are

$$-f_N \le f \le f_N, \tag{16}$$

where, for a spatial period Δx,

$$f_N = Vk_N = \frac{V}{2\Delta x}. \tag{17}$$

Having established the value of the folding frequency f_N, we find that the temporal sampling period Δt which preserves all information within the frequency band $(-f_N,\ f_N)$ becomes

$$\Delta t = \frac{1}{2f_N}. \tag{18}$$

The last two equations also show that the apparent cutoff velocity satisfies

$$V = \frac{f_N}{k_N} = \frac{\Delta x}{\Delta t}. \tag{19}$$

For any signal to pass through this velocity filter, the apparent velocity must be equal to or higher than V, i.e., for

a given sensor spacing the signal moveout τ must satisfy the condition

$$\tau \leq \Delta t. \tag{20}$$

Introducing the finite limits $\pm f_N$, $\pm k_N$, we have

$$h(t, x) = \int_{-f_N}^{f_N} \int_{-k_N}^{k_N} H(f, k) \exp \jmath 2\pi (ft - kx)\, df\, dk. \tag{21}$$

It may be expected intuitively that the sharpness between the pass and rejection zones increases and the amplitude of side lobes decreases with increasing number of array sensors.

C. Multidimensional Velocity Filters

Extensive documentation of design methods for multidimensional digital fan filters including the so-called cone and wedge filters is available in (Boll, 1992). For background information, see Dudgeon and Mersereau (1984).

V. FILTER BANKS

The journey from digital filters to banks of filters (filter banks) was motivated by applications that necessitated the decomposition of a discrete signal into frequency subbands (with the help of an analysis filter bank) prior to coding, transmission, and eventual reconstruction of the signal (with the help of a synthesis filter bank). The concept of subband decomposition originates from the expectation that if the input signal is decomposed into spectral bands,

(say, for topics of current interest like digital watermarking for copyright protection of images and even multimedia (Hartung and Kutter, 1999). The greater versatility of the filter bank can be further justified from the limitations inherent in a single filter. To wit, an IIR digital filter is unable to capture linearity of phase response, while a FIR filter cannot be made allpass. Thus, phase and amplitude cannot be manipulated simultaneously by either an FIR filter or an IIR filter. However, perfect reconstruction filter banks are capable of eliminating aliasing, phase, and amplitude distortions.

The idea of using multiband filter banks is crucial not only for encoding data from 1D signals like speech, but also including image and video. The generic analysis–synthesis system for perfect reconstruction consists of a bank of N filters each followed by a decimator or downsampler (with positive integer M as the downsampling factor in the uniform-filter-bank case) at the transmitting end and a corresponding bank of N synthesis filters each followed by an expander or upsampler (with upsampling factor M in the uniform case) at the receiving end. The maximally decimated or critically sampled case occurs when $N = M$ and the discussions, unless mentioned otherwise, pertain to this case. For more efficient implementation, the polyphase representation is generated from the generic structure. In the polyphase representation, decimation of the signal precedes filtering at the transmitting end and upsampling follows filtering at the receiving end. The conditions for perfect reconstruction (up to a delay) of the original signal is straightforward to derive in terms of the matrices characterizing the multiband filter banks. These conditions can even be satisfied with FIR filters that are guaranteed to be stable. When the FIR analysis polyphase matrix is restricted to be paraunitary, the perfect reconstruction (PR) condition for the polyphase representation is automatically satisfied and the synthesis filters are the time-reversed conjugate version of the analysis filters. The perfect reconstruction property may, however, be lost in practical subband coding systems, where the signals in each subband are quantized. The possibility of PR even when the coefficients and the results of multiplications in the various stages of design of ladder structures are quantized was first discussed by Bruekers and van den Emden (1992). The existence of structures that keep invariant the paraunitary property after coefficient quantization is also known (Vaidyanathan, 1990).

Generalizations as well as limitations of recent results which incorporate the perfect reconstruction as well as linear phase (LP) constraints in multidimensional multiband digital filter banks are discussed with several examples and counterexamples in a recent paper (Charoenlarpnopparut and Bose, 1999). The PR and LP properties in filter banks are important for the subband coding of images because the human visual system is known to be more robust to symmetric distortions. When the class of filter banks is restricted to satisfy the conditions for LP as well as PR, the problem is completely solved for the two-band n-dimensional case using Gröbner bases (Buchberger, 1985). Although the problem is satisfactorily solved for the multiband 1D case in Basu and Choi (1999) and the two-band multidimensional case in Charoenlarpnopparut and Bose (1999), the more general problem associated with the multidimensional multiband counterpart is still open.

VI. CONCLUSIONS

Building on the solid foundation provided by the available literature on the filtering of continuous-time and discrete-time one-dimensional (1-D) temporal signals, this article gives state-of-the-art information on wave digital filters that exploit our knowledge of classical passive synthesis

techniques in analog circuitry to offer digital simulations (accompanied with advantages of accuracy, robustness and dynamic range), multidimensional nonrecursive and recursive digital filters, and velocity discriminating filters (fan filters).

Digital filters are also used for time-frequency signal analysis and this has led to considerable research in the analysis and synthesis of digital filter banks. Multichannel filter banks have been used for signal compression, wavelet generation and related applications. This article provides a snapshot of the journey from filters to filter banks, the issues and implications of perfect reconstruction subject to constraints like linear phase and concludes with some frontiers of current research investigations in multidimensional filter banks and multidimensional wavelets.

The need for processing, coding, and transmitting multidimensional discrete signals in an expanding horizon of applications ensures the prominence of digital filters as an indispensable workhorse for electrical engineers. The emphasis placed on 2D digital filters is necessitated by the desired compromise between the extensive available documentation in the 1D case and the need for brevity in this presentation without sacrificing attention to some of the unique aspects of two- and higher dimensional digital filtering. Again, for the sake of brevity, attention here has been focused on linear shift-invariant filters. For an introduction to linear shift-variant multidimensional digital filters, see Bose (1985) [see in particular Valenzuela and Bose (1985), where other key references on the subject are available].

It is also important to point out that in spatiotemporal filtering, sometimes it is necessary to treat the temporal dimension in a manner different from the spatial dimensions. For example, in a baseball game the ball is thrown at a velocity higher than the motion of players. In that situation, a lowpass filter designed to preserve the details of the players and reduce noise would most likely have a cutoff frequency that would eliminate the baseball from the signal. Preservation of the baseball, however, would lead to unacceptable blurring of the players. To resolve the dilemma of temporal dependence on spatial variation, Dubois (1992) presented an approach to spatiotemporal filtering where the dimensions were treated unequally, unlike in conventional multidimensional filtering based, say, on least squares wavenumber domain-design.

Recent documentation of the theory for approximating magnitude only, group delay only, and both the magnitude and group delay of 1D as well as 2D recursive digital filters is available (Shenoi, 1999). There emphasis is placed on the designing of sharp-cutoff complexity digital filters for audio and image processing applications, where the need for approximating linear phase characteristics is crucial. Adaptive IIR filters remain as a potential tool for complexity reduction in several different areas of signal processing (Regalia, 1995). The most demanding problem in adaptive IIR filtering currently is related to convergence characterization. Adaptive IIR filters have been used as notch filters for fast tracking of sinusoidal inputs, channel equalization for wired and wireless communications, and short training interfering echo cancellation.

Nonlinear digital filters have been widely used during the last couple of decades. For research conducted before 1990 see Pitas and Venetsanopoulos (1990). Detailed documentation of polynomial filters with emphasis on quadratic filters is available (Sicuranza, 1992). A type of nonlinear filter called a median filter is very suitable for removing impulsive noise. Vector median filters (Astola *et al.*, 1990) have been used for color images, and multidimensional nonlinear filters have been suggested for suppressing impulsive noise from video sequences with low signal-to-noise ratio (SNR) while keeping intact information pertaining to fine details and textures.

ACKNOWLEDGMENTS

The writing of this article was completed when the author was visiting Ruhr Universitaet, Lehrstuhl fuer Nachrichtentechnik, Fakultaet fuer Elektroechnik, Universitaetsstrassee 150, 44780 Bochum, Germany as an Humboldt Awardee. The author is very grateful to Professor Alfred Fettweis for several useful discussions and for supplying some references.

SEE ALSO THE FOLLOWING ARTICLES

ANALOG ELECTRICAL FILTERS • CIRCUIT THEORY • DIGITAL ELECTRONIC CIRCUITS • IMAGE PROCESSING • SYSTEM THEORY

BIBLIOGRAPHY

Antoniou, A. (1993). "Digital Filters: Analysis, Design, and Applications," McGraw-Hill, New York.

Astola, J., Haavisto, P., and Neuvo, Y. (1990). "Vector median filters," *Proc. IEEE* **78,** 678–689.

Basu, S., and Choi, H. M. (1999). "Hermite reduction methods for generation of complete class of linear phase perfect reconstruction filter banks: Part I—Theory," *IEEE Trans. Circuit Syst. II Analog Digital Signal Processing* **46**(2), 434–448.

Boll, R. (1992). "Entwurfsverfahren für mehrdimensionale discrete Faecherfilter," Dr. Ing. Dissertation, Fakultät für Elektrotechnik, Ruhr-Universität, Bochum, Germany.

Bose, N. K. (1982). "Applied Multidimensional Systems Theory," Van Nostrand Reinhold, New York.

Bose, N. K. (ed.). (1985). "Multidimensional Systems Theory: Progress, Directions, and Open Problems," Reidel, Dordrecht, Holland.

Bose, N. K. (1993a). "Digital Filters: Theory and Applications," Krieger, Malabar, Florida.

Bose, N. K. (1993b). *In* "Handbook for Digital Signal Processing," (S. K. Mitra and J. F. Kaiser, eds.), Chapter 3, Wiley, Ney York.

Bose, N. K., and Rao, C. R. (eds.). (1993). "Signal Processing and Its Applications," North-Holland, Amsterdam.

Bruekers, F. A. M. L., and van den Emden, A. W. M. (1992). "New networks for perfect inversion and perfect reconstruction," *IEEE J. Select. Areas Commun*. **10,** 130–137.

Buchberger, B. (1985). *In* " Multidimensional Systems Theory: Progress, Directions, and Open Problems" (N. K. Bose, ed.), Chapter 6, Reidel, Dordrecht, Holland.

Carrai, P., Cortelazzo, G. M., and Mian, G. A. (1994). "Characteristics of minimax FIR filters for video interpolationdecimation," *IEEE Trans. Circuits and Syst. Video Technol.* **4,** 453–467.

Charoenlarpnopparut, C., and Bose, N. K. (1999). "Multidimensional FIR filter bank design using Gröbner bases," *IEEE Trans. Circuits Syst. II Analog Digital Signal Processing* **46,** 1475–1486.

Davis, P. J. (1975). "Interpolation and Approximation," Dover, New York.

Dubois, E. (1992). "Motion-compensated filtering of time-varying images" *Multidimensional Syst. Signal Processing*, **3,** 211–239.

Dudgeon, D. E., and Mersereau, R. M. (1984). "Multidimensional Digital Signal Processing," Prentice-Hall, Englewood Cliffs, NJ.

Fettweis, A. (1971). "Digital filter structures related to classical filter networks," *Arch. Elek. Uebertragung.* **25,** 79–89.

Fettweis, A. (1972). "Pseudopassivity, sensitivity, and stability of wave digital filters," *IEEE Trans. Circuit Theory* **19,** 668–673.

Fettweis, A. (1977). "Symmetry requirements for multidimensional digital filters," *Int. J. Circuit Theory Appl.* **5,** 343–353.

Fettweis, A. (1996). "Wave digital filters: Theory and practice," *Proc. IEEE* **74,** 270–327.

Hamming, R. W. (1989). "Digital Filters," Prentice-Hall, Englewood Cliffs, NJ.

Hartung, F., and Kutter, M. (1999). "Multimedia watermarking techniques," *Proc IEEE* **87,** 1079–1107.

Huang, T. S. (1972). "Two-dimensional windows," *IEEE Trans. Audio Electroacoustics* **20,** 88–89.

Huang, T. S., Burnett, J. V., and Deczky, A. G. (1975). "The importance of phase in image processing filters," *IEEE Trans. Acoustics Speech Signal Processing* **23,** (December).

McClellan, J. H. (1973). "The design of two-dimensional filters by transformations." *In* "Proceedings Seventh Annual Princeton Conference on Information Systems," pp. 247–251.

Mitra, S. K., and Kaiser, J. F. (eds.). (1993). "Handbook for Digital Signal Processing," Wiley, New York.

Petersen, D. P., and Middleton, D. (1962). "Sampling and reconstruction of wavenumber-limited functions for N-dimensional Euclidean spaces," *Information Control* **5,** 279–323.

Pitas, I., and Venetsanopoulos, A. (1990). "Nonlinear Digital Filters," Kluwer, Dordrecht, The Netherlands.

Regalia, P. A. (1995). "Adaptive IIR Filtering in Signal Processing and Control," Marcel Dekker, New York.

Sedlmeyer, A., and A. Fettweis, A. (1973). "Digital filters with true ladder configuration," *Int. J. Circuit Theory* **1,** 5–10.

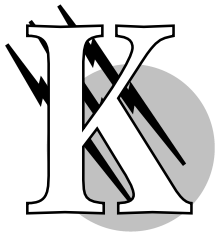
Shenoi, B. A. (1999). "Magnitude and Group Delay Approximation of 1-D and 2-D Digital Filters," Springer-Verlag, New York.

Sicuranza, G. L. (1992). "Quadratic filters for signal processing," *Proc. IEEE* **80,** 1263–1285.

Vaidyanathan, P. P. (1990). "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proc. IEEE* **78,** 56–93.

Valenzuela, H. M., and Bose, N. K. (1985). *In* "Multidimensional Systems Theory: Progress, Directions, and Open Problems" (N. K. Bose, ed.), Chapter 5, Reidel, Dordrecht, Holland.

Zayed, A. I. (1993). "Advances in Shannon's Sampling Theory," CRC Press, Boca Raton, FL.



Kalman Filters and Nonlinear Filters

Vytas B. Gylys
Texas Instruments Inc.

I. Introduction
II. Linear Estimation Problem and Kalman Filters
III. Modifications of Kalman Filters and Nonlinear Filtering
IV. Filter Implementation
V. Applications

GLOSSARY

Brownian motion process Process $\{x_t\}, t \geq 0$, is a (scalar) *constant-diffusion Brownian motion process* if (a) $\{x_t\}$ is a process with independent increments and (b) for any nonnegative t_1 and t_2, the increments $(x_{t_1} - x_{t_2})$ are Gaussian random variables with $E[(x_{t_1} - x_{t_2})] = 0$ and $E[(x_{t_1} - x_{t_2})^2] = q|t_1 - t_2|$, where $q > 0$.

Covariance function of a stochastic process If $\{x_t\}$ is a second-order, scalar- or vector-valued process (that is, if $E[x_t x_t^{\mathrm{T}}] < \infty$ for every t in the parameter space T_0), its covariance function is defined as

$$R(t, s) = E[x_t - Ex_t)(x_s - Ex_s)^{\mathrm{T}}]$$

for all t, s in the parameter set T_0. This function is also known as the autocovariance function of the process.

Estimation Process of observing the values taken on by random measurements and using these measurements to compute (with the assistance of an estimator) the values of some unknown parameters expressible as functions of measurements. In this article, the term *estimation* is used synonymously with *filtering*. See also Estimator.

Estimator An estimator $\hat{\theta}$ of some θ from a parameter space Θ is a function of random measurements $M_k \equiv \{m_0, m_1, \ldots, m_k\}$ (and hence itself is a random variable) constructed for inferring the unknown value of θ. The value taken on by the estimator of θ *after* the observation of a particular set of measurements is called an *estimate* of θ. To be consistent with the notational conventions set up for this article, we do not distinguish notationally between an estimator (a random variable) and a resulting estimate (a value taken on by this random variable as a function of actually observed measurements). The term *estimator* in this article also refers to a computational procedure (algorithm) for computing estimates. In general, it is used synonymously with the term *filter*.

Extended Kalman filter (EKF) Nonlinear filter (for a system with nonlinearities in the dynamics equation

or in the measurement equation of its model) that relinearizes at each measurement update time t_k the nonlinear parts of the system model, while using a vector-valued Taylor expansion around some reference trajectory $\{s_{0k}; k = 0, 1, \ldots\}$, and then applies Kalman filter equations. The reference trajectory used is obtained at each t_k from $\hat{s}_{k-1|k-1}$ (for the computation of $\hat{s}_{k|k-1}$ and $P_{k|k-1}$) and then from $\hat{s}_{k|k-1}$ (for the computation of $\hat{s}_{k|k}$ and $P_{k|k}$). Here, $\hat{s}_{k|j}$ and $P_{k|j}$, with $k \geq j$, denote the minimum-mean-square-error estimates of the state and the state error covariance matrix based on measurements $\{m_0, m_1, \ldots, m_j\}$. The discrete-time EKF algorithm is defined in terms of Eqs. (36)–(44) in the text. A slightly different version of the linearized extension of the Kalman filter is obtained by linearizing the model around some known trajectory. This version is often called a linearized Kalman filter (LKF).

Gaussian process A scalar- or vector-valued stochastic process $\{x_t\}$ is said to be Gaussian if for every finite subset $(t_1, t_2, \ldots, t_n)$ of points from the parameter set T_0, random variables $(x_{t_1}, x_{t_2}, \ldots, x_{t_n})$ are jointly Gaussian; that is, jointly they are distributed as a Gaussian random variable. It follows from the properties of Gaussian distribution that all finite-dimensional distributions of a Gaussian process $\{x_t\}$ are completely specified by its means $\bar{x}_t \equiv E[x_t]$ and its covariance functions $R(t, s) = E[(x_t - \bar{x}_t)(x_s - \bar{x}_s)^{\mathrm{T}}]$.

Linear system model Model in which both the stochastic differential (or difference) equations of system dynamics and the measurement equations relating the system state to the noisy measurements of the system are linear equations with respect to the state, measurement, and noise vector. The discrete-time linear system model, considered in this article for the Kalman filter and other related estimation schemes, is defined in the text in terms of Eqs. (2) and (3) and Assumptions 1D and 4D.

Markov process A process $\{x_t\}$ is a kth-order Markov process if for all $t_1 < t_2 < \cdots < t_n$ in the parameter set T_0 and some fixed integer k $(1 \leq k < n)$, the conditional probability $p[x_{t_n} < b_n \mid x_{t_1} = b_1, \ldots, x_{t_{n-1}} = b_{n-1}] = p[x_{t_n} < b_n \mid x_{t_{n-k}} = b_{n-k}, \ldots, x_{t_{n-1}} = b_{n-1}]$ holds for all n-tuples of real numbers $(b_1, b_2, \ldots, b_n)$. The forgetting-of-the-remote-past property of process $\{x_t\}$, defined by the above probability statement, is called the Markov property.

Minimum-mean-square-error (MMSE) estimator An estimator $\hat{\theta}$ of some parameter θ (fixed or random) from a parameter space is said to be a minimum-mean-square-error estimator of θ based (conditioned) on measurements $M_k \equiv \{m_0, m_1, \ldots, m_k\}$, if

$$E\left[(\hat{\theta} - \theta)(\hat{\theta} - \theta)^{\mathrm{T}} \mid M_k\right] \leq E\left[(\hat{\theta}_0 - \theta)(\hat{\theta}_0 - \theta)^{\mathrm{T}} \mid M_k\right],$$

where $\hat{\theta}_0$ is any other estimator of θ. If θ and M_k are two jointly distributed random variables, it can be shown that the MMSE estimator $\hat{\theta}$ must be of the form $E[\theta \mid M_k]$; that is, $\hat{\theta}$ is a conditional mean estimator.

Noise process Zero-mean stochastic process.

Positive definite and positive semidefinite matrices Matrix B is *positive definite* if and only if $\mathrm{x}^{\mathrm{T}}\mathrm{Bx} > 0$ for all nonzero vectors x of compatiable length; B is *positive semidefinite* if and only if $\mathrm{x}^{\mathrm{T}}\mathrm{Bx} \geq 0$ for all nonzero vectors x of compatible size.

Process with independent increments Process $\{x_t\}$ such that, for every ordered set

$$t_0 < t_1 < t_2 < \cdots < t_n$$

of parameter values from the parameter set T_0, the random variables $(x_{t_k} - x_{t_{k-1}})$ are mutually independent for $k = 1, 2, \ldots, n$.

Recursive estimation of system state Estimation scheme that does not require that all past measurements of the system and the past system state data be saved for estimation of the current state. To estimate the current system state. To estimate the current system state, a typical recursive estimator uses only the current measurements and the most recent estimate of the system state. In a more general case, a recursive estimator may use a finite, fixed-size set of most recent measurements and system state estimates. This property greatly facilitates the implementation of a recursive estimator for real-time operation on a digital computer. Nearly all filters considered in this article, including the Kalman filter, are recursive estimators.

Second-order process Scalar- or vector-valued stochastic process $\{x_t\}$ such that $E[x_t x_t^T] < \infty$ for every t in the parameter set T_0.

Square root filtering Collection of related computational algorithms for (measurement) updating, in a numerically stable manner, of estimates P of the state error covariance matrix in a Kalman filter and for computing the filter gain matrix. These algorithms utilize a result from matrix theory, according to which any nonnegative definite symmetric P can be factored as $P = SS^{\mathrm{T}}$, where S is usually a square matrix that is not necessarily nonnegative definite symmetric.

Stationary process A process $\{x_t\}$ is said to be stationary if for any set of values $t_1, t_2, \ldots, t_n$ from the parameter set T_0 and for any t_0 such that $t_j + t_0$ is in T_0 *for all* $j = 1, \ldots, n$, the joint distribution of $(x_{tj}, j = 1, 2, \ldots, n)$ is identical to that of $(x_{t_j + t_0}, j = 1, 2, \ldots, n)$.

System measurement *m* A vector that contains the values of system measurements, which are assumed to be noisy and associated with some instant of time. In this article, the letter m is reserved for the measurement vector. In particular, m_t or $m(t)$ refers to the measurement

vector m at time t; m_k refers to m at a discrete sampling epoch t_k. The measurement vector has two interpretations: Either it represents a scalar- or vector-valued random variable (RV) or it represents actually observed values taken on by this RV. Notationally we do not distinguish between these two meanings of m.

System measurements Sets of noisy measurements of the system. Each measurement set is associated with some instant or interval of time and contains the measured values, corrupted by random noise, of all (or of a selected subset of) observable system outputs. For example, if the sensor tracking an aircraft is a radar, the set of measurements obtained at the sampling epoch t_k may consist of the range and range-rate measurements at t_k.

System model (in recursive filtering) Mathematical specification of those characteristics of a system that pertain to the construction of a recursive filtering scheme for the system. Suppose that at any instant of time the state of a system is specified by a vector-valued quantity, known as the state vector. Then the model of such a system typically consists of (a) vector-valued stochastic differential (or difference) equations that specify the dynamics (evolution) of system state with respect to time, (b) vector-valued measurement equations that relate the system state to the noisy measurements of the system, and (c) probabilistic assumptions that specify the distributions of three stochastic components in the system model (the process noise term in system dynamics equations, the measurement noise term in measurement equations, and the initial distribution of state) and the probabilistic interdependence among these components. Often the process noise and the measurement noise terms additively enter the system dynamics and the measurement equations, respectively.

System state Complete, quantitative, typically time-dependent description of the system at some instant of time. What constitutes a complete description depends on the purpose for which the information about the system state is to be used. For example, if one is interested in predicting into the near-future the position and the velocity of a particle moving in a known gravitational field, it may suffice only to specify for the particle the current values of its position coordinates and its velocity and acceleration vectors relative to some suitable coordinate frame.

(System) state vector s Data array, usually structured as a vector, that contains the quantitative description of the system state at some time point. In this article, the letter s denotes the system state vector. In particular, s_t or $s(t)$ refers to the state vectors at time t; s_k refers to s at a discrete sampling epoch t_k. Note that the state vector has two interpretations: Either it represents a random vector [then it is a vector-valued random variable (RV)] or it represents a vector of specific values taken on by this RV. Notationally we do not distinguish between these two meanings of s.

Unbiased estimator An estimator $\hat{\theta}$ of some θ from a parameter space is said to be unbiased if $E[\hat{\theta}] = \theta$.

White noise process White process $\{x_t\}$ such that $E[x_t] = 0$ for all t.

White process Scalar- or vector-valued, wide-sense stationary stochastic process $\{x_t\}$ with $E[x_t] = \bar{x}$ for all t and the covariance function defined by

$$E\left[(x_t - \bar{x})(x_s - \bar{x})^{\mathrm{T}}\right] = C\delta(t - s),$$

where constant C is a positive scalar or a positive definite matrix and $\delta(\cdot)$ denotes the Dirac delta function defined by the property

$$\delta(t) = 0 \qquad \text{for} \qquad t \neq 0 \quad \text{and}$$

$$\int_{-\infty}^{\infty} \delta(t)\, f(t)\, dt = f(0)$$

for all functions f continuous at 0. Thus, the above definition requires $\{x_t\}$ to be an uncorrelated process with respect to t, while each x_t may be a vector-valued random variable with a nonzero covariance matrix $E[(x_t - \bar{x})(x_t - \bar{x})^{\mathrm{T}}] = C$. This definition of white process is not unique. For example, some authors replace the wide-sense stationarity assumption with the less restrictive "second-order process" property. In such a case, the time-invariant fixed (scalar or matrix) C, appearing above in the definition of covariance structure, is replaced by C_t, that is, with a fixed (i.e., nonrandom) scalar or matrix that may vary with time. In this article, we define (for operational convenience) a discrete-time white process $\{x_t\}$ by requiring only that x_t and x_s be independent random variables for any $t \neq s$.

Wide-sense stationary process Scalar- or vector-valued, second-order stochastic process $\{x_t\}$ such that (a) its mean $\bar{x} \equiv E[x_t]$ is independent of t and (b) its covariance function

$$R(t, s) \equiv E\left[(x_t - \bar{x})(x_s - \bar{x})^{\mathrm{T}}\right]$$

is a function of only the difference $t - s$, that is, $R(t, s) \equiv R_0(t - s)$.

FILTERING of a stochastic dynamic system (i.e., a system the dynamics of which can be adequately described by means of a mathematical model containing stochastic components and possessing certain well-defined properties) is a process of estimating the current state of the system while utilizing the recent estimates as well as

the past and current measurements (observations) of the system. The system model used in the filtering process describes the system dynamics by means of stochastic differential or stochastic difference equations and contains a transformation relating the system state to the measurements. These measurements are assumed to be noisy. The system dynamics part of the model may also be noisy.

A Kalman filter is a recursive, unbiased, minimum-mean-square-error (MMSE) estimator that uses linear models for system dynamics and state-to-measurement transformation. Both the dynamics part of the model and the state-to-measurement transformation incorporate the noise through additive stochastic components (terms). *Recursive* here means that new estimates are produced iteratively throughout the period of observation and that only the current and recent knowledge about the system is used to estimate its current state.

Nonlinear filters considered in this article are estimators of systems that can be adequately described only by means of nonlinear models. Such filters are often constructed by linearizing locally the system model and then applying a Kalman filter to this model.

I. INTRODUCTION

A. Filtering

Originally the term *filtering* was adopted in electrical engineering to refer to that aspect of signal processing which is concerned with the removal from signal of unwanted components such as noise. Until the advent of digital technology, especially of digital computers, filtering was performed exclusively by means of electrical circuits or devices called filters. A filter was tagged as linear or nonlinear depending on the type of circuits or devices (linear or nonlinear) from which it was built.

Presently the term *filtering* also refers to one of the three basic modes of information processing: filtering or estimation, smoothing, and prediction. Consider, for example, the description of the state of a time-varying system expressed in terms of a scalar or a vector quantity s. We call s the system state. Suppose that this system is observed (measured) by means of noisy measurements m, which, like s, may also be scalar- or vector-valued. (The letter t in this article always represents the time viewed as a variable. Specific values of t are indicated by means of a superscript or a subscript appended to t; in particular, t_0 always denotes the initial time.) We assume that s and m are related for all $t \geq t_0$ through the expression

$$m(t) = h[s(t)] + v(t),$$

where h is a function mapping $s(t)$ into $[m(t) - v(t)]$ and v represents the unobservable measurement noise (zero mean random error). For example, consider a particle moving in the xy plane. Assume that the state of the particle at time t is adequately described by means of its position coordinates (x, y) and the corresponding velocity components $(\dot{x}, \dot{y})$. Then the state vector $s \equiv s(t)$ can be written as $s^{\mathrm{T}} = (x, y, \dot{x}, \dot{y})$. Suppose that the system is observed at discrete time epochs $t_0, t_1, t_2, \ldots$ and that at each t_j one measures the polar coordinates (r, θ) of the particle. Thus, the measurement equation can be written

$$r = (x^2 + y^2)^{1/2} + v_1 \qquad \text{and} \qquad \theta = \arctan(y/x) + v_2,$$

where v_1 and v_2 represent the noise associated with the measurements of r and θ, respectively.

We use the terms *filtering* and *estimation* synonymously to describe inference about the value $s(t')$ of s at epoch t', which utilizes all information contained in measurements $m(t)$ accumulated over the period $t_0 \leq t \leq t'$. The term *smoothing* refers to recovery of the value $s(t')$ of s at time $t'(<t_e)$ from measurements $m(t)$ accumulated over the period $t_0 \leq t \leq t_e$. Finally, the term *prediction* refers to inference about the value $s(t')$ of s at time $t'(>t_e)$ made on the basis of measurements $m(t)$ accumulated over the period $t_0 \leq t \leq t_e$. This terminology is not completely universal; some authors use the term *estimation* to refer collectively to all three above-defined activities—filtering, smoothing, and prediction.

In general we write $\hat{s}_{t'|t_e}$ to denote the value of $s(t')$ inferred from the measurements $m(t)$ accumulated over the time period $t_0 \leq t \leq t_e$. Thus, $\hat{s}_{t'|t_e}$ represents an estimate (or a filtered value) of $s(t')$ if $t_e = t'$, a smoothened value of $s(t')$ if $t_e > t'$, and a predicted value of $s(t')$ if $t_e < t'$. In the case of measurements sampled at discrete epochs t_k $(k = 0, 1, 2, \ldots)$, if $t' = t_j$ and $t_e = t_h$, we simplify this notation by writing $\hat{s}_{j|h}$.

B. Historical Connections

The origins of the Kalman filter can be traced back to the ideas on least squares estimation formulated by K. F. Gauss. (Gauss first applied the estimation method around 1795, but his first published account of it appeared only around 1810.) He considered the problem of estimating essentially constant parameters of planetary orbits from noisy observations and derived a recursive estimation procedure for updating past estimates, without having to discard them, on the basis of newly available measurements. However, his recursive least-squares estimation method did not apply to a situation in which the estimated parameters would dynamically evolve over time. The extension of recursive estimation to dynamic system models is relatively recent.

Between the 1930s and the 1960s three major developments made a strong impact on the state of signal processing: (a) the advent of digital technology (digital circuits, devices, and stored-program computers); (b) advances in mathematics and in its application to the theory of linear systems, in particular, to communication and control systems; and (c) the infusion of statistical and probabilistic ideas into the theory of these systems. These advances were further stimulated by the demands of World War II and the geopolitical situation after the war. The Wiener filter is a well-known example from that era.

With the emergence of digital technology after World War II, it became possible for the first time and economically attractive to process complex signals digitally in real time. These advances in hardware had a revolutionary effect on both the theory and the implementation of filters. Software started to replace hardware as the medium of implementation; research shifted toward construction of algorithms for digital computing. The fast Fourier transform and the Kalman filter are examples of algorithms that were specifically designed for or naturally lend themselves to implementation on a digital computer. (To be precise, one should conceptually distinguish between an algorithm, the function it computes or approximates, and the applications for which it is used. Viewed as a mathematical function, the filters considered in this article are stochastic estimators of the system state. However, we often refer to them as algorithms and use expressions such as "the Kalman filter algorithm," especially when the function itself clearly indicates a computational procedure.)

The period between the turn of the century and the beginning of World War II witnessed important advances in the evolution of mathematics, especially in the generalization and abstraction of mathematical concepts, which later had an impact on the development of estimation theory and digital technology. Algebra, topology, functional analysis, and mathematical logic are a few examples of these developments. Two noteworthy developments from that era that were later used in filtering theory were (a) the formulation by A. N. Kolmogorov of the axiomatic theory of probability, based on the new theories of measure and (Lebesque) integration, and (b) the emergence of a generalized theory of linear spaces and operators. Mathematical measure and integration theories unified the conceptual setup of discrete and continuous probability spaces; they also furnished powerful tools, in the form of convergence theorems, for dealing with sequences of random variables. Axiomatic probability theory very soon led to a flurry of results in the theory of stochastic processes and in the application of this theory to practical areas such as time-series analysis.

C. Wiener Filter

The injection of probabilistic and statistical ideas into the filtering and control theories not only changed the view of how a system driven by randomly perturbed signals and observed by means of noisy measurements should be modeled, but also provided the tools for estimating the behavior of such systems. Wiener filtering stands as a major milestone in this process of adoption of the statistical approach to filtering and control. The basic problem considered in the 1940s by N. Wiener was as follows. Suppose that $\{x_t\}$ and $\{y_t\}$, where $-\infty < t < \infty$, represent two zero-mean, jointly wide-sense stationary stochastic processes with *known* joint autocovariances and cross-covariances. More precisely stated,

$$y_t = x_t + v_t, \qquad -\infty < t < t_f,$$

where

$$\begin{aligned} E\left[v_t v_s^{\mathrm{T}}\right] &= I_p \delta(t-s), \\ E\left[x_t v_s^{\mathrm{T}}\right] &= \text{arbitrary}, \qquad t \geq s, \\ &= 0, \qquad\qquad t < s, \end{aligned}$$

and $\delta(\cdot)$ represents the Dirac delta function defined by the property

$$\delta(t) = 0 \qquad \text{for} \qquad t \neq 0 \qquad \text{and}$$

$$\int_{-\infty}^{\infty} \delta(t)\, f(t)\, dt = f(0) \tag{1}$$

for all functions f continuous at 0.

Here y_t, x_t, and v_t represent p-valued vectors (Wiener originally considered only the scalar-valued case with $p = 1$) and I_p is a $p \times p$ identity matrix. Given sample observations $\{y_t(\omega), t_0 < t < t_f\}$ of process $\{y_t\}$, the problem is to find the MMSE estimate $\hat{x}(t_f + \Delta t \mid t_f)$ of x at time $t_f + \Delta t$ for some fixed Δt (≥ 0). It turns out that the task is to find a function $h(t)$ so that, for $t_0 < t_f < \infty$,

$$\hat{x}(t_f + \Delta t) = \int_{t_0}^{t_f} h(t_f - t') y(t')\, dt',$$

and the expectation $E[(\hat{x}(t_f + \Delta t) - x(t_f + \Delta t))^2]$ is minimized.

Wiener explicitly solved the above-stated estimation problem under the assumptions of a scalar observation process ($p = 1$) and a semiinfinite observation time interval ($t_0 = -\infty$). He handled this problem in the frequency domain by showing how to find a closed-form solution to a causal transfer function, say H, of h.

Actually, the optimal filtering theory, now known as the Wiener filter (WF), was independently and, at about the same time, developed by both N. Wiener and A. N. Kolmogorov. Whereas the continuous-time theory is mainly Wiener's work, Kolmogorov is credited with the

development of its discrete-time version. He introduced the notion of innovation sequence (a discrete-time white stochastic process associated with the measurement process, which we elaborate in our discussion of the Kalman filter) as a technique for orthogonalizing the data and then exploited the idea of orthogonal projections to simplify the estimation problem. These ideas were utilized years later to attain another view and deepen the understanding of the Kalman filter.

The publication of Wiener's work after World War II triggered a flurry of research in the 1950s and early 1960s. Attempts were made with some success to weaken (generalize) the underlying mathematical assumptions and to derive filtering schemes that would satisfy the requirements of the practical applications of that time period. Consequently, results were obtained extending the WF to nonstationary processes, to vector-valued processes, and to finite time intervals of observation. Despite these theoretical successes, certain difficulties with the WF remained. More precisely, (a) the results were mathematically complicated and thus were difficult to implement while using the signal-processing hardware of that period; (b) the WF did not lend itself easily to recursive estimation in which the estimates would be updated (improved) with each new batch of data—for example, with each pass of an earth satellite over the view of a tracking station; (c) implementational difficulties with the estimation of vector-valued processes persisted. Furthermore, for some applications, certain the oretical assumptions made in the formulation of the WF (such as the wide-sense stationarity of stochastic processes, the existence of spectral densities, and the knowledge of second-order moments) fail to hold. For example, the stochastic components of dynamic systems typically are nonstationary. Most important, the WF, being a product of the era of analog hardware, did not lend itself easily to the newly emerging mode of processing on a stored-program digital computer with finite memory and limited throughput capacity. The aerospace signal-processing problems in the 1950s started to point to the need for a robust algorithm for recursive estimation of vector-valued processes in dynamic systems, with the observations made and estimates updated at discrete time epochs. Furthermore, such an algorithm was to be implemented for real-time processing on a still primitive digital computer. The Kalman filter (KF) turned out to be an answer to these needs.

D. Transition to State Variable-Based Filtering

Although the use of state variables rather than impulse response is now largely credited to R. Kalman, P. Swerling actually was the first person (in a 1959 journal article) to describe an estimation scheme that essentially was identical to that presented the next year by Kalman. One reason for crediting Kalman with what today is known as the Kalman filter was that subsequently Kalman extended his results considerably beyond those obtained by Swerling.

In addition to practical applications of linear filtering theory developed in the 1960s, foundations for nonlinear estimation were laid during the same period. At the time when Kalman's (and also R. C. Bucy's) work was becoming known in the West, R. L. Stratonovich in the Soviet Union published results on his pioneering work in nonlinear estimation, that is, recursive estimation of the states of a nonlinear system driven by white noise. He also looked into the linearized system model and, while doing this, obtained the KF equations in 1960. Though Stratonovich's work in nonlinear estimation remained unknown for a while in the West, some of his results were independently replicated in early 1960s by H. J. Kushner and W. M. Wonham.

II. LINEAR ESTIMATION PROBLEM AND KALMAN FILTERS

The KF is a recursive, unbiased, MMSE estimator of the state s of a dynamic system observed (i.e., measured) by means of noisy measurements m. In general, both s and m are vector-valued. This system is assumed to be linear in the sense that (a) the dynamics of state s is described in terms of linear differential or difference equations that may be additively perturbed by a random process, called process noise; and (b) noisy measurements m are related to the state s by means of a linear transformation.

A. Discrete-Time Linear System Model

The system model used by Kalman is roughly of the following form. For $t_{k+1} = t_k + \Delta t_k$ $(\Delta t_k > 0$ and $k = 0, 1, 2, \ldots)$,

$$s_k = \Phi_{k-1} s_{k-1} + B_{k-1} u_{k-1} + G_{k-1} w_{k-1}, \qquad k = 1, 2, 3, \ldots, \quad (2)$$

and

$$m_k = H_k s_k + v_k, \qquad k = 0, 1, 2, \ldots. \quad (3)$$

Equation (2) is called the system dynamics equation, and Eq. (3) the measurement equation. Subscript k refers to the kth sampling epoch t_k. Then s_k denotes the value of state at time t_k. $\{w_k, k = 0, 1, 2, \ldots\}$, or simply written $\{w_k\}$, is a random sequence (a discrete-time stochastic process) that represents the process noise, also known as the input, driving, or plant noise of process $\{s_k\}$. The $\{m_k\}$ represents the measurements, also known as the ouput process,

which at time t_k are perturbed by random noise v_k. Matrix (operator) Φ_{k-1}, which propagates the state from time t_{k-1} to t_k, is called the system state transition matrix. The optional term $B_{k-1}u_{k-1}$ represents deterministic control inputs with $\{u_k\}$ being a known sequence. Matrix H_k defines the state-to-measurement transformation at t_k. The subscripts attached to Φ, B, G, and H indicate that these matrices may vary over time. It is assumed that these matrices are a priori known; that is, they do not depend on estimates of s or on measurements m.

As already noted the random processes $\{w_k\}$, $\{v_k\}$, $\{s_k\}$, and $\{m_k\}$ usually are vector-valued. We use n_s to denote the length of vector s_k and n_m to denote the lengths of v_k and m_k. As a special case we may have $n_s = 1$ or $n_m = 1$.

Notationally, we do *not* distinguish between scalar-valued and vector-valued quantities. Uppercase letters, English or Greek, are reserved mainly for matrices (operators). We also notationally do not distinguish between the random variables (RVs), or random functions, and the specific values taken on by these RVs. For example, depending on the context, w_k denotes either the RV representing the process noise at t_k or a specific value taken by this RV.

The system model given by Eqs. (2) and (3) is incomplete unless we specify the underlying probabilistic assumptions, which are as follows.

ASSUMPTION 1D. The process noise $\{w_k\}$ is a zero-mean, Gaussian (normal), white process with the covariance matrix

$$E\left[w_k w_j^{\mathrm{T}}\right] = Q_k\,\delta_{kj}, \tag{4}$$

where Q_k is a *known* $n_s \times n_s$ nonnegative definite matrix and δ_{kj} denotes the Kronecker delta, that is, $\delta_{kj} = 1$ if $k = j$; otherwise, $\delta_{kj} = 0$. (If $\{w_k\}$ is absent from the system model for some or all k, then for each such k, Q_k may be thought of as being a zero matrix.)

ASSUMPTION 2D. The measurement noise $\{v_k\}$ is a zero-mean, Gaussian, white process with the covariance matrix

$$E\left[v_k v_j^{\mathrm{T}}\right] = R_k\,\delta_{kj}, \tag{5}$$

where R_k denotes a *known* $n_m \times n_m$ positive definite matrix, and δ_{kj} is the Kronecker delta.

ASSUMPTION 3D. For all k and j, v_k and w_j are statistically independent RVs.

ASSUMPTION 4D. The value s_0 of system state s at the starting time t_0 is a Gaussian RV whose mean $\bar{s}_0$ and convariance matrix Σ_0, that is,

$$\bar{s}_0 \equiv E[s_0] \qquad \text{and} \qquad \Sigma_0 \equiv E\left[(s_0 - \bar{s}_0)(s_0 - \bar{s}_0)^{\mathrm{T}}\right] \tag{6}$$

are known. Furthermore, the RV s_0 is statistically independent of the the RVs w_k and v_k for all $k \geq 0$.

In Assumptions 1D–4D, the letter "D" indicates discrete time. In this article, a discrete-time white process is defined to be a discrete-time stochastic process, say $\{x_i\}$, for which x_k and x_j are statistically independent RVs whenever $k \neq j$. Collectively, all a priori distributional assumptions concerning the random components (s_0, $\{w_k\}$, and $\{v_k\}$) of the model are referred to as the a priori probability assumptions or, simply, as the a priori probabilities. All above-stated assumptions of Gaussian distribution, taken together, are referred to as the Gaussianity assumption. As a relaxation of the original assumptions in Wiener filtering, Assumptions 1D–4D do not require $\{w_k\}$ or $\{v_k\}$ to be a stationary process in the sense that matrices Q_k and R_k may be time dependent.

B. Properties

The system model defined in terms of Eqs. (2) and (3), restricted by the distributional assumptions (1D–4D), has the following easily verifiable properties.

a. System state s_k (for $k = 1, 2, \ldots$) can be expressed as

$$s_k = \Phi_{k,0}s_0 + \sum_{i=0}^{k-1} \Phi_{k,i+1}(B_i u_i + G_i w_i),$$

where the multistep state transition matrix $\Phi_{k,j}$, defined in terms of the single-step state transition matrix Φ_i as $\Phi_{i+1,i} \equiv \Phi_i$, possesses the following properties:

(i) $\Phi_{k,k} \equiv I_s (= n_s \times n_s$ identity matrix),

(ii) $\Phi_{k,j} = \Phi_{k-1}\Phi_{k-2}\cdots\Phi_j \quad$ for $\quad k > j \geq 0$, and

(iii) $\Phi_{k,i} = \Phi_{k,j}\Phi_{j,i} \quad$ for all $\quad k \geq j \geq i \geq 0$.

b. Each s_k is a multivariate Gaussian RV of length n_s; actually, we can say more: The process $\{s_k\}$ is a Gaussian random process.

c. $\{s_k\}$ is a Markov process, which implies that for any $0 \leq k_1 < k_2 < \cdots < k_n < k$, the conditional probability (density) function of s_k given $s_{k_1}, s_{k_2}, \ldots, s_{k_n}$ has the Markov property

$$p\left(s_k \,\middle|\, s_{k_1}, s_{k_2}, \ldots, s_{k_n}\right) = p\left(s_k \,\middle|\, s_{k_n}\right).$$

d. The measurement process $\{m_k\}$ is a Gaussian process, but it is *not* a Markov process.

e. Processes $\{s_k\}$ and $\{m_k\}$ jointly form a Gaussian process $\{(s_k^{\mathrm{T}}, m_k^{\mathrm{T}})^{\mathrm{T}}\}$.

The preceding properties of the system model are used in the derivation of the KF algorithm and in proving its

optimality properties. Recursiveness of the KF algorithm essentially is a consequence of (Markov) property c.

The first two moments of $\{s_k\}$ and $\{m_k\}$ follow directly from properties a–e. First, consider the process $\{s_k\}$ and, for $k \geq j \geq 0$, define

$$\begin{aligned} \bar{s}_k &\equiv E[s_k], \\ \Sigma_{k,j} &\equiv E\big[(s_k - \bar{s}_k)(s_j - \bar{s}_j)^{\mathrm{T}}\big], \\ \Sigma_k &\equiv \Sigma_{k,k}. \end{aligned}$$

It follows from properties a and b that

$$\begin{aligned} \bar{s}_k &= E[s_k] = \Phi_{k,0}\bar{s}_0 + \sum_{i=1}^{k-1} \Phi_{k,i+1} B_i u_i \quad \text{for} \quad k \geq 0, \\ \bar{s}_{k+1} &= \Phi_k \bar{s}_k + B_k u_k \qquad \text{for} \qquad k \geq 0, \\ \Sigma_k &= \Phi_{k,0}\Sigma_0(\Phi_{k,0})^{\mathrm{T}} + \sum_{i=0}^{k-1} (\Phi_{k,i+1}G_i) Q_i (\Phi_{k,i+1}G_i)^{\mathrm{T}}, \\ \Sigma_{k+1} &= \Phi_k \Sigma_k \Phi_k^{\mathrm{T}} + G_k Q_k G_k^{\mathrm{T}}, \\ \Sigma_{k,i} &= \Phi_{k,i}\Sigma_i \qquad \text{if} \qquad k \geq i. \end{aligned}$$

The first two moments of measurement process $\{m_k\}$ turn out to be

$$\bar{m}_k \equiv E[m_k] = H_k \bar{s}_k$$

and

$$E\big[(m_k - \bar{m}_k)(m_i - \bar{m}_i)^{\mathrm{T}}\big] = H_k \Phi_{k,i} \Sigma_i H_i^{\mathrm{T}} + R_k\, \delta_{kj} \qquad \text{for} \qquad k \geq i \geq 0,$$

where δ_{kj} is the Kronecker delta defined following Eq. (4). The above-listed expressions for the first two moments of s_k and m_k remain valid even if the initial state s_0 or at least one of the noise processes, $\{w_k\}$ or $\{v_k\}$, is non-Gaussian.

C. Discrete-Time Filtering Problem

In this subsection as well as in the next, it is important to view the measurements m_k, the state s_k, and the state estimator as RVs but not as the values taken on by these RVs. Let the RV $M_k \equiv \{m_0, m_1, \ldots, m_k\}$ represent the set of all measurements obtained over the time period $t_0 \leq t \leq t_k$. Use $\hat{s}_{k|j}$ to denote an estimator of s_k that is a function of (is conditioned on) M_j. As noted earlier, we have an estimation problem if $k = j$, a prediction problem if $k > j$ (in such a case, $\hat{s}_{k|j}$ is more appropriately called the predictor of s_k), and a smoothing problem if $k < j$. The KF is concerned only with the first two problems.

The optimality criterion used in the derivation of the KF algorithm is minimization of mean square error. Thus, the KF is an estimator $\hat{s}_{k|j}$ conditioned on M_j, which minimizes

$$E\big[(s_k - \hat{s}_{k|j})^{\mathrm{T}}(s_k - \hat{s}_{k|j}) \,\big|\, M_j\big] \qquad \text{for} \qquad k = 0, 1, 2, \ldots. \tag{7}$$

It can be shown that such $\hat{s}_{k|j}$ is of the form

$$\hat{s}_{k|j} = E[s_k \mid M_j]. \tag{8}$$

In the literature, the MMSE estimator is also alternatively known as the minimum variance (MV) or the least-squares (LS) estimator. The latter name is technically imprecise. Furthermore, these names are somewhat misleading because they fail to indicate that the resulting estimates depend on a priori probability assumptions and that these estimates are conditional in nature. Also note that the MMSE criterion can be applied to a more general system model than one defined by Eqs. (2) and (3) and Assumptions 1D–4D. Even then the minimization of Eq. (7) implies the conditional mean estimator of Eq. (8). However, the resulting estimator may not be mathematically or implementationally tractable.

Given the starting conditions $\bar{s}_0$, Σ_0, and the initial measurements M_0 (=$\{m_0\}$), the KF provides a procedure for computing recursively at each t_k the values of

$$\hat{s}_{k|k-1} = E[s_k \mid M_{k-1}] \quad \text{and} \quad \hat{s}_{k|k} = E[s_k \mid M_K] \tag{9}$$

and the associated error covariance matrices

$$P_{k|k-1} \equiv E\big[(s_k - \hat{s}_{k|k-1})(s_k - \hat{s}_{k|k-1})^{\mathrm{T}} \,\big|\, M_{k-1}\big] \tag{10a}$$

and

$$P_{k|k} \equiv E\big[(s_k - \hat{s}_{k|k})(s_k - \hat{s}_{k|k})^{\mathrm{T}} \,\big|\, M_k\big]. \tag{10b}$$

One noteworthy property of the resulting KF estimator is that it is *unbiased*. This follows from Eq. (8):

$$\begin{aligned} E[s_k - \hat{s}_{k|j}] &= E[s_k] - E[E[s_k \mid M_j]] \\ &= E[s_k] - E[s_k] = 0. \end{aligned}$$

The MMSE and unbiasedness properties of the KF imply that it also is an MV estimator. Another important property of the KF estimator is that it is *linear* in the sense that the estimators $\hat{s}_{k|k-1}$ and $\hat{s}_{k|k}$ are of the form

$$\hat{s}_{k|k-1} = \Phi_{k-1}\hat{s}_{k-1|k-1} + B_{k-1}u_{k-1}$$

and

$$\hat{s}_{k|k} = K_k(m_k - H_k\hat{s}_{k|k-1}) + \hat{s}_{k|k-1},$$

where K_k is the so-called Kalman gain matrix.

What happens if the Gaussianity assumptions for $\{w_k\}$, $\{v_k\}$, or s_0 are dropped? Then the KF algorithm still provides the best and unbiased estimator among all linear

estimators of the state. It is best in the sense that it produces the smallest mean square error among all linear estimators of the state. However, in such a situation there may exist a nonlinear filter which, in the MMSE sense, is better than the KF.

D. Alternative Approaches to Discrete-Time Linear Filtering

The MMSE optimality criterion [Eq. (7)] is not the only one used in recursive estimation. The LS principle of residual minimization, the maximum-likelihood (ML) principle, and various optimality principles used in conjunction with the Bayes method are examples of other possibilities. In the case of a linear estimation problem [i.e., for a system model of the form of Eqs. (2) and (3)], if the initial conditions and the noise processes have Gaussian distributions, as is implied by Assumptions 1D–4D, the methods of LS (with appropriate weights), ML, and Bayes (with a quadratic performance criterion) yield estimators of the same (KF) type.

The weighted LS method, which can be expressed in recursive form for sampled measurements [Eq. (3)], is the simplest of the three above-mentioned methods because it does not require making any explicit assumptions, with one exception, about the distribution of RVs involved in the estimation process. The exception concerns the assumption of a priori knowledge of the process and measurement noise covariance matrices, the inverses of which are used as weights in the LS estimation procedure. The optimality criterion at time t_k is the minimization of the sum of squares of the weighted residuals (i.e., the weighted differences between the measurements and their values predicted by the model) accumulated over the epochs $t_0, t_1, \ldots, t_k$. As already noted, this approach under the Gaussianity assumptions for noise processes leads to an estimator of KF type.

The LS method (recursively or nonrecursively formulated, weighted or unweighted) has been widely used since Gauss to estimate the parameters of static systems. A static system may be thought of as a special case of the system model [Eqs. (2) and (3)] in which Φ_k is an $n_s \times n_s$ identity matrix, and B_k and G_k are zero matrices, that is, the system is time invariant and so Eq. (2) is not needed to describe the system dynamics. The weighted or unweighted LS method, usually expressed in nonrecursive form, constitutes the backbone of regression analysis and currently is one of the principal methods in statistical inference.

The ML method has also been used for years in statistical inference as a point estimation technique. It is based on viewing the probability (density) function $p_s(M_k)$ of measurements $M_k = \{m_0, m_1, \ldots, m_k\}$ as a function $\lambda_s \equiv p_s(M_k)$ of the unknown parameters s (in our case, s represents the unknown system state) on which the form of this function depends. Then, given measurements M_k, the ML estimate of s is that value, say s^*, of s that maximizes λ_s. This principle has been extended to the construction of recursive estimators for linear dynamic systems of the type defined by Eqs. (2) and (3). As already noted, under Assumptions 1D–4D, the resulting recursive estimator is identical to the KF.

In engineering literature, an estimator derived from the Bayes conditional probability inversion formula

$$p(x \mid y) = p(x, y)/p(y) = p(y \mid x)p(x)/\int p(y \mid x)p(x)\,dx \quad (11)$$

by applying some optimality criterion is often, although somewhat imprecisely, called the Bayes estimator of x. More precisely, the resulting estimator is a Bayes estimator only if, in addition, the $p(x)$ appearing on the right-hand side of Eq. (11) is a Bayes prior (also known as Bayes a priori) probability of x. The term *prior* here means that $p(x)$ is postulated prior to observing the value of y. In estimation, y and x would represent the measurements and the unknown parameters (state), respectively. Hence, in the problems considered here, y and x would be replaced by variously subscripted M (or m) and s, respectively. The left-hand side of the Bayes inversion formula would then give the posterior (also known as Bayes a posteriori) probability of unknown parameters (state) s conditioned on measurements M (or m).

Using Eq. (11) and the RV independence conditions implied by Assumptions 1D–4D, and also noting that

$$M_k \equiv \{m_0, m_1, \ldots, m_{k-1}, m_k\} = \{M_{k-1}, m_k\},$$

one can derive the so-called Bayes probability recursion formulas,

$$p(s_k \mid M_k) = p(s_k \mid M_{k-1})p(m_k \mid s_k)/p(m_k \mid M_{k-1}) \quad (12a)$$

and

$$p(s_k \mid M_{k-1}) = \int p(s_k \mid s_{k-1})p(s_{k-1} \mid M_{k-1})\,ds_{k-1}, \quad (12b)$$

with the denominator in Eq. (12a) expressible as

$$p(m_k \mid M_{k-1}) = \int p(m_k \mid s_k)p(s_k \mid M_{k-1})\,ds_k. \quad (12c)$$

Equations (12a) and (12b) recursively define the measurement updating and the time propagation (also called time updating) of probability densities for $k = 0, 1, \ldots$. To get the recursive process started at t_0, one must interpret correctly the meaning of $p(s_0 \mid M_{-1})$. Since M_{-1} constitutes an empty set (i.e., no measurements), $p(s_0 \mid M_{-1})$ represents the Bayes prior $p(s_0)$. As an example, this prior

for a KF is specified in terms of the first two moments, $\bar{s}_0$ and Σ_0, of the Gaussian distribution of state s at the initial time t_0. It is worth noting that, in Eqs. (12a)–(12c), $p(s_k \mid s_{k-1})$ is obtained from the state time-propagation equations (2) with the assistance of the a priori known probability (density) $p(w_{k-1})$ of process noise w_{k-1} at t_{k-1}. Similarly, $p(m_k \mid s_k)$ is determined from the measurement equations (3) and with the assistance of the a priori known probability (density) $p(v_k)$ of measurement noise v_k at t_k. Knowledge of all these probability densities and $p(s_0)$ recursively determines $p(s_k \mid M_k)$ for all $k = 0, 1, \ldots$.

Using the recursive probability equations (12a)–(12c) in conjunction with various optimality criteria produces a variety of recursive estimation schemes. However, due mainly to mathematical and implementational considerations, the MMSE criterion, equivalent to Eq. (7) and yielding an estimator of the type in Eq. (8), is the most widely used criterion at present. As already noted, this criterion is applied to derive the KF equations.

E. Discrete-Time Kalman Filter

Given the discrete-time linear system model (including the distributional Assumptions 1D–4D) and the defining relation of Eq. (8) there are several ways in which the discrete-time KF algorithm can be derived. One technique, which is mathematically straightforward although tedious in detail, is based on the use of recursive probability equations (12a)–(12c) and the repeated application of conditional multivariate Gaussian distribution to express the probability densities $p(s_k \mid M_{k-1})$ and $p(s_k \mid M_k$ for all $k = 0, 1, \ldots$ in terms of m_k, $\hat{s}_{k-1|k-1}$, $P_{k-1|k-1}$, the matrices appearing in the system model, and the covariances of noise processes $\{w_k\}$ and $\{v_k\}$. Due to the regenerative properties of Gaussian distribution, densities $p(s_k \mid M_{k-1})$ and $p(s_k \mid M_k)$ remain Gaussian for all $k \geq 0$. By Eqs. (9), (10a), and (10b), the first two moments of $p(s_k \mid M_{k-1})$ and $p(s_k \mid M_k)$ are $\hat{s}_{k|k-1}$ with $P_{k|k-1}$ and $\hat{s}_{k|k}$ with $P_{k|k}$, respectively. Hence, the mathematical expressions for these quantities, giving the recursive estimation equations (13)–(18), are explicitly obtainable from the mathematical expressions for $p(s_k \mid M_{k-1})$ and $p(s_k \mid M_k)$.

Another technique for deriving the KF algorithm uses the notion of innovations, defined as

$$r_k = m_k - E[m_k \mid m_{k-1}] = m_k - H_k \hat{s}_{k|k-1}.$$

In the innovation process $\{r_k\}$, each r_k constitutes that part of the measurement at t_k that contains new information not provided by $m_0, m_1, \ldots, m_{k-1}$. Innovations $\{r_k\}$ constitute a zero-mean white process. It follows that r_k represents the error in the orthogonal projection of s_k on the subspace generated by $M_k = \{m_0, m_1, \ldots, m_k\}$. This projection, being a linear combination of the vectors in M_k, defines a linear MMSE estimator of s. Using this notion of orthogonality, the whiteness of $\{r_k\}$, and the system model [Eqs. (2) and (3)], especially the Markov property of state propagation, it is then rather simple to derive the KF recursion expressions.

Using the preceding system model and probabilistic assumptions, one can state the KF algorithm for a discrete-time linear system as follows.

Discrete-Time Kalman Filter Algorithm. At each $t_k = t_{k-1} + \Delta t_{k-1} (k = 0, 1, 2, \ldots)$, proceed as follows.

a. If $k = 0$ (i.e., at the initial time t_0), initialize the estimation process by setting

$$\hat{s}_{0|-1} = \hat{s}_0 \qquad \text{and} \qquad P_{0|-1} = \Sigma_0, \tag{13}$$

where $\bar{s}_0$ and Σ_0 are defined by Eq. (6).

b. If $k \geq 1$, propagate the estimates of system state vector s and state error covariance matrix P from t_{k-1} to t_k by computing

$$\hat{s}_{k|k-1} = \Phi_{k-1} \hat{s}_{k-1|k-1} + B_{k-1} u_{k-1} \tag{14}$$

and

$$P_{k|k-1} = \Phi_{k-1} P_{k-1|k-1} \Phi_{k-1}^{\mathrm{T}} + G_{k-1} Q_{k-1} G_{k-1}^{\mathrm{T}}, \tag{15}$$

respectively.

c. Use measurements $m_k = H_k S_k + v_k$ to update the propagated estimate of s by computing the Kalman gain matrix

$$K_k = P_{k|k-1} H_k^{\mathrm{T}} \left[H_k P_{k|k-1} H_k^{\mathrm{T}} + R_k \right]^{-1} \tag{16}$$

and the update of the state estimate

$$\hat{s}_{k|k} = \hat{s}_{k|k-1} + K_k [m_k - H_k \hat{s}_{k|k-1}]. \tag{17}$$

Next, update the estimate P of state error covariance matrix by computing

$$P_{k|k} = [I - K_k H_k] P_{k|k-1}. \tag{18}$$

The preceding algorithm assumes that one knows the values of $\bar{s}_0$ and Σ_0 and also knows how to compute for all k the matrices Φ_k, B_k, R_k, Q_k, and H_k. In the literature, $\hat{s}_{k|k}$ is often called the measurement update of $\hat{s}_{k|k-1}$ and $\hat{s}_{k+1|k}$ the time update of $\hat{s}_{k|k}$.

If the state estimate update equation (17) is rewritten

$$\hat{s}_{k|k} = K_k m_k + [I - K_k H_k] \hat{s}_{k|k-1}, \tag{17$'$}$$

an equivalent expression for $P_{k|k}$ is

$$P_{k|k} = K_k R_k K_k^{\mathrm{T}} + [I - K_k H_k] P_{k|k-1} [I - K_k H_k]^{\mathrm{T}}, \tag{18$'$}$$

which is known as the Joseph form (named after P. D. Joseph) of error covariance update. Since the right-hand side of Eq. (18′) is the sum of two symmetric matrices (the

first a nonnegative definite, the second a positive definite matrix), the state error covariance update in this form numerically is more stable, although computationally more expensive, than the same operation in the short form [Eq. (18)].

EXAMPLE 1. FILTERING OF AN AUTOREGRESSIVE PROCESS. This example illustrates the application of the discrete-time KF to a simple linear system in which both the state s and the measurement m are scalars. The system model is of the form

$$s_k = \phi s_{k-1} + w_k \qquad \text{and} \qquad m_k = h s_k + v_k,$$

where ϕ and h are scalar multipliers, that is, $n_s = n_m = 1$. Both noise processes, $\{w_k\}$ and $\{v_k\}$, and the initial conditions are assumed to satisfy Assumptions 1D–4D. In the present case, the process noise covariance matrices Q and R are 1×1; that is, Q and R represent the time-invariant variances of w_k and v_k, respectively, and are scalar quantities.

The KF equations now reduce to the following procedure.

a. If $k = 0$, initialize the estimation process by specifying s_0 and $P_{0|-1}$.

b. If $k \geq 1$, propagate the estimates by computing

$$\hat{s}_{k|k-1} = \phi \hat{s}_{k-1|k-1},$$
$$P_{k|k-1} = \phi^2 P_{k-1|k-1} + Q.$$

c. For all $k \geq 0$, update the estimates by computing

$$K_k = h P_{k|k-1} \big/ \left[h^2 P_{k|k-1} + R\right],$$
$$\hat{s}_{k|k} = \hat{s}_{k|k-1} + K_k[m_k - h\hat{s}_{k|k-1}],$$
$$P_{k|k} = K_k^2 R + [1 - hK_k]^2 P_{k|k-1}.$$

Figure 1 summarizes the results of a sample simulation of this KF. Figure 1a shows the generated time history of the true system state s. Figure 1b summarizes the time history of estimation error, which, at t_k, is defined as $e_k = \hat{s}_{k|k} - s_k$. The values of parameters of the true system model are

$$\phi = 1.02, \quad h = 100.0, \quad Q = 4.0, \quad \text{and} \quad R = 1.0.$$

The values of the corresponding parameters in the system model assumed by the KF are equal in value to those of the true model. The true initial value of state is $s_0 = 5.0$. The initialization values used by the KF are $\bar{s}_0 = 4.0$ and $P_{0|-1} = 25.0$.

Since, according to the true system model, the measurements are very precise compared with the level of process noise (i.e., the variance of m/h is $R/h^2 = 0.0001 \ll Q$) and the system model assumed in the KF agrees with the true model, the filter practically converges at t_0 after the first measurement update. One can easily see this by observing in Fig. 1b that (for all $k \geq 0$) the points $\pm(P_{k|k})^{1/2}$ lie on two horizontal lines that are symmetrically located with respect to the time axis.

In many respects, the above-described filtering problem for a random walk is trivial. Its only purpose was to illustrate the KF computations for a very simple (scalar) case. The following "reversion" of this problem has practical applications in time-series analysis and is known as a system identification problem. Suppose that the value of scalar multiplier ϕ is continuously undergoing random perturbations and so is not completely known. Assume that the dynamics of ϕ is adequately described by means of an equation of the form

$$\phi_k = \phi_{k-1} + w'_{k-1}$$

and that we are now directly observing the state s via measurement equation of the form

$$s_k = s_{k-1}\phi_k + v'_k,$$

which essentially is the state transition equation of the original problem with a modified noise term. Making the usual probabilistic assumptions for the noise processes $\{w'_k\}$ and $\{v'_k\}$ and the initial distribution of ϕ completes a system model for the KF of the system identification problem (estimation of the unknown and possibly time-varying values of parameters in the system model).

F. Continuous-Time Kalman Filter

A year after the publication of Kalman's seminal work on discrete-time filtering, R. Bucy and R. Kalman laid the foundation for continuous-time KFs. Although presently the KFs, probably without exception, are implemented as computer programs for processing on a digital computer, strong interest in the continuous-time KFs persists. Often the estimation problem for a stochastic dynamic system can be precisely formulated only in terms of stochastic differential equations. Thus, the continuous-time KF is an important modeling tool to a designer.

The system model for a continuous-time KF is summarized next. (As much as possible, the same notation as in the discrete-time model is used here. Also, to simplify the discussion of the continuous-time KF, the term representing the control inputs has been deleted from the system dynamics model; the presence of such a term would affect in an a priori known way only the mean value of the state, not its covariances.) The dynamics of the system state, at

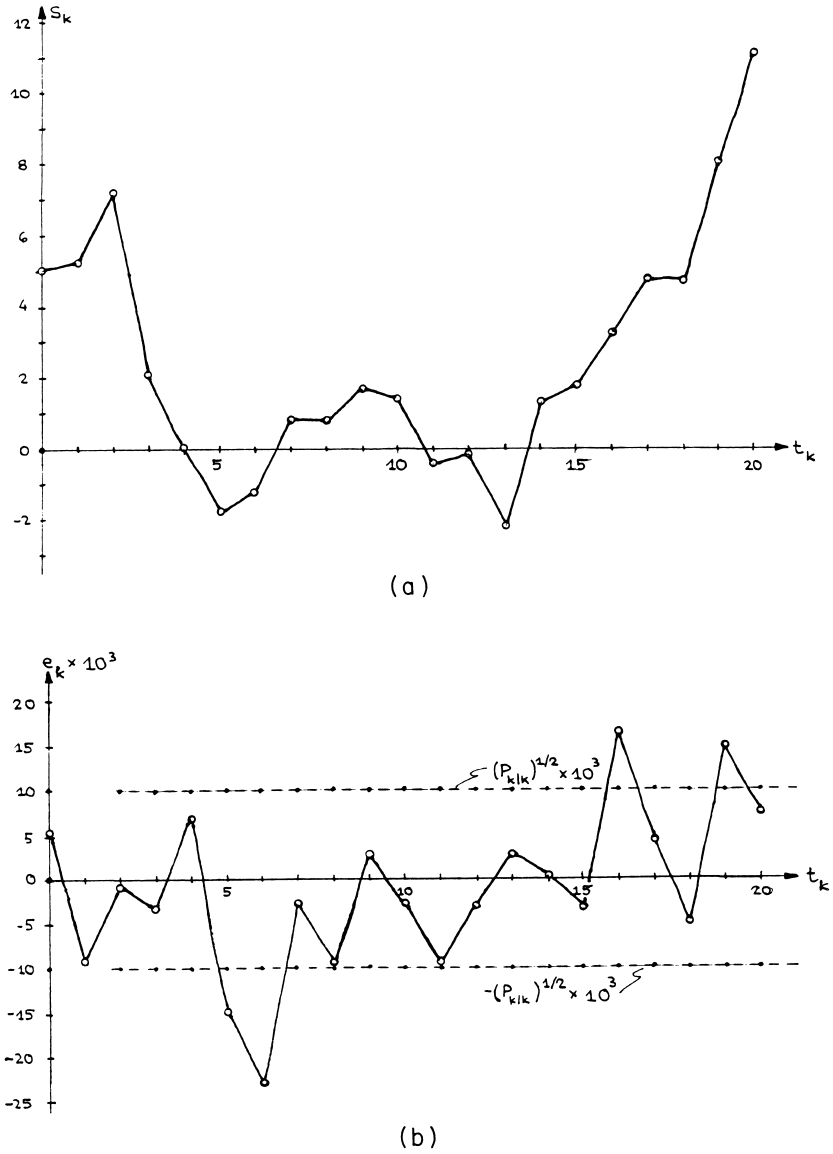


FIGURE 1 Results of a simulation run of the Kalman filter, Example 1. (a) Generated time history of the true system state and (b) time history of estimation error e (multiplied by 10^3).

least *formally*, is described in the continuous-time case by a stochastic differential equation of the form

$$ds_t/dt = F_t s_t + G_{(c)t} w_t, \qquad t \geq t_0. \tag{19}$$

The measurement equation is written

$$m_t = H_t s_t + v_t, \qquad t \geq t_0. \tag{20}$$

The time argument t presently ranges over a continuum (an interval) of real numbers and is, as indicated, finitely bounded from below. Thus, $\{w_t\}$, $\{s_t\}$, $\{v_t\}$, and $\{m_t\}$ are now continuous-time stochastic processes with $t \geq t_0$. With regard to vector and matrix sizes, the assumptions made and the notationused earlier continue to hold. The subscript "c" distinguishes a continuous-time entity from its discrete-time counterpart. The probabilistic assumptions about the model specified by Eqs. (19) and (20) are similar to those stated for the discrete-time version of the KF.

ASSUMPTION 1C. The process noise $\{w_t\}$ is a continuous-time, zero-mean, Gaussian white process with the covariance structure of the form

$$E\left[w_t w_{t'}^{\mathrm{T}}\right] = Q_{(c)t}\, \delta(t - t'), \tag{21}$$

where $Q_{(c)t}$ is a known $n_s \times n_s$ nonnegative definite matrix (which may optionally be a zero matrix) and $\delta(\cdot)$ denotes the Dirac delta function defined by Eq. (1).

ASSUMPTION 2C. The measurement noise $\{v_t\}$ is a continuous-time, zero-mean, Gaussian white process with

$$E\left[v_t v_{t'}^{\mathrm{T}}\right] = R_{(\mathrm{c})t}\,\delta(t-t'), \tag{22}$$

where $R_{(\mathrm{c})t}$ is a known $n_m \times n_m$ positive definite matrix and $\delta(\cdot)$ represents the Dirac delta function defined by Eq. (1).

ASSUMPTION 3C. For all t and $t' \geq t_0$, RVs w_t and $v_{t'}$ are independently distributed RVs.

ASSUMPTION 4C. The value s_0 of s at t_0 is distributed as a Gaussian RV whose mean and covariance matrix, that is,

$$\bar{s}_0 \equiv E[s_0]$$

and

$$\Sigma_0 \equiv E\left[(s_0-\bar{s}_0)(s_0-\bar{s}_0)^{\mathrm{T}}\right], \tag{23}$$

are known. Furthermore, the RV s_0 is statistically independent of the RVs w_t and $v_{t'}$ for all t and $t' \geq t_0$.

Mathematically the stochastic differential equation (19) is not meaningful because of the white noise assumption (Assumption 1C) for the continuous-time process $\{w_t\}$. It is well known, however, that white Gaussian noise is the formal derivative of a Brownian motion process $\{\beta_t\}$, with $t \geq t_0$, which has the property that the increments

$$\left\{\beta_{t_j} - \beta_{t_{j-1}}\right\}, \qquad j \geq 1,$$

are independent Gaussian RVs for any ordered set $t_0 < t_1 < t_2 < \cdots < t_n$ of epochs. Thus, the differential equation (19) can be viewed as being equivalent to

$$ds_t = (F_t s_t)\,dt + (G_{(\mathrm{c})t})\,d\beta_t, \qquad t \geq t_0, \tag{24}$$

and so to

$$s_t - s_{t_0} = \int_{t_0}^{t} F s_\tau\,d\tau + \int_{t_0}^{t} G_{(\mathrm{c})\tau}\,d\beta_\tau. \tag{25}$$

On the right-hand side of Eq. (25), the first integral is ordinary (Riemann or Lebesque), but the second is an Itô stochastic integral. (Integrals of this type were defined in the 1940s by the Japanese mathematician K. Itô.)

Replacement of Eq. (19) by Eq. (24) requires the corresponding modification of Assumptions 1C–4C. In these assumptions, $\{w_t\}$ must be replaced by $\{d\beta_t\}$, where $\{\beta_t, t \geq t_0\}$ is a Brownian motion process and

$$E\left[d\beta_t d\beta_{t'}^{\mathrm{T}}\right] = Q_{(\mathrm{c})}\delta(t-t'). \tag{21$'$}$$

The modified assumption are referred to as Assumptions 1C′–4C′.

Rigorous treatment of continuous-time stochastic processes and, in particular, of stochastic differential equations requires considerably more mathematical sophistication than that of discrete-time stochastic processes (sequences of RVs). Consequently, derivation of a continuous-time KF is technically more difficult than derivation of its discrete-time counterpart. Except for noting that mathematically it is more rigorous to use Eq. (24) in place of Eq. (19), we do not dwell any longer on these theoretical issues and proceed to an intuitive definition of a continuous-time KF.

According to whether the measurements are incorporated continuously over the time or are just sampled at discrete time points, one could distinguish two variations of continuous-time KF: a continuous-dynamics, continuous-measurement KF and a continuous-dynamics, discrete-measurement KF. Since the first is of little interest in practice, we do not consider it in this chapter. Hence, from now on the term *continuous-time KF* implies a continuous-dynamics, discrete-measurement KF.

Continuous-Time (Discrete-Measurement) Kalman Filter Algorithm. For the system model given by Eqs. (24) and (20) and the modified Assumptions 1C′–4C′, the evolution of the conditional mean $\hat{s}_{t|t}$ and the state error covariance matrix $P_{t|t}$, *during* an interval $t_k \geq t > t_{k+1}$ *between* two consecutive measurement sampling epochs t_k and t_{k+1}, satisfies the differential equations

$$d\hat{s}_{t|t}/dt = F_t \hat{s}_{t|t}$$

and

$$dP_{t|t}/dt = F_t P_{t|t} + P_{t|t}F_t^{\mathrm{T}} + G_{(\mathrm{c})t} Q_t G_{(\mathrm{c})t}^{\mathrm{T}}.$$

At a measurement epoch t_k (say $= t$), the estimates of the state vector and state error covariance matrix are

$$\hat{s}_{t|t} = \hat{s}_{t|t-} + K_t[m_t - H_t \hat{s}_{t|t-}] \tag{26}$$

and

$$P_{t|t} = P_{t|t-} - K_t H_t P_{t|t-}, \tag{27}$$

respectively, where

$$K_t = P_{t|t-}H_t^{\mathrm{T}}\left[H_t P_{t|t-} H_t^{\mathrm{T}} + R_{(\mathrm{c})t}\right]^{-1}. \tag{28}$$

Above, $\hat{s}_{t|t-}$ (or $P_{t|t-}$) denotes the value of $\hat{s}$ (or of P) measurement-updated at t_{k-1} and then time-propagated up to $t = t_k$. At the initial time $t = t_0$ the solution process is started while using

$$\bar{s}_0 = E[s_0]$$

and

$$P_{0|0-} = \Sigma_0 = E\left[(s_0-\bar{s}_0)(s_0-\bar{s}_0)^{\mathrm{T}}\right]$$

as the initial conditions, which are assumed to be known.

As in the discrete-time KF, the matrices F_t, $Q_{(\mathrm{c})t}$, and $R_{(\mathrm{c})t}$ are also assumed to be known, but they differ from the corresponding matrices of a discrete-time model even for the same dynamic system.

To handle (on a computer) the integration of differential equations implied by the propagation forward in time of estimates, a common approach is (a) to construct a discrete-time system model corresponding to the continuous-time system model defined by Eqs. (24) and (20) and (b) then to apply the discrete-time KF algorithm to this discrete-time system model. Heuristically this can be done by considering a time interval $t_{k-1} \le t \le t_{k-1} + \Delta t_{k-1} = t_k$ and replacing ds_t, dt, and $d\beta_t$ in Eq. (24) with $s_k - s_{k-1}$, Δt_{k-1}, and $\beta_k - \beta_{k-1}$, respectively, which yields

$$s_k \simeq \Phi_{k-1} s_{k-1} + G_{k-1} w_k$$

if, for all $k = 0, 1, 2, \ldots$, one defines $\Phi_k \equiv \mathrm{I} + \Delta t_k F_k$, $G_k \equiv \Delta t_k G_{(c)k}$, $Q_k \equiv \Delta t_k Q_{(c)k}$, and $w_k \equiv \beta_k - \beta_{k-1}$.

III. MODIFICATIONS OF KALMAN FILTERS AND NONLINEAR FILTERING

A. Filter Divergence

Discrepancies between the actual performance (the actual quality of estimates) of a filter and its apparent performance as indicated by the estimated state error covariance matrices $P_{k|k-1}$ and $P_{k|k}$ are called filter divergence. One variation of this phenomenon, known as apparent divergence, occurs when the actual estimation errors of state, although they remain bounded, become by at least an order of magnitude larger than indicated by P matrices. In such a case, performance of the KF can often be adequately improved by means of small parametric or structural adjustments in the linear system model, by changing the characterization of noise (R and Q matrices) or the initialization values s_0 and P_0 of estimates, or by switching to more accurate numerical procedures. Collectively such improvements are referred to as filter turning. What is important to note is that apparent divergence usually does not require changing from the linear system model or the basic KF algorithm.

A more serious situation, called true divergence, arises when the state error covariance estimates $P_{k|k-1}$ and $P_{k|k}$ remain bounded, while the actual errors in the state estimates start growing without bounds. True divergence may be caused by the numerical instability of filter algorithms or by modeling flaws. For example, certain critical factors or state components have not been included in the model, the system dynamics equation or the measurement-to-state transformation is incorrectly defined, or the noise model is structurally or distributionally incorrect. Numerical instability can usually be removed by resorting to better numerical algorithms. (This problem is briefly discussed in Section IV.) Modeling flaws can be detected by means of simulation in which the filter constructed on the basis of the assumed (say, linear) model is tested against a scenario generated while using a true (say, nonlinear) model. Model inadequacies arising from inherent nonlinearities in the modeled system, if they are not too strong, can often be compensated for by the techniques that do not require complete abandonment of the KF algorithm. One may be able to linearize locally the system model and then use an extended form of the KF algorithm. Otherwise, if they cannot be removed by such techniques, one must resort to nonlinear filtering, which is characterized by the use of nonlinear models for system dynamics and (or) measurements. By comparison with the KF, nonlinear filtering is considerably more difficult to attain both theoretically and practically.

B. Filter Tuning and Modifications

Certain deviations of the assumed linear system model or noise from the actual behavior of the modeled system (from its true model), leading to apparent divergence, can often be compensated for by means of a combination of the following techniques: (a) modifying, typically raising, the assumed levels of process or measurement noise through appropriate changes in noise covariance matrices Q and R; (b) bounding from below the main-diagonal elements of state error covariance matrix P and then appropriately adjusting the off-diagonal elements; (c) introducing the so-called fading (finite) memory filtering or overweighting the most recent measurements; (d) whitening (decorrelating) the process or measurement noise, also removing singularity or near-singularity from matrix R; (e) adding more states; that is, increasing the length of the state vector. The foregoing list of possibly useful measures is by no means complete.

Raising the level of process noise by an appropriate modification of matrix Q often compensates for small nonlinearities and factors, such as states, unaccounted for in the assumed model. Raising or lowering the assumed level of measurement noise by modifying matrix R forces the filter to decrease or increase, respectively, the dependence on measurements.

Bounding from below the estimates of state error variances (the main-diagonal elements of matrix $P_{k|k}$) and then correspondingly adjusting the estimated covariances (the remaining elements of $P_{k|k}$) usually increases the numerical stability of the estimation process and protects the filter from blowing up. For example, the estimation process can easily be destabilized by the cumulative effect of rounding errors that destroys the positive definiteness or symmetry of matrix P. Also, as the estimation process progresses, the elements of matrix P tend to become smaller; consequently, the filter starts to depend too

much on the assumed model and to little on the incoming measurements.

Fading memory techniques and the overweighting of the most recent measurements (discussed in Maybeck's Volume 2) help in situations where, for example, the true system model changes over time, perhaps in a manner dependent on the system state. In such a case we may want to suppress the effects of more remotely past estimates on the current estimates.

Next we briefly examine how to handle several situations that arise when some of the distributional assumptions 1D–4D (or 1C–4C) are violated. To compensate for nonwhiteness (autocorrelation over time) in the measurement noise $\{v_k\}$ or in the process noise $\{w_k\}$ one can resort, at the cost of increased computing load, to one of several known "whitening" techniques. For example, if the measurement noise $\{v_k\}$ is nonwhite so that the covariance matrix $E[v_j v_k^{\mathrm{T}}]$ is not zero for some $j \neq k$, one can conceptually model the process $\{v_k\}$ as a finite-dimensional system driven by white noise. This can be done only by adding new states to the overall system model and, thereby, increasing the processing cost. On the other hand, it saves the optimality properties of the filter.

A different situation arises when both noise processes $\{w_k\}$ and $\{v_k\}$ are white but the measurement noise covariance matrix R_k is nonnegative definite with nullity q (has q zero eigenvalues) for all k. Then (as shown by Anderson and Moore) q functionals of the state are exactly known and the length of the state vector can be shortened by q. If the covariance matrices are nonsingular, one can construct a suboptimal filter of reduced dimension.

Another pathological situation, violating the distributional Assumption 3D or 3C, comes up in feedback control applications when the process and measurement noise processes $\{w_k\}$ and $\{v_k\}$ are correlated or even identical. This problem can be solved by modifying the KF covariance processing equations (15), (16), and (18).

Filter tuning and stabilization techniques mentioned in the preceding paragraphs are critically important in applications work, that is, in designing filters for real-life situations and then making them work properly. Actually, every such development requires at least some tuning.

C. Nonlinear System Model

We now generalize the discrete-time system model defined in terms of Eqs. (2) and (3) to

$$s_k = f_{k-1}(s_{k-1}, u_{k-1}) + G_{k-1}(s_{k-1})w_{k-1} \tag{29}$$

and

$$m_k = h_k(s_k) + v_k. \tag{30}$$

We do it by replacing the linear terms $\Phi_{k-1}s_{k-1}$ and $B_{k-1}u_{k-1}$ of the linear dynamics model by a function $f_k(\cdot, \cdot)$, allowing the process noise-to-state transformation matrix G_{k-1} to depend on system state, and generalizing the state-to-measurement transformation matrix H_k to a function $h_k(\cdot)$. The subscripts attached to f, G, and h emphasize that these functions may be of time-dependent form. Furthermore, f may be nonlinear in s and possibly in u; h may be nonlinear in s. The same distributional assumptions as made for the linear discrete-time model (Assumptions 1D–4D) are used for the nonlinear model defined by Eqs. (29) and (30). Now as before, the noise is included in the model through additive terms; also, functions f and h in general are vector-valued so that their dimensions, respectively, agree with the lengths of vectors s and m. More general nonlinear system models are possible.

One could similarly define a continuous-time nonlinear system model, but we do not do it in this chapter.

D. General Approach to Nonlinear Filtering

Nearly all presently known approaches to nonlinear filtering are based, in one way or another, on the application of Bayes recursive probability equations (12a)–(12c) to a nonlinear system model, say to one of the form defined by Eqs. (29) and (30). The underlying problem is then to estimate from measurements $M_k = \{m_0, m_1, \ldots, m_k\}$ the evolution of the posterior probability densities $p(s_k \mid M_k)$ over t_k for $k = 0, 1, \ldots$. Knowing $p(s_k \mid M_k)$ at each sampling epoch t_k enables one to apply the optimization criterion of a desired type (such as the MMSE, the maximum a posteriori, or one of some other type) to obtain the corresponding estimates of state.

When the low-order moments or some other statistics are sufficient in the sense that they define or adequately approximate $p(s_k \mid M_k)$, optimal or nearly optimal filtering schemes can be derived for certain limited classes of problems. Typically, this can be done when a combination of the following conditions exists: (a) the nonlinearities in Eqs. (29) and (30) locally can be well approximated at each t_k by a linearized model (e.g., by a Taylor series approximation to the original model); (b) an adequate approximation to the state trajectory (known as a nominal state trajectory) is a priori available; (c) the actual noise levels are low and both noise processes are Gaussian. Several versions of the linearized extensions of the KF discussed below, are the most commonly used representatives of such nonlinear estimators. In general, although the manner in which $p(s_k \mid M_k)$ evolves can be described by means of difference (in discrete-time case) or differential (in continuous-time case) equations, these equations usually are too complex to be solved for practical implementation analytically in

closed form, or even numerically. This is especially true for real-time filter implementations. Because of these difficulties, much of the research thrust has been in the direction of developing tractable approximations to $p(s_k \mid M_k)$.

E. Linearized Extensions of Kalman Filters

If the functions f and h are sufficiently smooth and can be locally expanded in Taylor series with respect to the estimates or with respect to an a priori generated nominal time history of system states, then one can apply the KF algorithm to the linearized system model

$$s_k = F_{k-1}s_{k-1} + G_{k-1}w_{k-1} \tag{31}$$

and

$$m_k = H_k s_k + v_k, \tag{32}$$

with the matrices F, G, and H defined in terms of their (i, j)th elements as

$$\begin{aligned} F_k &\equiv F_k[s_{0k}, u_k] \equiv ((F_{i,j)t}))_{t=t_k} \\ &\equiv \left(\left(\delta f_{(i)t}(s, u)\big/\delta s_{(j)t}\right)\right)_{s=s_{0k},u=u_k,t=t_k}, \end{aligned} \tag{33}$$

$$G_k \equiv G_k(s_{0k}) \equiv \left(\left(G_{(i,j)t}(s)\right)\right)_{s=s_{0k},t=t_k}, \tag{34}$$

and

$$H_k \equiv H_k[s_{0k}] \equiv \left(\left(\delta h_{(i)t}(s)\big/\delta s_{(j)}\right)\right)_{s=s_{0k},t=t_k}. \tag{35}$$

Here we introduce the notational convention $X = ((X_{(i,j)t}))$ to denote matrix X at time t in terms of its (i, j)th element $X_{(i,j)t}$. For example, the right-hand side of Eq. (33) indicates that the (i, j)th element of matrix F_k is the partial derivative at $t = t_k$ of the ith component of f_k with respect to the jth component of s; the subscript expression to the right of double parentheses specifies at what point these partial derivatives are to be evaluated. In (33), this point is

$$s = s_{0k}, \qquad u = u_k, \qquad t = t_k,$$

where: $s = s_{0k}$ indicates the value of the nominal state vector s_0 at $t = t_k$; $u = u_k$ is the value of control u at $t = t_k$.

The two most common choices for the nominal state trajectory $\{s_{0k}\}$ are as follows: (a) $s_{ok} \equiv \hat{s}_{k|k}$, or $\hat{s}_{k|k-1}$, which yields the extended Kalman filter (EKF), or (b) $\{s_{0k}\}$ is a priori set to be a nominal state trajectory, which gives the so-called linearized Kalman filter (LKF). As an example, we next state in detail the algorithm for the discrete-time EKF, which is similar in form to the discrete-time KF algorithm defined in terms of Eqs. (13)–(18).

Discrete-Time Extended Kalman Filter Algorithm. At each $t_k = t_{k-1} + \Delta t_{k-1}$ $(k = 0, 1, 2, \ldots)$ proceed as follows.

a. If $k = 0$ (i.e., at the initial time t_0), initialize the estimation process by setting

$$\hat{s}_{0|-1} = \bar{s}_0 \qquad \text{and} \qquad P_{0|-1} = \Sigma_0, \tag{36}$$

where $\bar{s}_0$ and Σ_0 are as defined by Eq. (6).

b. If $k \geq 1$, propagate the estimates of system state vector s and the state error covariance matrix P from t_{k-1} to t_k by computing

$$\hat{s}_{k|k-1} = f_{k-1}(\hat{s}_{k-1|k-1}, u_{k-1}) \tag{37}$$

and

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^{\mathrm{T}} + G_{k-1}Q_{k-1}G_{k-1}^{\mathrm{T}}, \tag{38}$$

respectively.

c. To update the estimates, first compute

$$H_k = H_k[\hat{s}_{k|k-1}] \tag{39}$$

according to Eq. (35) and evaluate the Kalman gain matrix

$$K_k = P_{k|k-1}H_k^{\mathrm{T}}\left[H_k P_{k|k-1}H_k^{\mathrm{T}} + R_k\right]^{-1}. \tag{40}$$

Next use the current measurements m_k to update the estimate of s by means of

$$\hat{s}_{k|k} = \hat{s}_{k|k-1} + K_k[m_k - h_k(\hat{s}_{k|k-1})]. \tag{41}$$

Then update the estimate of state error covariance matrix P by computing

$$P_{k|k} = [I - K_k H_k]P_{k|k-1}. \tag{42}$$

d. Finally, compute the matrix

$$F_k \equiv F_k[\hat{s}_{k|k}, u_k] \tag{43}$$

according to Eq. (33) and evaluate the process noise-to-state transformation matrix

$$G_k \equiv G_k[\hat{s}_{k|k}]. \tag{44}$$

The above-stated discrete-time EKF reduces to the standard discrete-time KF with its all-optimality properties if the *true* system model is linear, that is, if the true model is defined by Eqs. (2) and (3). When the EKF algorithm is applied to a nonlinear system whose true model is of the form given by Eqs. (29) and (30), the resulting estimator is no longer linear or optimal, which makes it more prone to various problems. For example, if the time step Δt used is too large to accommodate nonlinearities in system dynamics or in measurements, the estimates may be biased or even the entire estimation process may be destabilized by filter divergence. Thus, the tuning and the performance validation of an EKF or an LKF typically require considerably more work (simulation and analysis) than performing the same tasks for a KF applied to a truly linear model.

In addition to the LKF and the EKF several other types of linearized extensions of the KF are known. One class

of variations can be derived by including more than the first-order term in the Taylor series expansions of functions $f_k(s_k, u_k)$ or $h_k(s_k)$. For example, the estimation algorithm obtained by including the first- and second-order terms is known as the second-order extended (or linearized) KF. Another variation can be constructed by using several, say I, iterations to perform the measurement update of state with each set of measurements m_k. In such a scheme, step c of the discrete-time EKF algorithm is repeated I times. One enters the ith pass of this step with $\hat{s}_{k|k}(i-1)$, then computes $\hat{s}_{k|k}(i)$ and all other quantities dependent on $\hat{s}_{k|k}(i)$. The first iteration is entered by defining $\hat{s}_{k|k}(0) \equiv \hat{s}_{k|k-1}$. At the end of the Ith iteration, one defines $\hat{s}_{k|k} \equiv \hat{s}_{k|k}(I)$.

Although the measures described in the foregoing two paragraphs often stabilize the EKF and produce satisfactory results, they fail for some nonlinear or non-Gaussian systems. The cause of such failures is usually ascribed to two kinds of approximations used in the construction of EKF: (a) the linearization of system equations relative to some reference values and (b) the assumption of Gaussianity for all a priori distributions in the system model. Approximations of the second kind enforce the retention of a false illusion that the probability densities $p(s_k \mid M_k)$ continue to be Gaussian for all k. For example, if the true conditional distribution of state is multimodal [i.e., densities $p(s_k \mid M_k)$ have multiple peaks], the EKF functions more as a maximum likelihood than as an MMSE estimator and the estimated mean may end up following one (if any) of the peaks.

EXAMPLE 2. ESTIMATING THE MOTION AND THE DRAG CHARACTERISTICS OF A BALLISTIC PROJECTILE ENTERING THE ATMOSPHERE. Consider the problem of estimating the motion and the drag (air resistance) characteristics of a *ballistic projectile* (BP). To simplify the problem without robbing from it its illustrative value, assume that the BP is aimed at the North Pole and is approaching the earth along the polar axis. We denote this axis the Y axis of the one-dimensional coordinate system whose origin is assumed to coincide with the North Pole and whose positive direction points "upward" from the pole.

Notation. In this example, the individual components of a vector are referred to by means of parenthesized subscripts. For example, if z is a column vector of length 2, we write $z = [z_{(1)}, z_{(2)}]^{\mathrm{T}}$. In particular, to refer to the values of these components at time t, we write either $[z_{(1)t}, z_{(2)t}]$ or $[z_{(1)}, z_{(2)}]_t$. If t represents a sampling epoch, say t_k, we write either $[z_{(1)k}, z_{(2)k}]$ or $[z_{(1)}, z_{(2)}]_k$. Another notational convention adopted here is the use of dots above a symbol to denote the derivatives with respect to time of the quantity represented by the symbol. For example, $\dot{y}_k$ and $\ddot{y}_k$, respectively, denote the values of dy/dt and d^2y/dt^2 at time t_k. In addition, we use "hats" over lowercase letters to denote unit vectors. For example, $\hat{v}$ denotes the unit vector in the direction of velocity vector v.

Physical Model. We model the earth as a perfect sphere of radius

$$R_E = 6{,}378{,}135\,(\mathrm{m}).$$

For the present problem, the gravitational field at any point P on the polar (Y) axis, y meters above sea level, is approximated by

$$-g(y)\hat{y} = -(g_0)(C)(1.0 + y/R_E)^{-2}\hat{y}\ (\mathrm{m/sec}^2),$$

where

$$g_0 = 9.7803327\,(\mathrm{m/sec}^2)$$

and

$$C = 1.00530246.$$

The air density at P (y meters above sea level) is approximated by

$$\rho \equiv \rho(y) = \rho_0 \exp(-\gamma_y)\,(\mathrm{kg/m}^3),$$

where

$$\rho_0 = 1.22\,(\mathrm{kg/m}^3),$$
$$\gamma = 1.6404 \times 10^{-4}\ (\mathrm{m}^{-1}).$$

The drag force on the BP, divided by its unknown mass and moving with velocity $\dot{y}$, is approximated by

$$d \equiv d(-\hat{v}) = (\dot{y})^2[\rho(y)](D_F)\ (\mathrm{m/sec}^2),$$

where $\hat{v}$ is the unit vector in the direction of velocity, and D_F is the unknown drag factor (m/kg). In the present case, $\hat{v} = -\hat{y}$. Hence, the drag can be written

$$d(y, \dot{y})\hat{y} = (\dot{y})^2[\rho(y)](D_F)\hat{y}.$$

It follows that the differential equation of motion for the BP is of the form

$$\ddot{y} = -g(y) + d(y, \dot{y}) + \text{process noise}.$$

System Model. Analysis of the physical model suggests the use of a four-dimensional ($n_s = 4$) state vector (SV) defined by letting

$$s_{(1)} \leftrightarrow y, \qquad s_{(2)} \leftrightarrow \dot{y}, \qquad s_{(3)} \leftrightarrow \ddot{y}, \qquad s_{(4)} \leftrightarrow D_F.$$

The state transition equations for this SV can be directly written from the time-propagation equations,

$$\begin{aligned} y_{k+1} &= y_k + h\dot{y}_k + (0.5)h^2\ddot{y}_k + w_{(1)k}, \\ \dot{y}_{k+1} &= \dot{y}_k + h\ddot{y}_k + w_{(2)k}, \\ \ddot{y}_{k+1} &= \ddot{y}_k + h\dddot{y}_k + w_{(3)k}, \\ (D_F)_{k+1} &= (D_F)_k + w_{(4)k}, \end{aligned}$$

where

$$h \equiv \Delta t_k = t_{k+1} - t_k,$$
$$\ddot{y}_k = \{(d/dy)[-g(y)]\dot{y} + (d/dy)[d(y, \dot{y})]\dot{y} + (d/d\dot{y})[d(y, \dot{y})]\ddot{y}\}_k,$$

is a function of y, $\dot{y}$, $\ddot{y}$, and D_F (all evaluated at $t = t_k$); $w_{(j)k}$ are components of process noise vector w_k. Hence writing

$$s_{(j)k+1} = f_{(j)}(s_k) + w_{(j)k} \qquad \text{for} \qquad j = 1, \ldots, n_s,$$

we get

$$f_{(1)}(s_k) = \left(s_{(1)} + hs_{(2)} + (0.5)h^2 s_{(3)}\right)_k,$$
$$f_{(2)}(s_k) = \left(s_{(2)} + hs_{(3)}\right)_k,$$
$$f_{(3)}(s_k) = \left(s_{(3)} + h\dot{s}_{(3)}\right)_k,$$
$$f_{(4)}(s_k) = \left(s_{(4)}\right)_k,$$

where, as it follows from a previous expression for $\ddot{y}_k$, the time derivative $\dot{s}_{(3)k}$ can be expressed in terms of $s_{(1)}$, $s_{(2)}$, $s_{(3)}$, and $s_{(4)}$ (all evaluated at t_k).

We assume that at each observation epoch t_k a ground-based sensor measures both the position (y) and the velocity ($\dot{y}$) of the incoming ballistic missile. Hence the measurement equations are of the form

$$m_{(1)k} = y_k + v_{(1)k},$$
$$m_{(2)k} = \dot{y}_k + v_{(2)k},$$

or (in vector/matrix form)

$$m_k = Hs_k + v_k.$$

Here v_k represents the measurement noise and

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Filter implementation requires a procedure for evaluating the F matrix. As noted in the description of the EKF algorithm, the (i, j)th element of F_k is obtained by differentiating $f_{(i)}$ with respect to $s_{(j)}$ and then evaluating this derivative at t_k. For example, in the present case, we get

$$F_{(1,2)k} = \left(\delta f_{(1)} / \delta s_{(2)}\right)_k = h = \Delta t.$$

In our implementation of the filter for the present problem, we included an algorithm for bounding the state error covariance matrix P. (This algorithm is described in Maybeck's Volume 2.) We found it to be effective in stabilization of the filtering process.

Results of a Simulation Run. Figures 2A–E illustrate the results of a computer simulation run performed for the parameter values and the initial conditions summarized in Fig. 2A. The filter was tuned up to respond rapidly to relatively accurate measurements. Note that the drag factor D_F (expressed as square meters per kilogram) is treated as an unknown constant parameter that is estimated along with the dynamic variables of system state.

F. Other Approaches to Nonlinear Filtering and the Gaussian Sum Approximation Filter

Since it is generally impossible to construct for a nonlinear or non-Gaussian system an exact representation of a posteriori probability density functions (PDFs) $p(s_k \mid M_k)$, various types of finite approximations to $p(s_k \mid M_k)$ have been investigated. As an illustration of this approach we outline next one specific technique, known as the Gaussian sum approximation (GSA).

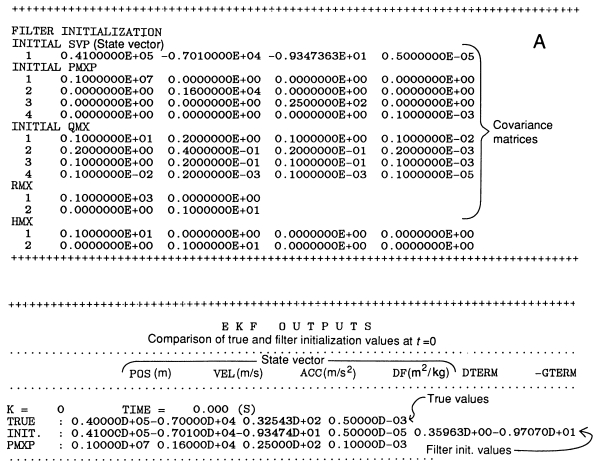
Let R^n denote the n-dimensional Euclidean space. The GSA method is based on a theoretical result, according to which any probability function p of a vector (or scalar-)-valued RV x, say of length n, can be approximated as closely as desired in the $L_1(R^n)$ space (i.e., the integral $\int_{R^n} | p(x) - p_A(x) | dx$ can be made arbitrarily small) by a function of the form

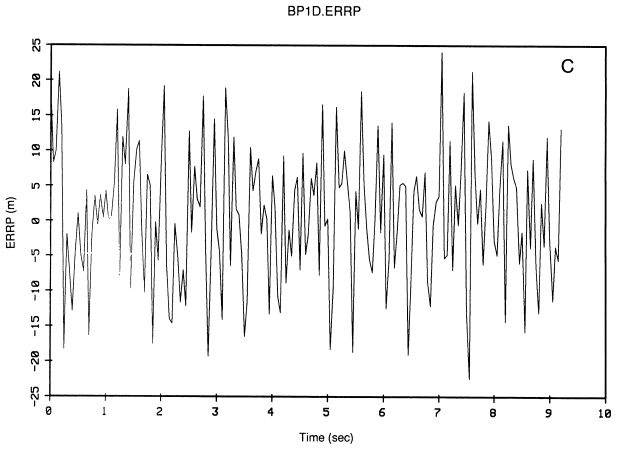
$$p_A(x) = \sum_{i=1}^{L} \alpha_i N[x - a_i, B_i] \tag{45}$$

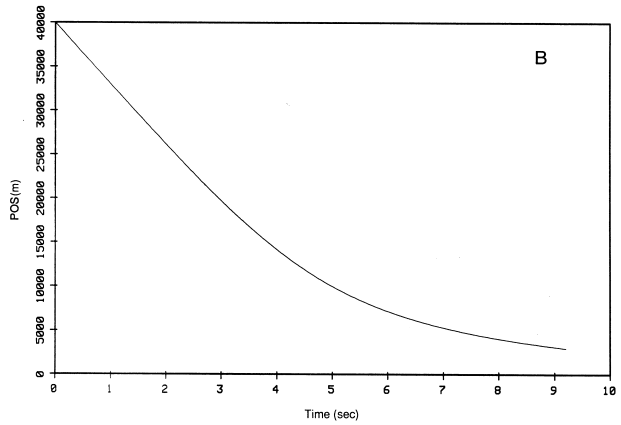
for some positive integer L, positive scalars α_i such that $\alpha_1 + \alpha_2 + \cdots + \alpha_L = 1$, suitably selected n-dimensional vectors a_i, and $n \times n$ positive definite matrices B_i. In Eq. (45), $N[x - a_i, B_i]$ represents the n-variate Gaussian PDF

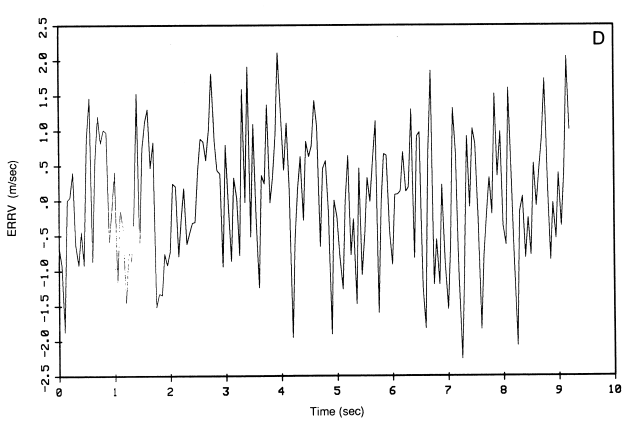
$$(2\pi)^{-n/2} |B_i|^{-1/2} \exp\left\{-\tfrac{1}{2}(x - a_i)^{\mathrm{T}} B_i^{-1} (x - a_i)\right\}, \tag{46}$$

with $|B_i|$ denoting the determinant of B_i. Furthermore, it can be shown that $p_A(x)$ converges uniformly in x to any PDF of practical interest as the covariance matrices B_i tend to a zero matrix and L increases. There are many ways of selecting the parameters a_i, B_i, and α_i. For construction of a filtering scheme, it is convenient (a) to choose the values of a_i so as to build up in the state space a grid covering at least a substantial part of the mass of the PDF to be approximated and (b) to set $B_i = bI$ (with I representing an identity matrix) for some small positive scalar b. Then each term in the Gaussian sum on the right-hand side of Eq. (45) converges to an impulse function centered at a_i as $b(>0)$ converges down to zero. Hence, for a small $b > 0$, each term in the Gaussian sum is practically zero outside a small open neighborhood (hypersphere) in R^n containing a_i. Since $p_A(x) \geq 0$ and $\int p_A(x)\, dx = 1$, p_A is a PDF.









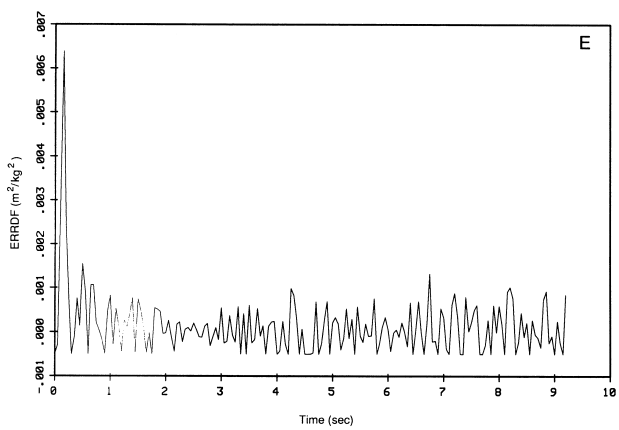


FIGURE 2 (A) The upper part shows the state vector and filter matrices at the initialization time t_0. The lower part compares the true and initialized state vectors at t_0. (B) True position (the Y coordinate) of the ballistic projectile vs time. (C) Error in position estimates vs time. Because of the compression of time scale, the initialization error of 1 km is invisible in the graph. (D) Error in velocity estimates vs time. Because of the compression of time scale, the initialization error of 10 m/sec is not visible on the graph. (E) Error in D_F estimates vs time.

Next suppose that $p(s_k \mid M_k)$ is approximated by

$$p_A(s_k \mid M_k) = \sum_{i=1}^{L} \alpha_{ik} N[s_k - \bar{s}_{ik}, B_{ik}]. \quad (47)$$

Then $p_A(s_k \mid M_k)$ is also a PDF, and while using appropriately selected values of $\bar{s}_{ik}$, one can show that

$$\hat{s}_{k|k} \equiv E[s_k \mid M_k] = \sum_{i=1}^{L} \alpha_{ik} \bar{s}_{ik} \quad (48)$$

and

$$P_{k|k} \equiv E\big[(s_k - \hat{s}_{k|k})(s_k - \hat{s}_{k|k})^{\mathrm{T}} \big| M_k\big] = \sum_{i=1}^{L} \alpha_{ik}\big[B_{ik} + (\hat{s}_{k|k} - \bar{s}_{ik})(\hat{s}_{k|k} - \bar{s}_{ik})^{\mathrm{T}}\big]. \quad (49)$$

This suggests that perhaps the conditional MMSE estimator $\hat{s}_{k|k}$ can be approximated as a weighted sum of conditional mean estimators, each representing an EKF. It can be proved that this is indeed the case. Combining the foregoing ideas with the nonlinear system model given by Eqs. (29) and (30) and Assumptions 1D–4D, the latter now possibly relaxed by dropping some or all Gaussianity assumptions, leads to a filtering scheme, which we outline next.

Discrete-Time Gaussian Sum Approximation (GSA) Filter Algorithm. This algorithm applies to the system model defined by Eqs. (29) and (30) and by the probabilistic assumptions 1D through 4D. Let n_s be the dimension of state vectors.

a. Let $k = 0$ (i.e., initialize the sampling time t_k to t_0) and select appropriate values of the parameters $\bar{s}_{ik}$ and $\alpha_{i,k-1}$ (all $\alpha_{i,k-1} \geq 0$ and $\sum_{i=1}^{L} \alpha i, k-1 = 1$) for $i = 1, \ldots, L$.

b. To complete a GSA,

$$\sum_{i=1}^{L} \alpha_{i,k-1} N[s - \bar{s}_{ik}, \bar{B}_{ik}]$$

to $p(s_k \mid M_{k-1})$, where $M_k = \{m_k, M_{k-1}\}$ and M_{k-1} is an empty measurement set when $k = 0$, initialize (reinitialize) all L covariance matrices $\bar{B}_{ik}$ to bI, with b representing a small positive constant and I the $n_s \times n_s$ identity matrix.

c. Approximate $p(s_k \mid M_k)$ by means of the following measurement-update procedure. For $i = 1, \ldots, L$, compute

$$\begin{aligned} H_{ik} &= ((\delta h_k(s)/\delta s))_{s=\bar{s}_{ik}}, \\ D_{ik} &= \big(H_{ik}\bar{B}_{ik}H_{ik}^{\mathrm{T}} + R_k\big)^{-1}, \\ K_{ik} &= \bar{B}_{ik}H_{ik}^{\mathrm{T}}D_{ik}, \\ B_{ik} &= \bar{B}_{ik} - \bar{B}_{ik}H_{ik}^{\mathrm{T}}D_{ik}H_{ik}\bar{B}_{ik}^{\mathrm{T}}, \\ s_{ik} &= \bar{s}_{ik} + K_{ik}[m_k - h_k(\bar{s}_{ik})], \end{aligned}$$

with the new weights determined by

$$S_k = \sum_{j=1}^{L} \alpha_{j,k-1} N\big[m_k - h_k(\bar{s}_{jk}), D_{ik}^{-1}\big]$$

and

$$\alpha_{ik} = \alpha_{i,k-1} N\big[m_k - h_k(\bar{s}_{ik}), D_{ik}^{-1}\big] \big/ S_k.$$

Finally, compute $\hat{s}_{k|k}$ and $P_{k|k}$ by means of Eqs. (48) and (49), respectively.

d. Approximate $p(s_{k+1} \mid M_k)$ by means of the following time-update procedure:

$$\sum_{i=1}^{L} \alpha_{ik} N[s_{k+1} - \bar{s}_{i,k+1}, \bar{B}_{i,k+1}],$$

where

$$\begin{aligned} F_{ik} &= ((\delta f_k(s)/\delta s))_{s=s_{ik}}, \\ \bar{s}_{i,k+1} &= f_k(s_{ik}), \\ \bar{B}_{i,k+1} &= F_{ik}B_{ik}F_{ik}^{\mathrm{T}} + G_k(s_{ik})Q_kG_k(s_{ik})^{\mathrm{T}}. \end{aligned}$$

e. Let $k \leftarrow k+1$ (i.e., increment the sampling time by setting $t_k \leftarrow t_{k+1}$). If the new $t_k > t_{\max}$ (= estimation end time), then *quit*; otherwise, proceed to step f.

f. Test whether $\bar{B}_{ik} < bI$ (i.e., whether $[bI - \bar{B}_{ik}]$ is a positive definite matrix) for all i. If yes, go to step c; otherwise, go to step b.

In the preceding algorithm, use the same rule for computing matrices H_{ik} and F_{ik} as given for matrices H_k and F_k [see the text following Eqs. (33)–(35)]. As the above algorithm indicates, both the measurement-update and the time-propagation equations for each constituent filter in the Gaussian sum are essentially those of an EKF; besides, when any covariance matrix $\bar{B}_{ik}$ becomes too "big" (in the sense explained in step f), all L covariance matrices $\bar{B}_{ik}$ must be reinitialized. If $L = 1$, the GSA filter reduces to an EKF.

The following two important advantages of the GSA come from the choice of the Gaussian weighting PDFs and the positive weighting coefficients that add to 1: (a) any finite Gaussian sum is a PDF; (b) the presence of Gaussian weighting PDFs allows the utilization of certain Gaussian properties in the computation of approximations to $p(s_k \mid M_k)$, though the resulting approximation function p_A is not Gaussian. However, the GSA approach has one disadvantage: The series obtained from the Gaussian sum expansion cannot be orthogonalized. Consequently, it is difficult to obtain useful error bounds. Thus, as an alternative to the GSA, the use of approximations based on orthogonal series expansions (e.g., the Edgeworth and Gram–Charlier series) has been investigated. Despite the attractiveness of orthogonality, such approaches have one

distinct disadvantage. The resulting truncated sum approximations are not PDFs.

Finally, we should mention the nonlinear filtering methods based on the assumption that $p(s_k|M_k)$ is Gaussian for all k. The EKF was one example of this approach. As already noted, this assumption often leads to discarding a great deal of information present in the actually true PDF, especially in the multimodal case.

IV. FILTER IMPLEMENTATION

Filters of the type considered in this chapter are usually implemented in software form. Besides a filter is typically designed for ultimate implementation as part of a real-time system, which means that it will be subject to timing constraints of that system. Thus, in addition to tasks already discussed (e.g., defining the system model, selecting a filter algorithm, and making a trial filter work), one must (a) select appropriate numerical procedures to be used with the filter algorithm and (b) adapt the program implementing the filter to the real-time processing environment in case the filter is to become part of a real-time system. In the latter case, one must also time and size the algorithms for a particular real-time computer.

A. Numerical Procedures

The accuracy, stability, and computing efficiency of a filter depend critically on the adopted numerical procedures. For example, the unnecessary use of multiple precision arithmetic may be prohibitively expensive in some real-time applications; on the other hand, careless computation of dot products while using operands of insufficient length may quickly lead to an unbounded accumulation of roundoff error and destabilization of the estimation process. As another example, consider a continuous-time nonlinear filter. It often uses numerical integration to propagate the estimates of the state to the next time point, and thus one faces the problem of selecting an appropriate method for propagating numerically the solution of underlying differential equations.

There are several simple rules of thumb with regard to matrix operations in Kalman filtering of linear systems: (a) Dot products should in general be computed using arithmetic of a higher precision than used in the remaining matrix operations; (b) matrices that theoretically are symmetric should be periodically resymmetrized [this simply requires replacing each member of every pair of elements a_{ij} and a_{ji}, where $i \neq j$, of a theoretically symmetric matrix A with $(a_{ij} + a_{ji})/2$]; (c) positive definiteness of theoretically positive definite matrices should be preserved. Furthermore, in many applications, such as in precision navigation, the state must be known to many more significant digits than the gain and the state error covariances. Hence, nearly all gain–covariance processing can be performed at a lower precision than the processing of the state vector, but this approach is safe only if the gain–covariance processing is computationally stable. Square root filtering, discussed next, provides a numerically stable method for preserving the positive definiteness in state error covariance matrices.

B. Square Root Filtering

By far the greatest trouble spot in computer mechanization of the KF is the updating of state error covariance matrix P, that is, the computation of $P_{k|k}$ according to Eq. (18). As the estimation process progresses, the elements of $P_{k|k}$ typically continue to decrease in magnitude and so matrix $P_{k|k}$ keeps approaching the zero matrix, although theoretically it should forever remain positive definite, no matter how small in magnitude its elements become. Hence, unless special measures are taken, accumulation of roundoff error in the repetitive use of Eq. (18) may cause the computed $P_{k|k}$ to lose its positive definiteness. As suggested by the matrix inversion operation appearing on the right-hand side of Eq. (16) for computing the Kalman gain, this situation is aggravated if several components of the measurement vector are very accurate and consequently the positive definite measurement error covariance matrix R is ill conditioned, that is, if R has eigenvalues of both relatively very large and small magnitudes.

Let A be a nonnegative definite symmetric matrix; then there exists a matrix S such that $A = SS^{\mathrm{T}}$. Matrix S is often called the square root of A. The Cholesky decomposition algorithm provides a method of constructing from A the matrix S so that S is lower triangular; that is, all elements of S above the main diagonal are zero. Square root filtering is motivated by the observations that, if the state error covariance matrix $P = SS^{\mathrm{T}}$, then (a) since SS^{T} is always nonnegative definite, matrix P expressed as SS^{T} cannot become negative definite, and (b) matrix S is generally less ill conditioned than matrix P.

Several versions of the square root filtering algorithm are known. The earliest form was developed by J. E. Potter in 1964 for applications in which the process noise is absent (i.e., covariance matrix Q is zero) and the measurements are sequentially processed as scalars. In 1967 J. F. Bellantoni and K. W. Dodge extended Potter's results to vector-valued measurements. A. Andrews in 1968 and then S. F. Schmidt in 1970 published two alternative procedures for handling the process noise. In 1973 N. A. Carlson described a procedure that considerably improved the speed and decreased the memory requirements of square root filtering and in which, as in Potter's algorithm,

vector-valued measurements are processed sequentially as scalars. Finally, the so-called UDU^{T} covariance factorization method is the most recent major milestone in numerical handling of KFs. This method, developed by G. J. Bierman and C. L. Thornton, represents the state error covariances before and after the measurement update step as

$$P_{k|k-1} = (U_{k|k-1})D_{k|k-1}(U_{k|k-1})^{\mathrm{T}}$$

and

$$P_{k|k} = (U_{k|k})D_{k|k}(U_{k|k})^{\mathrm{T}},$$

with D being a diagonal matrix and U an upper triangular matrix with 1's on its main diagonal. In this method, the square root of the covariance matrix, which now would correspond to $UD^{1/2}$, is never computed explicitly, which avoids numerical computation of square roots. Like Carlson's algorithm, the UDU^{T} factorization method maintains the covariance matrix in factored form and so (like Carlson's algorithm) is considerably more efficient in processor time and storage than the original Potter algorithm.

As a quick comparison of computational efficiency, the conventional Kalman method, the less efficient form of Carlson's algorithm, and the UDU^{T} factorization method are roughly equal: The processing of each time step (consisting of one time propagation and one measurement update) requires of the order of $\frac{1}{6}[9n_s^3 + 9n_s^2n_m + 3n_s^2n_w]$ adds and about the same number of multiplies, plus a relatively modest number of divides and square roots (square roots are required only in some, as in Potter's or Carlson's square root algorithms). Here, as before, n_s is the length of the state vector, n_m the length of the measurement vector, and n_w the lenght of the process noise vector w. The faster version of Carlson's algorithm is more efficient and requires only of the order of $\frac{1}{6}[5n_s^3 + 9n_s^2n_m + 3n_s^2n_w]$ adds and $\frac{1}{6}[5n_s^3 + 12n_s^2n_m + 3n_s^2n_w]$ multiplies, plus $2n_sn_m$ divides and n_sn_m square roots, at each time point. The stable (Joseph) form of the KF [as given by Eq. (18′)] fares more poorly: At each time step, it requires of the order of $\frac{1}{6}[18n_s^3 + 15n_s^2n_m + 3n_s^2n_w]$ adds and about the same number of multiplies.

As a summary, (a) a square root filter is a numerically stable form for performing the KF covariance–gain processing defined by Eqs. (15), (16), and (18); (b) the efficiency of its more recent versions roughly compares with that of these three equations; (c) the increased stability allows one to use relatively low-precision arithmetic in the KF gain–covariance processing, with a possible exception of some dot products.

Real-time implementation of a filter involves additional issues that are unimportant in the non-real-time environment. Besides the adequacy of functional performance, the most important of these issues is the requirement to produce timely responses to external stimuli. Thus, resorting to a parallel or concurrent processing may be the only way out. This usually implies the use of special hardware architectures such as parallel, vector pipelined, or systolic processors.

As one example, consider the use of a filter in the tracking of multiple objects in a hard real-time environment characterized by strict deadlines. In such a case one may want to maintain simultaneously many estimation processes, each handling a single object. Parallel processors may seem to be a suitable hardware architecture for this problem, but if separate estimation processes in such an application progress at different rates and at any time some of them require a great amount of special handling, then parallel architecture, such as a single-instruction multiple-data stream computer, may not be the best choice. As another example, consider a KF to be implemented as part of a navigation system on a small airborne computer (uniprocessor). Suppose that the navigation measurements come at a certain fixed rate. If the filtering process cannot keep up with the arrival rate of measurements and so not all of them can be utilized, the estimation performance may deteriorate. In this problem, if there is an upper bound on hardware resources, the only solution may be to decompose the estimation algorithm into concurrently executable processes. For instance, the time-propagation step (which, say, is to be executed at a relatively high rate) may constitute one process and the measurement-update step (which needs to be executed only at some lower rate, say, at the rate of measurement arrivals) may constitute another. Such a decomposition of an estimation algorithm into concurrent procedures often creates a surrogate algorithm that performs more poorly than the original algorithm.

The effects of the finite-length word computing is another issue that must be considered in filter implementation for real-time applications. The computer on which a filter is developed and validated through successive off-line simulations is often more powerful and uses higher-precision arithmetic and number representations than the ultimate real-time processor. Hence, one must in advance determine in advance what effect a shorter word length will have on performance.

V. APPLICATIONS

A good part of present-day technology would be unthinkable without recursive estimators, especially modern computer-based control and communication systems. Kalman and other types of recursive filters have been widely adopted by the defense and aerospace communities for applications such as navigation, aircraft flight control, satellite tracking, and orbit determination.

Early applications of recursive filtering were mainly in the estimation and identification of uncertain dynamic or control systems. Many of these applications, although in general successful, were hastily executed in the sense that insufficient thought was given to various theoretical, numerical, and purely implementational issues. The first two decades were also marred by a flood of papers describing specific applications, published in the technical literature or presented at various engineering conferences. However, during that period many important theoretical issues, especially gaps in theoretical knowledge, were identified and resolved.

In contrast to the quick, widespread adoption of recursive estimators to control and of dynamic systems during the 1960s and 1970s, their penetration of signal processing was considerably less decisive. One reason for that was the inadequacy of computer hardware and software with respect to the stringent timing constraints of on-line signal processing, although there are several application areas of signal processing, such as the analysis of economic time series, in which real-time constraints are virtually nonexistent. Another factor was the technological culture of that period, which was also reinforced by the nature of the problem. In communication problems, one is usually interested in steady-state behavior. Alternative signal processing techniques, notably those based on frequency domain analysis, such as the fast Fourier transform, were developed and then adapted to digital signal processing. However, developments in computer hardware and software technologies (e.g., the introduction of gallium arsenide devices, very large-scale integration, and data-flow architectures such as systolic processors) promise to change this situation.

Several interesting types of functional extensions of filtering are known. We briefly mention only one of them, which is known as adaptive filtering. Adaptive filtering is concerned with the generalization of estimation algorithms, especially of KFs, to state estimation situations in which the system model is not completely known. For example, one may want to estimate the state of a stochastic dynamic system for which some parameters, such as the covariance matrices of noise processes or a few coefficients in the system dynamics equation, are unknown or are changing in an unknown way in time. A filter that estimates the state while concurrently estimating the values of the unknown components of the system model is called an adaptive filter. Example 2 illustrates a relatively simple case of adaptive filtering: estimation of the parameters that define the drag force while estimating the trajectory. In that example, the adaptive filtering problem is solved by introducing an additional component into the state vector.

SEE ALSO THE FOLLOWING ARTICLES

AIRCRAFT INSTRUMENTS • ANALOG ELECTRICAL FILTERS • APPROXIMATIONS AND EXPANSIONS • CIRCUIT THEORY • DIGITAL FILTERS • SPATIAL OBJECTIVE ANALYSIS • STOCHASTIC PROCESSES

BIBLIOGRAPHY

Anderson, B. D. O., and Moore, J. B. (1979). "Optimal Filtering," Prentice-Hall, Englewood Cliffs, NJ.

Davis, M. H. A., and Vinter, R. B. (1985). "Stochastic Modelling and Control," Chapman and Hall, New York.

Jazwinski, A. H. (1970). "Stochastic Processes and Filtering Theory," Academic Press, New York.

Kailath, T. (1974). *IEEE Trans. Info. Theory* **IT-20**(2), 146–179.

Kalman, R. E. (1960). *Trans. ASME Ser. D J. Basic Eng*. **82,** 34–35.

Kalman, R. E., and Bucy, R. S. (1961). *Trans. ASME Ser. D J. Basic Eng*. **83,** 95–107.

Maybeck, P. S. (1979, 1982). "Stochastic Models, Estimation, and Control," Vol. 1, Vols. 2 and 3, Academic Press, New York.

Schmidt, G. T. (1976). "Linear and Nonlinear Filtering Techniques," Control, and Dynamic Systems, Vol. 12, pp. 63–98, Academic Press, New York.

Thornton, C. L., and Bierman, G. J. (1980). "UDU^T Covariance Factorization for Kalman Filtering," Control and Dynamic Systems, Vol. 16, pp. 177–248, Academic Press, New York.



Power Electronics

Adrian Ioinovici
Holon Academic Institute of Technology

I. Requirements for a Power Electronics Circuit
II. Principles of Operation of a Switching-Mode Circuit
III. DC-to-DC Converters
IV. AC-to-DC Rectifiers. Power Factor Correction Circuits
V. DC-to-AC Inverters. Uninterruptible Power Supply
VI. Analysis and Design of Power Electronics Circuits

GLOSSARY

Conduction losses Energy losses due to parasitic DC resistances of inductors, capacitors, transistors, and diodes.
Current-mode/voltage-mode control A control method in which the variable that is fed to the feedback circuit is either a current (input current or the current through the main switch) or the load voltage.
Duty cycle The ratio between the duration of conduction of the main switch and the switching period.
Duty-cycle control Operation in which the duty cycle is the control variable.
Hard-switching Switching of a transistor or diode that takes place with switching losses.
Line The input voltage in a converter or inverter.
Soft-switching Switching of a transistor or diode that takes place without switching losses.
Switching frequency The frequency of turning-on the main switches.
Switching losses Energy losses incurred in the transient process of turning on/off a switch.
Switching period Inverse of the switching frequency. It is the duration of a cycle.
Switching topologies (stages) Circuits through which the converter goes cyclically during a switching period.

POWER ELECTRONICS systems are cyclically switching circuits for processing power. They convert an unregulated input power into a regulated output power. According to the type of input and output voltages, there are DC-to-DC converters (used, for example, as DC switching-mode power supplies or in electrical DC motor drives), AC-to-DC rectifiers (used, for example, for power factor correction circuits), DC-to-AC inverters (used, for example, in AC uninterruptible power supplies [UPS] or in electrical AC motor drives), and AC-to-AC

cycloconvertors (in which the frequency is converted from one value to another). By using a feedback circuit, the output voltage is regulated despite variation in the line (input voltage) and/or load.

I. REQUIREMENTS FOR A POWER ELECTRONICS CIRCUIT

If the power electronics circuit is to provide energy of a high power level, as in motor drives or in DC or AC UPS, then the efficiency of the energy conversion is of greatest interest. As all the energy is processed through the electronic circuit, one cannot afford large losses in this process. As a matter of principle, resistors are excluded from such circuits. Special power electronics devices (power insulated-gate bipolar transistors [IGBTs], power metal oxide field effect transistors [MOSFETs], diodes, thyristors, etc.) are used as switches to carry large currents and support large voltages (kilovolts in high-voltage applications). From efficiency requirements, these devices must feature small on-parasitic resistances in order to reduce the energy losses when they are in the conduction state. They have to switch with high speed in order to diminish the switching losses in the transient process between the states on–off and off–on.

In low-power switching-mode power supplies, the size and weight of the converter have to be minimized, i.e., the power density has to be maximized. To reduce the size of the passive devices (inductors, capacitors, transformers) it is desirable to use a high frequency in the conversion process. This requirement is particularly important in portable devices. One recent development is the use of only switches and capacitors in the structure of a converter in order to avoid bulky magnetic elements.

Recent challenges have been imposed by very large scale integration (VLSI) circuits operating at low voltage (under 3 V, and even lower in the future), which put a heavy constraint on the designer of a power electronic supply to get a high ratio between the input and output voltages. Computers present rapid load changes; their switching-mode power supplies need a fast dynamic response in order to bring the supply voltage back very quickly to its nominal value.

A recent development in manufacturing electronic switches is the use of galium arsenide power semiconductors instead of silicon ones. Such transistors (metal semiconductor field effect transistors, MESFETs) have extremely short transition times, which makes it possible to increase the switching frequency to the 100-MHz level without increasing the switching losses. For driving such power MESFETs, a gate driver providing a pulse of values 0 and −3 V is used. Such an ultrahigh frequency permits the reduction of the size of passive elements, resulting in an ultracompact converter suitable for portable devices which require a very small power.

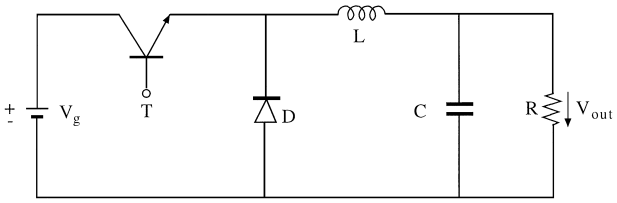


FIGURE 1 Basic buck DC–DC converter.

II. PRINCIPLES OF OPERATION OF A SWITCHING-MODE CIRCUIT

The basic structure of a DC–DC converter is given in Fig. 1. The driving signal of the transistor T is cyclical of the form shown in Fig. 2.

The cycle period T_s is dictated by the switching frequency f_s ($f_s = 1/T_s$) provided by a clock in the feedback circuit. When $d(t)$ has the logical value 1, T is on, and consequently D is switched off (Fig. 3a). When $d(t)$ drops to the logical value 0, T is turned off (and, consequently, D is turned on) (Fig. 3b). Here D denotes the ratio between the time the transistor is on and the switching period. It is called the duty ratio.

By assuming 100% efficiency, i.e., $P_{\text{input}} \approx P_{\text{output}}$, one can find an approximate relationship between V_g (the line voltage) and V_{out} (the load voltage). One has $P_{\text{input}} = V_g I_{\text{in}}$ and $P_{\text{output}} \approx I_L V_{\text{out}}$. In stage (a), of duration DT_s, $I_{\text{in}} = I_L$. In stage (b), of duration $(1 - D)T_s$, $I_{\text{in}} = 0$, i.e., $I_{\text{in}} \approx DI_L$. Thus

$$V_g D I_L \approx I_L V_{\text{out}} \qquad \text{or} \qquad V_{\text{out}} \approx D V_g.$$

Therefore, any change in V_g can be compensated by variation of the duty cycle to keep V_{out} constant.

The operation of the converter takes place through a permanent switching between the two topologies (a) and (b) at a high frequency (from tens of kilohertz to megahertz in modern converters). If the current I_L drops to zero before

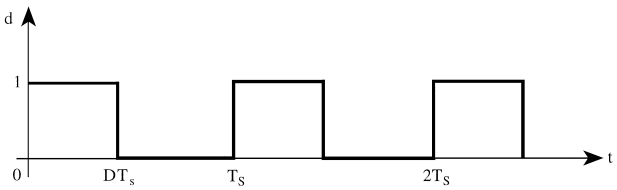


FIGURE 2 Driving signal of transistor T in the basic DC–DC converter.

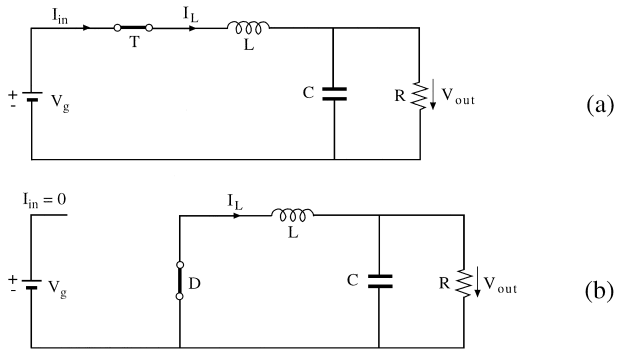


FIGURE 3 Topological switching stages of the cyclical operation of the basic buck converter.

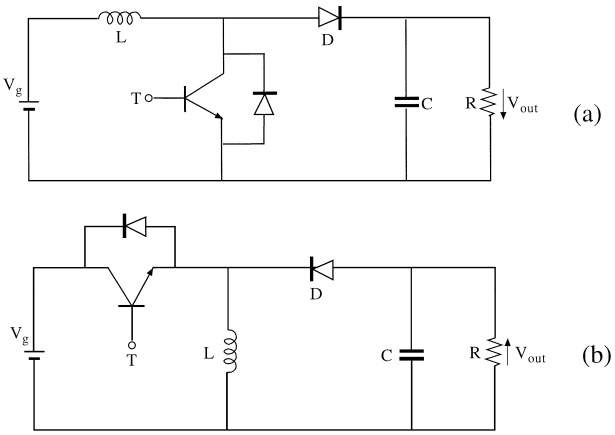


FIGURE 4 Basic DC–DC structures: (a) boost; (b) buck–boost converter.

the end of the cycle T_s, i.e., the inductor energy which was accumulated during topology (a) is completely released to the load, the converter enters the third topology, in which D is turned off. This mode is called the discontinuous inductor current operation mode, and the converter exhibits different properties.

The role of the capacitor C in parallel to the load is to ensure a permanent voltage across the load. The block LC plays the role of a low-pass filter, with the corner frequency $f_c = 1/(2\pi\sqrt{LC})$. It eliminates all the harmonics from the square-wave type of voltage that appears at the output of the switching block T–D. In order to get a clean DC output, it is necessary that $f_c \ll f_s$. As f_c cannot be reduced, since then the size of L, C, i.e., of the converter, would be increased, one has to increase f_s. Research in the last decade has had to solve the problem of how to increase the switching frequency without increasing the switching losses. (When a switching device is turned on/off, there is a short transient period in which the voltage across it reaches a theoretical zero and the current increases from zero to the nominal value, or the current drops to zero and the voltage increases to the maximum off-value, respectively, i.e., during this period there is an energy loss, which accumulates at every cycle.)

III. DC-TO-DC CONVERTERS

A. Square-Wave Hard-Switching Converters

1. Single-Ended Converters

The buck converter shown in Fig. 1 gives voltages across the switching devices and currents through them of a square form. As the turn-on/off is accompanied by transient losses, this type of converter is called a hard-switching circuit.

The buck converter is one of three basic structures. The other two are the boost (Fig. 4a) and the buck–boost (Fig. 4b). Each basic structure contains an inductor, a capacitor, a transistor (IGBT or MOSFET) with its antiparallel built-in diode, and a rectifier diode. The difference among the three structures lies in placement of the inductor: connected to the load (buck), which ensures an output with small ripple, to the line (boost), which ensures an input current with small ripple, or switched from line to load (buck–boost). For the boost converter, $V_{out} \approx [1/(1-D)]V_g$; for the buck–boost, $V_{out} \approx -[D/(1-D)]V_g$, i.e., the buck is a stepdown voltage circuit (as $0 < D < 1$), the boost is a stepup, and the buck–boost, with an inverse output voltage, can either step up or step down the line.

By connecting a buck converter as the load of a boost converter and reducing the redundant switches, one gets a Ćuk converter (Fig. 5a), which gives a small ripple both in the input and output currents. Thus, no additional input or output filter is necessary for reducing the ripple, which is one of the causes of electromagnetic interference (EMI) emission. By combining other basic structures, one gets the Sepic converter (Fig. 5b) and the Zeta converter (Fig. 5c).

In all these structures, there is no DC separation between line and load; undesirable disturbances in line can affect the user and conversely. By adding a transformer to a buck structure and to a buck–boost structure, respectively, one gets the forward converter (Fig. 6a) and the flyback converter (Fig. 6b). Of course, the transformer adds new conduction losses and switching voltage spikes.

The presence of an inductor in series with the transistor (as in the circuit realized when T is turned on in a boost or buck–boost converter) limits di/dt when the device is turned on, thus limiting the switching loss and the emission of EMI. A parallel capacitor to the transistor slows dv/dt across it when the device is switched

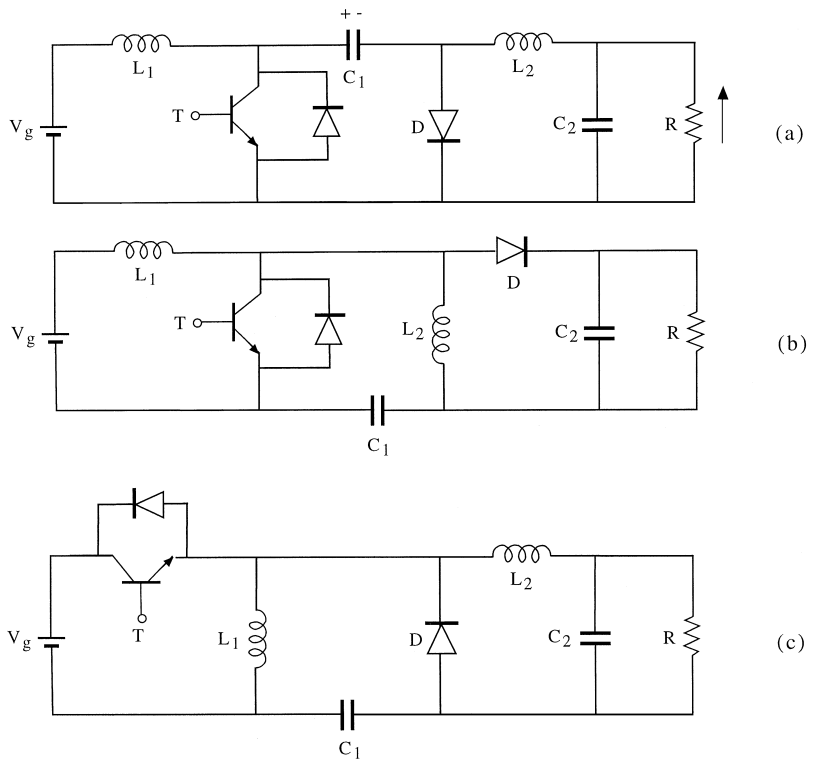


FIGURE 5 Derived structures: (a) Ćuk, (b) Sepic, (c) Zeta converter.

off. A MOSFET has a built-in parallel capacitance. This helps the turn-off operation, but when the device is off, the parasitic capacitance is charged at the voltage across the switch; when the switch is turned on, this energy is lost. Sometimes there is need for an additional circuit (snubber) to dissipate the turn-on parasitic capacitance energy.

When T is turned on and D turned off, for an instant (see, for example, the boost converter), the capacitor *C* is shot through to the ground, creating a sharp negative

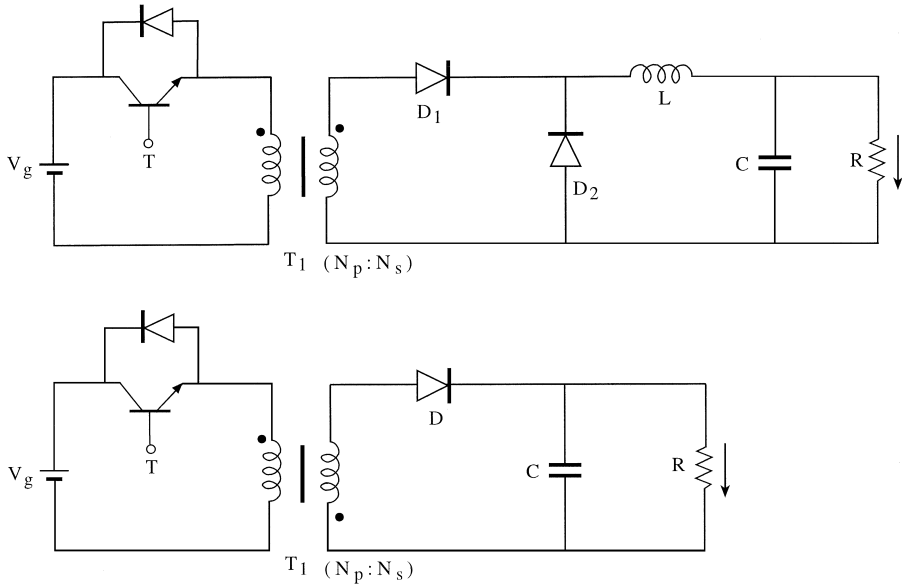


FIGURE 6 Converters with DC line-load isolation: (a) forward; (b) flyback.

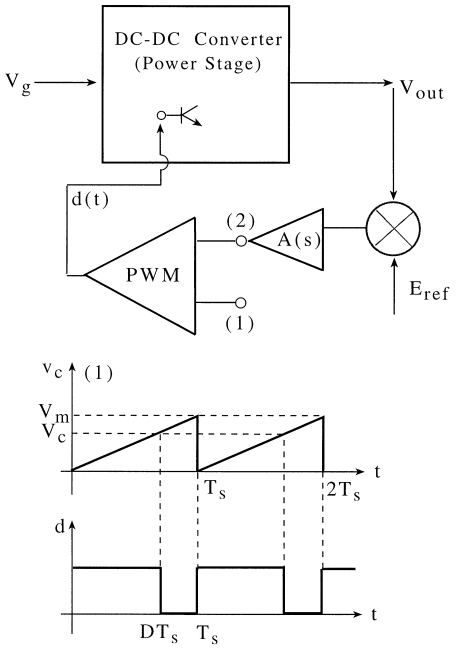


FIGURE 7 PWM-type feedback circuit.

spike in the recovery diode current (i.e., switching loss, EMI).

The above problems have been dealt with by recent research in power conversion.

2. Pulse Width Modulator (PWM) Control

The above converters can be easily controlled by a PWM-type feedback circuit (Fig. 7). The load voltage V_{out} is compared with the nominal desired (reference) voltage E_{ref} and the error is processed through an amplifier–compensator circuit (which is designed for ensuring closed-loop stability) and is applied to the entry (2) of the PWM. At the entry (1), a sawtooth voltage v_c of maximum value V_m is superimposed on a control DC voltage V_c. In the absence of any error, the PWM works as a comparator between v_c and V_c and supplies the pulse-form signal $d(t)$ and the nominal value D of the duty cycle. Any error moves the V_c level up or down, thus increasing or decreasing the duty ratio D by modifying the width of the pulses.

The converters in Figs. 1, 4, 5, and 6 are useful for low- to medium-power applications.

3. Double-Ended Converters

For large-power applications, one uses double-ended converters: push-pull (Fig. 8a) and full-bridge converters (Fig. 8b). These are variations of the forward converter, but present a double load voltage $V_{out} = 2D(N_s/N_p)V_g$. In the case of the full-bridge converter, the voltage rating of each transistor is only V_g, which is advantageous for high-voltage applications. N_s/N_p is the turns ratio.

These converters operate with a half-cycle symmetry. For example, for the push-pull converter, in the first topology, of duration $DT_s/2$, SW1 is on and SW2 is off; in the second topology both transistors are off, the output energy being delivered by LC. The operation is inverted in the second half-cycle. A duty-cycle control (of PWM type) is therefore used.

4. Voltage-Mode and Current-Mode Control

Figure 7 shows a typical voltage-mode control. A better dynamic response is obtained by using a current-mode control (Fig. 9). In Fig. 9, the set-pulse generator provides, at the beginning of a cycle, a pulse V_{set} which sets the Q output of the R–S flip-flop circuit to a high value, determining the turn on of the transistor. The converter operates in the first topology until the controlled current I_L reaches the maximum value determined by the control voltage V_c: $V_c = KI_LR_f$, where K is the DC gain of amplifier A_1. At this instant, the output of the comparator A_2 sets Q of the flip-flop circuit at a low value, implying the turnoff of T. The converter operates in the second topology until the end of the cycle. The role of the compensation sawtooth wave is to eliminate instability that may be caused by noise.

The current-mode control is very helpful in the case of parallel connection of converters: the output load current can be shared by each converter by adjusting the value K of each one. Of course, even a small resistor R_f used for sensing the current results in a conduction loss.

A combination of voltage- and current-mode control is often used by building an outer voltage feedback loop and an inner current loop (Fig. 10). Here A_2 is a comparator between the control voltage v_c (determined by the disturbance in the output voltage) and the voltage $R_s i_L$. The control circuit plays the role of a pulse-width modulator by adjusting the width of the pulses $d(t)$. Often, the controlled current is sensed at the converter input; such a feedforward control rapidly senses any change in line.

5. Switched-Capacitor Converters

A special class of square-wave (hard-switching) circuit is represented by switched-capacitor-based (SC) converters; for the sake of miniaturization, they are built of switches and capacitors, but no inductors or transformers. A stepdown SC DC–DC converter is shown in Fig. 11.

The converter is formed of two symmetrical SC blocks. Each block contains two capacitors (C_1, C_2; and C_3, C_4), and two transistors and three diodes (S_1, S_2, D_1, D_2, D_3; and S_3, S_4, D_4, D_5, D_6). The two SC circuits operate in

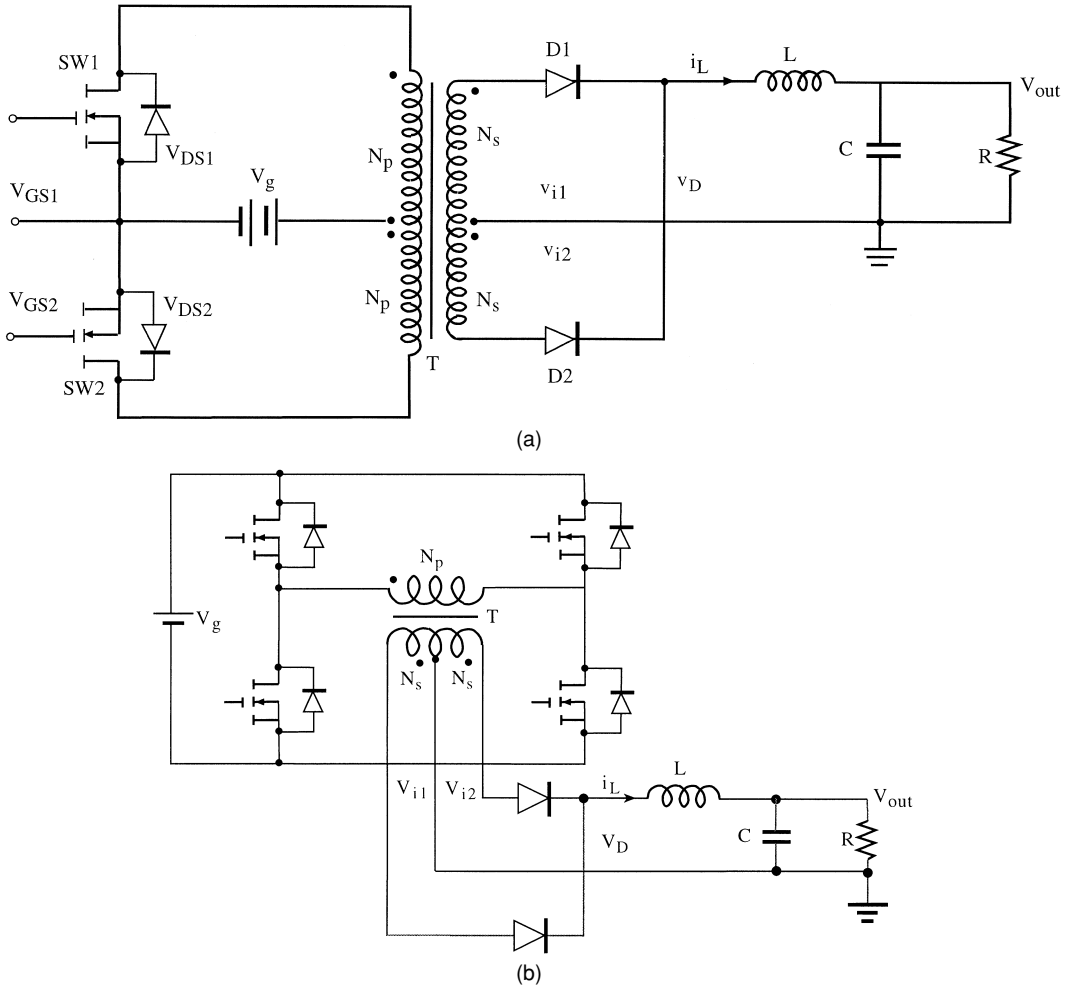


FIGURE 8 Double-ended DC–DC converters: (a) push-pull; (b) full-bridge. [Reproduced, with permission, from Lee, Y. S. (1993). "Computer Aided Analysis and Design of Switch-Mode Power Supplies," Dekker, New York.]

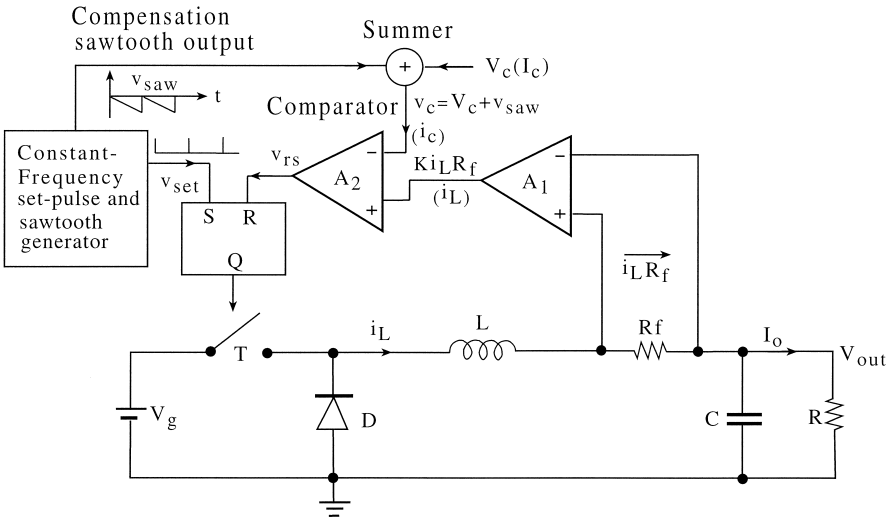


FIGURE 9 Current-mode-controlled converter.

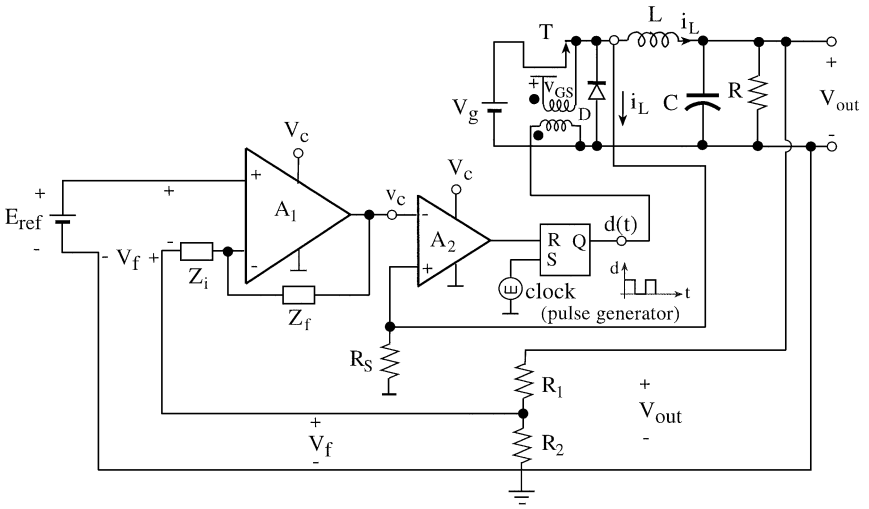
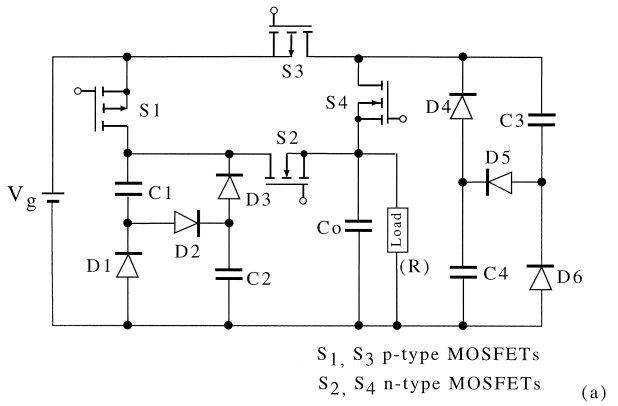


FIGURE 10 Voltage- and current-controlled DC–DC converter. [Reproduced, with permission, from Kazimierczuk, M. K. (2000). "Transfer function of current modulator in PWM converters with current-mode control," *IEEE Trans. Circuits Systems I* **47,** 1407–1412. © 2000 IEEE.]



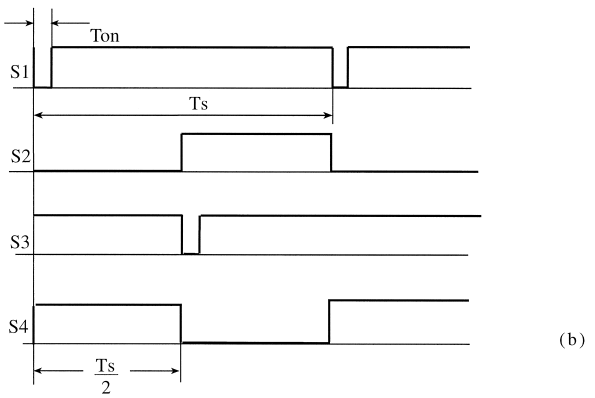


FIGURE 11 (a) Switched-capacitor-based DC–DC converter and (b) its timing diagram for driving the switches. [Reproduced from Cheong, S. V., Chung, H., and Ioinovici, A. (1994). "Inductorless DC-to-DC converter with high power density," *IEEE Trans. Industrial Electronics* **41,** 208–215. © 1994 IEEE.]

antiphase in each half-cycle. In the first topology, of duration DT_s, C_1 and C_2 are charged in series from the line, and C_3 and C_4 are discharged on the load. In the second topology, C_1 and C_2 are disconnected from the line, for the sake of line and load regulation; C_3 and C_4 continue their discharge on the load. The role of the capacitors is inverted in the second half-cycle. We have

$$V_{out} = \frac{V_g - 3V_d}{2 + (1/R)[1/g_2 + 1/(4g_1D)]},$$

$$g_1 = \frac{1}{2r + r_1}, \qquad g_2 = \frac{1}{2r_2 + r},$$

where V_d is the forward voltage of a diode, r is the DC parasitic resistance of the capacitors (multilayer ceramic capacitors are used for their small r), r_1 is the on-resistance of the transistors S_1 and S_3, and r_2 is the on-resistance of the transistors S_2 and S_4. Thus, by using two capacitors in each SC bloc, one gets $V_{out} \approx V_g/2$. Other ratios can be realized by using some other number of capacitors.

Similar stepup DC–DC converters or DC–AC inverters can be designed. A simple, robust PWM control is used. A high power density is obtained: for example, 23 W/in.^{3} for a 10-W converter.

B. Sinusoidal-Wave Soft-Switching Converters

1. Resonant Converters

By adding a resonant circuit of inductor–capacitor type after the switching devices, the square-wave voltages and currents are converted into sinusoidal waveforms.

A resonant DC–DC converter is therefore formed of an inverter, a resonant circuit, a high- frequency transformer, and a half-bridge or a bridge rectifier, as these converters are used for a high power level.

The existence of sinusoidal waveforms allows the switching to take place when either the voltage across the switch or the current through it has a zero value, implying theoretical zero switching losses, thus, the designation "soft-switching."

The resonant circuit can be a simple series or parallel *LC* circuit, or a circuit formed by an inductor and two capacitors (parallel *LCC*) or two inductors and one capacitor (parallel *LLC*), or a parallel *LLCC* circuit (Fig. 12a).

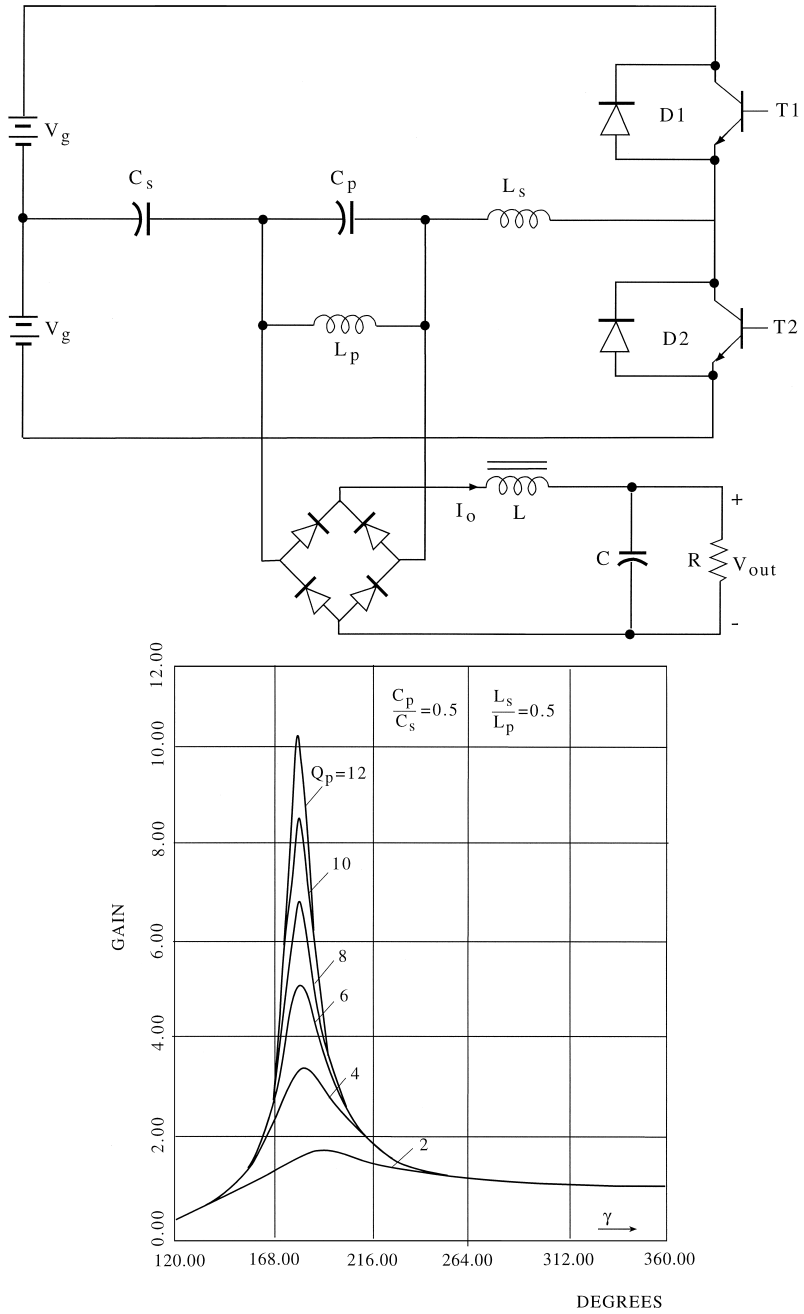


FIGURE 12 (a) *LLCC*-type parallel resonant converter and (b) its control characteristics. [Reproduced, with permission, from Batarseh, I., and Lee, C. Q. (1991). "Steady-state analysis of the parallel resonant converter with LLCC type commutation network," *IEEE Trans. Power Electronics* **6,** 525–538. © 1991 IEEE.]

As the switches have to turn on/off depending on the value of the associated voltage/current and not according to the decisions of a PWM feedback circuit, a duty-cycle-type regulation is not possible. This is why the resonant converters are regulated by switching frequency control. Figure 12b gives the control characteristics in terms of gain (output voltage over line voltage) versus the frequency ratio, in degrees ($\pi f_r/f_s$, with f_r being the resonant frequency), for different values of the load $R_0(Q_P = R_0/Z_0, Z_0 = 2\pi f_r L_s)$. By varying f_s, one can keep a constant load voltage.

The resonant converters can be controlled either in the "above-resonance" mode, when the range of the switching frequency is always above the resonant frequency of the resonant block, or in a "below-resonance" mode, when the switching frequency remains always below the resonant frequency.

The last mode has the advantage that the switches turn off when the current through them is zero, i.e., there are no turn-off losses. This feature allows for the use of SCRs as switching devices; SCRs have a slow turn-off and a large power rating. Consequently, these converters can be used in high-power applications. Since for regulation purposes, however, one needs to reduce the switching frequency below its nominal value, one has to design all the magnetic elements at the smallest frequency in the range of the switching frequency (otherwise the transformer could reach saturation), resulting in a larger size of the magnetic elements. As the parasitic capacitances of the switches are fully charged before the transistors are turned on, switching losses are present and snubbers are necessary to dissipate this energy.

If the converter is operated in the "above-resonance" mode, the switches are turned on when the voltage across them is zero. Consequently, no turn-on losses are present. These converters operate at a high switching frequency, and thus the size of the magnetic and filter components is reduced. The existence of turn-off losses, however, requires very fast turn-off switches. At light load, the switching frequency has to be increased significantly for regulation purposes; such a large frequency range makes it more difficult to design the control circuit.

2. Quasi-Resonant Converters

a. Zero-current switching and zero-voltage switching. In resonant converters, the resonant block is present in the circuit in all topological stages. A great deal of energy is circulated in the resonance process. This is not acceptable in low- to medium-power converters. The quasi-resonant converters (QRC) introduced by the Power Electronics Center of Virginia University make use of the block LC only for creating the zero-current-switching (ZCS)/zero-voltage-switching (ZVS) condition when the switch is turned on/off. In the largest part of a switching period, the converter behaves like a square-wave converter, converting to a sinusoidal-wave type only in the stages that prepare the switching moments. The first QRCs were frequency controlled. By introducing an additional transistor and by controlling the period between switching on/off the two transistors, a duty-cycle (PWM control) was created.

The ZCS converters create a zero-current turn-off of the transistor (suitable for transistors with a slow turn-off speed, like IGBTs). The maximum voltage stress on the switch is $2V_g$; the current through the switch, larger than the load current, increases with the line, creating an additional stress which requires an overrated switching device, with a larger parasitic on-resistance, i.e., with increased conduction losses. As the transistor turns on at full line voltage, parasitic-capacitance turn-on losses appear. When the transistor turns on, the current through the recovery diode is reduced gradually to zero and the diode turns off naturally, without losses and noise .

In ZVS converters, the transistors turn on when the sinusoidal voltage across them goes through the value zero. This is useful for devices with a slow turn-on speed, like MOSFETs. The voltage across the transistor (off state) is larger than the line voltage and increases with the load current, thus creating stress on the device. When the diode switches off, the voltage across it changes abruptly from zero to line, introducing oscillations between the resonant inductor and the parasitic capacitance of the diode, which could lead to instability.

As ZCS or ZVS allows for almost zero turn-off/turn-on losses, the switching frequency of QRCs can be very large (megahertz), allowing for reduction in the size of the passive elements of the converter.

The ZCS condition is lost at low line and large load current. The ZVS condition is lost at high line and light load current.

b. Zero-current transition and zero-voltage transition. The above problems have been solved by ZCT (zero-current-transition) and ZVT (zero-voltage-transition) converters. A ZVT converter is shown in Fig. 13. Here C_d and C_D are the parasitic capacitances of the transistor S and diode D, respectively. L_r and C_r from the resonant block, and S_1 and D_1 are additional switching devices. Before turning on S, the voltage across it, V_{DS}, is brought to zero in a resonant process. In the most of the period T_s (T_5–T_6, T_7–T_0), the converter behaves like a square-wave converter. The auxiliary switch S_1 handles a small transition energy and operates with ZCS. ZVT is insensitive to line or load values. There are some stresses

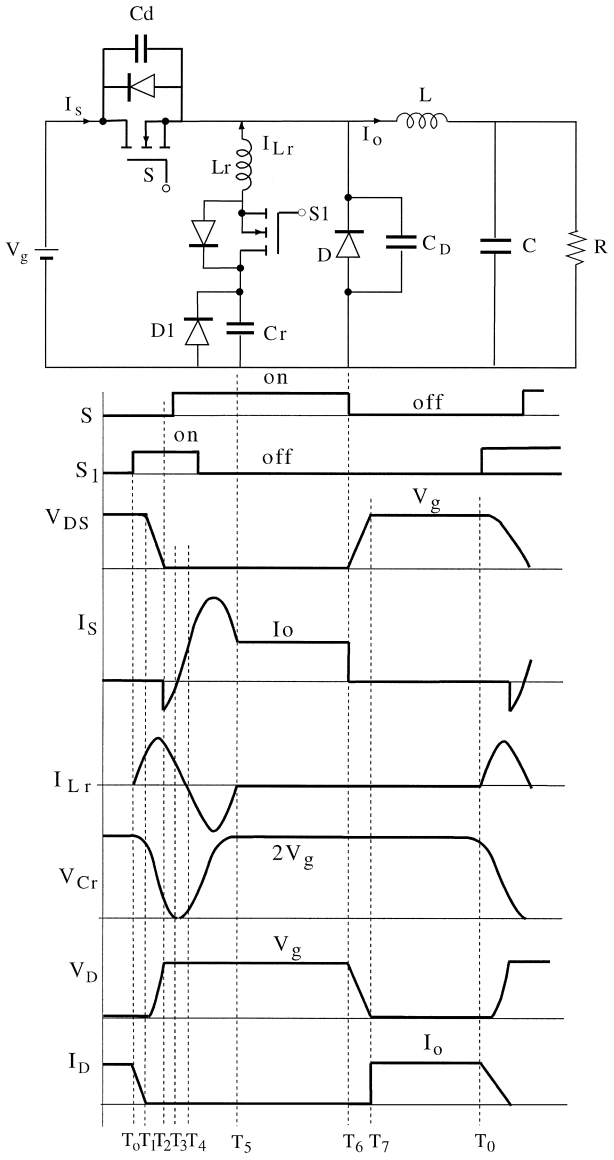


FIGURE 13 (a) ZVT PWM buck converter and (b) its main waveforms. [Reproduced, with permission, from Hua, G., Leu, C. S., and Lee, F. C. (1991). "Novel zero-voltage transition PWM converters." *In* "High-Frequency Resonant and Soft-Switching PWM Converters," Vol. IV, p. 208 (Lee, F., ed.), VPEC Publications, Blacksburg, VA.]

in the switch current I_s and in the voltage across C_r, V_{cr}. A turn-on parasitic capacitance loss accompanies S_2.

The circuit in Fig. 14 does not present these inconveniences, but its auxiliary switch is in the main route of processing energy, which gives some conduction loss. This circuit uses the parasitic capacitances of the transistors as resonant elements. The resonant block C_1LC_2 ensures that the voltages across S and S_1 drop to zero before S and S_1 are turned on with ZVT. They are turned on also with ZCT, as the current through them, I_L, starts increasing slowly

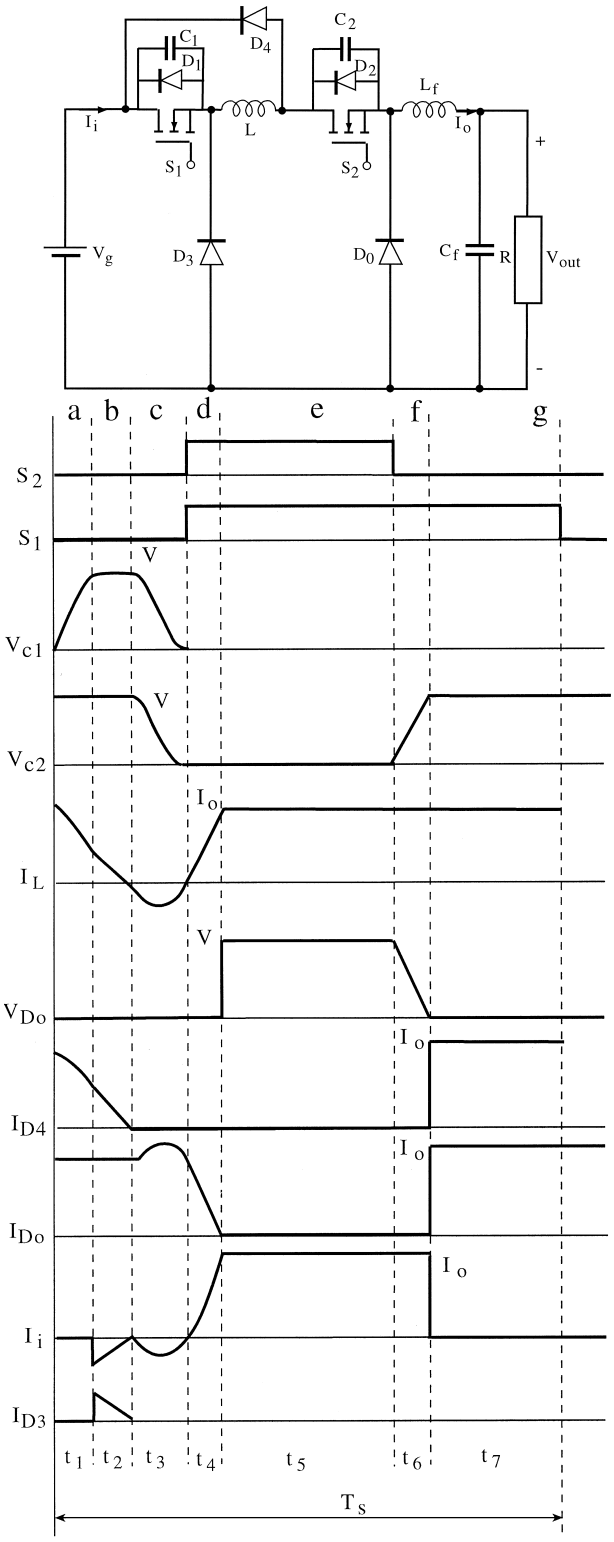


FIGURE 14 (a) ZVT PWM buck converter with soft turn-on/off for all switches and (b) its key waveforms. [Reproduced from Divakar, B. P., and Ioinovici, A. (1997). "PWM converter with low stresses and zero-capacitive turn-on losses," *IEEE Trans. Aerospace Electronic Systems* **33,** 913–920. © 1997 IEEE.]

from zero due to the presence of L in the circuit. S and S_1 are turned off with ZVT due to the parallel capacitances C_1 and C_2, which prevent the voltages across S and S_1 from rising sharply. D_0 turns on at the end of time interval t_6, when the voltage across it reaches zero (ZVT) and turns off naturally (ZCS) when I_{D0} reaches zero at the end of time interval t_4. Similarly for D_3 and D_4. The elements are under no voltage/current stress.

ZCS and ZVS have been introduced in all PWM converters, including high-power ones (push-pull, bridge), allowing for a high-efficiency conversion of the energy. Efficiency of 95% has been reported even for low- to medium-power converters. In most full-bridge PWM converters, soft-switching is realized by creating ZVS for the leading leg switches and ZCS for the lagging leg switches. In the circuit shown in Fig. 15, S_1 and S_3 are the leading leg switches. In the first half-cycle C_3 is completely discharged in a resonant process (in which C_1, C_3, L_{1k}, C_h participate) before S_3 is turned on with ZVS. ZCS of lagging leg switches is created by the use of an auxiliary circuit formed by an auxiliary winding (N_3) with a small turns ratio, four diodes d_1–d_4, a holding capacitor C_h, and a diode D_h. Before turning off S_2, C_h is completely discharged, the rectifier diodes d_1–d_4 start to conduct, and the load current flows in a circuit formed by the rectifier diodes and load (L_0–C_0–R) (typical freewheeling stage of a hard-switching full-bridge PWM converter). As a result, there is no current through S_2, and S_2 can be turned off with ZCS. After a dead time, S_4 is turned on with ZCS. In the second half-cycle, the process is repeated and S_1 is turned on with ZVS. A tradeoff requirement is imposed in the design of C_h: it has to be large enough to provide the load voltage in one of the stages, yet not too large because it has to be fully discharged in the freewheeling process.

Converters able to recuperate the circuit resonant energy by transferring it, at least partially, to the load have been proposed recently. Such a circuit is shown in Fig. 16. The additional elements for recuperating part of the resonant energy are the transformer T_x and diodes D_{x1} and D_{x2}. The main switch S_1 is turned on/off with ZVS. The auxiliary switch S_2 is turned on with ZCS (due to the series inductor L_r) and turned off with ZVS. When the auxiliary circuit is present in the power stage, the secondary voltage of T_x determines the conduction of either D_{x1} or D_{x2}, depending on the direction of the current i_{Lr} through the primary of T_x. In both cases, part of the resonant energy is fed to the load, improving the efficiency. The primary voltage on T_x is always clamped to V_0/N_x, where V_0 is the load voltage and N_x the turns ratio of T_x. In order to recuperate the maximum amount of resonant energy, one needs a very small N_x. But then the voltage across L_r, and consequently i_{Lr}, is reduced, and in the resonant stage, the energy available to discharge C_{s1} is lowered. If C_{s1} is not completely discharged, ZVS is lost when turning on S1. This is why only a part of the circulating resonant energy

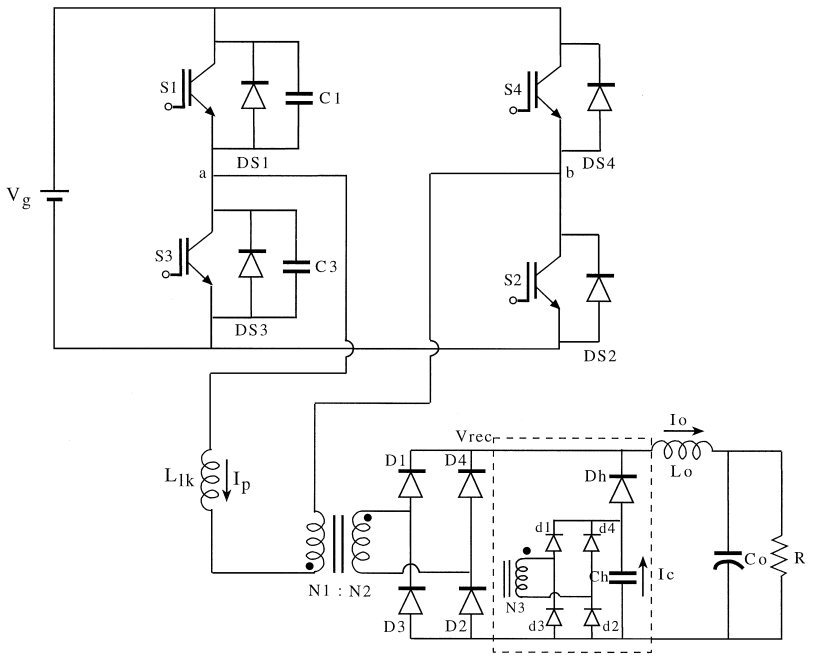


FIGURE 15 PWM soft-switching full-bridge converter. [Reproduced, with permission from Cho, J. G., Baek, J. W., Jeong, C. Y., Yoo, D. W., and Joe, K. Y. (2000). "Novel zero-voltage and zero-current switching full-bridge PWM converter using transformer auxiliary," *IEEE Trans. Power Electronics* **15,** 250–257. © 2000 IEEE.]

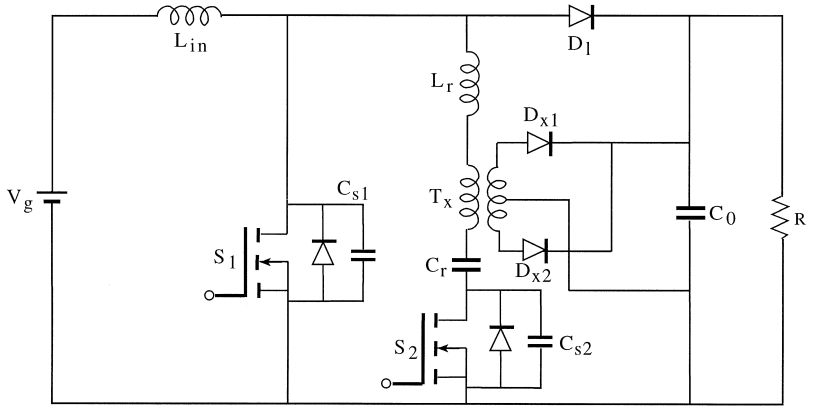


FIGURE 16 ZVT PWM soft-switching boost converter with partial recuperation of the resonant energy. [Reproduced, with permission, from Moschopoulos, G., Jain, C. K., Liu, Y. F., and Joos, G. (1999). "A zero-voltage-switched PWM boost converter with an energy feedforward auxiliary circuit," *IEEE Trans. Power Electronics* **14,** 653–663. © 1999 IEEE.]

can be recuperated. Another penalty of this circuit is a resonant current peak in both transistors, which requires their overrating. An overall improvement in efficiency of 2% is reported when this recuperation method is used.

C. Parallel Connection of Switching Converters

In high-power applications, paralleling techniques allow the converters to share the input current. One method of paralleling is through interleaved power conversion, which consists in a phase shifting of the control signals of several converters operating at the same switching frequency. As a current-mode-controlled (CCM) converter is equivalent to a gyrator, it is preferred to connect CCM cells in parallel, the output current of each converter k being in the form of a current source $g_k V_g$ (g_k is the equivalent gyration conductance). The converters share an equal amount of input current and operate at the same duty cycle, the output voltage being a multiple of the input voltage. The ripple in the output voltage of the interleaved converters is smaller than the ripple which would be present in the output voltage of a single converter, allowing for the reduction in size and losses of the filter stage. A good review of this technique is given in Giral *et al.* (1999). Another reliable technique is based on a frequency-domain method for encoding and distributing current-sharing information among cells; the method is reviewed in Perreault *et al.* (1998).

IV. AC-TO-DC RECTIFIERS. POWER FACTOR CORRECTION CIRCUITS

If a diode full bridge is connected to the input of any of the previous DC–DC converters, a switched-mode AC-to-DC rectifier is obtained. The load voltage is regulated despite variations in the AC input voltage or load.

A. Power Correction Factor (PFC) Circuit

Of practical interest is to have the input voltage and current waveforms in the same phase, i.e., to obtain a unity power factor circuit, in order to satisfy the requirements of the IEC 1000-3-2 or IEEE 519 specifications. One of the methods to get this is to use a ZVT boost AC-to-DC converter with a feedback circuit whose focus is a chip of type UNITRODE controller UC3855BN, thus realizing load voltage regulation, unity power factor, and soft-switching (ZVS) of the transistors. Despite its simplicity, this structure determines a large output voltage ripple, which requires a large output capacitor. The bandwidth of the feedback circuit is affected.

It is interesting to note that a buck converter with an input filter L_1C_1 exhibits a natural unity power factor if the periodic voltage across C_1 reaches zero before the end of a steady-state cycle, i.e., the converter is operated in discontinuous-capacitor-voltage mode. The disadvantage of this approach is the high peak voltage stress on the transistor.

B. Two-Stage Solution

A two-stage structure is another solution: a preregulator (a PFC stage) ensures a high input PFC and low line-current harmonic distorsion. An intermediate DC bus is created, and a DC–DC regulator is the second stage; it can assure a fast load voltage regulation with a good bandwidth in the feedback circuit. The PFC can be a boost converter operating either in continuous-inductor-current mode for high-power applications (this mode provides lower conduction

losses) or in discontinuous-inductor-current mode for low- to medium-power applications (in this mode, the converter naturally draws a near-sinusoidal current). This approach has both a high cost (due to the device count) and low efficiency (two converters connected in cascade).

C. Single-Stage, Single-Switch Solution

The most used approach is the single-stage, single-switch PFC converter, in which the active switch is shared by two converters. Different combinations have been used. For example, a boost, frequency-controlled converter operated in discontinuous-current mode in cascade with a forward, PWM-controlled converter operated in continuous-current mode. Frequency modulation ensures power factor correction; the second cell is immunized against frequency variation. Between the two stages is a buffer capacitor that suffers from a high voltage stress when the load becomes light, due to the power unbalance between the input and output (in the continuous-current mode, the duty ratio almost does not change upon load variation). To avoid this high bus voltage, the second stage also can be operated in discontinuous-current-conduction mode, but with the penalty of higher losses.

Another option is to use a small resonant tank, called a charge pump, as the PFC preregulator (Fig. 17). In Fig. 17, the input stage operates in a continuous-current mode. The charge-pump capacitors C_r, C_s absorb energy from the AC line in half of the switching period and deliver it to the bulk capacitor C_B in the other half-cycle. The charge-pump circuit also contains the resonant inductor L_r, the output rectifier diode D_x, and the diode D_s. These two diodes are replaced by a full diode bridge in high-power applications. C_s also serves as a snubber capacitor, by allowing a ZVS turn-off of the transistor S. The C_s also helps to reduce the DC bus voltage stress on C_B. The second stage is a flyback circuit which operates in continuous mode at full load and discontinuous mode at light load, as a tradeoff design from the efficiency point of view.

V. DC-TO-AC INVERTERS. UNINTERRUPTIBLE POWER SUPPLY

In many applications (for example, hospital systems, computers, industrial control), a permanent energy supply is critical. An uninterruptible power supply (UPS) acts as a line conditioner and an output voltage stabilizer. In case of loss of input power, the UPS supplies the load with a controlled sinusoidal voltage, drawing power from a battery.

A modern UPS is based on two voltage source inverters (VSI), one in series with the input and the other in parallel with the load. The system operates in two modes:

(a) The charging mode, with both converters active.
(b) The backup mode, when, in the absence of input power, the series converter is disabled and the parallel converter supplies power to the load using the battery as the source. Kaneran and Habetler (1998) present an overview of this type of UPS.

A buck DC–AC inverter can be used as VSI. Its instantaneous average output voltage is lower than the input DC voltage. If one needs an output voltage larger than the input one, the boost VSI shown in Fig. 18 can be used. The load is connected differentially across the converters. Each of the circuits A and B produces a unipolar DC-biased sinusoidal voltage. The modulation of each converter is 180°

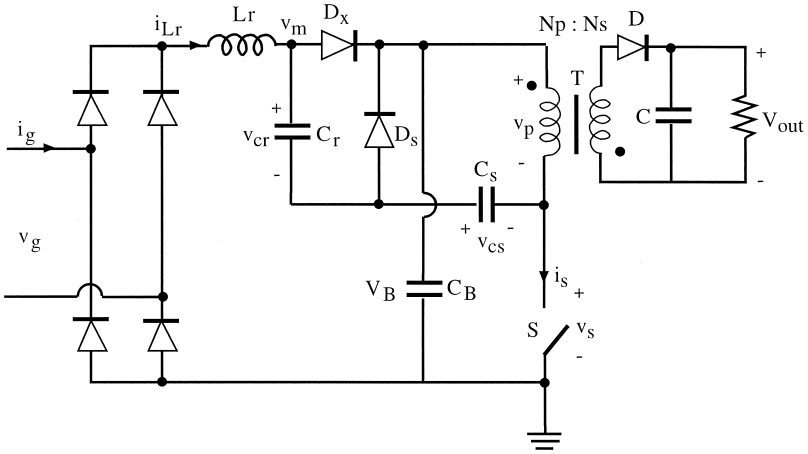


FIGURE 17 A single-stage, single-switch charge-pump PFC regulator. [Reproduced, with permission, from Qian, J., and Lee, F. C. Y. (1998). "A high-efficiency single-stage single-switch high-power-factor AC/DC converter with universal input," *IEEE Trans. Power Electronics*, **13,** 699–705. © 1998 IEEE.]

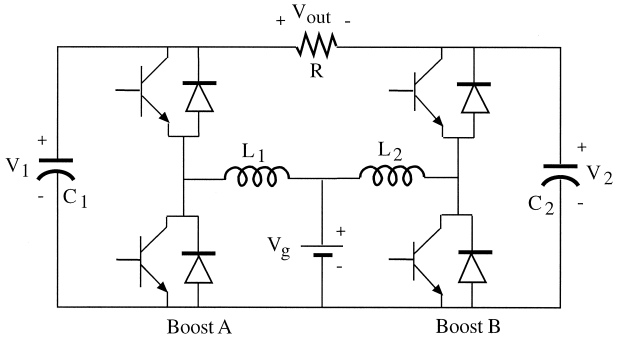


FIGURE 18 Boost voltage source DC–AC inverter. [Reproduced, with permission, from Caceres, R. D., and Barbi, I. (1999). "A boost DC–AC converter: Analysis, design and implementation," *IEEE Trans. Power Electronics* **14,** 134–141. © 1999 IEEE.]

out of phase with the other, which maximizes the voltage across the load. A bipolar voltage appears across the load.

In order to achieve better efficiency and reduce the switching losses, soft-switching can be used in DC–AC inverters, too; the aim is to realize a zero-voltage condition at the instant of switching . One method is to create a resonant DC link inverter in which the sinusoidal voltage passes periodically through the zero value, creating a ZVS condition. As the peak of the sinusoidal voltage is twice the DC link voltage in a hard-switching inverter, overrated devices are needed. Clamp circuits have been used to limit the peak voltage, but overstresses are still present. A second method is to create a front stage consisting of a QRC. The first half of the resonant cycle brings the DC link voltage to zero. At that instant, switching takes place. In the second half of the resonant cycle, the DC link voltage is set back. Therefore, the inverter operates as a hard-switching one, except for the period around the instant of switching. The design is difficult, as the resonant cycle has to start exactly at half-cycle before the instant of switching, despite variation in the system parameters.

VI. ANALYSIS AND DESIGN OF POWER ELECTRONICS CIRCUITS

A. Time-Domain Simulation

1. Challenges

As seen previously, any power electronics circuit operates by switching cyclically from one topological stage to another. The number of switching topological stages varies from two (basic PWM DC–DC converters operating in continuous-current mode) to a large number in the case of soft-switching converters. The transition from one stage to another one is determined by the turn on/off of switching devices. Transistors switch on/off according to the signals provided by the feedback circuit: a clock determines the beginning of a switching cycle by turning on a transistor; the output of a PWM determines the driving of a transistor for duty-cycle control, etc. Therefore, the transistor can be seen as an externally controlled switch. The diodes turn on/off depending on the values of the currents/voltages within the circuit; they can be seen as internally controlled switches that switch when some threshold conditions are fulfilled. A practical switching device is never a short-circuit when the switch is on, and never a perfect open-circuit when the switch is off. A switching device features an on-resistance, a forward-voltage drop when in conduction, etc. If the simulator considers the switch as an ideal one, then it has to handle Dirac impulses at transient instants.

2. Accurate Algorithm

Theoretically, a simulation algorithm for a transient state can look as follows: start with the first topological state in the first cycle after the change has occurred. Write and solve the differential equations (state variable equations, for example) for this switching circuit. Find the moment when this topological state ends (a signal from the feedback circuit? a certain threshold condition fulfilled before an external command from the feedback circuit occurs?). Calculate the state of the circuit at the end of the first topology. Use these values as the initial state for the second switching circuit. Repeat the process for each stage in a cycle and for each cycle, until the steady state is reached (the waveforms in a steady-state cycle begin and end at the same values). With realistic models for each device, such an approach would be quite accurate, but of course completely impractical from the point of view of the duration of the simulation process. Approximate methods are available.

3. Averaged Continuous and Sampled-Data Approaches

By using a small-ripple approximation (i.e., by neglecting the ripple within a switching cycle, that is, by considering its period T_s small enough to neglect the values $T_s^2, T_s^3, \ldots$), averaged models have been obtained. The averaged continuous method consists in writing the state-space equations in each switching stage in the form

$$\dot{x}(t) = A_k x(t) + B_k v_g,$$

where k is the number of a topological stage in a cycle, $x(t)$ is the vector of the converter state variables (the feedback circuit is included), and A_k and B_k are matrices depending on the values of the converter elements (inductors,

capacitors, load). By linearizing the expression $\exp(A_k T_s)$, we obtain an averaged model of the form

$$\dot{x}(t) = A_{av}x(t) + B_{av}v_g,$$

where A_{av} and B_{av} are the respective averages of the matrices A_k and B_k in a switching cycle; for example, in a duty-cycle-controlled converter operating in three stages $k = 1$, 2, and 3, of respective duration dT_s, d_2T_s, and$(1 - d - d_2)T_s$, we have $A_{av} = dA_1 + d_2A_2 + (1 - d - d_2)A_3$. Despite its simplicity and extension to resonant and quasi-resonant circuits, the averaged method suffers from inaccurate results. Another approach is based on sampled-data models: the continuous-state equations are transformed into discrete equations; in their solution, however, a truncated Taylor series expansion is used, causing inaccuracy similar to that in the averaged continuous models.

4. Laplace Transform Approach

As the Laplace transform cannot be directly applied to such time-variable systems with internally controlled switching instants, different tricks have been used, for example, the use of a forward–backward integration procedure based on numerical Laplace inversion. These methods can be used for converters operating in a few switching stages.

5. Algebraic Modified Nodal Equations Approach

An iterative approach which requires solving algebraic modified nodal equations instead of differential state equations consists in the following steps: Choose a time step σ smaller than any time constant (Cr_C, Lr_L, with r_C, r_L being the on-resistances of capacitor C and inductor L, respectively) in the converter. Approximate any nodal voltage with a straight line segment for the step duration. With $v_C(t_n)$ and $i_L(t_n)$ denoting, respectively, the capacitor voltage and inductor current at the beginning of a topological stage at instant t_n, calculate v_C and i_L after the step σ, at time $t_{n+1} = t_n + \sigma$:

$$v_C(t_{n+1}) = v_{bC}(t_n) + [v_C(t_n) - v_{bC}(t_n)]e^{-\sigma/(r_C C)}$$

$$i_L(t_{n+1}) = \frac{1}{r_L}v_{bL}(t_n) + \left[i_L(t_n) - \frac{1}{r_L}v_{bL}(t_n)\right]e^{-\sigma/(L/r_L)},$$

where v_{bC} is the voltage across the branch formed by C and r_C in series, and v_{bL} is the voltage across the series branch L, r_L. Replace each capacitor and inductor at time t_{n+1} by DC sources of values $v_C(t_{n+1})$, and $i_L(t_{n+1})$, respectively. Solve the resistive circuit obtained in such a way by the modified nodal method. Find the voltages across all the switches and currents through them. Check the compatibility of the values of these voltages and currents with the assumed position (on/off) of each switch. If a contradiction appears, which points to a change of topology within step σ, return to t_n, decrease σ, and repeat the process until the switching instant is found with the desired accuracy. If no contradiction exists, check if the driving signals calculated at t_{n+1} impose a change in the state of any externally controlled switch. Update the switch position if necessary and continue with the next step.

B. AC Small-Signal Transfer Functions

Small-signal transfer functions of a converter are obtained by considering small disturbances in the line and load from their steady-state values (which imply small changes in the state variables and load voltage/current), perturbing the average differential equations of a converter, separating the DC and AC components, using a small-signal approximation for neglecting products of AC components, and taking the Laplace transform of the AC linearized equations. These functions can be used for the design of the feedback circuit in the voltage- or current-controlled mode. As a small-signal approximation is used, no large-signal stability is guaranteed.

In a basic duty-cycle-controlled converter, the switching block is formed by a three-terminal circuit consisting of a transistor and diode. By using a perturbation process in the circuit equations, one gets an equivalent circuit as in Fig. 19, where D is the nominal (steady-state) duty ratio, r_{eq} is the equivalent resistance of the DC resistance of the capacitor and load resistance, $D' = 1 - D$, $\hat{d}$ is the perturbation in the controlling variable D, I_0 is the DC output current, and $V_{eq} = V_1 + I_0(D - D')r_{eq}$. This model replaces the switching block in the s-domain equivalent circuit of any power electronics circuit.

By using this model or its equivalent, "the averaged inductor" (where the three-terminal circuit also includes the inductor, which takes different positions in each basic DC–DC structure), AC small-signal open-loop transfer functions are obtained which are equivalent to those of the averaged models. For example, in a voltage-controlled mode, the AC small-signal line-to-output and control-to-output open-loop transfer functions present two stable poles; the control-to-output transfer function of the boost (and its derivatives) converter presents a right-half-plane zero, etc. The transfer functions of current-mode-control converters without compensating ramp present a single stable pole and no right-half-plane zero.

C. Large-Signal Stability-Oriented Design

Different methods have been proposed for finding sufficient conditions for ensuring the large-signal stability of a

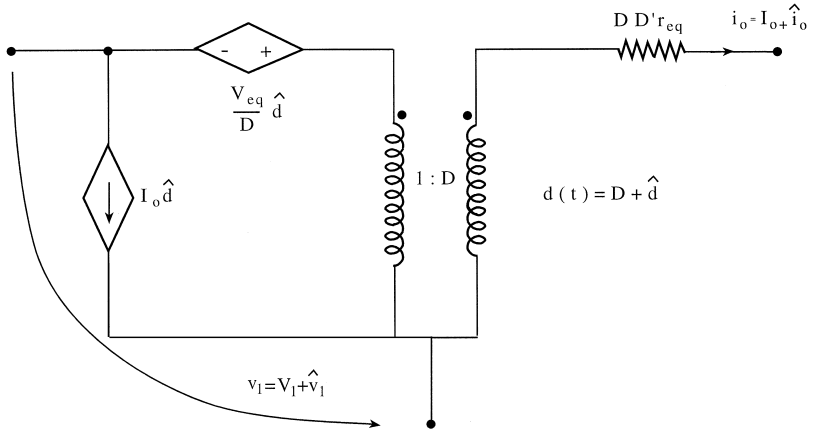


FIGURE 19 Equivalent DC and AC small-signal model of a PWM switch. $x(t) = X + \hat{x}$, where X is the steady-state DC value and $\hat{x}$ is the disturbance. [Adapted, with permission, from Vorperian, V. (1989). "Simplified analysis of PWM converters using the model of the PWM switch." *In* "Modeling, Analysis and Design of PWM Converters," Vol. II, p. 89 (Lee, F. C., ed.), VPEC Publications, Blacksburg, VA.]

switching-mode power circuit. Design criteria have been found by using the Lyapunov method.

Another method is based on a multiple-parameter, bifurcation-theory-oriented approach: the condition for local stability at the equilibrium point of interest in ensured by fulfillment of the Hurwitz criterion applied to the closed-loop averaged analytical model. The eigenvalues of the Jacobian of the averaged system of equations of the regulator are determined analytically, and the type of bifurcation (either static, pointing to a jump phenomenon, or dynamic, pointing to an onset of limit cycles) is studied at critical points at which the Hurwitz inequalities become equalities.

The complete picture of the transient behavior of a regulator can be drawn by using a state-plane portrait in which trajectories of the state variables are plotted and the position of the equilibrium points are determined.

A method for designing constraints on the feedback gain vector for global stability has, as a first step, a procedure to assure local stability around the operating point (the discrete perturbed equations of the regulator are linearized, the z-transform is used, and the condition for the eigenvalues of the system of equations to be inside the unit circle are found). Then, nonlinear terms, including nonlinearity of PWM due to possible saturation, are considered. A new constraint in terms of feedback gains is found in order to ensure that one and only one real equilibrium point (i.e., the quiescent operating point) exists. A sufficient condition in terms of boundaries on the feedback gain vector is found, ensuring a large-signal stability of the design. An overview of methods for large-signal stability can be found in Ioinovici *et al.* (1997).

SEE ALSO THE FOLLOWING ARTICLES

CIRCUIT THEORY • ELECTRICAL AND ELECTRONIC MEASUREMENTS • ELECTRIC MACHINES, DESIGN • ELECTRIC MACHINES, DRIVES • ULTRA LARGE SCALE INTEGRATION DESIGN • VACUUM SWITCHES

BIBLIOGRAPHY

Erickson, R. W. (1997). "Fundamentals of Power Electronics," Chapman and Hall, New York.

Giral, L., Martinez-Salamero, L., and Singer, S. (1999). "Interleaved converters operation based on CMC," *IEEE Trans. Power Electronics* **14,** 643–652.

Ioinovici, A., Martinez-Salamero, L., and Vlach, J. (1997). "Special Issue on simulation, theory and design of switched-analog networks," *IEEE Trans. Circuits Systems I* **14,** 657–776.

Kaneran, F., and Habetler, T. G. (1998). "A novel on-line UPS with universal filtering capabilities," *IEEE Trans. Power Electronics* **13,** 410–418.

Perreault, D. J., Selders, R. L., and Kassakian, J. G. (1998). "Frequency-based current sharing techniques for paralleled power converters," *IEEE Trans. Power Electronics* **13,** 626–634.